

**AZƏRBAYCAN RESPUBLİKASI TƏHSİL NAZİRLİYİ
AZƏRBAYCAN DÖVLƏT İQTİSAD UNİVERSİTETİ**

“MAGİSTRATURA MƏRKƏZİ”

Əlyazması hüququnda

İbrahimova Xanım Ağa Mehdi qızı

**“Tətbiqi proqramlar paketinin layihələndirilmə vasitələrinin təşkili
problemlərinin həlli”**

mövzusunda

MAGİSTR DİSSERTASIYASI

İxtisasın şifri və adı : 060632 “İnformasiya texnologiyaları və sistemləri mühəndisliyi”
İxtisaslaşma: “Tətbiqi proqram təminatı”

Elmi rəhbər:

f.-r.e.n., dos. Əliyeva T.Ə.

Kafedra müdiri:

Magistr proqramının rəhbəri:

akad. Abbasov Ə.M.

akad. Abbasov Ə.M.

BAKİ - 2020

Mündəricat

Giriş.....	3
Fəsil I. TƏTBİQİ PROQRAMLAR PAKETİNİN ARXİTEKTURASI.....	7-54
1.1. Paket sisteminin xarakteristikaları	7
1.2. Proqramların modullaşmasına paket yanaşması.....	25
1.3. Funksional məzmunun modullaşdırılması qaydaları (modullaşmasının reqlamenti) və modulların konfigurasiyaları	30
1.4. Hesablamların planlanması və informasiya təminatı.....	38
Fəsil II. Hesablama təcrübəsi məsələlərinin proqramlaşdırılması.....	55-64
2.1. Hesablama təcrübəsi, hesablama təcrübəsi dövrü və nəticələrin emalı.....	55
2.3 Hesablama proqramlarının hazırlanması.....	58
2.4 OLYMPUS layihəsi.....	61
Fəsil III. Xüsusi tətbiqi proqramlar paketlərinin birləşdirilməsinin dialoglu sistemi.....	65-82
3.1. Tətbiqi proqramlar paketi generasiya edən sistemlər.....	65
3.2 . DİSUPPP kompleksinin tətbiq sahəsinin formal modeli.....	70
3.3. DISUPPP sisteminin arxitekturu.....	74
Nəticə və təkliflər	83
İstifadə edilmiş ədəbiyyat siyahısı	85
Резюме.....	86
Summary.....	87

GİRİŞ

Mövzunun aktuallığı. Uzun müddət sistem proqramlaşdırılmasında əsas diqqət kompüterlərin ümumi proqram təminatı məsələlərinin həllinə yönəldildi ki, bu da hesablama maşınlarının instrumental və idarəetmə səviyyəsinin yüksəldilməsi ilə əlaqəli idi. Bu, kompüter texnologiyasının inkişafında baş verən iki tendensiya ilə izah olunur. Bunlardan birincisi, hesablama maşınları parkının və tətbiq sferasının böyüməsi idi ki, bu da tətbiqi məsələlərin həlli proqramlarının hazırlanması və sazlanmasını daha səmərəli şəkildə həyata keçirməyə imkan verən bir sıra universal avtomatlaşdırılmış proqramlaşdırma sistemlərinin yaradılması tələbini ortaya çıxardı. Xüsusilə altmışıncı illərdə bu istiqamətdə intensiv işlər aparıldı.

Digər tendensiya - hesablama sistemlərinin aparat hissəsinin imkanlarının, iş rejimlərinin və təşkilinin keyfiyyətə təkmilləşdirilməsi idi. Burada avadanlıqların, proqramların və məlumatların səmərəli istifadəsini təmin edən tapşırıqların yerinə yetirilməsini təşkil edən inkişaf etmiş əməliyyat sistemlərinin yaradılması tələb olunurdu. Bu istiqamətdə işlərin intensivləşmə dövrü yeni arxitektura malik maşınların yaranması ilə əlaqələndirilir və 1965-1975-ci illərə təsadüf edir.

Kompüterlərin tematik ixtisaslaşma səviyyəsini artırmaq vəzifəsinə gəldikdə, ilkin mərhələdə sistem proqramlaşdırma bu sahədən translyasiya, düzəltmə və s. kimi fundamental problemlərin formalaşdırılmasını öyrənsə də, gələcəkdə o "hesablama xidmətləri" təminatıyla bağlı məsələlərin bilavasitə həllində marağını zəiflətdi. Kompüter sistemlərinin tematik ixtisaslaşma səviyyəsinin artımı əsasən müxtəlif əlavələrdə məsələlərin həlli üçün alqoritmlər və proqramlar yaratmış tətbiqi proqramlaşdırma qrupları tərəfindən həyata keçirilirdi.

Bir neçə il ərzində sistem proqramlaşdırma tərəfindən kompüter sistemlərinin tematik ixtisas səviyyəsini artırmaq məsələsinə kifayət qədər diqqət yetirilməməsi hesablama işinin səmərəliliyinə ciddi təsir göstərməmişdir. Bu onunla izah olunur ki, sistem proqramlaşdırmasının əsas məsələlərindən biri - məsələlərin həlli üçün alqoritmlərin formal təsviri vasitələrinin yaradılması və bu

cür təsvirləri maşın dilinə tərcümə edən aparatların - tətbiqi məsələlərin proqramlaşdırılması zərurəti ilə əlaqədar olaraq əvvəlcədən qoyulmuşdur. Daha əvvəl qeyd etdiyimiz kimi, bir sıra digər sistem proqramlaşdırma məsələləri də analogi mənşəyə malikdir. Təcrübənin əsas ehtiyaclarının nəzərə alınması, tətbiq olunan və sistem proqramçıları tərəfindən yerinə yetirilən işlər arasındakı məhdud əlaqələrə, həmçinin sistem dəstəyinin standart vasitələrinin ümumi təyinatlı təbiətinə baxmayaraq, bu vasitələr tətbiqi-proqramçıların sistemlə bağlı bütün sorğularını təmin edirdi. Mövcud sistem vasitələri hər hansı tətbiqi işləmələr üçün kifayət etmədiyi hallarda, müvafiq sistem imkanları tətbiqçilər tərəfindən müstəqil həyata keçirilirdi və bir qayda olaraq, aparılan işlər mövcud sistem vasitələrinin (bu və ya digər dərəcədə) modifikasiyasına gətirilirdi.

Yetmişinci illərin ortalarında bu vəziyyət ciddi dəyişikliklərə məruz qaldı. Hesablama maşınlarının sistem təminatı o qədər çətinləşdi ki, hətta onlara verilən bütün imkanların peşəkar səviyyədə istifadəsi, həm də onun modifikasiyası və ya inkişafı üzrə işlər istifadəçilərdən çox yüksək səviyyədə peşəkar bilik tələb edirdi. Bununla belə məlum oldu ki, ümumi təyinatlı sistem dəstəyi vasitələri əksər əlavələrdə təsvir olunan hesablama işlərinin yeni formalarına və onların inkişaf meyllərinə həmişə adekvat olmur.

Tədqiqatın məqsədi və başlıca vəzifələri: Tətbiqi proqramlar paketinin layihələndirilməsində istifadə olunan vasitələrin strukturu birinci növbədə kadrların- işçilərin birlikdə göstərdikləri fəaliyyətin əlaqələndirilməsi və təşkili kimi məsələlərin həlli ilə bilavasitə bağlıdır. Elə bu cəhətdən işin əsas məqsədi mövcud strukturun dəyərləndirilməsi, ondan səmərəli şəkildə istifadə olunması, problemli məqamların nəzərə alınmasından ibarətdir. Magistr dissertasiyasının əsas məqsədi tətbiqi proqramlar paketinin layihələndirilməsində istifadə olunan vasitələrin təşkili texnologiyasının bəzi problemlərinin hərtərəfli surətdə təhlilidir. Bu məqsədə nail olmaq üçün aşağıdakı vəzifələrin həyata keçirilməsini məqsədəuyğun hesab etmişik:

- 1) layihələndirilmə vasitələrinin təşkili mexanizminin araşdırılması və öyrənilməsi;

- 2) proqram təminatının kefiyyət göstəricilərinin müqayisəli analizi;
- 3) tətbiqi proqram paketlərinin layihələndirilməsinin əsas prinsiplərinin tədqiq edilməsi və hesablama təcrübəsi məsələlərinin proqramlaşdırılması metodlarının araşdırılması.

Arifmetik problem yönümlü sistemlərin, o cümlədən ağıllı tətbiqi proqram paketlərinin dizayn metodologiyasını öyrənmək, tətbiqetmə proqram paketlərini və bu sistemlərdə tətbiqi sahəsini izah etmək üçün istifadə olunan müxtəlif modelləri, tətbiqi proqram paketlərinin layihələndirilməsində proqramlaşdırma sistemlərini öyrənmək və təsnif etmək də tədqiqatın məqsədlərinə daxildir.

Məqsədlərin həyata keçirilməsi tətbiqi proqramlar paketinin layihələndirilməsi vasitələrinin təşkilinin gələcək vəziyyətini müqayisəli şəkildə qiymətləndirməyə, problemə müqayisəli qaydada yanaşmağa zəmin yaradacaqdır.

Tədqiqat işinin obyektı və predmeti: Tədqiqat işinin obyektı tətbiqi proqramlar paketinin layihələndirilməsində istifadə olunan vasitələrin təşkili, predmeti isə bu sahədə meydana çıxıb biləcək problemlərin həlli prosesindən ibarətdir. Əsas etibarlı ilə mövzunun tədqiqatında əsas istiqamətlər və həmin istiqamətlər üzrə mövcud olan nəzəri-metodoloji biliklər əsas götürülür.

Tədqiqatın elmi yeniliyi: Dissertasiyada sadalanan elmi – praktiki məlumatlar mövzu üzrə çoxlu sayda elmi ədəbiyyatlara və elmi nəzəriyyələrə əsaslanır. Bu mövzunun tədqiq olunmasından hasil olunan nəticələr, tətbiqi proqramlar paketinin layihələndirilməsində istifadə olunan vasitələrin təşkili və bu sahədə olan problemləri öyrənmək baxımından səmərəli elmi mənbə hesab oluna bilər. Tədqiqat işinin elmi yeniliyinin əsasını aşağıdakılar təşkil edir:

- 1) Tətbiqi proqramlar paketinin layihələndirilməsində istifadə olunan vasitələrin təşkilinin konseptual əsasları müəyyən olunmuşdur;
- 2) xüsusi tətbiqi proqram paketlərinin birləşdirilməsi modelləri və arxitekturasının mövcud vəziyyəti müəyyən olunmuşdur;

3) Hesablama təcrübəsi məsələlərinin proqramlaşdırılması və tətbiqi proqram paketlərinin arxitekturası sahəsində olan və meydana çıxacaq problemlərin aradan qaldırılması üçün həyata keçirilən tədbirlərin təkmilləşdirilməsi istiqamətləri müəyyən olunmuşdur.

Tədqiqat işinin praktiki əhəmiyyəti. Tədqiqat zamanı tətbiq proqram paketlərinin layihələndirilməsi vasitələrinin təşkili prosesində yaranan problemlər nəzərdən keçirildi, hesablama təcrübəsinin proqramlaşdırılması modeli, tətbiqi sahə modelinin yaradılması üsulları irəli sürülmüşdür ki, bu da mövzu üzrə tədqiqat aparan tələbələr, müəllimlər və bu sahəylə maraqlananlar üçün səmərəli vasitə hesab oluna bilər. Tədqiqat işi üzrə əldə olunan nəticələr, elmi məlumatlar sistemləşdirilmiş, son dövrlərin elmi ədəbiyyatı əsasında yeni elmi faktlar və məlumatlar əlavə edilmişdir. Sadalananlar böyük elmi-praktiki əhəmiyyətə malikdir və bu sahədə fəaliyyət göstərənlər üçün dəyərli təcrübi mənbə rolunu oynaya bilər.

Tədqiqat işinin strukturu: Dissertasiya işi giriş, üç fəsil, 9 paragraf, nəticə və təkliflər və istifadə edilmiş ədəbiyyat siyahısından ibarət olmaqla tərtib edilmişdir. Tədqiqat işi ümumilikdə 89 səhifə təşkil edir.

İşin girişində mövzunun aktuallığı, tədqiqatın məqsədi və başlıca vəzifələri ,tədqiqat işinin obyektı və predmeti ,tədqiqatın elmi yeniliyi və pratiki əhəmiyyəti verilmişdir.

I Fəsildə paket sisteminin xarakteristikaları, proqramların modullaşmasına paket yanaşması, funksional məzmunun modullaşdırılması qaydaları (modullaşmasının reqlamenti) və modulların konfigurasiyaları və hesablamaların planlanması və informasiya təminatı göstərilmişdir.

II Fəsildə hesablama təcrübəsi, hesablama təcrübəsi dövrü və nəticələrin emalı, hesablama proqramlarının hazırlanması, OLYMPUS layihəsindən bəhs edilir.

III Fəsildə tətbiqi proqramlar paketi generasiya edən sistemlər, DİSUPPP kompleksinin tətbiq sahəsinin formal modeli və DISUPPP sisteminin arxitekturu verilmişdir.

FƏSİL 1. TƏTBİQİ PROQRAMLAR PAKETİNİN ARXİTEKTURASI

1.1. Paket sisteminin xarakteristikaları

Tətbiqi proqramlar paketləri tətbiq olunan məsələlərin həllində insanın kompüterlə ünsiyyətinin səmərəliliyini artırmaq üçün bir vasitə rolunu oynayır. Hər bir xüsusi paketin hazırlamasının müvəffəqiyyəti avtomatlaşdırılmış tətbiqetmə fəaliyyətinin xüsusiyyətlərinin hərtərəfli nəzərə alınmasına əsaslanır.

Əlavələrin müxtəlifliyi (predmet sahələri və işlərin nizamı) təbii olaraq paketlərin də müxtəlifliyinə gətirib çıxarır. Bununla yanaşı, bu müxtəliflikdə iki komponenti ayırmaq lazımdır: paketin *funksional* məzmunu (dolması) və *sistem məzmunu* (dolması).

Paketlərin funksional məzmununun müxtəlifliyi, xidmət etdikləri predmet sahələrinin təbiətindən irəli gəlir. Burada sistemləşdirmə cəhdi (fiziki –texniki proqramların tətbiq sahələri ilə məhdudlaşsaq da) bizi bu məsələlərin çərçivələrindən çıxaracaqdır.

Əksinə, paketlərin qurulmasında istifadə olunan müxtəlif sistem metodları və vasitələri bu gün o qədər də böyük deyildir. Burada paket tərtibatçılarının qarşılaşdığı əsas sistem problemləri təhlil edilir. Buraya daxil edilmiş anlayışlar və terminologiya sonrakı fəsillərdə təsvir olunan bir sıra xüsusi istehsal paketlərində qəbul edilmiş sistem həllərini vahid baxımdan qiymətləndirməyə imkan verir.

Paketdə həyata keçirilməsi onun sistem xüsusiyyətləri ilə müəyyənləşdirilən ən vacib anlayışlara aşağıdakılar daxildir:

- tapşırıqların dili (paketlə istifadəçinin ünsiyyət vasitələri);
- funksional məzmunun modullaşması reqlamenti (proqram materialının hissələrə ayrılması üsulları, tətbiqi proqramlar paketinə daxil olan dizayn qaydaları);

- hesablamaların planlaşdırılması (hesablama zəncirinin qurulması üsulu, yəni istifadəçi tapşırığının icrasını təmin edən funksional və sistem modullarının ardıcılığı);

- əməliyyat imkanları (hesablama zəncirinin yerinə yetirilməsi üsulu, ştat proqram təminatı ilə interfeys, modullararası interfeysin təşkili, hesablama prosesini idarə etmə, proqram materialı və hesablama məlumatlarının saxlanması, paketlə istifadəçinin interaktiv qarşılıqlı əlaqəsi, sistem xidmətləri);

- informasiya xidmətləri (istifadəçilərdən və ya paketin proqram komponentlərindən alınan informasiya sorğularının təmin olunması).

Növbəti bölmələrdə bu anlayışların fiziki -texniki proqramlara yönəldilmiş paketlərdə həyata keçirilməsinə müxtəlif yanaşmalar nəzərdən keçirilir.

Hesablama texnikasından istifadə sahəsində əhəmiyyətli problemlərdən biri müxtəlif tətbiqi məsələlərin həlli zamanı hesablama maşını ilə insanın ünsiyyət problemi olur. Bu cür ünsiyyətin effektivliyini artırmaq üçün müvafiq alqoritmlərin və proqram təminatlarının inkişafı və təkmilləşdirilməsi sahəsində daimi səylər tələb olunur. Burada işlərin üç istiqamətini ayırmaq olar:

- proqram təminatının işlənməsini avtomatlaşdırmaq üçün istifadəçiləri müxtəlif vasitələrlə təmin edən proqram vasitələrinin yaradılması;

- mühəndis-dispetçer heyəti tərəfindən maşınların istismarı prosesini asanlaşdıran, eyni zamanda hesablama (kompüter) sisteminin bütün qaynaqlarından səmərəli istifadəni təmin edən proqram vasitələrinin yaradılması;

- tətbiq olunan problemlərin həllində istifadəçilərə müxtəlif "hesablama xidmətləri" təqdim edən proqram vasitələrinin yaradılması.

Bu üç sahə (istiqamət) müvafiq olaraq kompüterin (hesablayıcı maşının) instrumental, icraedici və tematik keyfiyyətlərinin yüksəldilməsinə gətirilir.

Bir çox mühüm əlavələrdə kompüter texnologiyasından istifadəni ləngidən bu mənfi hallar, insanın maşınla ünsiyyəti probleminin tematik aspektinə tətbiqi və sistem proqramlaşdırma baxımından kifayət qədər uyğunlaşdırılmamış yanaşmanın nəticəsi idi.

Bununla əlaqədar, maşınla insanın ünsiyyətinin təşkili vasitələrinin hazırkı inkişaf mərhələsində kompüter sistemlərinin tematik ixtisas səviyyəsinin yüksəldilməsi məsələnin həlli formalarına nisbətən əvvəlkindən daha mükəmməl və xüsusi diqqət yetirilir. Konkret olaraq bu o deməkdir ki, proqramlaşdırmanın vacib məqsədlərindən biri istifadəçilərin maşın tərəfindən təqdim olunan "hesablama xidmətləri" nə daha rahat girişini təmin edən və minimal peşəkar – proqramçı səviyyəsi tələb edən "dost" vasitələrin inkişafı olmuşdur.

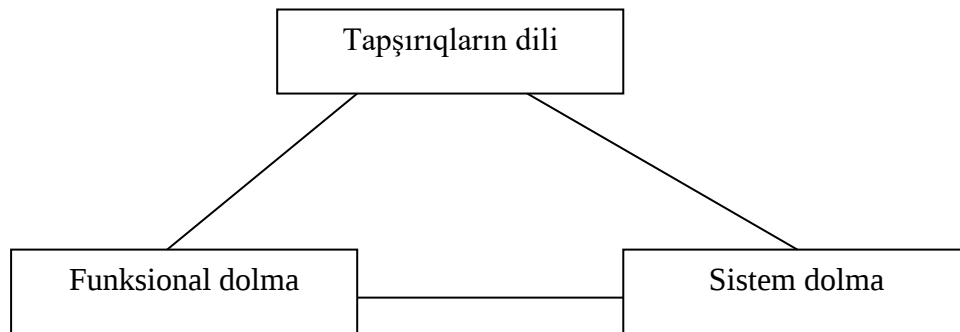
Hesablama sistemlərinin tematik ixtisaslaşma səviyyəsinin yüksəldilməsi fərdi əlavələrə yönəldilmiş proqram təminatının ixtisaslaşdırılmış komponentlərini yaratmaqla həyata keçirilir [1–11]. Bu halda sistemin səviyyəsinin yüksəldilməsi məsələsinin həllinin tamlığı və səmərəliliyi proqram təminatının hazırlanan komponentində nəzərdə tutulan və tətbiq olunan fəaliyyətin xüsusiyyətləri nəzərə alınmaqla əldə edilir. Bu cür ixtisaslaşdırılmış proqram komponentləri üzərində işin təşkilində vacib məqam tətbiqi və sistem proqramçılarının bütün inkişaf mərhələlərində birgə iştirakıdır. Bu yanaşma bizə yalnız sistem vasitələrinə əsaslanan "hesablama xidmətləri"ni səmərəli şəkildə həyata keçirməyə imkan vermir, həm də inkişaf zamanı ortaya çıxan tətbiq olunan problemlərin və sistem problemlərinin əlaqələndirilmiş peşəkar həlli sayəsində kompüterin tematik ixtisaslaşmasını əhəmiyyətli dərəcədə artırmağa imkan verir.

İxtisaslaşdırılmış proqram təminatının əsas formalarından biri tətbiqi proqramlar paketləridir.

Tətbiqi proqramlar paketi – bu, qarşılıqlı əlaqəli tətbiqi proqramların və məsələlərin müəyyən sinifinin həllinin avtomatlaşdırması üçün nəzərdə tutulmuş (proqramların və dillərin) sistem təminatı vasitələrinin kompleksidir [12].

Gələcəkdə alqoritmlərin işlənməsi və məsələlərin həlli proqramları, həmçinin hesablamaların hazırlanması və aparılması ilə bağlı bütün işləri *tətbiqi fəaliyyətin* avtomatlaşdırılması adlandıracağıq. Qeyd edək ki, istənilən xüsusi avtomatlaşdırılmış tətbiqi fəaliyyəti iki amil ilə xarakterizə olunur. Birincisi, *predmet sahəsi*, yəni həll olunan tətbiqi məsələlər toplusu və bunun üçün istifadə olunan ədədi metodlar və ikincisi, *iş nizamı*, yəni proqramların hazırlanması,

müzakirəsi və istismarı zamanı, o cümlədən hesablamalar zamanı qəbul edilən qaydalar, razılaşmalar, texnoloji yanaşmalar və üsullar sistemi. Tətbiqi proqramlar paketinin strukturunda üç əsas komponent ayırmaq olar: funksional məzmun (dolma), tapşırıq dili və sistem məzmunu (dolma) (şəkil 1) [13].



Şəkil 1.1. Tətbiqi proqram paketinin strukturu

Funksional məzmun paketin mövzu (predmet) sahəsinin xüsusiyyətlərini əks etdirir və modullar toplusunu təşkil edir. Burada *modul* dedikdə paketin fəaliyyətinin müxtəlif mərhələlərində istifadə olunan konstruktiv element başa düşülür. Etibarlı modul tiplərini, təqdimat formalarını, modullar arasındakı münasibətləri, modulların saxlanması və istifadəsi metodlarını, habelə onların birləşmə formalarını təyin edən bir modulun formal tərifini yalnız tapşırıq dilinin və sistem məzmununun xüsusiyyətlərinə əsaslanaraq müəyyən paketə tətbiq edildiyi kimi veriləcəkdir. Funksional dolma modullarının yazıldığı dil (dillər) paketin *baza dili* adlandırılacaqdır.

Paketin funksional məzmununun (doldurulmasının) tərkibi, gücü və ya mövzu sahəsinin əhatə dairəsinin bütövlüyü paketə daxil edilmiş tətbiqi biliklərin həcmi, yəni paketin tematik ixtisaslaşmasının potensial səviyyəsini əks etdirir. *Paketin iş dili* paket ilə istifadəçinin ünsiyyət vasitəsidir. O məsələnin həllini təmin edən müxtəlif əməliyyatların icra ardıcılığını və ya bu ardıcılığın avtomatik qurulduğu məsələnin şərhini təsvir etməyə imkan verir. Bundan əlavə, tapşırıq dili paketin fəaliyyəti çərçivəsində digər işlərlə bağlı sorğuları hazırlamağa imkan verir. Mümkün əməliyyatlar dəsti, leksika və tapşırıq dilinin sintaksisi paketin xidmət etdiyi predmet sahəsi və həyata keçirdiyi iş intizamı ilə müəyyən edilir.

Sistemin arxitekturası (yəni istifadəçiyə təqdim olunan zahiri görünüş) sistemin hansı məsələləri həll edə biləcəyi və istifadəçiyə hansı imkanlar verməsi ilə müəyyən edilir.

Aydındır ki, tapşırıq dili müəyyən tətbiq sahəsindəki hesablama sisteminin ixtisaslaşma səviyyəsini yüksəltməyə çalışan paketin tərtibatçıları tərəfindən verilən əsas arxitektura qərarlarını əks etdirir. İstifadəçi tapşırıq dili vasitəsilə "hesablama xidmətləri" nin nə olduğunu və onlardan istifadənin nə qədər rahat olduğunu, yəni bir hesablama sisteminin tematik ixtisaslaşmasının faktiki səviyyəsini necə olduğunu qiymətləndirir. *Sistem məzmunu* (doldurulması), bu tətbiqi fəaliyyətdə iş nizamına adekvat olan tapşırıqlar və paketlə istifadəçinin qarşılıqlı əlaqəsini təmin edən proqramlar toplusudur.

Sistem məzmunu (doldurulması), paketin tapşırıq dilində verilən imkanlara uyğun olaraq, funksional məzmunu daxil edilmiş bilik potensialının istifadəsini təşkil edir. Sistem məzmunu funksiyalarının yerinə yetirilməsi a) ümumi təyinatlı sistem təminatı vasitələrinin, b) standart proqram (ştat) komponentlərinin imkanlarını genişləndirən və uyğunlaşdıran sistem məzmunu vasitələrinin və c) paketi əhatə edən tətbiq xüsusiyyətləri nəzərə alınmaqla idarəetmə, arxivləşdirmə və işləmə prosedurlarını yerinə yetirən xüsusi sistem məzmun vasitələrinin əlaqələndirilmiş istifadəsi əsasında həyata keçirilir. Sistem doldurma proqramlarının yazıldığı dil (dillər) paketin *instrumental dili* adlanır.

Beləliklə, paketin quruluşu ilə bağlı qısa bir araşdırmadan sonra görmək olar ki, paket müəyyən tətbiqi fəaliyyətlə əlaqədar yerinə yetirilən hesablamalar haqqında bilikləri və bu biliklərin müxtəlif işləmə kontekstlərində genişlənməsinin və istifadəsinin müxtəlif formalarını təmin edən sistem vasitələrini özündə birləşdirdirir. Paketin bu keyfiyyətləri ən müasir əlavələrində izlənən hesablama işlərinin yeni formalarına olan tələblərə cavab verən proqram təminatı hazırlayarkən xüsusilə cəlbedici görünür.

Hesablama texnologiyalarından istifadənin ənənəvi olaraq daha sürətli templə həyata keçirildiyi fiziki - texniki əlavələrdən danışırıqsa, ilk növbədə aşağıdakı meylləri qeyd etmək lazımdır [2,9,14]:

- tədqiqatın müxtəlif fiziki və mühəndis-texniki sahələrini birləşdirən kompleks problemlərin həllinə keçid;
- təbiətşünaslıq təcrübələrinin aparılmasında hesablama maşınlar rolunun artması;
- hesablama sistemlərinin xidmətlərinə müraciət edən mütəxəssislər dairəsinin genişlənməsi;
- tətbiq fondlarının yaradılması və inkişafı;
- riyazi modellərin və alqoritmlərin modul quruluşu əsasında hesablamların səmərəli təşkili və proqramların dizaynı ilə əlaqəli məsələlərin proqramlaşdırılmasının ön plana çəkilməsi;
- həm proqramın işlənməsi baxımından, həm də hesablama baxımından müxtəlif yeni hesablama işlərinin meydana çıxması.

Baxılmış tendensiyaaların fiziki - texniki tətbiqlərdə (əlavələrdə) istifadə olunan hesablama sistemlərinin proqram təminatına qarşı nəhəng bir proqram layihəsi çərçivəsində belə təmin edilə bilməyəcək bir tələblər irəli sürdüyünü görmək asandır.

Burada, müxtəlif problemlə bağlı maraqları olan və hesablamları aparmaqda fərqli bacarıqlara sahib istifadəçilərin ehtiyaclarını ödəyə biləcək bir proqram təminatının yaradılması məsələsi meydana gəlir. Başqa sözlə, müxtəlif yönlü ixtisaslı proqram təminatının yaradılması vəzifəsindən bəhs edirik və yuxarıda qeyd edildiyi kimi bu məsələ, əlavələrdəki işlərin müəyyən hissəsinə yönəlmiş müxtəlif tətbiqi proqram paketlərini inkişaf etdirməklə effektiv şəkildə həll edilə bilər.

Hər bir paketin avtomatlaşdırılmış tətbiqi fəaliyyətinin fərdi ehtiyaclarına yönəldilməsi həm arxitektura baxımından, həm də paketin quruluşu baxımından həllərin müstəsna müxtəlifliyə malik olmasını izah edir. Bunu fiziki-texniki əlavələrlə bağlı iki misal ilə qısaca təsvir edə bilərik.

Beləliklə, kompleks problemlərin həlli ilə bağlı işə xidmət etmək üçün predmet sahəsinin əhatə dairəsində böyük ehtiyata malik və güclü funksional

məzmunu olan paketlər hazırlamaq lazımdır. Bu halda, çox vaxt mövcud tətbiqi proqram fondlarına əsaslanan funksional məzmun yaratmaq imkanı mövcud olur.

Tapşırıqların dili və bu cür paketlərin sistem məzmunu, ilk növbədə, funksional məzmun modullardan proqram sistemlərinin inkişafı və qurulmasında rahatlığı və səmərəliliyi, funksional məzmunun inkişaf etdirilməsini və proqram kompleksinin icrası zamanı modul qarşılıqlılığının müxtəlif formalarını dəstəkləməyi və uzun və böyük həcmli hesablama məlumatlarının təşkilini təmin etməlidir. Kompleks problemlərə xidmət göstərməyə yönəlmiş paketlərin hazırlamasında, proqram sistemlərinin qurulması və müntəzəm hesablamaların təşkili texnologiyasına diqqət yetirilir [1,9,14,15]. Bundan başqa nəzərə almaq lazımdır ki, bu paketlərin əsas istifadəçiləri tətbiq proqramçılardır, yəni bir proqram məhsulu hazırlayan və istifadə edən mütəxəssislərdir.

Əsasən təbii-elmi təcrübələrə xidmətə yönəlmiş paketlər fərqli keyfiyyətlərə sahib olmalıdır. Elmi müşahidələrin işlənməsinin müasir metodlarından istifadə [16-21], tam miqyaslı eksperiment dövrü ərzində hesablama maşının funksiyalarının spektrini, yəni təcrübi nəticələrin quraşdırılması və ilkin emalına nəzarət prosesinin avtomatlaşdırılmasından təcrübi məlumatların riyazi şərhinə qədər bu cür fiziki xüsusiyyətlərin dəyərlərini əldə etməyi əhəmiyyətli dərəcədə genişləndirməyə imkan verir. Bununla əlaqədar olaraq fiziki təcrübənin bütün mərhələlərində maşının "ixtisaslı iştirakını" təmin edən ixtisaslaşdırılmış proqram sistemləri və ya proqram paketləri böyük praktiki əhəmiyyət daşıyır. Belə paketlərin funksional məzmununun tərkibi nisbətən mühafizəkardır, çünki təcrübənin xüsusiyyətlərini əks etdirən məhdud tapşırıqlar dəsti ilə müəyyən edilir. Çox vaxt proqramların səmərəliyinə, xüsusən real vaxt rejimindən istifadə zamanı sərt tələblər qoyulur. Paketin istifadəçiləri ümumiyyətlə hesablama texnikası sahəsində peşəkar bacarıqlara malik olmayan mütəxəssislər (eksperimentatorlar, layihəçilər) olduğundan, hesablama maşını və bütövlükdə eksperimental quraşdırma ilə təbii ünsiyyət formalarının təmin edilməsinə xüsusi diqqət yetirilməlidir.

Baxdığımız misallar yalnız paketlərin hazırlanmasında müxtəlif yanaşmaların qısa şəkildə təsvir edilməsi üçün nəzərdə tutulsa da, bir paketin yaradılması bu tətbiq sahəsinin mütəxəssislərinin və sistem proqramçılarının birgə iştirakını tələb edən mürəkkəb bir iş olduğunu göstərir. Məhz bu tələbin reallaşdırılması və praktiki dəstək alması, hesablama sistemlərinin tematik ixtisas səviyyəsinin artırılması məsələnin həllində yeni bir mərhələ başladı.

Proqramlaşdırma sahəsində müstəqil bir sahə kimi problemlərin son on il yarım ərzində inkişaf etdirildiyini hesab etmək olar (və bu, paketlərə həsr olunmuş nəşrlərin tarixləri ilə təsdiqlənir). Bu dövrdə proqram paketi sadə bir tematik seçimdən və ya rudiment sistemli dəstəyi olan böyük bir proqram paketindən xüsusi tətbiqi hesablama fəaliyyətlərinin kifayət qədər tam avtomatlaşdırılmasını təmin edən inkişaf etmiş bir proqram sisteminə çevrilmişdir. [6,8,22]

Aşağıdakı qısa icmalda proqram paketlərinin təkamülündəki xarakterik məqamları vurğulamağa, həmçinin bu problemlərdə mövcud vəziyyəti əks etdirən paketlərin hazırlanmasına dair yanaşmaları qeyd etməyə çalışacağıq. Bunu paketin əsas struktur komponentlərinə uyğun üç aspektdə aparacağıq: funksional məzmun, tapşırıq dili və sistem məzmunu.

Hazırlanmış paketin funksional məzmununun sələfləri elementar hesablama prosedurlarını həyata keçirən standart alt proqram kitabxanaları və tipik, bir-biri ilə əlaqəli olmayan məsələlərin həlli üçün sadə tematik proqram dəstləridir. (Qeyd edək ki, "paket" terminini meydana gətirən məhz bu dəstlərdir.)

Belə proqram kolleksiyaları adətən aparıcı hesablama təşkilatları tərəfindən hazırlanır və kompüter istehsalçıları tərəfindən standart proqram təminatı ilə komplekt şəkildə tədarük olunur [23-29].

Alt proqramlar kitabxanaları və sadə paketlər həm geniş tətbiq olunan məsələlərin həlli üçün proqramların yaradılmasında kömək kimi, həm də paketin funksional tərkibinə daxil olan proqram materialı kimi istifadə olunur. Ancaq kitabxana və sadə paket arasında bir tərəfdən keyfiyyətə fərq, digər tərəfdən isə paketin funksional tərkibinə görə fərq mövcuddur.

Məsələ ondadır ki, kitabxananın və sadə paketin hazırlaması konkret tətbiqi fəaliyyətin bütün həcmi nəzərə alınmadan həyata keçirilir və ümumiyyətlə bir sıra predmetlər arasındakı sahələrə ümumi “hesablama xidmətləri” təqdim etmək məqsədi daşıyır. Bir qayda olaraq, bu aralıq sahələrin hər birinin spesifik məsələləri kitabxanada və sadə paketdə əks olunmur. Əksinə, hazırlanmış paketin funksional məzmununu təkmilləşdirərkən məqsəd konkret predmet sahəsinin tam əhatə olunmasına nail olmaqdır. Başqa sözlə, funksional məzmunlu proqram materialının tərkibi paketin təkmilləşdirilməsinin bu mərhələsində tətbiqi fəaliyyətdə nəzərdən keçirilənlər arasında istənilən tətbiqi məsələnin həllini təmin etməlidir. Bu tələblə əlaqədar olaraq, paketin funksional tərkibinin hazırlaması predmet sahəsinin modul təhlili ilə başlayır [14,30-33].

Modul təhlilin məqsədi sadə alqoritmlərin belə bir bazisini yaratmaqdır ki, onun superpozisiyası bu predmet sahəsində baxılan bütün məsələlərin həlli üçün alqoritmlər təqdim edə bilər [9]. Funksional yüklənmə baxımından alqoritmik bazisin elementləri həm riyazi, yəni riyazi metodların reallaşdırılmasını təsvir edən, həm də fiziki (və ya tətbiqi), yəni konkret fiziki məsələlərə və ya alt məsələlərə uyğun elementlər ola bilər [34].

Modul analizdən sonra funksional məzmunun proqram təminatına keçirlər. Bu mərhələdə modullar - paketin işinin müxtəlif mərhələlərində istifadə olunan konstruktiv elementlər hazırlanır. Modulların hazırlanması paketin funksional məzmununun tərkibinə daxil olan materialın modullaşdırılmasının müəyyən reqlamentinə uyğun olaraq aparılır. Tətbiq və sistem proqramçıları tərəfindən birlikdə hazırlanan belə bir reqlament, baza (xatırladaq ki, funksional məzmun modullarının dili baza adlanır) paket proqramlaşdırma dilini (dillərini), həmçinin modulların məqbul təmsil və qarşılıqlı formalarını müəyyən edir. Paketin modulları, ilk növbədə modul analiz mərhələsində ayrılmış əsas alqoritmlərin proqram təminatıdır. Belə modullar üçün modullaşma reqlamenti, məsələn, hamısına müstəqil translyasiyanın icazə verilməsini və ya baza dilinin prosedurları şəklində təqdim olunmasını tələb edə bilər və ya makro vasitələrin istifadəsinə icazə verə bilər və s. İnformasiya əlaqələrinin qurulması üsulları da tənzimləne

bilər: parametrlər vasitəsilə, ümumi dəyişənlər vasitəsilə, xüsusi interfeys vasitələrinin köməyi ilə və s. Modullaşmanın işlənilib hazırlanan reqlamentinə modulun təhlili yanaşı hesablama işlərinin planlaşdırılan təşkili də, yəni tətbiqi fəaliyyətin nizamı da təsir edir. Bununla ilkin məlumatlar toplusu, bəzi tipik tapşırıqların həlli sxemi, sənədləşmənin fraqmenti və s. kimi modulların meydana çıxması izah edilir.

Funksional məzmunun proqram reallaşması həm modulların orijinal işlənilib hazırlaması yolu ilə, həm də mövcud mənbələrdən proqram materialının uyğunlaşdırılması yolu ilə həyata keçirilə bilər. İnkişaf etmiş kitabxanalardan və böyük proqram fondlarından materialın uyğunlaşdırılması, onun modul proqramlaşdırma konsepsiyaları əsasında təşkil olunması ilə asanlaşdırılır [35-39].

İlk ixtisaslaşmış xüsusi proqramlaşdırma sistemlərinin və proqram paketlərinin tapşırıq dilləri, demək olar ki, maşınlarda yerinə yetirilən işlərin təsvir olduğu Fortran, Algol, assembler və ya əməliyyat və monitor sistemlərinin tapşırıqlarının idarəetmə dilləri kimi baza proqramlaşdırma dilləri ilə uyğun gəlir [18,40]. Belə dillərin problem yönümü əsasən dəyişənlərin, funksiyaların və prosedurların müəyyənləşdirilməsi zamanı müvafiq mnemonikanın tətbiqi ilə əldə olunmuşdur. Aydınır ki, belə bir yanaşma adi bir proqramlaşdırma ilə müqayisədə bir paket üçün tapşırıq tərtib edərkən xüsusi üstünlüklər vermədi, eyni zamanda heç bir xüsusi ixtisaslaşdırılmış sistem dəstəyini də tələb etmədi. Ona görə də təəccüblü deyildir ki, ilk paketlərin istehsalçı və istifadəçiləri paket tapşırıq dilinin çatışmazlıqlarını konkret əlavənin xüsusiyyətlərinə uyğun olmasına cəhd edərək kompensasiya edən tətbiqi proqramçılar idi.

Paketlərin hazırlanmasına sistem proqramçılarının cəlb edilməsi standart proqramlaşdırma dillərindən birinin əsasında paket tapşırıqları dilinin yaradılmasına yanaşmanı əsaslı şəkildə dəyişməyə imkan verdi. Hazırda tapşırıq dilinin problem yönümü təkcə istifadə olunan mnemonika ilə deyil, əsasən konkret predmet sahəsindən olan məsələlərin həlli prosesinin əsas mərhələlərini kompakt və aydın şəkildə formalaşdırmağa imkan verən xüsusi dil makrokonstruksiyaları ilə ifadə edilir. Bəzən daxili və ya quraşdırılmış (qoyulmuş) adlandırılan tapşırıq

dillərini inkişaf etdirmək üçün bu cür ideologiya, indi çox sayda tərəfdarına malikdir və müxtəlif problem yönümlü modul sistemlərinin qurulması təcrübəsində geniş yayılmışdır. İnteqrasiya edilmiş dillərin məşhurluğunun səbəbi, görünür ki, istifadəçidən yaxşı bir proqramlaşdırma bacarığının tələb olunsa da, dilləri genişləndirən güclü makrokonstruksiyalarla onların sistem reallaşmasının dəyəri arasında asanlıqla əldə edilə bilən səmərəli kompromisdir [41,42].

Paketlərin tapşırıq dillərinin inkişafındakı növbəti mərhələ *müstəqil* (yəni heç bir proqramlaşdırma dilləri ilə bağlı olmayan) problem yönümlü və ya *ixtisaslaşdırılmış* tapşırıq dillərinin yaranması ilə xarakterizə olunur [17.43, 44]. Belə bir dil istifadəçisinin peşəkar bir proqramçı olması mütləq deyildir. Bir qayda olaraq, tapşırıqların problem yönümlü dili paketin predmet sahəsinin lüğətinə əsaslanır və çox sadə bir sintaksisə malikdir. Bundan əlavə, işin xarakterini və istifadəçi proqramlaşdırma səviyyəsini nəzərə alaraq, paketin tərtibatçıları yüksək səviyyədə qeyri-prosessual problem yönümlü bir dil təmin etməyə çalışırlar. Bu o deməkdir ki, problem yönümlü bir dildə tərtib edilmiş paket üçün tapşırığın əsas hissəsi həll olunan məsələnin qoyuluşunun təsviridir (yəni "nə etmək lazım olduğunu" göstərir), daha az hissəsi isə bu problemin həlli prosesinin təsviridir (yəni "necə etmək lazımdır").

Xüsusi proqramlaşdırma proqramlarının aparatlarına əsaslanan problem yönümlü dillər konsepsiyası hələ 1959-cu ildə V.M. Qluşkov tərəfindən təklif edilmişdir [45]. Lakin, yalnız əməli təcəssümünü əldə edən paket problemlər üzərində iş aparıldı.

Bu, problem yönümlü bir dilin quruluşunun böyük semantik dolğunluğunun onların həyata keçirilməsi üçün inkişaf etdirilmiş proqram vasitələrinin tələb olunması ilə izah olunur. Paketdə göstərilən çətinlik müvəffəqiyyətli bir struktur həlli hesabına aradan qaldırılır: əsas yük funksional məzmunun modullarına, predmet sahəsində aparılan hesablamalar haqqında bilikləri tətbiq edən proqramların üzərinə düşür və tapşırığın interpretasiyası müvafiq modulları çağırmaq və onların arasında informasiya əlaqələrinin təşkilinə yaxınlaşır.

Hal-hazırda bu məqsəd üçün xüsusi instrumental - baza sistemlərin köməyi ilə paketlərin yeni gələcək paketlərin yaradılması üçün instrumental vasitələr, həmçinin bu paketlərə daxil olmaq üçün baza tədarükləri olan sistemlərin yaradılması məsələsi geniş yayılmışdır. Bu halda paketin tapşırıq dilinin hazırlamasına iki yanaşma tətbiq edilir [6].

Birincisi, instrumental baza sisteminin köməyi ilə yaradılan bütün paketlərin tapşırıq dillərinin ümumi bir hissəsi olan dil-nüvədən istifadə etmə fikrinə əsaslanır. Nüvə -dili kimi adətən sistemin özünün tapşırığının giriş dili çıxış edir.

Bu dil, paketlərin yaradılacağı əlavələrdə ümumi olanları əks etdirən vasitələrlə yanaşı genişləndirmə vasitələrini də özündə ehtiva edir. Baxılan yanaşmada, konkret paketin tapşırıq dili nüvə-dilə genişləndirmə vasitələri ilə müəyyən olunan və əlavənin xüsusiyyətlərini də nəzərə alan yeni anlayışlar və konstruksiyalar əlavə edilməklə yaradılır [46-48].

Başqa bir yanaşma, instrumental baza sisteminin giriş dili meta-dildir (yəni dilləri təsvir etmək üçün bir dil), bunun vasitəsilə hər yaradılmış paketin tapşırıq dilinin sintaksisi və semantikasi müəyyənləşdirilir. Kifayət qədər inkişaf etmiş bir metadildə bu yanaşma, instrumental baza sisteminin əlavədən asılı olmamasını təmin edir və paket tapşırıqlarının tamamilə fərqli forma dillərini müəyyənləşdirməyə imkan verir [49.50].

Təkmilləşdirilmiş paketin sistem doldurma vasitələri istifadəçinin əsas proqramının ayrı-ayrı hissələrdən və ona daxil edilən kitabxana altproqramlarından yığılması proseslərini avtomatlaşdıran ilk tərtib və şərh sistemlərindəndir [51-54].

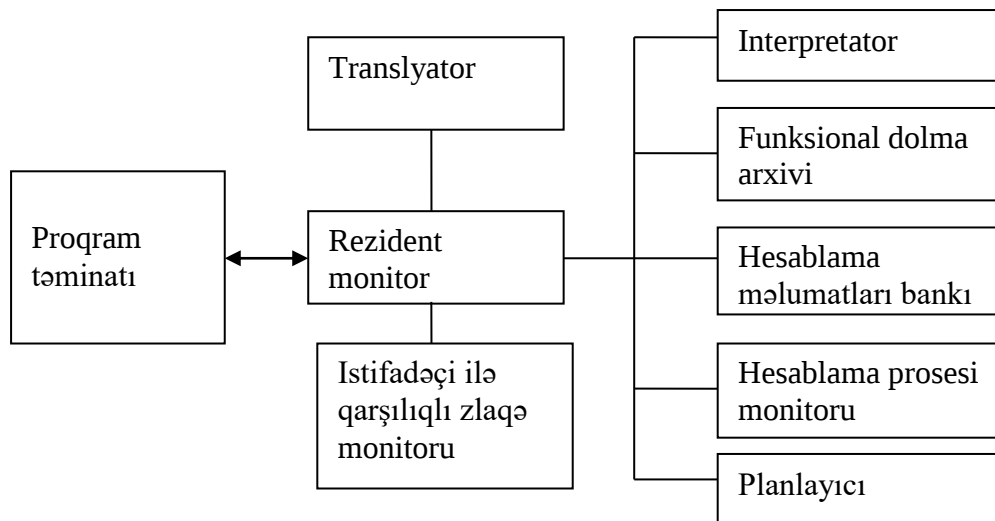
İlk ixtisaslaşdırılmış proqramlaşdırma sistemlərində və proqram paketlərində tapşırıq dilləri baza proqramlaşdırma dilləri ilə üst-üstə düşür, bu məqsədlə ya kompilyatorlar, makrogeneratorlar, xarici link redaktorları, mətn redaktorları kimi ştat proqram təminatlarının ayrı-ayrı vasitələrindən, ya da onlara əsaslanan proqramlaşdırmanın çoxfunksiyalı avtomatlaşdırma sistemlərindən istifadə edilmişdir [56,57].

Bununla yanaşı, ilk paketlərdə sistem doldurma funksiyaları yalnız paketin yuxarıda sadalanan proqram təminatının komponentləri ilə qarşılıqlı əlaqəsini

təşkil etməklə məhdudlaşmadı, həm də paketdəki xüsusi proseslərə də şamil edildi. Məsələn, sistem alt proqramlarının köməyi ilə HYDRA xüsusi proqramlaşdırma sistemində əməli yaddaşın dinamik bölgüsü məsələləri, verilənlərin spesifik strukturlarının formalaşması və onlara giriş, həmçinin modulların qarşılıqlı təsirinin təşkili məsələləri həll edilirdi.

Gələcəkdə predmet sahəsinin daha çox müxtəlif formalarının meydana çıxması ilə konkret məsələlərin həllində problem üzrə proqramçı tərəfindən yerinə yetirilən bütün fəaliyyət dəstəyini əhatə edən vasitələr də daxil olmaqla tapşırıq dilləri geniş yayılmışdır. Məlum oldu ki, dilin ixtisaslaşması problem üzrə proqramçının işinin istənilən mərhələsində əmək məhsuldarlığını əhəmiyyətli dərəcədə yüksəltməyə imkan verir. Belə yanaşmanın həyata keçirilməsi, tərkibinə digər müasir paket formalarına xas olan geniş sistem proqramlaşdırma vasitələrinin daxil edilməsini tələb edir [15.34.58-61].

Həqiqətən, bu paketlərin daxili quruluşları ilə tanışlıq, artıq ənənəvi hal alan sistem tərkibindəki bu cür komponentləri ayırmağa imkan verir (şəkil 2):



Şəkil 1.2. Paketin sistem dolmasının (məzmununun) komponentləri

- həm sistem məzmununun fərdi komponentləri arasında, həm də onlar ilə proqram təminatı arasında interfeysi təmin edən rezident monitor;
- tapşırığın daxili təqdimatını formalaşdıran və ümumiyyətlə makrogenerator və ya preprosessor kimi həyata keçirilən giriş tapşırıqlarının translyatoru;

- tapşırıqın daxili təqdimatının interpretatoru (şərhcisi);
- funksional məzmun (dolma) arxivi (proqram materialının saxlanması altsistemi);
- hesablama məlumatları bankı (ilkin və aralıq məlumatları saxlayan altsistem, həmçinin hesablamaların nəticələri);
- hesablama prosesinin təşkilinin monitoru (məlumat və nəzarət üçün modulların qarşılıqlı əlaqəsi);
- tapşırıq paketini həyata keçirən modulların icrasının ardıcılığını təyin edən hesablama prosesinin planlayıcısı;
- istifadəçi ilə interaktiv qarşılıqlı təsirin təşkilinin monitoru.

Müxtəlif paketlərdə sadalanan komponentlər fərqli bir inkişaf səviyyəsinə malik idi, bəzən onlar sadəcə yox idi, fərqli sistem yanaşmaları əsasında həyata keçirildi və müxtəlif cür adlandırılırdı (paket problemlərindəki terminologiya, təəssüf ki, tamamilə qurulmadı). Buna görə, yuxarıda göstərilən strukturu müəyyən və ya tipik bir paketin görüntüsü kimi deyil, sistemin tərkibindəki müxtəlif məlum komponentlərin birləşdirilmiş siyahısı kimi qəbul etmək lazımdır.

Paketlərin inkişafında digər müasir istiqamət proqramlaşdırma ilə tanış olmayan mütəxəssislərin rahatlığını təmin etməkdir. Burada əsas diqqət tapşırıq dilinin yüksək dərəcədə qeyri-prosedurluğunun əldə edilməsinə yönəldilmişdir ki, bu da istifadəçidən onun məsələsinin həllinin alqoritmik təfərrüatının böyük hissəsini gizlədirdi. Sistem məzmunu komponentləri arasında giriş tapşırığı translyatoru, hesablama planlayıcısı və istifadəçi ilə interaktiv qarşılıqlı əlaqəni təşkil edən monitor burada ön plana çıxır. Bundan başqa, bu cür paketlərdə informasiya - arayış xidmətləri inkişaf etdirilirdi [44,46].

Artıq qeyd etdik ki, paket problemlərindəki mühüm tendensiyalardan biri instrumental-baza sistemlərinin köməyi ilə paket qurulmasına keçiddir. Bir paketin funksional tərkibini formalaşdırarkən çox vaxt mövcud tətbiqi proqramlardan və alt proqramlar kitabxanalarından istifadə olunduğundan, adətən instrumental-baza sistemlərində bu cür proqram materiallarının qoşulmasını dəstəkləyən vasitələr nəzərdə tutulur. Bütün digər aspektlərdə, instrumental-baza sisteminin proqram

hissəsi elə reallaşdırılır ki, yaradılmış paketin sistem məzmununda baza qismində istifadə oluna bilsin.

Paket problemlərinin inkişafı ilə bağlı qısa araşdırmamızı yekunlaşdıraraq, diqqət çəkmək istəyirik ki, paketlərin təkamülü iki ölçmə fəzasında – tətbiqi və sistem fəzasında baş verir və paket yaratmaq üçün birtərəfli yanaşma uğurlu ola bilməz. Həqiqətən, predmet sahəsini əhatə edən həm alt proqramlar kitabxanası, həm də tətbiqi proqramların tematik dəsti yalnız paketin funksional məzmunu ilə müqayisə edilir və hər bir konkret məsələnin həlli üçün müxtəlif kombinasiyalarda alt proqram və proqramların birgə icrasının təşkili işini istifadəçiyə həvalə edir. Digər tərəfdən, müəyyən bir tətbiqi fəaliyyətin təhlili xaricində həyata keçirilən hər hansı bir universal proqramlaşdırma dili və onun reallaşdırılması bu sahədə hesablamaların avtomatlaşdırılması üçün müvafiq vasitə kimi çıxış edə bilməz. Birincisi, bu dildə, konkret tətbiqi fəaliyyətdə mütəxəssislərə tanış olan xüsusi terminologiya və texnoloji hesablama üsulları əks olunmayacaqdır. İkincisi, belə bir yanaşma, ümumiyyətlə, hər bir məsələnin həlli üçün müvafiq hesablama prosedurlarının hazırlanmasını tələb edir. Xüsusilə vurğulanmalıdır ki, hər iki birtərəfli yanaşma istifadəçidən yüksək peşəkar proqramlaşdırma bacarığının olmasını tələb edir.

Paket məsələləri ekstensiv inkişaf dövrü keçir. Fəaliyyətində paket mövzusu qabaqcıl yerlərdən birini tutan hesablama təşkilatlarının sayı, həmçinin proqram paketlərinin hazırlanması, yaradılması və ya hazırlanması üçün əlavələrin sayı artır. Eyni zamanda, paketlərin intensiv inkişafı mövcuddur ki, bu da öz əksini daha səmərəli və etibarlı alqoritmlərin və tətbiqi proqramların yaradılmasında, həmçinin sistem məzmununun təşkili üçün müxtəlif metod və formaların işlənməsində tapır.

Bu mövzuda tətbiq proqram paketlərinin qarşılaşdığı problemlər sistemləşdirilir və onların həlli yolları tədqiq olunur. Təqdimat bir sıra xüsusi istehsal paketlərinə əsaslanır. Təsvir olunan paketlər üçün predmet sahələrini, bir qayda olaraq, hesablama fizikasının müxtəlif məsələlər sinfi təşkil edir, lakin

bunlarda istifadə olunan sistem həlləri digər predmetyönümlü paketlərdə də tətbiq edilə bilər.

Material təqdim edərkən aşağıdakı məsələlərə xüsusi diqqət yetirilir:

- proqram paketlərinin sistem təminatının proqram və vasitələrinin təsnifatına və paketin sistem məzmununun arxitekturasını (xarici funksional imkanlarını) təyin edən əsas funksiyalarının aşkar edilməsinə;
- tətbiq olunan fəaliyyətin bütövlükdə paketin arxitekturasına təsirini və tətbiq olunan fəaliyyətin konkret xüsusiyyətləri ilə paketin bu və ya digər sistem funksiyasının həyata keçirilməsi formaları arasındakı əlaqələrin təhlilinə;
- paketlərin seçilmiş sistem funksiyalarının müxtəlif kombinasiyalarının effektiv şəkildə həyata keçirilməsinin və istehsal proqram paketlərinin praktik həyata keçirilməsinin mümkünlüyünün araşdırılmasına (əsasən hesablama fizikasının tətbiqi sahəsində).

Sistem təminatının hazırlanmasında istifadə olunan əsas anlayışlar: tapşırıq dili, funksional məzmunun modullaşmasının tənzimlənməsi, hesablamanın planlaşdırılması, əməliyyat imkanları, informasiya xidmətləri daxil edilir. Mühəndis-fiziki məsələləri həll edən paketlərdə bu anlayışların həyata keçirilməsinin xarakterik formaları təhlil edilməlidir.

Baxılan paketlərin əsas xüsusiyyəti, həll etdikləri məsələlərin müxtəlifliyi, müxtəlif sistem həllərinin geniş spektrindən istifadə olunmasına səbəb olmuşdur.

IPM istehsal paketlərində həyata keçirilməyən yeganə əhəmiyyətli funksiya - hesablamaların planlaşdırılmasıdır. Müzakirə olunan metod və vasitələr yeni tətbiq paketləri üçün arxitektura və sistem dəstəyinin seçilməsində faydalı olacaqdır.

Paketin tapşırıq dilinin ümumi quruluşu və üslubu, paketin xidmət etdiyi predmet sahəsində qəbul edilmiş iş intizamından çox asılıdır.

Hesablama işinin iki əsas (müəyyən mənada əksinə) nizamını qeyd etmək olar:

- funksional məzmun üçün mövcud modulların qurulması və tənzimlənməsi, habelə konkret hesablaşma proqramları yaradarkən yeni modulların işlənilib hazırlanmasını əhatə edən *aktiv nizam*;

- paketin funksional məzmununun modifikasiyası olmadan hesablamaları təmin edən *passiv nizam*.

Aktiv nizam, müvafiq predmet sahəsindən məsələlərin həlli metodlarına malik olan və bu məsələlərin həlli üçün proqram təminatına malik olan tətbiqi riyaziyyatçıların fəaliyyəti üçün xarakterikdir.

Passiv nizam son istifadəçilərin fəaliyyəti üçün yeni hesablama texnikası sahəsində yüksək səviyyədə təhsil alması mütləq olmayan və xüsusi tətbiqi məsələləri həll etmək bacarığı olan və maşından yüksək ixtisaslı bir kompüter kimi istifadə edən mütəxəssislər xarakterikdir.

İş nizamının bu cür iki yerə ayrılması kifayət qədər şərtidir (hesab etmək olar ki, onlar real mövcud olan nizamların spektrinin kənarlarına uyğundur) və hesablama işinin müxtəlif üslubları ilə əlavələrin (proqramların) avtomatlaşdırılmasında istifadə olunan sistem yanaşmalarının ziddiyyətini vurğulamaq məqsədi daşıyır.

Beləliklə, aktiv nizam rejimində hesablama işlərinin həyata keçirilməsini dəstəkləyən iş paketlərinin dilinin xarakterik xüsusiyyəti onun tətbiq olunan məsələlərin həlli üçün proqram sxemlərinin təsvirinə yönəldilməsidir.

Həm də bu dillərdə mərkəzi yeri (biz onları *montaj dilləri* adlandıracağıq) universal prosedur yönümlü proqramlaşdırma dillərinə xas olan verilənləri təsvir və onları manipulyasiya etmə vasitələri deyil, aşağıdakı vasitələr tutur:

- müəyyən bir məsələni həll edərkən modulların icra ardıcılığı və qarşılıqlı əlaqəsinin göstərildiyi proqram sxemlərinin tərtib edilməsi vasitələri;
- funksional məzmunun inkişafı və ya dəyişdirilməsi vasitələri;
- istifadəçinin tapşırığını həyata keçirən hesablama proqramının yaradılması və icrası proseslərinin idarə edilməsi vasitələri.

Bir tapşırıq çərçivəsində müxtəlif istiqamətli vasitələri tətbiq etmə imkanını təmin edən quraşdırma dilləri, bir qayda olaraq, istifadəçilərin kifayət qədər

peşəkar proqramlaşdırma bacarıqlarını tələb edir. Bununla belə, bu cür istifadəçilərin tətbiqi proqramçı olduqlarını nəzərə alsaq, bu tələblər olduqca məqbul görünür [18,56,62-66].

Qeyd edək ki, tətbiqi proqramçılara məsələlərin paketin köməyi ilə həll edilməsi üçün tələbat səviyyəsi ümumiyyətlə adi proqram təminatından istifadə edərək eyni işi yerinə yetirərkən onlara qoyulacaq tələblər səviyyəsindən bir az aşağı olur. Lakin ön plana, hesablamaları apararkən aktiv nizamlı paketlər üçün çox vaxt diqqət yalnız istifadəçilər üçün peşəkar tələblərin azaldılmasına deyil, həm də məhsuldarlığının artırılmasına yönəldilir. Burada məhsuldarlığın artmasına, xüsusən də universal ştat proqram təminatı mühitində olmayan proqramların modifikasiyası və yığılması üçün ixtisaslaşmış (xüsusi) sistem vasitələrindən istifadə etməklə nail olunur. Tələblərin səviyyələri və əmək məhsuldarlığında bu fərqlər, bu proqram paketi nəzərdən keçirilmiş məsələlər sinifini həll etmək üçün istifadə olunursa, hesablama sisteminin tematik keyfiyyətinin nə qədər artdığını xarakterizə edir.

Paketin tapşırıq dilinin tərtibatçılarının əsas məqsədi passiv intizam rejimində tətbiq olunan məsələlərin həllini təmin edən məsələnin həllinin alqoritmik təfərrüatlarının böyük hissəsini son istifadəçidən “gizlətmək” və ya başqa sözlə desək, dilin qeyri-prosedurluq səviyyəsini artırmaqdır [67]. *Sorğu dilləri* adlanan bu dillər adətən məsələlərin mənalı ifadələrini, yəni aydın şəkildə “necə əldə ediləcəyini” göstərmədən “alınması lazım olanı” göstərən sorğuları formalaşdırmağa yönəldilmişdir [48]. Bununla da istifadəçi onun məsələsinin həll etmə üsullarını və vasitələrini müəyyənləşdirmək ehtiyacından qurtulur ki, bu da onun proqramlaşdırma bacarığı səviyyəsinə olan tələb həddini kökündən azaltmağa imkan verir.

Hesablama sisteminin tematik ixtisaslaşma səviyyəsinin yüksəldilməsi burada sorğu dilindən istifadə edərək formalaşdırıla və həll edilə bilən çoxsaylı tətbiqi məsələlərlə müəyyən edilir. Paketin funksional doldurulması təkmilləşdikcə onun həll etdiyi məsələlərin dairəsi genişlənə bilər və buna görə də sorğuların dili, adətən, genişləndirilə bilən hesab edilir, yəni yeni sorğuların əlavə edilməsinə

icazə verilir. Dilin genişlənməsi müvafiq instrumental-baza sistemi ilə dəstəklənə bilər [49,50] və ya sistem məzmununun imkanlarını birbaşa genişləndirməklə əldə edilə bilər.

Paketin tapşırıq dili həm sərbəst bir dil şəklində, həm də inteqrasiya edilmiş (quraşdırılmış) dil, yəni mövcud proqramlaşdırma dilinin genişləndirilməsil şəklində həyata keçirilə bilər. Reallaşdırmanın formasından asılı olmayaraq, dil tərtibatçısı dil quruluşlarının leksika, sintaksis və semantikalarının tətbiq olunan məsələlərin istifadəçi qavrayışına mümkün qədər yaxın olmasını təmin etməyə çalışır. Predmet sahəsinin anlayışlarına və terminologiyasına və özünə xas olan iş intizamına güvənməklə yanaşı, avtomatlaşdırılan tətbiqi fəaliyyətin əsas iş aktlarını dildə əyani şəkildə əks etdirmək vacibdir.

1.2. Proqramların modullaşmasına paket yanaşması

Tətbiqi proqramlar paketinin hazırlanmasında əsas problemlərdən biri modulyarlaşma, yəni paketin funksional məzmununu modullara bölmək olur. Predmet sahəsinin hərtərəfli yerinə yetirilmiş modul təhlili və onun əsasında həyata keçirilən modulyarlaşma, paketi reallaşdırmaq üçün iş həcmi azaltmağa, etibarlılığını artırmağa və sonrakı təkmilləşməni asanlaşdırmağa imkan verir.

Xatırladaq ki, *modul* dedikdə paketin fəaliyyətinin müxtəlif mərhələlərində istifadə olunan bir konstruktiv element nəzərdə tutulur. Paketlərdə istifadə olunan müxtəlif modulların sayı çox böyükdür. Əvvəlcə proqram modulları, məlumat modulları və sənəd modulları ayırmaq lazımdır. Proqram modulları üçün, məsələn, belə (birindən digərinə tez-tez keçən) formalar: alt proqram; müstəqil translyasiyaya (keçidə) imkan verən alqoritmik dilin konstruksiyası; makrotərif; tədqiq etmək və / və ya redaktə etmək üçün proqramın müstəqil bir obyekt kimi qəbul edilən bir parçası kimi bir mətn olan bir fayl; proqramın müəyyən bir versiyasının necə qurulacağını göstərən təlimatlar dəsti; bir abstrakt məlumat tipinin həyata keçirilməsi və s. məlumdur.

Modulların bu və ya digər formaları, həmçinin modullara tətbiq olunan əməliyyatların məcmusu hər bir paketdə onun avtomatlaşdırdığı tətbiqi fəaliyyətin

xüsusiyyətlərinə əsasən seçilir. Proqram paketlərində modullaşma xüsusiyyətlərini daha dəqiq müəyyən etmək üçün əvvəlcə daha geniş tətbiq olunan modulların ayrılmasının bəzi motivlərinə baxaq, yəni yalnız paketlərdə deyil, digər proqram hazırlamalarında da istifadə olunur.

Bu motivlərdən ən sadəsi və aydın olanı ***qeydlərə qənaət etməkdir***. Proqramın təkrar və ya oxşar hissələri ayrı bir modul şəklində icra olunur (alt proqramlar, makrotəriflər və s.). Proqramın mətni qısaldılmışdır, çünki seçilmiş hissələrin yerində yalnız seçilmiş modula müraciətlər qeyd olunur.

Translyasiyanın ümumi vaxtının ixtisarı, müstəqil translyasiyaya imkan verən modulların ayrılması hesabına əldə edilir. Onda dəyişiklik edərkən, bütün proqramı deyil, yalnız toxunulmuş modulları təkrarlamaq kifayətdir.

Proqramın mətninin redaktəsinin (düzəltməyin) rahatlığı böyük proqramları görünən kimi işləyən və buna görə redaktənin daha rahat obyektləri olan kiçik hissələrə ayırmağı tələb edir.

Aydınlıq mülahizələri və ya proqramın qəbul edilməsinin asanlıığı da onun hissələrə bölünməsinə səbəb olur. Burada modullar "bir baxışla tutmaq" və ya siyahı səhifəsində və ya ekranda göstərilə bilən modullardan tamamilə fərqlənir.

Bir neçə icraçının arasında ***bir proqram yazma işi paylandıqda***, fərdi icraçılar tərəfindən yerinə yetirilən hissələr modul rolunu oynayır. Adətən bu cür hissələr arasındakı əlaqələri minimuma endirməyə çalışırlar ki, bu da təkmilləşmə zamanı aparılan birgə müzakirələrin sayını və intensivliyini azaldır.

Uzun bir proqramın məhdud həcmi əməli yaddaşda yerləşdirmək tələbi çox vaxt proqramın icrası zamanı əməli yaddaşda bir-birini əvəz edən modulların (seqmentlərin) ayrılmasına səbəb olur. Segmentasiya zamanı modullaşmanın müvəffəqiyyətinin əsas meyarı -xarici yaddaşda tələb olunan sayda dəyişdirmə seqmentini minimuma endirməkdir.

Yuxarıda modullaşmanın sadalanan motivləri proqram təsərrüfatının fərdi aspektləri ilə əlaqələndirilir və buna görə də texniki xarakter daşıyır. İstənilən böyük proqram layihəsində bu və ya digər dərəcədə bu motivlərin hər biri ilə rastlaşmaq lazım olur. Bununla yanaşı, proqramların modullara

dekompozisiyasının səbəblərini təhlil edərkən mürəkkəb proqramların hazırlamasının *sistemləşdirmə və sadələşdirilməsi* mülahizələri ümumiyyətlə ön plana çıxır. Bu mülahizələrə əsasən funksional olaraq müstəqil proqram hissələrini modul olaraq ayırmaq tövsiyə olunur.

Deykstranın qeyd etdiyi kimi [68], müəyyən modullaşmanın əsasını düşüncə tərzini təşkil edir. Bu nizam əsasən, mürəkkəb bir problemi həll etmək, hər biri ayrıca bir araşdırmaya imkan verən nisbətən müstəqil aspektləri bir sıra halına gətirməklə sadələşdirmək olar. Qeyd olunan ayrılma təbii üsulla bu problemi həll edən proqramın quruluşunda öz əksini tapmışdır: müəyyən edilmiş aspektlərin hər birinə ayrı bir proqram modulu uyğundur.

Modullaşmanın texniki motivləri, bir qayda olaraq, bir-biri ilə yaxşı uzlaşır. Məsələ ondadır ki, onların əksəriyyəti üçün modulun konkret sərhədlərinin seçilməsində müəyyən sərbəstliyə yol verilir. Bu sərbəstlik çərçivəsində sərhədlər seçilsə, funksional sərbəstlik və modulların müstəqilliyi mülahizələrini rəhbər tutaraq müxtəlif motivlərin tələblərini uzlaşdırmaq mümkündür.

Bu halda xüsusən əyanilik artır, çünki modul tərəfindən emal edilən obyektlərin sayı azalır və ən əsas modulun təyinatını dəqiq şəkildə müəyyənləşdirmək mümkün olur. Funksional modullaşma əsasında icraçılar arasında iş bölgüsü nisbətən müstəqil modulların ayrılmasına səbəb olur və buna görə də onlar arasındakı əlaqələrin sayını minimuma endirir. Nəhayət, müəyyən anlayışa aid olan proqram dəyişiklikləri bir modulda lokallaşdığından redaktənin rahatlığı və translyasiya vaxtına qənaət yalnız bir modulun redaktə edilməsi və yenidən transkyasiya edilməsi sayəsində əldə edilə bilər.

Burada funksional modullaşmasının xeyrinə daha bir əhəmiyyətli arqumentə yaxınlaşdıq. Onun köməyi ilə proqramın inkişafını asanlaşdırmaq mümkündür, çünki ümumiyyətlə, proqrama edilən dəyişikliklərin lokalizasiyasını təmin edir. Parnas [69], proqramın hazırlanması zamanı verilmiş hər hansı bir həlli modul şəklində tərtib etməyi təklif edir və bununla nəticələrini proqramın qalan hissəsindən "gizlədir" və bu həllin sonradan

nəzərdən keçirilməsini asanlaşdırır. Praktiki olaraq bütün böyük proqramlar (tətbiqi proqram paketləri daxil olmaqla) mövcud olduqları müddətdə çoxsaylı dəyişikliklərə məruz qalırlar və belə dəyişikliklərin həyata keçirilməsini asanlaşdırmaq olduqca aktual məsələdir.

Funksional modullaşmanın tətbiqi proqramlar paketlərini qurarkən yeni məzmun verilir. Burada yalnız funksional olaraq sərbəst hissələri ayırd etmələr, həm də xidmət olunan predmet sahəsini "əhatə edən" bir dəstə yaratmağa çalışırlar. Sahənin əhatə dairəsi, orada qoyulmuş hər hansı bir məsələ üçün, seçilmiş dəstdən modulların zəruri şəkildə təşkil edilmiş alt çoxluğunu təşkil edən həlledici hesablama proqramının qurula biləcəyi deməkdir.

Şəkil 1.1-də modullaşmaya paket yanaşma ilə adi bir "universal" proqramın hazırlaması arasındakı fərqi təsvir edilmişdir, burada bir proqramın köməyi ilə mümkün olan məsələlərin ən geniş sinifləri əhatə etməyə çalışırlar. İlk baxışdan fərq ondadır ki, hər iki halda proqramın eyni hissələri yaradılır, lakin bütün yığılmış proqram materialı yerinə yetirilən universal proqrama daxil edilir və yalnız bəzi seçilmiş modullar paketin hesablama proqramına daxil edilir.



Şəkil 1.1. Modullaşmada iki yanaşma:

a) universal proqram; b) paket

Modullaşmada paket yanaşmasının əsas üstünlüyü - proqram fondunun *ağrısız inkişaf* imkanının olmasıdır. Paketlə həll olunan məsələlər sinfinin genişlənməsinə əsasən yenidən yaradılmış modulları paketə əlavə olaraq bağlamaqla nail olmaq olar; bu halda, ümumiyyətlə, əvvəllər mövcud olan modulları dəyişdirmək tələb olunmur. Əksinə, "universal" bir proqramın

inkişafı həmişə proqramın daha əvvəl yazılmış hissələrinin mətnlərindəki dəyişikliklər ilə əlaqələndirilir ki, bu da dəyişiklik edilməzdən əvvəl hesabın silinməsi zamanı uğursuzluğa səbəb ola bilər. Artıq qeyd edildiyi kimi, dəyişikliklərin icra edilməsinin sadəliyi böyük bir proqramın ən vacib keyfiyyətlərindən biridir və buna görə modullaşmada paket yanaşma geniş yayılmışdır.

Eyni zamanda, ciddi bir texnoloji problemin paket yanaşması ilə əlaqəli olduğunu qeyd etmək lazımdır: mövcud modulların yeni yaradılanlarla qarşılıqlı əlaqəsini necə təşkil etmək olar?

Bu problem paketlərlə aralıq sahədə - proqram kitabxanalarında asanlıqla həll olunur. Kitabxanaların spesifikliyi ondadır ki, onların yaradılması zamanı, paketlərdən fərqli olaraq, məqsəd predmet sahəsini əhatə etməməkdir, yəni yalnız kitabxanada saxlanılan modullardan hesablama proqramları yaratmaqdır. Adətən hesablanmada kitabxana modulları qarşılıqlı əlaqənin təşkil olunduğu kitabxanadan kənarda hazırlanan və saxlanılan modullara qonşu sahədə yerləşir. Beləliklə, kitabxana modullarına olan tələblər paket modullarına olan tələblərdən bir qədər fərqlidir. Kitabxananın tərtibatçısı, əsas etibarilə, modulları yalnız onların imkanlarının vahid təsvirinə diqqət yetirərək "toplayır". Kitabxana modulları kolleksiyasının mexaniki şəkildə doldurulması, proqramçı tərəfindən nadir hallarda rast gəlinən bir vəziyyətə nümunədir, bu zaman tərtibatçı tərəfindən xüsusi səy göstərilmədən, proqram fondunun inkişafı əvvəllər yığılan proqramlara təsir etmədən olur.

Paketlərdə hesablama proqramı tamamilə paket modullarından ibarətdir (bu da paketi proqramlaşdırma ilə tanış olmayan insanlar tərəfindən istifadə etməyə imkan verir). Buna görə, yeni modullar paketinə çətinlik çəkmədən qoşulmanın həyata keçirilməsi üçün müəyyən təşkilati səylər tələb olunur.

Süni intellekt texnologiyası çərçivəsində paket modullarının qarşılıqlı əlaqəsini təşkil etmək probleminin məlum həlli var. Giriş / çıxış məlumatlarının və tətbiq olunma şərtlərinin formal təsviri hər bir modulla əlaqələndirilir və funksional məzmunu yazılır və birbaşa modul çağırışları əvəzinə müəyyən bir

modulu deyil, yalnız modulun qarşısında duran vəzifəni təyin edən " nümunə kimi " çağırışlar istifadə olunur [48,70,71].

Buna görə, yenidən ortaya çıxan modul modulların mətnlərinə - potensial istehlakçılara təsir göstərmir. Bu modullar təsvirinin onların təyin etdikləri sorğulara uyğun olduğuna görə avtomatik olaraq tapacaqdır.

Mövcud proqram paketlərində süni intellekt texnologiyası nisbətən nadir hallarda istifadə olunur, çünki bu, çox vaxt effektivliyin əhəmiyyətli dərəcədə itirilməsinə səbəb olur və paketin tərtibatçılarından qeyr-trivial məntiqi aparatda səlis olmağı tələb edir. Əksər hallarda, bununla birlikdə ağrısız dəyişiklik etmək xüsusiyyətini təmin edən, daha sadə bir proqram quruluşu əsasında paketin qarşısındakı məsələlərə daha qənaətli bir həll tapmaq mümkündür.

1.3. Funksional məzmunun modullaşdırılması qaydaları

(modullaşmasının reqlamenti) və modulların konfigurasiyaları

Paketin fəaliyyətinin müxtəlif mərhələlərində istifadə olunan bir konstruktiv element kimi modulun tərifinə qayıdaq. Burada modulun *konstruktivliyi* nə deməkdir?

Əvvəla, əsasən *alqoritmik* konstruktivliyi nəzərdə tuturuq. Modul, tətbiq olunan fəaliyyətin ixtiyari sorğularını təmin edən proqramların qurulması üçün əsas olan predmet sahəsinin modul təhlili nəticəsində əldə edilmiş alqoritmik əsas elementidir. Bundan başqa, modulların alqoritmik konstruktivliyinə tipik hesablama alqoritmlərinin strukturları, bu strukturlarda istifadə olunan alqoritmik bazisin elementləri arasındakı münasibətlər və müxtəlif hesablama kontekstlərində yaranan məlumat axını təsir göstərir.

Alqoritmik konstruktivliklə yanaşı, paketdə işlənib hazırlanan və tətbiq olunan iş mühitində və sistem mühitində tətbiq olunan modulların *texnoloji* konstruktivliyini də vurğulamaq lazımdır. Texnoloji konstruktivliyə aşağıdakı amillər təsir edir:

- proqram modullarının təqdimat formaları;

- proqram sistemlərinin ayrı-ayrı hissələri arasındakı idarəetmə əlaqələrinin növləri (açıq və qapalı alt proqramlar, müştərək proqramlar);
- informasiya əlaqələrinin təşkili üsulları (parametr aparatları, ümumi yaddaş sahələri, ümumi fayllar vasitəsilə);
- tətbiqdə istifadə olunan proqram sistemlərinin hazırlama metodları (yuxarıdan aşağıya, aşağıdan yuxarıya və s.);
- tətbiqi proqramların hazırlanmasında istifadə olunan baza dili və ya proqramlaşdırma dilləri;
- tətbiqi proqramların ölçüsünə məhdudiyyətlər;
- proqram sistemlərinin əlaqələndirilməsi, yüklənməsi və seqmentləşdirilməsini, mətnlərin redaktəsini təmin edən ştat sistem vasitələrinin imkanları.

Modulların alqoritmik və texnoloji konstruktivliyindən, habelə aşağıda nəzərdən keçirilən modulların bəzi digər xüsusiyyətlərindən irəli gələn tələblər birlikdə *modullaşma qaydasını (reqlamentini)*, yəni paketin tərtibatçıları tərəfindən qəbul edilmiş funksional məzmununda materialın təqdimat forması, həmçinin onun yaradılması və inkişafı üsulları təşkil edir. Əgər tapşırıq dilinin təsviri proqram paketi ilə istifadəçinin əlaqəsinin spesifikliyi kimi qiymətləndirilsə [72], onda tərtibatçılar modullaşmanın qaydalarından (reqlamentindən) istifadə edərək paketə (daha dəqiq desək, paketin funksional məzmunu ilə əlaqə) qoşulmanı təyin edirlər.

Modullaşma reqlamenti ilə təmin olunan modulların vacib xüsusiyyətlərindən biri *uyğunluqdur*, yəni hesablama proqramları çərçivəsində onların birgə işləmələrinin mümkünlüyüdür.

Modul uyğunluğunun ən kəskin problemi, müəyyən bir paket üzərində iş başlandıqda, müstəqil tərtibatçı qrupları tərəfindən yaradılmış böyük bir proqram fondunun artıq mövcud olmasıdır. Daha əvvəl yaradılan fond çox vaxt standartlaşdırılmamış olur. Bu fond paketin funksional məzmununa daxil edildikdə, funksional baxımdan predmet sahəsini əhatə etməyən alt proqramlar məcmusu yaranır və onlar eyni proqram çərçivəsində bir-biri ilə əlaqə yarada

bilmirlər, çünki onlar əvvəlcədən müstəqil şəkildə yaradıldığından birgə iş üçün yazılmamışdır.

Funksional olaraq paketin alqoritmik bazasına daxil olan, lakin birbaşa hesablama proqramlarında qarşılıqlı əlaqə qura bilməyən belə alt proqramları *kvazimodullar* adlandıracağıq. Kvazimodulları birləşdirmək üçün əsas maneə ümumiyyətlə birgə istifadə edilən məlumatların təqdimatı və saxlanması arasındakı fərkdir.

Standartlaşdırılmamış fondun probleminin iki müxtəlif həlli var. Birincisi, çox vaxt aparan (buna görə ümumiyyətlə qəbul edilməzdir) və paketdə qəbul edilmiş standartlara uyğun olaraq bütün kvazimodulları yenidən yazmaqdır. İkinci həll proqram materialının yalnız cüzi dəqiqləşdirilməsini tələb edən funksional məzmun paketi yaratmaq üçün belə bir texnologiyayı təmin etməkdir.

Beləliklə, reqlamentin formalaşmasına iki yanaşma ayırd edilə bilər: daxili və xarici modullaşma.

Daxili modullaşmada, modulların uyğunluğu modulların proqram gövdəsinin hazırlanması zamanı reqlamentin bütün tələblərinə riayət etməklə əldə edildiyi fərz edilir. Reqlament, həm idarəetmə, həm də məlumat üçün modullar arasındakı interfeysin təşkili metodlarını müəyyən edir. Bir paket sıfırdan hazırlanarkən daxili modullaşma istifadə olunur. Bundan başqa, əgər proqram fondu artıq mövcuddursa, lakin bəzi reqlamentə uyğun olaraq standartlaşdırılırsa, paketin modullaşma qaydaları proqram fondunun reqlamentinə sahib ola bilər və bununla da modulların uyğunluğunu əvvəlki-lərdəm miras olaraq ala bilər. Daxili modullaşma zamanı adətən modullar arası interfeysləri dəstəkləmək üçün xüsusi sistem vasitələri tələb olunmur, modulların uyğunluğu modullararası əlaqələrin müntəzəm redaktə vasitələri ilə təmin olunur [17, 46, 65].

Qeyri-standart bir fondla işləyərkən *xarici modullaşma* istifadə edilə bilər. Kvasimodulların mənbə mətni dəyişməz olaraq qalır, lakin hər kvasimodula pasport və ya modulun *başlığı* adlanan müşayiət olunan bir sistem qeydləri ilə

əlavə olunur, bu da bütün giriş və çıxış obyektlərinin formal təsvirini ehtiva edir [47, 73–77]. Başlıqda qeyd edilən məlumatlara sahib olaraq, paketin sistem məzmunu bütün lazımi modullararası interfeysi təmin edə bilər. Beləliklə, “kvasimodul – başlıq” cütü paket tam hüquqlu moduluna çevrilir.

Xarici modullaşmada adətən kvasimodulların rəsmiləşdirilməsinə heç bir xüsusi tələb qoyulmur (“qara qutu” kimi onlara baxmağa imkan verir) və beləliklə xaricdən gələn standart olmayan proqram materiallarını nisbətən asanlıqla uyğunlaşdırmağa kömək edir. Uyğunlaşmanın sadəliyini əvəzində - daxili modullaşma ilə müqayisədə daha çox təkmilləşdirilmiş sistem təminatı vasitələrinə ehtiyac yaranır. Burada lazım olan şey modulların başlıqlarının təsvir edən dildir və modullararası interfeysi təşkil edərkən bu başlıqları “anlayan” sistem vasitələridir.

Modul başlıqlarını və sistem vasitələrini təsvir etmək üçün bir dilin hazırlanması yalnız o halda özünü doğruldur ki, uyğunlaşan kvasimodullar bir çox fərqli konfigurasiyada birləşdirilə bilsinlər. Formalaşdırılmış hesablama proqramlarında kvazimodulların kombinasiyalarının planlaşdırılmış sayı azdırsa və əvvəlcədən dəqiq müəyyənləşdirilə bilərsə, hər kvasimodul üçün funksional dolmanın konkret modulları ilə uyğunluğu təmin edən xüsusi bir proqram (proqram adapteri) hazırlamaqla xarici modullaşmanı həyata keçirmək əhəmiyyətə malik olar.

Hesablamaların planlanmasını həyata keçirən paketlərdə, hər iki yanaşmanın imkanlarını birləşdirən *qarışıq modullaşma* tətbiq olunur. Burada, funksional dolma üçün proqramlaşdırma qaydalarının (reqlamentinin) müvafiq daxili modullaşmasına uyğun olaraq, modullararası interfeysin təşkili ilə bağlı xərclər minimuma endirilir və modulların semantikasını, tətbiq olunma şərtlərini, qiymətləri və hesablamaların avtomatik planlaşdırılması üçün lazım olan digər göstəriciləri təsvir etmək üçün xarici modulizasiya vasitələri istifadə olunur [71].

Beləliklə, predmet sahəsinin modulyar analizinin necə aparıldığını və onun əsasında paketin əsas proqram modullarının baza dəsti necə qurulduğunu

öyrəndik. İndi bu modulların hesablama proqramlarında hansı konfiqurasiyalar yarada biləcəyini və konkret konfiqurasiyaları dəqiqləşdirmək üçün hansı metodların mövcud olduğunu nəzərdən keçirək.

Bir çox paketlərdə, bütün hesablamalar az dəyişən bir sxemə görə aparılır. Bu sxemə paket ilə formalaşdırılan bütün hesablama proqramlarının müəyyən strukturu uyğundur. Burada aparılan hesablamaların müxtəlifliyi hesablama sxemlərinin sayı ilə deyil, müəyyən olunmuş proqram strukturunun yuvalarının – karkasın (proqramın gövdəsinin) doldurulmasının çoxsaylı variantları ilə əldə edilir. Məsələn, paketin funksional tərkibində hesablamanın müxtəlif metodlarını reallaşdıran bir neçə modul ola bilər. Lakin proqramın strukturunda istifadəçilərin göstərişlərinə əsasən hesablama proqramlarının formalaşmasında bu və ya digər modulun daxil olduğu yeganə "METOD" yuvası nəzərdə tutulmuşdur.

Karkas adlandıracağımız bu yanaşma ilə modulların qarşılıqlı təsirinin üsullarına heç bir məhdudiyyət qoyulmur. Bununla birlikdə, paketin tərtibçisi qarşılıqlı əlaqə metodları seçmək azadlığından yalnız proqram çərçivəsinin tərtib edilməsi mərhələsində istifadə edir və gələcəkdə bütün modullar çərçivənin karkasın planlaşdırılması zamanı hazırlanmış yuvaların doldurulması tələblərinə ciddi şəkildə uyğunlaşdırılmalıdır.

Karkasın (Çərçivənin) bir yuvasına aid modulların vahid rəsmiləşdirməsi hesablama proqramlarının variantlarında modulların müxtəlif kombinasiyalarını olduqca sərbəst birləşdirməyə imkan verir. Karkas yanaşması ilə paketin yaratdığı bir çox hesablama proqramının gücünü sayca qiymətləndirə bilərsiniz. Tutaq ki, layihələşdirilmiş karkasda n yuvalar var və i yuvalar ($1 \leq i \leq n$) üçün onu reallaşdıran müxtəlif m_i modullar var.

Onda mövcud ola bilən hesablama proqramlarının sayı R , ümumiyyətlə, verilmiş funksional dolma əsasında yarana bilər və aşağıdakı düsturla ifadə edilir

$$P = \prod_{i=1}^n m_i.$$

Qeyd edək ki, real hesablamalarda, modulların hamısının potensial etibarlı kombinasiyaları yaranmır. Buna baxmayaraq, bir çox istehsal paketlərində faktiki istifadə olunan variantların sayı verilən qiymətə yaxındır.

Asanlıqla görmək olar ki, karkas yanaşması paketin proqram fondunun inkişafının ağrısızlığını təmin edir.

Həqiqətən, fondun inkişafı, bir qayda olaraq, ayrılmış proqramın karkasının seçilmiş yuvaları üçün yeni modul-reallaşdırmaların yazılması ilə əlaqədardır. Belə bir yeni modulun yaranması nə qonşu yuvalardakı modulların mətnlərinə, nə də onu çağıran modula təsir etmir, yəni ağrısız keçir. Yuvaya görə qonşu olanlar hesablama proqramlarında heç vaxt qonşu olmurlar, amma çağıran modullar razılaşdırılmış karkas çərçivəsində proqramlaşdırılır və buna görə də istənilən yuvanın (həmçinin yeni daxil olmaqla) reallaşdırılmasına müraciət edə bilər.

Karkas yanaşmada, konkret konfigurasiyanı təyin etmək üçün, proqram karkasının hər yuvasına uyğun ailə yuvasından hansı modulun bu yuvaya yerləşdirilməli olduğunu göstərmək lazımdır. Bunun üçün paketin tapşırıq dilində aşağıdakı cütlüklərin məcmusu yazılır

$$\langle \text{yuvanın adı} \rangle = \langle \text{modulun adı} \rangle,$$

hansı ki, hesablama variantını tamamilə müəyyən edir. Karkas yanaşması ilə konfigurasiyanın tapşırığının sadələşdirilməsinin bəzi üsulları, istehsal paketlərini nəzərdən keçirərkən irəlidə müzakirə ediləcəkdir.

Çox az rast gəlinən predmet sahələri vardır ki, orada idarəetmə üzrə modulların əlaqələrinin müxtəlifliyini onları ardıcıl olaraq yerinə yetirmək yolu ilə məhdudlaşdırmaq olur və bu halda nə proqramın tərtib edilməsinin rahatlığına, nə də onun icrasının səmərəliliyinə xələl gəlmir. Burada hesablama proqramı bir-birinin ardınca yerinə yetirilən modullar zənciridir, buna görə də bu yanaşmanı *zəncirvari* adlandıracağıq.

Adətən bir zəncirdə olan modulların tərkibi və ardıcılığı hesablama planı adlanan bəzi xarici quruluşla (konstruksiya ilə) müəyyən edilir. Ancaq zəncirin

gedişində hər növbəti modulun özü dinamik olaraq bir **varisi** seçdiyi paketlər də var.

Zəncirdəki modulların hər biri istənilən alt proqramı çağıra bilər. Bununla birlikdə, zəncirvari yanaşma nöqteyi-nəzərindən zəncirvari bağlantıların daxili nəzarət strukturları əhəmiyyətli deyildir, yəni modullar tərəfindən çağırılan alt proqramlar simasızlaşdırılmışdır və zəncir xətti bir səviyyəli bir quruluş kimi qəbul edilir. Proqramların hesablama variantlarının formalaşması zamanı edilən bütün manipulyasiyalar ümumiyyətlə, eyni, ən yüksək səviyyədə çağırılan müxtəlif modul ardıcılıqları çərçivəsindən kənara çıxmır. Xatırladaq ki, karkaslı yanaşma zamanı hesablama variantını formalaşdıran modullar, idarəedici əlaqələrin iyerarxiyasının istənilən səviyyəsində yerləşə bilən proqramın karkasının yuvalarını doldurur.

Zəncirvari yanaşma ilə idarə etmələrə qoyulan belə bir ciddi məhdudiyyət, karkas yanaşmasından fərqli olaraq zəncirdə modulun mövqeyinin sabit olmaması ilə qismən kompensasiya olunur. İstənilən bir modul, ümumiyyətlə, zəncirin istənilən mövqeyinə daxil edilə bilər.

Zəncirdə modulların dəfələrlə təkrarlanacağını fərz etsək, onda hər hansı bir boş olmayan modul dəstindəki potensial zəncirvari yanaşmanın sonsuz sayda hesablama proqramı yaratmağa imkan verdiyini başa düşə bilərik.

Əyər (hətta) çox olmayan zəncirə modulun bir dəfə əlavə etməsiylə (qoşmasıyla) kifayətlənmək və zəncirdə modulların sırasını (nizamını) əhəmiyyətsiz hesab etmək (real paketlərdə tez-tez nə o, nə başqa məhdudiyyət yerinə yetirilmir), onda bununla belə R sayının qiyməti (qiymətləndirilməsi) potensial olaraq mümkün hesab proqramları təşkil edəcək

$$R = 2^M, \quad (2)$$

burada M - paketdə modulların sayıdır.

Bir modulun zəncirə daxil edilməsini bir dəfədən çox məhdudlaşdırsaq və zəncirdəki modulların sırasını əhəmiyyətsiz hesab etsək (real paketlərdə, çox vaxt bir və ya digər məhdudiyyət yerinə yetirilmir), buna baxmayaraq,

potensial olaraq mümkün hesablama proqramlarının P sayının qiymətləndirilməsi aparılacaqdır

$$P = 2m, \quad (2)$$

burada M - paketdəki modulların sayı. Hətta belə bir ucuzlaşdırılmış qiymətləndirmə karkas yanaşmasının (1) qiymətləndirməsindən daha təsirli görünür. Lakin, praktikada kifayət qədər böyük M ilə, potensial zəncirlərin böyük əksəriyyəti heç vaxt həyata keçirilmir.

Zəncirvari yanaşmanın populyarlığı, bir çox müasir əməliyyat sistemlərinin (ilk növbədə UNIX) məsələlər arasındakı əlaqələrin təşkilində zəncirvari yanaşmanı tətbiq etmək üçün rahat vasitə təmin etməsi ilə asanlaşır. Xüsusilə, əməliyyat sistemlərinin tapşırıqlarının idarəetmə dillərində bir sıra məsələlərin (proqramların) ardıcıl icrasına başlamış bir konstruksiya tapılır və hər bir əvvəlki məsələnin çıxış axını sonrakı üçün giriş axını kimi istifadə olunur. Bu konstruksiya, məsələn, çap olunmuş bir sənədin formalaşması üçün nəzərdə tutulmuş müxtəlif standart məlumat dəyişikliklərinin bir zəncirini kompakt və vizual şəkildə müəyyənləşdirməyə imkan verir.

UNIX sisteminin konsepsiyalarının təsiri altında yazılmış monoqrafiyada [77], hesablama riyaziyyatında məsələlərinin masına zəncirvari bir yanaşma tətbiq etməyə cəhd edilir. Müəllif bu yanaşmanı proqramların təkrar istifadəsinə ("yenidən istifadə") nail olmaq üçün vasitələrdən biri hesab edir. Bununla belə, hesablama məsələlərinin həlli üçün böyük proqram sistemlərinin əksəriyyəti təkrar istifadə olunmasını təmin edən bir vasitə rolunu iddia etmək üçün heç bir səbəb olmayan karkas yanaşmasına əsaslanır. Zəncirvari yanaşmanın nisbətən dar yayılması, açıqca, modulların idarəetmə əlaqələri tərəfindən qoyulmuş ciddi məhdudiyyətlərin, proqramların icrasında əhəmiyyətli proqramlaşdırma çətinliklərinə və bəzi səmərəliliyin itirilməsinə səbəb olması ilə izah olunur.

Eyni zamanda, bəziləri üçün əvvəlcədən zəncirvari yanaşma çox səmərəlidir. Zəncir sxemi natura və ya hesablama təcrübələrinin nəticələrinin emalını təşkil etmək üçün əlverişlidir. Burada fərdi modullar eksperimentatoru

maraqlandıran məlumatları ayıra bilər, onları müxtəlif yollarla çevirə və müxtəlif vizual cihazlara çıxara bilər. Nəhayət, zəncirvari yanaşma süni intellekt məsələlərində faydalıdır, məsələnin birbaşa proqramlaşdırılması nədənsə çətin və ya mümkün olmadıqda və paket funksiyaları paketdəki modullara və onların xüsusiyyətlərinə əsaslanaraq həll alqoritminin qurulmasını əhatə edir.

Zəncirvari yanaşma proqram fondunun zərərsiz inkişafını təmin edir. Əgər yenidən yazılmış modul, zəncirlərin formalaşdırılmasının aparatın tələblərinə cavab verərsə, onda o asanlıqla əvvəllər yazılmış modulların heyətinə qoşulacaqdır, çünki onun yaranması onların mətnlərinin dəyişdirilməsinə səbəb olmayacaqdır.

Zəncirvari yanaşmada paket tapşırıq dilinin ən sadə quruluşu zəncirdə olan modulları saymaqdır. Əgər formalaşdırılan zəncir kifayət qədər uzundursa və ya modulların açıq sayılması hansısa bir səbəblərdən narahatdırsa, sonrakı bölmədə müzakirə ediləcək tapşırıq dilinin daha inkişaf etmiş konstruksiyaları istifadə olunur. Paketlərdə modulların konfigurasiya sxemlərinin müxtəlifliyi karkaslı və zəncirvari yanaşmalarla məhdudlaşmadığını yalnız qeyd etmək qalır. Digər sxemlərdə mümkündür, bundan başqa, qarışıq karkas-zəncir formaları istifadə olunur. Lakin baxılmış iki yanaşma daha maraqlıdır və buna görə onlar fərqli sahələrin ətraflı təhlili üçün seçilmişdirlər, xüsusi halda elektrotexniki hesablamaları üçün zəncirvari yanaşmanın tətbiqi çox səmərəlidir. Zəncirvari sxem təbii və ya hesablama təcrübələrinin nəticələrinin emalını təşkil etmək üçün əlverişlidir. Burada ayrı-ayrı modullar eksperimentatoru maraqlandıran məlumatları ayıra bilər, onları müxtəlif yollarla çevirə və müxtəlif cihazlarda vizuallaşdırı bilər.

1.4. Hesablamaların planlanması və informasiya təminatı

Yuxarıda qeyd etdiyimiz kimi, ən sadə halda tapşırıqların dili modulun hesablama zəncirinin bütün tərkib elementlərini lazımi ardıcılıqla sadalamağa imkan verir. Əgər bu konstruksiyanın yerinə yetirilməsində makrogenerasiya

texnikasından istifadə olunarsa, onda tapşırıqların dilinin leksika və sintaksisinin xidmət olunan sahə üçün səciyyəvi olan formalara yaxınlaşması baş verir.

Bəzi paketlərdə tapşırıqların dilinin operatorlarının semantikasi qismən nizamlanmış modullar çoxluğu vasitəsilə müəyyən olunur. Məsələn, AED, CAП -2 və АИПОКC sistemlərində tapşırıqların dilinin hər bir operatoruna tabelik ağacı uyğun gəlir və onların təpəsində funksional modullar yerləşir ki, bunlar da ağacla müəyyən olunmuş birləşmələrdə bu operatorun müxtəlif konkretləşdirmələrini həyata keçirir. Hesablama zənciri tərtib edilərkən tapşırıqların təkcə mətni deyil, həm də emal edilən operatorun tabelik ağacı da təhlil edilir və tabelik ağacının operatorun verilən konkretləşməsinə aid olan fraqmenti zəncirin bəndi qismində daxil edilir. Bu halda karkas və zəncirvari yanaşmadan söhbət gedir. Lakin zəncirvari yanaşmadan söhbət gedirsə, yuxarıda göstərilən dil konstruksiyaları deyil, bu yanaşmaya əsaslanan paketlərdə hesablamaların avtomatlaşdırılmış planlanması imkanları xüsusi maraq kəsb edir. Avtomatlaşdırılmış planlamada tapşırıqların dilində modullar zənciri deyil, yalnız istifadəçini maraqlandıran məsələnin qeyri-prosedur qoyuluşu yazılır. Bu cür qeyri-prosedur qoyuluşa əsasən paketin sistem doldurulması məsələnin həlli üçün zəruri olan modullar zəncirini avtomatlaşmış şəkildə qurur və icra edir.

Tapşırıqın qeyri-prosedur formasının üstünlüklərini yada salaq. Hər şeydən öncə, bu, tapşırığı daha əyani və daha kompakt şəkildə formalaşdırmağa imkan verir.

Paketdən istifadə edənlərin dairəsi əhəmiyyətli dərəcədə genişlənir, belə ki, bu halda onlardan məsələnin həllində hər hansı proqramçı bilikləri tələb olunmur. Eyni zamanda çox vaxt istifadəçi tələb olunan zənciri özü müstəqil şəkildə layihələndirə bilər, lakin mexaniki səhvlərə yol verməmək, yaxud da çox vaxt sərf etməmək məqsədilə o, avtomatlaşmış planlamadan istifadə edir.

Hesablamaların avtomatlaşmış planlanması paketin funksional baxımdan müəyyən dərəcədə mürəkkəbləşdirilməsini tələb edir. Bu halda paketdə modulların təkcə proqramları deyil, həm də onların semantikasi haqqında biliklər, eləgə də predmet sahəsinin ümumi qanunauyğunluqları haqqında biliklər yerləşdirilir. Bu

biliklər həm deklarativ formada, yəni təsdiqləmə sistemi və nəticə qaydaları sistemi şəklində, həm də prosedur formada, yəni proqramlar çoxluğu kimi təqdim oluna bilər.

Ekspert planlama. Planlayıcı iyerarxik altsistem kimi reallaşır, hansı ki, planlaşdırma proqram-administratorundan və ekspert adlanan genişləndirilmiş xüsusi proqram dəstindən ibarət olur. Hər bir ekspert konkret tiptən olan tapşırıqlar üzrə hesablamalar planının tərtib edilməsinə yönəldilmişdir, başqa sözlə desək, hər bir ekspert paketin predmet sahəsindən götürülmüş məsələlərlə bağlıdır. Planlayıcının işi sintaksis baxımından idarə edilən translyasiyadır, tapşırığın mətnini administrator ayırd edir, lakin seçilmiş sintaksis konstrukusiyaların semantik emalını ekspertlər həyata keçirir ki, bu da hesablamalar planının qurulmasına gətirib çıxarır.

Bu halda hesablama zəncirinin formalaşdırılması prosesini aşağıdakı şəkildə baş verir: planlama administratoru tapşırığın mətnini təhlil edərək növbəti planlaşdırılan hesablama kontekstini ayırd edir və hansı ekspertin bu konteksti emal edəcəyini müəyyən edir. Ekspert paketin informasiya bazasında yerləşən məlumatları və məsələnin tapşırıqda göstərilən parametrlərindən istifadə edərək tapşırığın göstərilən kontekstinə müvafiq hesablama zəncirinin fraqmentini qurur, ya da MEI WEP paketində olduğu kimi, konteksti məzmunca dəyişdirə bilər, yəni formalaşdırılmış məsələni altməsələlərin supermövqeyi şəklində formalaşdırma bilər ki, bu da öz növbəsində müvafiq şəkildə emal oluna bilər. Öz funksiyaların icra etdikdən sonra ekspert idarəetməni administratora ötürür.

Ekspert planlamada paketin informasiya bazasında bu və ya digər məsələnin mümkün həlləri barədə və onun mümkün parametrləri haqqında məlumatlar, alqoritmin ilkin verilənlər üzrə tətbiq olunması meyarları, alqoritmin hesablama sxemləri və s. saxlanır. Məsələnin parametrləri qismində matris və xətti cəbri tənliklər sisteminin sağ tərəfi, onların elementlərinin dəqiq tapşırıqları, axtarılan həllin tələb olunan dəqiqliyi, matrisin bəzi topoloji xassələri və s. göstərilə bilər.

Qeyd etmək lazımdır ki, ekspert planlamanın təşkili üçün məsələnin həll qaydalarının qurulmasını və alqoritmlər sxeminin seçilməsini formalaşdırmaq və

bunun əsasında məsələnin qoyuluşuna görə müvafiq hesablama zəncirini axıra qədər tərtib edə biləcək ekspertləri hazırlamaq tələb olunur.

Hesablama modellərinə görə planlama. Paket problemi nəzəriyyəsinə və praktikasında hesablama modellərinin istifadəsinə əsaslanan hesablama planlaması metodları daha çox diqqət çəkir.

Hesablama modeli dəyişənlər və münasibətlər çoxluğundan ibarət obyektidir ki, bu dəyişənləri birləşdirir. Hesablama modelinin hər bir dəyişəni paketin predmet sahəsindən olan məzmun anlayışına uyğun gəlir və müəyyən ada və şəkllə malik olur. Münasibət dəyişənlər arasında əlaqəni əks etdirir və digər dəyişənlərin məlum qiymətlərinə görə onların bəzisi hesablanı bilər. hər bir əlaqənin (münasibətin) semantikasi paketin funksional dolma modulu ilə müəyyən olunur. Hesablama modeli funksional modullar arasında informasiya və məntiqi əlaqəni müəyyən edən qraf şəklində, yaxud da modulların tətbiqi və uzlaşması barədə təqdiqləmələr məcmusu şəklində verilə bilər.

Planlamanın göstərilən texnikasını reallaşdıran paketlərin işlənilməsində predmet sahəsinin təsvirinin mürəkkəbliyindən asılı olaraq paketin informasiya bazasına ya predmet sahəsinin hesablama modelini, yaxud da məsələlərin hesablama modelləri dəstini daxil edirlər. Birinci yanaşma predmet sahəsi nisbətən dayanıqlı struktura malik olduğu və vahid hesablama modeli ilə təsvir oluna bildiyi halda tətbiq edilir ki, bu halda ilkin verilənlərin müxtəlif variantlarından istifadə etməklə paketin bütün məsələləri həll edilir. Bu yanaşmadan iri enerji qurğularının, texnoloji proseslərin layihələndirilməsi zamanı aparılan hesablamalara yönəldilmiş paketlərdə istifadə olunur.

İkinci yanaşmadan o halda istifadə olunur ki, həll edilən məsələlər sinfi o qədər müxtəlifdir ki, onu bir hesablama modeli ilə əks etdirmək mümkün olmur, ya da bu məsələlərin vahid model ilə həlli qeyri-mümkün vaxt xərcləri ilə bağlı olur. Hər bir məsələ üçün ayrıca hesablama modelinin qurulması planlama prosesində təhlil edilən verilənlərin həcmi azaltmaq imkanı yaradır ki, bu da planlayıcının hərəkətlərinin sürətini əhəmiyyətli dərəcədə artırır. Bundan başqa, biliklərin təqdim edilməsinin vahid modeli ilə müqayisədə bu cür çevik model modellərin

cəbri əsasında yeni modellər (daha mürəkkəb məsələlər üçün (sintez etməyə imkan yaradır.

« $W_1, W_2, \dots, W_r$ -in verilən qiymətlərinə görə $V_1, V_2, \dots, V_k$ -ni hesablayın» şəklində məsələ üçün hesablama modeli əsasında hesablama zənciri müxtəlif üsullarla yerinə yetirilə bilər, burada bütün kəmiyyətlər predmet sahəsinə aiddir.

Onlardan birincisi dalğa metodu adlanır. Bu metod ondan ibarətdir ki, ilkin olaraq plana modullardan biri daxil edilir və bu modulun giriş qiymətləri əvvəlcədən verilən kəmiyyətlər çoxluğuna aiddir. Modul hesablama zəncirinə daxil edildikdə onun hesabladığı kəmiyyətlər əvvəlcədən verilənlər çoxluğuna əlavə edilir və axtarılanlar çoxluğundan çıxarılır. Daha sonra proses o vaxta qədər davam edir ki, axtarılan kəmiyyətlər çoxluğu bitəcək və yaxud da aydın olacaqdır ki, daha hesablanacaq bir kəmiyyət qalmamışdır.

Digər metod əks dalğa metodu adlanır və axtarılan kəmiyyətlərdən verilən kəmiyyətlərə doğru hesablamlar planının formalaşdırılmasını təmin edir, yəni hesablamlar zəncirinin formalaşdırılması sağdan sola doğru aparılır. Əvvəlcə modeldən giriş kəmiyyətləri axtarılan kəmiyyətlər çoxluğuna daxil olan modullar seçilib modullar zəncirinə daxil edilir. Sonra isə plana daxil edilmiş və verilənlər çoxluğuna aid olmayan modulların tətbiq üçün zəruri olan kəmiyyətlər axtarılan kəmiyyətlər sırasına daxil edilir. Bu modullarla hesablanan kəmiyyətlər axtarılan kəmiyyətlər sırasından çıxarılır. Proses axtarılan kəmiyyətlər çoxluğu bitənə qədər davam edir. Bu modellərlə iş təcrübəsi göstərir ki, ən yaxşı nəticə hər iki modeldən istifadə zamanı əldə edilir.

Semantik şəbəkədə planlama. Bu yanaşma hesablamların avtomatlaşdırılmış planlanması üçün süni intellekt metodlarından istifadəni nəzərdə tutur. İlk növbədə burada biliklərin təqdim olunması sistemləri və hallar fəzasında həllərin axtarılması metodları nəzərdə tutulur. Planlayıcının istifadə etdiyi semantik şəbəkə biliklərin 3 bölməsini əhatə edir: predmet sahəsi; hesablama metodları; hesablama metodlarının tətbiqi və bu metodların reallaşması.

Böyük verilənlər bazasının məntiqi təhlilini aparmaq qabiliyyətinə malik olan və iş zamanı prosedur planlaşdırılmasını tətbiq etməyə imkan verən deduktiv

informasiya sistemlərindən istifadə olunması yuxarıla göstərilən bilikləri 3 şəkildə toplamağa imkan yaradır: faktlar şəklində, nəticə qaydaları və prosedurlar şəklində. Məsələn, predmet sahəsi haqqında biliklər paketin informasiya bazasında obyekt haqqında təsdiqləmələr (fikirlər) və nəticə çıxarmaq qaydaları dəsti şəklində göstərilə bilər ki, çox vaxt bu qaydalar predmet sahəsinin obyektləri arasında qarşılıqlı əlaqəni müəyyən edir. Digər tərəfdən predmet sahəsində tətbiq edilən hesablama metodları haqqında biliklər prosedur şəklində, yəni funksional modullar şəklində də verilə bilər. Nəhayət, hesablama metodların tətbiqi və bu metodların reallaşması haqqında biliklər faktlar və nəticə qaydaları sistemi şəklində verilə bilər ki, bunlar da modulların problem və sistem semantikasını təsvir edir.

Semantik şəbəkədə mövcud olan informasiyanın çoxaspektliliyi planlama prosesinin özünü keyfiyyətə təkmilləşdirməyə imkan yaradır. Semantik şəbəkədə planlamanın aşağıdakı üstünlükləri daha çox əhəmiyyətlidir. Modulun problem semantikasi bu modulun baxılan sahə çərçivəsində aparılan hesablamalarda necə istifadə edilə bilməsi haqqında məlumatdan ibarətdir. Mühüm olan odur ki, problem semantikasının təsvirində modulun parametrlərinin xarici identifikasiyası adlar vasitəsilə deyil, faktlar vasitəsilə aparılır və bu faktlar predmet sahəsinin obyektlərinin xassələrini təsvir edir ki, onları həmin parametrlər modelləşdirir. Modulun problem semantikasının bu şəkildə təqdim olunması yuxarıda müzakirə olunan yanaşmalarla müqayisədə planlamanın daha çox çevikliyi təmin etməyə imkan yaradır, göstərilən yanaşmalarda isə hesablama zəncirini əmələ gətirən $\subseteq \subseteq$ modulların bir-birinə bağlanması onların giriş və çıxışlarının adlarına əsaslanırdı.

Modulların sistem semantikasi modul haqqında onu bir prosedur kimi əks etdirən məlumatdan, yəni müxtəlif proqram parametrləri və istismar xarakteristikaları haqqında məlumatdan ibarətdir, planlayıcıya modulun sistem semantikasına çıxışı hesablamaların sadəcə planlanmasını deyil, həm də optimallaşdırma meyarlarını nəzərə almaqla optimal planlamayı aparmağa imkan verir. Xüsusən də, modulların sistem semantikasının təsvirindən istifadə etməklə

planlayıcı hesablama zəncirləri tərtib edə bilər ki, onların icrası minimal zaman alar, yaxud da ölçülərinə görə yaddaşda minimal sahəni tutur. Semantik şəbəkədən istifadə etməklə optimal hesablama zəncirinin qurulması məsələsini hallar fəzasında minimal yolun axtarışı məsələsi kimi formalaşdırmaq və həll etmək olar.

S halı dedikdə funksional dolma modulunun məlum parametrləri (yəni verilmiş, yaxud da hesablanmış) çoxluğu başa düşəcəyik. Hər bir m modulu semantik şəbəkədə ad, giriş parametrləri çoxluğu $in(m)$, çıxış parametrləri çoxluğu $out(m)$ və qiymətlə $price(m)$ xarakterizə olunur. Əgər $in(m) \subseteq S$ olarsa, onda modul S halına tətbiq edilə bilər. m modulunun S halına tətbiqi S' halını yaradır $S' = S \cup out(m)$. S_0 halından S_k halına keçid funksional dolma modulları zənciri $(m_1, m_2, \dots, m_k)$ adlanır, onların S_0 halına ardıcıl tətbiqi $(S_1, S_2, \dots, S_k)$ hallar ardıcılığını doğurur, həm də modullar zəncirinin hər bir bəndi $in(m_i) \subseteq S_{i-1}$ şərtini ödəyir. Zəncirə daxil olan modulların qiymətlərinin cəmi S_0 halından S_k halına gedən yolun qiyməti adlanır. Planlama alqoritminin vəzifəsi verilmiş ilkin S_{beg} halına və S_{end} məqsəd halına görə müəyyən S' vəziyyətinə minimum qiymətə keçidin qurulmasıdır, belə ki, $S \supseteq S_{end}$ olsun. Bu məsələ hallar fəzasında seçib ayırma metodları və yaxud da bu metodların modifikasiyası ilə həll edilə bilər.

Modulun qiyməti qismində onun bu və ya digər atributundan (icra müddəti, tələb olunan operativ yaddaş və s.) istifadə etməklə planlayıcının bu və ya digər parametr üzrə tərtib etdiyi proqramı optimallaşdırmaq olar.

Planlayıcının istifadəçi tapşırığını reallaşdıran hesablama prosesinin hansı dövrdə işləməsindən asılı olaraq statik və dinamik planlamayı fərqləndirirlər.

Statik planlama hesablama prosesinin icrasına qədər həyata keçirilir və əksər paketlərdə giriş informasiyasının emalı mərhələlərindən biri və işçi proqramın generasiyası kimi həyata keçirilir. Lakin bir sıra hallarda hesablama prosesini səmərəli həyata keçirmək üçün təkcə statik planlama kifayət etmir. Onda statik

planlama hesablama prosesində dinamik planlama ilə tamamlanır, yaxud əvəz olunur. Bir qayda olaraq, dinamik planlama internet texnologiyasına əsaslanır.

Hesablamaların planlanmasının nəzərdə keçirdiyimiz təşkili metodları təkmil paketlərin planlayıcılarında rast gəlinən çoxsaylı həllərin müxtəlifliyini əlbəttə ki, tam olaraq əks etdirmir. Müxtəlif planlama metodlarının üstünlüklərini özündə birləşdirən sxemlər (xüsusən də ekspertlərdən istifadə etməklə semantik şəbəkədə planlama) ı xüsusən də hesablama zəncirinin dialoq formasında qurulması (bu halda hesablamaların istifadəçi tərəfindən verilən ilkin planı dialoqun gedişatında paketin informasiya bazasında yerləşdirilmiş biliklər əsasında müvafiq şəkildə planlayıcı tərəfindən dəqiqləşdirilir) olduqca perspektivli görünür.

Paketin sistem doldurulması ilə təqdim olunan təşkilati, idarəedici və xidmət (servis) funksiyalarının məcmusu onun əməliyyat imkanları adlanır. Bu funksiya bir tərəfdən, proqram təminatının bütün alətlərinin paketin marağında istifadə etməli, digər tərəfdən isə bu təminatın rutin mexanizmlərini istifadəçidən gizlətməli və yaxud da heç olmasa istifadəçinin bu mexanizmlərə çıxışını sadələşdirməlidir. Nəticədə xüsusi əməliyyat mühiti formalaşır ki, verilən əlavədə hesablamaların aparılması üçün əlverişli şərait təmin olunur. Paketin əməliyyat imkanlarının reallaşması zamanı meydana gələn bəzi məsələlər ənənəvi məsələlərdir, ona görə ki, məxsusi giriş dilinə malik olan hər bir proqram sisteminin işlənilib hazırlanması zamanı rast gəlinən məsələlərdir; digər məsələlər isə məhz proqram paketləri üçün səciyyəvidir. Bu məsələlərdən daha vacib olanlarını qısaca şərh edək və onların həllərinin tətbiq üsullarını nəzərdən keçirək.

Modullararası interfeys. Modullar arasında informasiya (verilənlər üzrə interfeys) və idarəetmə (idarəetmə üzrə interfeys) əlavələrinin təşkili modular arasılı interfeys adlanır. İnterfeys verilənlər və idarəetmə üzrə modulların bir-birinə birləşdirilməsi dəstəyinin sistem vasitələri və funksional doldurma ilə modullaşdırılması üzrə qəbul edilmiş reqlament ilə müəyyən olunur.

Problemyönümlü modul sistemlərin və proqram paketlərinin işlənilib hazırlanması təcrübəsinin təhlili verilənlər üzrə modellərarası interfeysin 3 formasını ayırmağa imkan verir. Sistemin reallaşması baxımından ən sadə

interfeys sərt interfeys hesab olunur. Bu interfeysdə modullar reqlamentə uyğun şəkildə proqramlaşdırılır, hər bir model üçün emal olunan verilənlərin strukturu, olduğu yer və digər xarakteristikaları qeydə alınır. Bununla da müxtəlif hesablama konfigurasiyalarında uzlaşa bilən modulların informasiya və proqram uyğunluğu tam təmin edilir. Sərt interfeysin sistem dəstəyi modullar arasındakı əlaqənin redaktə vasitələri ilə həyata keçirilir.

Yarım sərt interfeysdə modulların proqram hissəsinin işlənilib hazırlanması da həmçinin müəyyən reqlamentə uyğun həyata keçirilir və bu reqlamentə görə verilənlərin mümkün strukturları və saxlanma üsulları məcmusu müəyyən olunur. Bununla yanaşı hər bir modula qeyd olunan mümkün strukturlar və saxlanma üsulları məcmusu çərçivəsində seçim sərbəstliyi verilir. Yarım sərt interfeysin sistem dəstəyi verilənlər bazasının idarə edilməsi sistemi tərəfindən yaxud da paketdə sistem tərəfindən doldurulmuş tərkibdə olan verilənlərin xüsusi saxlanması sistemi tərəfindən təmin edilir. Bu cür sistemlər dil və proqram vasitələri təqdim edir ki, onların köməyi ilə interfeys prosedurları müəyyən olunur (modulların proqram hissəsində) və reallaşır.

Qeyd edək ki, həm sərt, həm də yarım sərt interfeys halında proqram materialının adaptasiyası daxili modullaşma yolu ilə həyata keçirilir, yəni modullaşma reqlamentinin bütün bəndlərinə (müddəalarına) modulların proqram hissələrinin işlənilib hazırlanması mərhələsində riayət olunmalıdır. Çevik interfeys proqramlaşdırılmış modullardan asılı olmayaraq verilənlər üzrə birləşməni həyata keçirməyə imkan verir. Çevik interfeysin sistem dəstəyi yarım sərt interfeysdə olduğu kimi verilənlərin idarə olunması üzrə ştat sistem yaxud da verilənlərin saxlanması və çevrilməsi üzrə xüsusi sistem tərəfindən həyata keçirilir. Lakin çevik interfeysə malik olan sistemin prinsipial fərqi ondan ibarətdir ki, o, xarici modullaşmaya imkan verir, yəni proqram materialının sistemə adaptasiyasına proqramın tərtib edildiyi vaxtda deyil, onların funksional dolma zamanı paketə daxil edilməsi mərhələsində sistemə adaptasiya olmasına imkan verir. Bu isə informasiya üzrə əhəmiyyətli dərəcədə uzlaşmayan proqramları paketə daxil etməklə onun funksional dolmasını sürətlə təkmilləşdirməyə imkan yaradır.

İdarəetmə üzrə modullararası interfeysin təşkili zamanı modulların birbaşa və dolayı kommutasiyası sxemi, həmçinin bu sxemlərin birləşməsi tətbiq olunur.

Birbaşa kommutasiya halında idarənin çağırılan modula ötürülməsi çağıran modul hissəsindən birbaşa həyata keçirilir. İdarəetmənin bu cür sxem üzrə ötürülməsi, adətən, əlaqələrin redaktə vasitələri (ştat üzrə) ilə təmin edilir və paket tərəfindən proqramlarda yaradılan idarəetmə strukturları kifayət qədər dayanıqlı olduğu və moduldan modula idarəetmənin ötürülməsi zamanı vaxt sərfinin aradan qaldırılmasının zəruri olduğu hallarda geniş tətbiq edilir. Birbaşa kommutasiya adətən, modulların yaradılmasına karkas yanaşmada tətbiq edilir.

Dolayı kommutasiya sxemi zəncirvari yanaşma üçün xarakterikdir. Bu sxemdə modullar arasında idarəetmə əlaqələri müəyyən vasitəçi proqramların iştirakı ilə qurulur. Bu cür vasitəçi qismində paketdə nəzərdə tutulmuş və ya xüsusi dinamik yükləyici və ya paket monitoru (idarəedici proqram) çıxış edə bilər. Dolayı kommutasiya modullararası idarəedici əlaqənin qeyd oluna bilmədiyi (belə ki, onlar hesablama prosesinin gedişatında dinamik olaraq meydana gəlir) hallarda tətbiq edilir.

Bir sıra paketlərdə hər 2 kommutasiya sxeminin üstünlüklərini özündə birləşdirən yanaşma tətbiq olunur. Bu halda idarəedici əlaqə öncə dolayı kommutasiya sxem üzrə qurulur, sonra birbaşa kommutasiya sxemi üzrə dəstəklənir (yaddaşın əlçatan resursları imkan verənə qədər). Bu yanaşmanın orijinal reallaşması İS -2 sistemində həyata keçirilmişdir.

Hesablama zəncirinin icrası. Planlayıcının tərtib etdiyi hesablama zənciri modullarının icrasının təşkili üçün kompilyasiya, yaxud da interpretasiya texnikasından istifadə olunur.

Birinci halda proqramın yaradılması və hesablamaların aparılması prosesi icra vaxtına görə ciddi şəkildə bölünür: əvvəlcə verilmiş hesablama zənciri üzrə paketin baza dilində olan translyator (çevirici) üçün tapşırıq formalaşdırılır və tam işçi proqram yaradılır, sonra isə bu proqram üzrə bir və ya bir neçə hesablama icra olunur. Kompilyasiya rejimindən istifadənin 2 qeyd-şərtsiz üstünlüyü vardır. Birincisi, məlum olduğu kimi, kompilyasiya texnikası proqramın zamana görə

səmərəli icrasına imkan yaradır, ikincisi, bu proqramlar paketindən müstəqil şəkildə də istifadə oluna bilər. Bununla əlaqədar olaraq kompilyasiya texnikası bir qayda olaraq uzun və külli miqdarda hesablamaların təmin edilməsinə yönəldilmiş paketlərdə tətbiq olunur.

İnterpretasiya rejimində modulların istifadəsi zamanı işçi proqramın yaradılması və hesablama prosesləri monitorunun ümumi idarəsi ilə kvazi paralel icra olunur. Paketdə interpretasiya texnikasını reallaşdırmaq kompilyasiya ilə müqayisədə, ümumiyyətlə desək, olduqca sadədir. İnterpretasiya çox vaxt hesablamaların icrası zamanı paket tərəfindən istifadə edilən operativ yaddaş həcmi azaltmağa imkan verir. Nəhayət, məsələnin həlli dialoq formasında aparıldıqda hesablama zəncirinin interpretasiya rejimi çox rahatdır.

Bəzi hallarda hesablama zəncirinin icrası zamanı kompilyasiya və interpretasiya texnikalarını birləşdirmək daha məqsəduyğun olur. Bu halda hesablama zəncirinin icra vaxtı baxımında daha kritik olan hissələri kompilyasiya olunur və artıq bütün hesablama zənciri deyil, qismən yeni yaradılmış proqram interpretasiya olunur. Bu cür yanaşma tapşırıqın icra müddəti və tələb olunan operativ yaddaş resursları arasında əlverişli kompromisə nail olmaq imkanını təmin edir.

İnformasiyanın saxlanması. Tətbiqi proqram paketləri vasitəsilə saxlanan və emal edilən informasiya olduqca müxtəlifdir: proqram materialı, hesablamaların ilkin verilənləri, hesablamaların aralıq verilənləri (nəzarət nöqtələrinin qeydiyyatı), hesablamaların nəticələri, sənədlər.

Bu informasiyanın saxlanması sistem dəstəyi vasitələri arasında ənənəvi olaraq 2 böyük bölməni ayırır: proqram materialının saxlanması və hesablama verilənlərinin saxlanması.

Proqram materialının saxlanması üçümi məqsəd daşıyan redaktorlar, translyatorlar və yükləyicilər tərəfindən istifadə olunan vasitələr əsasında təşkil edilə bilər. Lakin əgər işçi proqramın yaradılması zamanı proqram mətninin yaradılması üzrə mükəmməl alqoritmdən istifadə olunarsa, onda xüsusi saxlama vasitələrinin reallaşması zərurəti yaranır.

Hesablama məlumatlarının saxlanması təşkili üçün adətən, fayl sistemlərinin imkanlarından istifadə olunur. Bu imkanlar verilənlərin operativ yaddaşa xüsusi vasitələrlə buferə yerləşdirilməsi ilə tamamlana bilər və bu vasitələr əlavədə aparılan hesablama proseslərinin səciyyəsinə nəzərə alır və bununla da vaxt itkisini (sərfini) əhəmiyyətli dərəcədə azaltmağa imkan yaradır.

Bununla yanaşı bir sıra əlavələrdə məlumatların və proqramların qarşı-qarşıya qoyulması çox süni şəkildir, bundan əlavə, işlərin səmərəli təşkilinə maneə törədir.

Çox vaxt seriyalarla hesablamlar apararkən hər bir hesablama variantında təkcə ilkin məlumatların bir hissəsinin deyil, həm də hesablama alqoritminin müəyyən fraqmentlərinin dəyişdirilməsi tələb olunur. Bu halda tələb olunan variantın formalaşdırılması üçün ənənəvi yanaşma zamanı 2 müxtəlif saxlama sisteminə müraciət etmək lazım gəlir ki, bu da müəyyən narahatlığa səbəb olur.

Oxşar çətinliklər proqram materialı kimi saxlanan müəyyən proqram variantını bu proqram üzrə hesablama nəticələri ilə (hansı ki, verilən olaraq saxlanır) əlaqələndirməyə cəhd edilərkən də meydana çıxır. Bununla da proqramın işlənilib hazırlanması və düzəldilməsinə dəstək üzrə inkremental yanaşma kimi müasir metodların tətbiqində maneələr meydana gəlir.

Nəhayət, proqram sənədlərinin müstəqil saxlanması çoxlu sayda nöqsanlara səbəb olur. Əgər bu müstəqilliyi aradan qaldırmaq mümkün olarsa, onda proqram və verilənlərdən qoşma sənədlərə və əksinə müraciətlərdən geniş istifadə etmək olardı. Göstərilən arqumentlər onu göstərir ki, bütün informasiyanın saxlanması təşkili üçün vahid sistem tətbiq edilməlidir. Bu cür sistemin istifadəsi yuxarıda göstərilən çətinliklərin aradan qaldırılması ilə yanaşı, müxtəlif cinsli informasiyaları çıxışı vahid şəkildə gətirməyə və bütün cəhdləri bir saxlama sistemində cəmləməklə onun imkanlarının zənginləşdirməyə imkan yaradır. Qeyd etmək lazımdır ki, bu cür sistemlərin işlənilib hazırlanması tətbiqi proqram paketlərinin səciyyəsinə görə çox vaxt verilənlər bazasının idarə edilməsi sistemlərinin ənənəvi problemləri çərçivəsindən kənara çıxır.

İstifadəçinin paketlə qarşılıqlı əlaqəsi. Proqram təminatı, adətən, istifadəçinin paketlə əlaqəsinin təşkili üçün kifayət qədər geniş vasitələr dəstindən (tapşırığın idarə edilməsi üçün dil operatorundan və əməliyyat sistemi direktivlərindən tutmuş dialoq sisteminə qədər) ibarətdir. Bu, lazım gəldikdə paketin müxtəlif iş mərhələlərinin məzmununu əks etdirən bir deyil, bir neçə əlaqə formalarını, başlıcası isə, hər mərhələdə istifadəçinin müxtəlif aktivlik səviyyəsini asanlıqla təmin etmək mümkün olur.

Proqramlar paketində istifadəçi ilə aşağıdakı əlaqə formaları, başqa sözlə desək, tapşırıqların icra rejimləri həyata keçirilir. İstifadəçinin paketlə ən az fəal qarşılıqlı əlaqəsi tapşırığın qapalı emalı zamanı baş verir. Bu rejimdə istifadəçinin həm tapşırığın hazırlıq mərhələsində, həm də onun icra mərhələsində paketlə birbaşa əlaqəsi olmur və bütün qarşılıqlı əlaqələr tapşırıq hesablama sistemindən uğurla keçdikdən sonra istifadəçinin paketdən aldığı məlumatlarda və ya listinqlərdə ifadə olunur. Bəzi paketlər tapşırığın hazırlıq mərhələsində istifadə olunan müəyyən dialoq mühiti təqdim edirlər, lakin onun köməyi ilə formalaşdırılan tapşırıqlar qapalı rejimdə icra olunurlar. Qapalı rejimdə emal uzun sürmə müddətinə malik olan işçi proqramlar (fon proqramları) generassiya edən proqram paketlərdə tətbiq olunur. Məhdud sayda sadə məsələlər dəstinə xidmət üçün təyin edilmiş paketlərdə çox vaxt «sorğu-cavab» rejimindən istifadə olunur. Bu rejimdə terminalın arxasında olan istifadəçi formalaşdırdığı tapşırıqlar (sorğu) hesablamaların nəticələrini operativ şəkildə lədə edir. Bəzi paketlərdə üzrə «sorğu-cavab» rejimində çox da mürəkkəb olmayan cəbri hesablamalar aparmaq, paketin imkanları və yaxud da onun cari vəziyyəti ilə tanış olmaq olar.

Daha təkmil dialoq formaları da tətbiq olunur və bu dialoq istifadəçiyə və paketə bütün fəaliyyət mərhələlərində görülməli işi istiqamətləndirməyə və dəqiqləşdirməyə yönəldilmiş mütəmadi informasiya mübadiləsinə imkan verir. Bəzi hallarda istifadəçiyə müəyyən həll qovşaqlarında hesablama prosesinin sonrakı istiqamətini seçməyə imkan yaratmaqla hesablama proqramının icrasının gedişatını idarə etmək vasitələrini təqdim etmək olar.

Servis. Hər bir təkmil proqram sisteminin servis vasitələri kimi paketin sistem tərəfindən doldurulmasının servis komponentləri istifadəçini paketin istismarında, təkmilləşdirilməsi və müşayiət olunmasında meydana çıxan məsələlərin operativ həlli üçün alətlərlə təmin edir. Ən geniş yayılmış servis imkanlarından biri paketin funksional dolma modullarının ilkin mətnlərinin redaktəsidir. Bu xidmət hesablamaların aktiv rejimdə aparıldığı zaman xüsusilə zəruridir, çünki istifadəçi proqramlar kompleksinin yeni versiyasını paket vasitəsilə yığılmasını icra edərkən bu kompleksə daxil olan bir sıra modulları öncədən modifikasiya etməlidir. Redaktə etmək üçün ya universal mövcud redaktorlardan istifadə olunur, yaxud da paketin baza dilinə yönəldilmiş və paket üçün tapşırıq mühitindən kənara çıxmadan modulları redaktə etməyə imkan verən xüsusi vasitələr yaradılır.

Servis imkanlarının digər mühüm bölməsi sazlama funksiyalarını və istifadəçi tapşırığının icrası prosesini müşahidə funksiyalarını əhatə edir. Buraya səhvlərin diaqnostikası və səhvləri lokallaşdırma məlumatların verilməsi vasitələri, işçi proqramın gedışatını izləyə və onun idarəedici parametrlərini çap edən vasitələr daxildir.

Sistem və funksional dolma zamanı intensiv inkişaf etməkdə olan proqram paketləri ilə işləyərkən istifadəçi sənədlərində proqramlara daxil edilən bütün dəyişikliklərin daim izlənməsi məsələsi olrtaya çıxır. Təkmilləşən paketlər üçün sürətlə köhnələn çap təlimatlarının buraxılışını operativ sənədləşmənin servis vasitələri əvəz edir. Bu vasitələr birinci növbədə, paketin cari vəziyyətini əks etdirən çapa sənədini istənilən anda əldə etməyə imkan verir, ikincisi, mövcud sənədlərin mətninə dəyişikliklərin sadə və asan yolla daxil edilməsi imkanlarını təmin edir. Bundan əlavə, cari sənədlərin çapa hazırlanması prosesini sadələşdirir.

Yüksək səviyyədə avtomatlaşdırılmış tətbiqi fəaliyyətə nail olmaq üçün informasiya xidməti vasitələrinə xüsusi diqqət yetirilməlidir. Paketi hazırlayanların böyük əksəriyyəti tərəfindən bu şərt qəbul edilsə də, paket problemləri sahəsində informasiya xidmətləri kifayət qədər işlənməmişdir. Vəziyyətin bu cür olması, görünür onunla izah olunur ki, paketləri hazırlayanları universal proqram təminatından istifadə formaları qane edir. Lakin xüsusi proqram sistemləri həm

daxili təşkilinə və həm də təyinatına görə proqram təminatının digər ümumi məqsəd daşıyan komponentlərindən əhəmiyyətli dərəcədə fərqlənir və işlərin yerinə yetirilməsinin müxtəlif forma və üsullarını əmələ gətirirlər. Sənədləşməyə gəlincə, baxmayaraq ki, bu baxımdan paketlərə qarşı yüksək tələblər irəli sürülür, sənədlərin çoxluğu paketdən istifadəni və sadəliyi təmin etmir, bundan əlavə, yuxarıda qeyd etdiyimiz kimi, təkmil paketlərdə çap olunan sənədlər praktiki olaraq heç vaxt paketin cari vəziyyətinə uyğun olmur.

İnformasiya təminatı sistem tərəfindən doldurulmanın tərkib hissəsidir və onun köməyi ilə istifadəçi və paket arasında bütövlükdə, eləcə də paketin ayrı-ayrı komponentləri arasında «qarşılıqlı anlaşma» əldə olunur. Ona görə də biz paketin informasiya təminatından danışarkən 2 tip xüsusi xidməti nəzərdə tuturuq: daxili və xarici.

Xarici informasiya xidmətinin məqsədi istifadəçilərdən daxil olan informasiya sorğularının təmin olunmasıdır, yəni, dialoqun təşkilidir ki, bunun gedişatında istifadəçi tapşırığın icrası, yaxud da paketin iş qabiliyyətini təmin etmək üçün zəruri olan məlumatları paketə göndərir və yaxud da ondan alır. Başqa sözlə, paketin xarici informasiya təminatı məhdud sayda informasiya məsələlərinin həlli vasitəsi kimi düşünülür. Buna misal olaraq paketin funksional dolma modullarının proqram-istismar xarakteristikalarının öyrənilməsinin informasiya məsələsini və yaxud da konkret hesablamaların qoyulmasını reallaşdıran proqram kompleksinin layihələndirilməsi məsələsini göstərmək olar.

İnformasiya məsələsinin reallaşmasında menyu texnikası geniş tətbiq edilir, burada dialoqun məzmunu reqlamentləşdirilir və ağacşəkilli struktura malik olan və konkret informasiya məsələlərinin həllinə yönələn məlumat kitabçası tərəfindən istiqamətləndirilir. Bu halda dialoqun aktiv tərəfi paket olur və bu paket istifadəçidən bir neçə suala (verilən suallara verilə biləcək cavablar ekranda menyu şəklində təqdim olunur) cavab aldıqdan sonra ona arayış və yaxud göstəriş verir ki, verilən anda paketlə iş zamanı necə hərəkət etsin.

Hipermətn texnikasından istifadə zamanı informasiya şəbəkə şəklində təşkil olunur. Dialoqun istənilən anında ekranda müəyyən sənəd səhifəsi olur ki, onun

mətnində əsas (baş) anlayışlar qeyd olunur (məsələn, işıqlandırılaraq) İstifadəçi funksional düymələrin (klaviş) köməyi ilə ixtiyari şəkildə şəbəkədə hərəkət edə bilər: növbəti/əvvəlki səhifələri oxuya bilər, yaxud da onu maraqlandıran əsas anlayışları göstərərək ona aid olan sənədlərə baxa bilər, ya da ekrana əvvəlki sənədi qaytara bilər və s. Sənədləri əyani şəkildə əks etdirmək üçün maşın qrafikasından və yaxud da informasiyanın təqdim edilməsinin digər müasir vasitələrindən geniş istifadə olunur.

Menu və hipermətn adətən, paketin cari vəziyyəti və ya onun imkanları ilə tanışlıq üzrə informasiya məsələsinin həlli üçün istifadə olunur. Digər və həm də az əhəmiyyətli olmayan məsələ – hesablama proqramlarının tələb olunan versiyasının qurulması məsələsi üçün bir qədər fərqli dialoq vasitələri tələb olunur. Əgər paketin imkanları ilə tanışlıq zamanı sənədlər şəbəkəsi üzrə istifadəçinin keçdiyi yol yadda saxlanırdısa, lakin hesablama proqramının qurulması üçün istifadəçi qarşısında duran problem haqqında informasiyanı aydınlaşdırmaq və yadda saxlamaq lazım gəlir. Bu halda dialoqun aktiv tərəfi qismində həm istifadəçi, həm də paket çıxış etməli olur. Bu cür dialoqun gedişatında paketin informasiya xidmətinin istifadəçinin susduğu və üstündən keçdiyi konteksti ixtisaslı mütəxəssis kimi qəbul etmək qabiliyyətinə malik olması arzu olunandır.

Paketin informasiya təminatı konkret informasiya məsələsinin həlli ssenarisini əldə rəhbər tutaraq ikitərəfli aktiv dialoqu dəstəkləyə bilər. Bu cür ssenarini, məsələn, məhsul (maddə) terminlərində verməklə əvvəlcədən paketin informasiya bazasına daxil etmək olar. Maddə (məhsul) müəyyən dildə yazılmış və 2 hissədən ibarət olan qaydadır. Birinci hissə dialoqun gedişatında meydana gələn bu və ya digər vəziyyəti xarakterizə edən faktların siyahısından ibarətdir, ikinci hissə isə bu vəziyyətdə (situasiyada) icrası lazım cələn hərəkətlər siyahısından ibarətdir. İnformasiya məsələsi mürəkkəbləşdikcə bu cür ssenari və yaxud maddələr (məhsul) sistemi ayrı-ayrı maddələr əlavə edilməklə genişləndirilə bilər, həm də bunu yeni maddənin köhnə maddələrlə əlaqələrini nəzərə almaqla icra etmək lazımdır.

İkitərəfli dialoqun təşkilinə daha universal yanaşma (ümumiyyətlə desək, istifadəçi tərəfindən praktiki düşünülmüş məlumat və ya sorğunun emal edilməsinə yol verən) ünsiyyət dilini istifadəçi üçün təbii ünsiyyət dilinə yaxınlaşdıran inkişaf etmiş informasiya-məntiq sistemindən və linqvistik prosessordan birgə istifadəni nəzərdə tutur. Qeyd edək ki, informasiya-məntiq sistemi ilə ünsiyyət dili kifayət qədər məhdud leksikaya və sərt sintaksisə malik olsa da, adətən, paket istifadəçiləri tərəfindən kifayət qədər asanlıqla qavranılır. Buna görə də informasiya-məntiq sistemi yalnız təkmil informasiya təminatı bazası kimi deyil, həm də onun müstəqil istifadə olukna bilər ilkin reallaşma forması kimi tətbiq edilə bilər.

Proqramın təşkili baxımından xarici informasiya xidməti təminatı vasitələri paketin sistem tərəfindən doldurulmasına quraşdırıla bilər, ya da müstəqil işlənilib hazırlana bilər. Sonuncu halda informasiya vasitələrinin ДИЛОС, VOPROS-OTVET tipli güclü ümumi məqsədli informasiya-məntiq sistemlərinə əlavə kimi quraşdırılması asanlaşır.

Daxili informasiya xidmətinin məqsədi paketin fəaliyyəti zamanı onun müxtəlif informasiya komponentlərindən daxil olan sorğuları təmin etməlidir. Bu cür sorğular, məsələn, hesablama planı tərtib edilərkən, proqram kompleksi yaradılarkən, proqram-material arxivinin və hesablama məlumatlarının nizamlaşdırılması zamanı və s. meydana gəlir.

Bu qədər müxtəlif cinsli sistem proseslərini təmin etmək üçün paketin daxili informasiya xidməti predmet sahəsi haqqında, funksional dolma modulları semantikasını, paketin istismarının həyata keçirildiyi hesablama mühiti haqqında biliklərə malik olmalıdır.

Sonda qeyd edək ki, paketin informasiya vasitələri işlənilib hazırlanarkən, hər şeydən əvvəl, terminoloji bazis yaradılmalıdır ki, ona əsaslanaraq biliklərin informasiya xidmətinin təşkili üçün tələb olunanları təsvir etmək mümkün olsun.

Fəsil 2. Hesablama təcrübəsi məsələlərinin proqramlaşdırılması

2.1. Hesablama təcrübəsi, hesablama təcrübəsi dövrü və nəticələrin emalı

Hazırkı dövrdə hesablama maşınlarının köməyiylə həll edilən mühüm məsələlər hesablama təcrübəsi məsələləridir. Qurğu və proseslərin riyazi modelinin qurulması və bu modelin hesablamalarla tədqiqi vasitəsilə öyrənilməsi metodu hesablama təcrübəsi adlanır. Riyazi modeli reallaşdıran proqrama təbii təcrübədə istifadə olunan təcrübə qurğusunun analoqu kimi baxmaq olar. Modelin ədədi (hesablamalarla) tədqiqi nəinki layihələndirilən qurğuların layihəsini və ya fəaliyyət rejimini optimallaşdırmaq, proseslərin müxtəlif xarakteristikalarını müəyyən etmək olar, həm də hesablama təcrübəsindən öncə tədqiqatçıya məlum olmayan yeni prosesləri və xassələri aşkar etmək olar. Hesablama təcrübəsi analitik tədqiqat və təbii təcrübə arasında aralıq mövqe tutur. Analitik tədqiqatda da riyazi model qurulur, lakin model o qədər sadə olur ki, onun öyrənilməsi üçün analitik hesablama kifayət edir və hesablama texnikasının istifadəsinə əsaslanan ədədi metodların cəlb olunmasına ehtiyac qalmır. Analitik tədqiqat tədqiq olunan prosesin sərt xarakteristikasını çox qısa bir zamanda əldə etmək lazım gəldikdə tətbiq edilir. Real proseslərin bu cür öyrənilməsində bəzi qanunauyğunluqları nəzərə almadan çox ciddi sadələşdirmələr aparmaq lazım gəlir. Baxmayaraq ki, verilən model üçün analitik yanaşma çox vaxt dəqiq həll verir, bu sadələşdirilmiş model adətən praktiki hesablamalarda qiymətləndirmələrin tələb olunan dəqiqliyini əldə etməyə imkan vermir. Hesablama təcrübəsində əldə edilən ədədi metodlar tədqiq olunan prosesləri kifayət qədər dolğun əks etdirən və daha mürəkkəb olan modelləri öyrənməyə imkan verir.

Hesablama təcrübələrini təbii (fiziki) təcrübələrlə müqayisə edərkən hesablama təcrübəsinin 2 üstünlüyünü qeyd etmək lazımdır: birincisi, onun aparılması fiziki təcrübələrlə müqayisədə iqtisadi baxımdan bir qayda olaraq, daha ucuz başa gəlir. İkincisi, bir sıra hallarda təbii tədqiqatlar həddən ziyadə parametrlərin ölçülməsi, kritik (böhran) rejimlərin yaradılması və saxlanması və s. ilə bağlıdır, ona görə də hesablama təcrübəsinin tədqiqatın praktiki olaraq yeganə mümkün üsulu olduğu hallar az deyildir.

Hesablama təcrübəsinin göstərilən üstünlükləri onu hazırda nüvə energetikası, kosmik fəzanın öyrənilməsi və s. kimi iri fiziki və mühəndis-texniki problemlərin tədqiqatında əsas metodlar sırasına çıxarmışdır.

Fiziki məsələlərin həlli üçün hesablama təcrübəsinin aşağıdakı mərhələlərini ayırd etmək olar:

1. Riyazi modelin qurulması (tədqiq olunan hadisəni təsvir edən tənliklərin tərtib edilməsi)
2. Ədədi hesablama metodlarının seçilməsi (ilkin riyazi məsələni approksimasiya edən diskret modelin qurulması, fərq sxeminin qurulması, hesablama alqoritminin hazırlanması və s.)
3. Hesablama alqoritmini reallaşdıran proqramın yaradılması
4. Hesablamaların aparılması və alınan informasiyanın emalı.
5. Hesablama nəticələrinin təhlili, fiziki təcrübə ilə müqayisəsi.

Adətən, son (5-ci) mərhələdə məlum olur ki, 1,2 və ya 3 mərhələlərində qəbul olunan qərarlarda müəyyən dəyişikliklər edilməlidir. Belə ki, aydın ola bilər ki, qurulmuş model tədqiq olunan hadisənin xüsusiyyətlərini kifayət qədər əks etdirmir. Bu halda modeldə düzəlişlər aparılır, ədədi metodlarda və onları icra edən proqramlarda müvafiq düzəlişlər aparılır və yeni hesablama yerinə yetirilir. Beləliklə, hesablama təcrübəsi dövrü tam həcmdə təkrar olunur. Nəticələr təhlil edilərkən istifadə edilən ədədi metodlarda nöqsanlar aşkar edilə bilər ki, bunlar çox vaxt dəqiqlik və səmərəlilik mülahizələri ilə bağlı ola bilər. Metodların dəyişdirilməsi müvafiq proqramların da dəyişdirilməsinə səbəb ola bilər. Başqa sözlə, tsikl (dövr) bir qədər ixtisar olunmuş şəkildə təkrarlanacaqdır (2-5 mərhələləri). Ən nəhayət, hər hansı proqramın həll uğursuz ola bilər, məsələn, xarici yaddaşa seçilmiş iş üsulu. Bu cür həllərin yenidən nəzərdən keçirilməsi 3-5 mərhələlərinin təkrarlanmasına gətirib çıxarır.

Burada 3-cü mərhələyə - hesablama proqramlarının hazırlanması mərhələsinə xüsusi diqqət ayrılır və məhz bu mərhələnin avtomatlaşdırılması üçün müəyyən paketlər müəyyən olunur.

3-cü mərhələnin vəzifələrinin geniş səciyyələndirilməsinə keçməzdən öncə 4 və 5-ci mərhələlərdə hesablamaların emalı və nəticələrin təhlili xüsusiyyətləri üzərində qaçıcı dayanaq.

Hesablama təcrübəsi proqramlarının icra etdiyi proqramların bilavasitə nəticələri adətən, böyük verilənlər massivi olur. Əgər bu verilənlərin təhlilini aparan tədqiqatçı yalnız kağızda çap edilmiş sonsuz sayda ədədlərdən ibarət sütunlarla işləyərsə, onda onun əməyi olduqca səmərəsiz olacaqdır. Burada köməyə təcrübənin nəticələri ilə tanışlığın dialoq formaları, məsələn, maşın qrafikası imkanlarından geniş istifadə gələ bilər.

Nəticələrin təhlilinin xarakterik məsələsinə misal göstərək. Tutaq ki, hesablamalar nəticəsində bir neçə dəyişənin müəyyən funksiyasının düzbucaqlı torda hesablanmış qiymətlərini əks etdirən böyük həcmdə çoxölçülü müəyyən matris alınmışdır. Bəzi dəyişənlərin qiymətlərini qeyd etməklə bu matrisin ikiölçülü kəsiyini almaq və 2 dəyişənin onunla təyin olunan vəziyyətinin qrafik qurğuda əksini almaq tələb olunur.

Bu konkret məsələ kifayət qədər sadə təqdim olunur. Lakin təcrübə hesablamasının real nəticələrinin emalı üçün universal alətlərin qurulması cəhdləri çox çətinliklə aradan qaldırıla bilən maneələrə rast gəlir ki, bu da hesablama nəticələrinin tədqiq olunması formalarının çoxsaylı müxtəlifliyi ilə bağlıdır.

Burada yekun funksiyanın hesablandığı torun qeyri-müntəzəmliyini, çox seyrəldilmiş matrisin xüsusi təqdim olunma formalarını, funksiyanın xüsusiyyətlərinin müxtəlif verilmə üsullarını xatırlatmaq olar. Xüsusi təqdim etmə formalarından istifadə olunması hesablama nəticələrinə maşın qrafikasının mövcud alətlərini tətbiq etməyə imkan vermir, belə ki, bu vasitələr, adətən, verilənlərin təqdim olunmasının tam müəyyən olunmuş ardıcıl üsuluna yönəlir.

Ona görə də hesablama nəticələrinin emalı üçün alətlərin qurulması asan məsələ deyildir. Konkret avtomatlaşdırılmış tətbiqi fəaliyyətdə istifadə olunan verilənlərin təqdim olunmasının bütün formaları aşkar edilməli və sistemləşdirilməlidir və bunlardan hər biri üçün çevirici proqram yazılmalıdır ki,

bu formanı qrafikası proqramlarının qəbul edə biləcəyi formaya gətirsin. Bundan sonra istifadəçiyə onu maraqlandıran nəticə məlumatlarından ibarət altçıoxluqları lazım gəldikdə hərf-rəqəm və qrafik qurğulara çıxarmaq üçün dil yaratmaq çətinlik törətmir.

2.2. Hesablama proqramlarının hazırlanması

Hesablama proqramlarının hazırlanması hesablama təcrübəsinin ən çox əməktutan mərhələlərindən biridir. Hesablama təcrübəsinin hər bir dövründə (tsikl) proqramın yeni versiyasının qurulması icra olunur və bu cür fəaliyyətin avtomatlaşdırılmasına yönəldilmiş hər bir proqram paketinin sistem tərəfindən doldurulmasına daim təkrarlanan bu prosedurun dəstəklənməsi vasitələri daxil edilir.

Proqramın reallaşması baxımından hesablama təcrübəsi metodunun çox modellik və çoxvariantlılıq kimi xüsusiyyətləri daha çox əhəmiyyətlidir. Çox modellik odur ki, hesablama təcrübəsinin gedişatında riyazi model dəfələrlə dəyişdirilir və dəqiqləşdirilir, çoxvariantlılıq isə o deməkdir ki, modeldə aparılan dəyişikliklər sonradan müxtəlif birləşmələrdə kombinasiya edilir.

Bu xüsusiyyətlər o halda aşkar edilir ki, birincisi, hesablama təcrübəsinin aparılması üzrə işlərin əksəriyyəti birinci icrada deyil, yuxarıda göstərilən dövrün çoxsaylı təkrarlanması prosesində yerinə yetirilir. Bu cür münasibət tamamilə bütün proqrama da şamil edilir: proqramı hazırlayanların əsas cəhdləri hesablama proqramının ilkin versiyasının yaradılmasına deyil, model haqqında yeni qərarların qəbul edilməsini, proqramın təşkili və hesablama metodlarını əks etdirən müxtəlif modifikasiyalara yönəldilir.

İkincisi, hesablama təcrübəsinin növbəti dövründə yenidən baxılan qərarlar (onlarla yanaşı onlara uyğun proqramlar) birqayda olaraq, kənara atılmır və sonradan digər hesablamalar üçün istifadə oluna bilər. Məsələn, bir riyazi model üçün yararsız olan metod digər model hesablamaları üçün yararlı ola bilər və s.

Beləliklə, hesablama təcrübəsi məsələlərinin proqramlaşdırılması üzrə fəaliyyətin əsas istiqaməti yeni proqramların deyil, mövcud proqramların

təkmilləşdirilməsidir. Lakin bu inkişaf bir qayda olaraq, mövcud modulların daha təkmil versiyalarla əvəz edilməsi deyil, proqram fonduna təcrübənin gedişatında qəbul edilən müxtəlif qərarları əks etdirən yeni-yeni modulların daxil edilməsi hesabına baş verir.

Modulların bu cür toplanması və reallaşdırılan riyazi modellərin yuxarıda qeyd etdiyimiz mürəkkəbliyi ona səbəb olur ki, proqram fondu hədsiz dərəcədə böyüyür. Məsələn, SSRİ Elmlər Akademiyasının tədqiqat institutunda birölçülü qaz dinamikası məsələləri üçün proqram fondunda 1000-dən çox proqram vardı ki, 7104 cümlədən ibarət idi, radiasiya yadan müdafiə məsələləri üçün isə Fortran dilində 23105 cümlədən ibarət proqram var idi. Fond daim yeni-yeni modullarla dolur və onlar müxtəlif kombinasiyalar şəklində hesablamalarda istifadə olunur.

Bu cür böyük və mürəkkəb proqram çoxluğunda istiqamət götürmək o qədər də asan iş deyildir. Burada işin əsas məzmunu yeni proqram modullarının yazılmasından deyil, baxmayaraq ki, buna həmişə ehtiyac vardır, hesablama proqramlarının modullarından yenilərinin qurulmasından və proqram fondunun inkişaf etdirilməsi və müşahidə edilməsi ilə bağlı bir sıra sırf texniki problemlərin həllindən ibarətdir. Bu cür işləmələrin mövcud təminatı proqram vasitələrinin bazasında yerinə yetirilməsi əhəmiyyətli dərəcədə əmək sərfi tələb edir.

Hesablama təcrübəsi məsələlərinin proqramlaşdırılması üçün mövcud proqram təminatından istifadənin uyğun olmamasının əsas səbəbləri aşağıdakılardan ibarətdir:

- tətbiqi proqramçı öz gündəlik işində çoxlu sayda müxtəlif cinsli sistem vasitələri ilə: mətn və yükləmə modulları kitabxanası, redaktorlar, yükləyicilər və s. ilə işləmək məcburiyyətindədir;

- mövcud proqram təminatının bəzi komponentləri ilk növbədə kiçik və orta həcmli proqramlara yönəlir; hesablama təcrübəsi məsələlərinin proqramlaşdırılmasında bu komponentlərdən bilavasitə istifadə olunması qeyri-məhsuldar xərclərin artmasına gətirib çıxarır;

- mövcud proqram təminatı «ümumi istifadəçiyə» istiqamətlənib, yeni mümkün qədər geniş məsələlər dairəsini əhatə etməyə istiqamətlənir və deməli,

ümumilikdə desək, xüsusi vasitələrdən, yəni hesablama təcrübəsinin çoxmodellik və soxvariantlılıq kimi xüsusiyyətlərini nəzərə alan xüsusi vasitələrdən ibarət olmamalıdır.

Mövcud proqram təminatı əsasında hesablama təcrübəsi məsələlərinin proqramlaşdırılmasının təşkili üzrə göstərilən çətinliklər qanunauyğun olaraq xüsusi sistem vasitələrinin yaradılması fikrinə gətirib çıxarır. Bu vasitələr proqram fondunun doldurulması, saxlanması və modifikasiyasını, eləcə də fondun ayrı-ayrı modullarından konkret hesablamalar üçün müxtəlif proqram variantlarının sadə, çevik və sürətli yığılma mexanizmini təmin etməkdir.

Xüsusi sistem vasitələrinin köməyiylə həll edilə biləcək digər məsələlər sırasında qeyri-standart proqram fondundan və informasiya təminatından istifadənin dəstəklənməsini göstərmək olar.

Hesablama təcrübəsi çox vaxt «0-dan başlayır» və proqram fondunun müəyyən hissəsi artıq mövcud olmuş olur. Bu hissəyə əvvəlki hesablama işlərinin gedişatında yaradılmış proqramlar daxil olur. Onların arasında qeyri-standart proqramlara, yəni bir mürəkkəb məsələnin həlli çərçivəsində müştərək istifadəsi planlaşdırılmayan proqramlara da rast gəlinir. Sistem vasitələri bu cür proqramların hesablamalarda istifadəsini asanlaşdıra bilər.

Proqram fondun həcmi və mürəkkəbliyi kifayət qədər böyük olduğundan təcrübəni aparan tətbiqi proqramçı bu fondu xarakterizə edən çoxlu sayda müxtəlif verilənlərlə işləməli olur. Çox vaxt ona hesablama təcrübəsinin müxtəlif «həlqə»lərində istifadə olunan hesablama proqramlarının müxtəlif versiyalarının xüsusiyyətləri və tərkibi haqqında məlumatlar fonda daxil olan ayrı-ayrı modulların funksional-istismar xarakteristikaları, bu və ya digər hesablamanın aparılması zamanı verilən parametrlərin qiymətləri haqqında məlumatlar tələb olunur. Xüsusi sistem vasitələri bir tərəfdən istifadəçini proqram fondunun tərkibi və xarakteristikaları ilə təmin edə, digər tərəfdən isə proqramda dəyişikliklər aparıldıqca sənədlərdə düzəlişləri yüngülləşdirmək və stimullaşdırmaqla «proqram-sənədlər» cütünün tamlığını (bütövlüyünü) dəstəkləyə bilər. Hesablama

təcrübəsinin daim doldurulan və dəyişilən proqram fondu mühitində bu cür xidmətlərinin əhəmiyyətini yenidən qiymətləndirmək çətinlik törədir.

Hesablama təcrübəsi məsələlərinin proqramlaşdırılmasında başlıca problem proqramların hesablama versiyalarının layihələndirilməsinin təşkilidir.

Bu problemin daha inandırıcı həllərindən biri OLYMPUS layihəsi tərəfindən irəli sürülür.

2.3. OLYMPUS layihəsi

Hesablama təcrübəsi məsələlərinin EHM vasitəsilə həlli artıq uzun illərdir ki, davam edir. Bu dövr ərzində onların proqramlaşdırılmasının təşkili üzrə kifayət qədər təcrübə yığılmışdır ki, hesablama təcrübəsi məsələlərinin proqramlaşdırılması texnologiyasının mövcudluğu haqqında fikir söyləməyə əsas verir. Bu texnologiyanın bir çox xüsusiyyətləri OLYMPUS layihəsində ümumiləşdirilmiş və təkmilləşdirilmişdir.

OLYMPUS layihəsi proqramların qarşılıqlı əlaqəsinin rəsmiləşdirilməsini, sənədləşdirilməsini və təşkilini reqlamentləşdirən razılaşmalar məcmusundan ibarətdir. Bu razılaşmaların əsas məqsədləri aşağıdakılardan ibarətdir: proqramın əyaniliyinin yüksəldilməsi; mürəkkəb proqram fondunun istismarının sadələşdirilməsi, proqram materialının mübadiləsi imkanlarının genişləndirilməsi. OLYMPUS layihəsinin bu məqsədə çatmaq üçün təklif etdiyi vasitələri nəzərdən keçirək.

Əyaniliyin yüksəldilməsinə ilk növbədə proqramların rəsmiləşdirilməsi və sənədləşdirilməsi ilə bağlı razılaşmalar kömək edir. OLYMPUS layihəsi çərçivəsində işlənilən hazırlanan proqramlarda onun təyinatını müəyyən edən şərhlərin forması və yeri, istifadə olunan alqoritmlər və s. ciddi şəkildə qeyd olunur. Proqramın daxili rubrikasının əyani təqdimat vasitələri nəzərdə tutulur ki, bu da onun müstəqil fraqmentlərini dəqiq şəkildə bir-birindən ayırır. Təcrübədə istifadə olunan bütün proqramların təsnifatı metodları işlənilib hazırlanmışdır. Proqramda istifadə olunan identifikatorların forması haqqında razılaşmalar

mövcuddur: identifikatorun ilk hərfinə görə müvafiq dəyişənin yalnız tipini deyil, həm də funksional xarakteristikalarını da müəyyən etmək olar.

Bunlar və eləcə də, sənədləşdirmə və rəsmiləşdirmə üzrə bəzi digər razılaşmalar layihə ilə tanış olan istifadəçiyə öz proqramında və digər proqramlarda asanlıqla istiqamət götürməyə, çətinlik çəkmədən istənilən proqramın təyinatını müəyyən etməyə, ona lazım olan alqoritmlər və yaxud onun fraqmentlərini və s. axtarıb tapmağa imkan yaradır. Bundan əlavə, sənədləşmənin sistem tərəfindən dəstəklənməsi vasitələri işlənib hazırlanmışdır ki, bu da çoxsaylı proqramlar üzrə dağınıq olan və bir növdə olan bütün şərhləri formalaşdıran vahid bir sənəddə toplamağa imkan verir. Bununla da proqramların mətnləri sənədləşmənin əsasını təşkil edir və bunun sayəsində bir çox hallarda proqramlar və onları müşayiət edən sənədlər arasında tez-tez rast gəlinən uyğunsuzluqlardan yayınmaq mümkün olur.

OLYMPUS layihəsinin ikinci məqsədi, həm də hesablama təcrübəsinin sistem proqram təminatı baxımından daha əhəmiyyətli olan məqsədi mürəkkəb proqram fondundan istifadəni sadələşdirməkdir. Buna fondada saxlanan proqramların istifadəsi və onlar arasında qarşılıqlı əlaqəyə təşkili haqqında razılaşmalar vasitəsilə nail olunur.

Bu razılaşmalar proqramların modullardan layihələndirilməsinə karkas yanaşmanın tətbiqinin əyaniliyini inandırıcı şəkildə sübut edir.

Tədqiq olunan fiziki hadisəni modelləşdirən konkret hesablamanı icra edən proqram bu proqramın gövdəsini (karkas) təşkil edən və onun tərkib hissəsi olan qeydə alınmış funksional hissələrin məcmusu şəklində təqdim olunur və hesablama sxemi adlanır. Bu hissələrin hər birinə hesablama üzrə aparılan işlərin bir hissəsi təhkim edilir və ona icra edəcəyi işi xarakterizə edən ad verilir ki, buna onun funksional adı deyilir. Məsələn, ENERGY – enerji hesablamlarını, STEPON – hesablamların əsas addımını təmin edən hissələrdir və s. bundan başqa, bu cür müəyyən edilmiş hissələr arasında qarşılıqlı əlaqə üsulları işlənib hazırlanır və qeydə alınır. Bundan sonra funksional hissələrin hər biri üçün bir neçə

reallaşma və yaxud modullar yazıla bilər və hər hansı konker hesablama bu modullardan yalnız biri iştirak edə bilər.

Nəticədə hesablama təcrübəsi dövrünün (tsikl) növbəti bəndinə keçid proqramın reallaşması baxımından, əvvəlki bəndlərdə istifadə olunmuş modulun (modulların) dəyişdirilməsinə gətirib çıxarır. Əgər verilən bənddə qəbul edilən qərar ilk dəfə reallaşarsa, onda müvafiq modul yenidən yazılmalıdır; əgər əvvəl tətbiq olunmuş metod tələb olunursa, onda onun üçün əvvəlki dövrdə yazılmış və proqram fondunda saxlanmış moduldan istifadə etmək olar.

Göstərilən iş rejimi (qaydası), şübhəsiz ki, hesablama təcrübəsinin proqramlaşdırılmasının ilkin mərhələsində proqramın gövdəsinin dəqiqliklə hazırlanması ilə bağlı müəyyən əmək sərf olunmasını tələb edir. Lakin bu əmək sərfi proqram fondunun növbəti genişlənməsi və istifadəsinin sadələşdirilməsi ilə özünü doğruldur. Çünki, fondun inkişaf etdirilməsi (təkmilləşdirilməsi) əvvəlcədən yazılmış proqramların mətninə toxunmadan yerinə yetirilmişdir.

OLYMPUS layihəsinin üçüncü məqsədi eyni bir sahədə, yaxud da yaxın sahələrdə hesablama təcrübəsi məsələlərinin həlli ilə məşğul olan müxtəlif şəxslər və təşkilatlar arasında proqram materiallarının mübadiləsi imkanlarını genişləndirməkdir. Məsələ ondadır ki, bu layihə meydana gəlməmişdən öncə mübadilə obyektini kimi, əsasən, ya konker məsələni və ya məsələlər sinfini həll edən tamamlanmış proqramlar, yaxud da ayrı-ayrı riyazi metodları reallaşdıran alqoritmlər və onlardan tərtib olunmuş kitabxanalar çıxış edirdi. OLYMPUS razılaşmalarının tətbiqi praktiki olaraq ixtiyari proqram fraqmentlərini mübadilə etməyə imkan verir və əlbəttə ki, istehsalçının və istehlakçının eyni bir proqram gövdəsindən istifadə etməsi əsas şərtidir. Məhz verilənlərin daxil edilməsi və ilkin emalını təmin edən modullar, enerji tənliyinin hesablanması və ümumiyyətlə proqramın gövdəsində (karkasında) nəzərdə tutulmuş istənilən hissənin reallaşması mübadilə obyektini ola bilər. Burada alınmış hissələrin artıq istifadəsində olan proqramlara uyğunlaşdırılması, hər şeydən əvvəl, proqram materialının mübadiləsində iştirak edənlərin vahid proqram gövdəsindən, yəni hesablama sxemindən istifadə etməsi sayəsində əldə edilir. OLYMPUS razılaşmaları

sistemində işləyən bütün proqramların hamısı mövcud translyatorların (çeviricilərin) əksəriyyəti tərəfindən qəbul edilən Fortran dilində yazılır. Nəhayət, konker əməliyyat sistemi ilə əlaqədar olan proqram hissələri qeydə alınmış şəkildə ayrılır və sənədləşdirilir.

Beynəlxalq təcrübədə OLYMPUS layihəsi geniş tətbiq edilərək qəbul olunmuşdur. Computer physics communications jurnalında müntəzəm şəkildə bu layihə çərçivəsində işlənilib hazırlanmış və müxtəlif ölkələrdə bir çox təşkilatlar tərəfindən istifadə edilən alqoritmlər nəşr edilir.

OLYMPUS razılaşmaları sistem dəstəkləmə vasitələri olmadan da təcrübə hesablamalarını aparmağı əsaslı şəkildə asanlaşdırır. Lakin bu vasitələrin meydana gəlməsi bu razılaşmaların əhəmiyyətini dəfələrlə artırır. Bunu həm proqramların əyaniliyinin artmasına, həm də hesablama variantlarının layihələndirilməsinin asanlaşdırılmasına aid etmək olar. OLYMPUS layihəsinin müəllifləri tərəfindən proqramları daha aydın etmək, proqram fondu ilə tanışlıq və sənədlərin tərtib edilməsi proseslərini asanlaşdırmaq üçün bir sıra sistem işləmələri hazırlanmışdır. Eyni zamanda proqramların hesablama variantları layihə müəllifləri tərəfindən mövcud sistem vasitələri əsasında layihələndirilmişdi. Ona görə də variantların formalaşdırılması üçün onlar hər hansı bir hesablama üçün bir neçə sətirə qədər olan tapşırıqlar tərtib etməklə universal mətn prosessorundan istifadə edirdilər.

Fəsil III. Xüsusi tətbiqi proqramlar paketlərinin birləşdirilməsinin dialoglu sistemi

3.1. Tətbiqi proqramlar paketi generasiya edən sistemlər.

Sistem proqram təminatı (SPT) (Riyazi dəstək sistemləri (RDS)) kompüterlərin yaradılması üzrə iş həcminin yüksək artımına səbəb olan kompüter texnologiyalarının sürətlə böyüməsi və onun tətbiqi sahəsinin daim genişlənməsi ilə əlaqədar olaraq müasir kompleks kibernetikada böyük komplekslər, xüsusən də alətlər istehsal etmək məqsədi ilə proqramlaşdırmanın avtomatlaşdırılması metodları intensiv inkişaf edir. Tətbiq olunan **proqram təminatı** (RT), problem yönümlü sistemlər (proqram sistemləri) və fərdi yönümlü sistemlərlə təyin edilə bilər. Sonuncular, bu sistemin tətbiqi sahəsində çalışan insanların kontingentinin peşə lüğətini əks etdirən bir dildə istifadəçilərlə dialog vasitələri ilə xarakterizə olunur.

Tətbiq olunan **SPT** (RDS) vasitələrinin yaranmasının mürəkkəb və çoxölçülü probleminin həllində əsasən instrumental sistemlərin inkişafı ilə əlaqəli ayrı-ayrı istiqamətlər xüsusən tətbiqi proqram paketlərinin generatorları /TPP/ ilə əlaqələndirilmişdir. TPP generatorları, bir meta-alqoritm quraraq, problemlərin müəyyən bir hissəsini həll etmək üçün daha dar bir sinif sinfinə yönəlmiş bir proqram sisteminin müəyyən bir versiyasını seçməyə imkan verir.

Mövcud **SPT** (RDS)-nin bir sıra komponentlərindən, bu seriyanın ayrı-ayrı komponentləri tərəfindən həll olunan siniflərin böyüdülməsindən ibarət olan problemlər üçün proqram versiyalarını tərtib etməyi mümkün edən genişləndirici proqram sistemlərini çağırırıq.

Problemin belə bir həlli DISUPPP sistemində tətbiq olunan yeni bir riyazi dəstək yönümlü proqram təminatının layihələndirilməsinin texnologiyasının əsasını təşkil edir. Sistem, həll olunan problemlər sinfinin genişlənməsi məqsədi ilə ixtisaslaşmış TPP-lərini birləşdirmək üçün hazırlanmışdır. Bu sistemin yaradılmasının aktuallığı aşağıda göstərilən cəhdlərdən irəli gəlirdi:

- istifadəçini seçilmiş predmet sahəsində tətbiq olunan problemləri həll etmək və həm də maşınların quruluşunun dərinliyinə varmaq zərurətindən xilas edən

rahat vasitələrlə təmin etmək, **kompüterlərdə** (EHM-larında) problemlərin həllinin təşkili və proqramlaşdırma metodları ilə təmin etmək. Əslində, DISSUPP, TPP üçün xüsusi bir xidmət göstərmək və EHM-larında ƏS-də tapşırıqların idarəetmə dilini DISPPP sisteminin dialoq dili ilə əvəz etmək üçün nəzərdə tutulmuşdur;

- texnoloji bir üsul olaraq, həll ediləcək problemlər sinfi və TPP-də dəyişməz olan tapşırıqların həlli prosesini həyata keçirmək və burada hər bir tapşırıq ayrı-ayrı paketlərlə həll edilən alt məsələlərə bölünə bilər. Xüsusi halda, bu genişlənmiş sinfə ayrı-ayrı TPP-lər tərəfindən həll olunan problemlərin birliyi olan bir alt sinif kimi daxil edilir.

Sistem arxitekturası işlənilib hazırlanarkən, birləşdirilən paketlərə olan təbii tələblər nəzərə alınır. Xüsusilə də, birləşdiriləcək paketlər qəbul edilmiş instrumental proqramlaşdırma sistemindən yaranan vahid sistem konvensiyalarına cavab verməlidir və ayrı-ayrı məsələlərin (sabit bir sinifdən) həlli üçün paketin meta-alqoritminin planlaşdırılması paketin mövzu sahəsinin modelinə uyğun olaraq həyata keçirilməlidir.

Bu sistem çərçivəsində proqram təminatı – interpretasiya proseslərinin təsviri üçün ixtisaslaşdırılmış xüsusi dilləri - giriş paketi dilləri və direktivlərin daxilolma dillərində dialoqun təşkili metodlarını təsvir etmək üçün seçilmiş formalizmə əsaslanır. Burada sonuncu, əlavə informasiyanın istifadəçi tərəfindən daxil edilməsinə operativ xidmətə və yaxud da onun tərəfindən paketlərdə tapşırıqların idarə olunması üzrə qərarların qəbul edilməsinə yönəldilmişdir.

Dillərin semantik cəhətlərini təsvir etmək üçün paketin mövzu hissəsinin qrafik modeli xüsusi marşrutlaşdırma qrafiki şəklində istifadə olunur, bununla əlaqədar DESUPP tərəfindən yaradılan dialoq sistemlərinə marşrutizasiya deyilir. DESUPP tərəfindən həll olunan tapşırıqlar sinfinin genişlənməsi yeni paketlər birləşdirməklə həyata keçirilir. Bu məqsədlə müvafiq domen modelləri qurulur və məlumatların idarə olunması üçün sistem konvensiyalarına uyğun olaraq plug-in paketləri dəyişdirilir.

TPP nəsil sistemləri. Tətbiqi RDS üçün alətlər yaratmaq probleminin həllində sistem proqramçılarının məhsuldarlığının əhəmiyyətli dərəcədə artmasına

ehtiyac yaranır, buna isə sistem proqramlaşdırmasının intellektləşdirilməsi yolu ilə nail olmaq mümkün idi ki, bu sahə də 1990-cı illərin ortalarına qədər olduqca geniş nəzəri bazaya malik idi. Bu vəziyyətdə, müəyyən bir sinifin giriş dili haqqında informasiyadan müvafiq proqramlaşdırma sistemini qurmaq üçün tətbiq olunan metasistemlərdə proqramlaşdırma dillərinin sintaksisi və semantikasının formallaşdırılmasında dərin nəzəri nəticələrin praktiki təcəssümü nəzərə alınmalıdır. Metasistemlər arasında üç sinif fərqləndirilir: metakompilyatorlar, çoxdilli sistemlər və parametrik sistemlər.

Metakompilyatorlar əsasən RDS-nin komponentlərinin təsvirinə yönəldilmiş, xüsusi dillərdən ibarət olan və sistem proqramlaşdırma dilləri olan kompilyatorlardır. Onlara klassik nümunə kimi Brucker və Morrisin tərtib etdiyi kompilyatorları və ünvanların translyatorlarını göstərmək olar.

Kiyev instrumental EHM-da tətbiq olunan ünvan dili translyatorlarından birini istifadə edərək, SSRİ-də ilk dəfə bu dildə kompilyatorun rəsmi təsviri alındı və Dnepr-1 EHM-da translyator həyata keçirildi. Sonralar, SSRİ-də konkret dil prosessorlarının yaradılması üzrə ümidverici nəticələr verən bir sıra metakompilyatorlar yaradıldı məsələn, СПУТНИК, Т-СЕМОЛ, СПИТ, СПАРТАК, ЛОРД, СУПЕР)

Çoxdilli proqramlaşdırma sistemi, hər bir prosessorda istifadə olunan altsistemlərin inteqrasiyasının vahid alqoritmlərini özündə birləşdirən müxtəlif giriş dillərinə malik olan bir neçə dil prosessorunun birləşməsidir və sistemin nüvəsi adlanan bu hissə ümumi hissədir. Çoxdilli sistemlərə misal olaraq BETA sistemlərini, START, DTS/EU, BASIC -6, ISP göstərmək olar.

Daxilolma dili ilə əlaqədar bəzi məlumatlar üzrə parametrik bir proqramlaşdırma sistemini konkret dil prosessoru, həm də onun içərisində yerləşmiş meta-alqoritmə uyğun olaraq yaradır və bu da vahid translyasiya sxemində ifadə olunmaqla vahid informasiya bazası şəklində həyata keçirilir ki, ümumiyyətlə translyasiya alqoritmləri onun üzərində işləyir. Belə bir sistemin seçilmiş sinifdən müəyyən bir dildən tərcüməyə yönəldilməsi yaradılan dil

prosessorunu xarakterizə edən verilmiş parametrlərə görə avtomatik köklənməsindən ibarətdir.

Əlbəttə ki, parametrik sistemlər yaradılan prosessorların formal təsvirinin mövcudluğu və bu təsvirə uyğun olaraq sistemlərin avtomatik tənzimlənməsi ilə çoxdilli sistemlərdən fərqlənir. Digər tərəfdən, metakompilyatorlarla müqayisədə parametrik sistemlərdə yaradılan prosessorların formal təsvir vasitələri sadələşdirilir, çünki bütün işləmə prosesini təsvir etmək lazım deyil, yalnız bu prosesin fərdi parametrlərini təyin etmək lazımdır. Beləliklə, parametrik sistemlər çoxdilli sistemlər və metakompilyatorlar arasında aralıq mövqe tutur. Bu sinifdən olan sistemlərin müxtəlifliyi parametrlərin və onların qurulma vasitələrinin müxtəlifliyi, eləcə də parametrik sistemlərin reallaşması üsulları ilə əlaqədardır, hansı ki, bu və ya digər dərəcədə ya çoxdilli sistemlərə yaxud da metakompilyatorlara yaxınlaşa bilər (məsələn, ДИСПРОМ, ДИФИИ, АСП-РВ).

Son illərdə həm EHM resurslarının, həm də məsələlərin həllinə xidmət edən proqram təminatı vasitələrinin inteqrasiyası prinsipindən istifadə ilə əlaqədar olaraq, eləcə də tətbiqi məsələlərin həllində RDS-nin ayrı-ayrı komponentlərinin yerinə yetirdiyi funksiyaların xüsusiyyətlərinin aydınlaşdırılması ilə əlaqədar olaraq, ənənəvi proqramlaşdırma sistemi konsepsiyasına dil prosessoru və bir-biri ilə əlaqəli olmayan servis altsistemlərinin birləşməsi kimi baxılır.

Tətbiqi məsələlərin proqramlaşdırılması texnologiyasının keyfiyyətə yaxşılaşdırılmasına ehtiyac duyulan xüsusi sahələrdə RDS-nin idarəetmə məsələlərinin həllində əsas rolu TPP oynayır. TPP istifadəçilərlə dialoqun daha səmərəli təşkilini həyata keçirir ki, bu da onlardan TPP-lər, ƏS və xarici mühit arasında informasiyanın ötürülməsinin mürəkkəb prosesini gizlətməyə və paketlərin giriş hissəsini, yəni proqramlaşdırmanı, xüsusən də giriş dillərini asanlaşdırmağa imkan verir.

Üstəlik, maşına daxil olmaq imkanı, istifadəçinin problemlərin həlli üçün mümkün qədər az zaman ərzində və nisbətən az sayda parametr və variantlardan istifadə etdiyi təqdirdə məqbul miqdarda düzgün həll tapmasına kömək edir.

Metasistemlərlə müqayisədə, TPP generatorları yaradılan TPP-nin giriş dilini tənzimləmək üçün makrogeneratorlar (bax, məsələn, РАДУГА və ya МАКРОБОЛ sistemləri) və ya genişləndirilmiş sistemlər kimi daha sadə vasitələrə malikdir (məs. ППОЕКТ). Lakin bununla birlikdə, bu sistemlərdə TPP-nin xarici mühitlə qarşılıqlı təsir məsələlərinə xüsusi diqqət yetirilir, məsələn, ƏS, terminal cihazlar, xarici yaddaş və s. Qarşılıqlı əlaqə məsələləri, ayrı-ayrı paketlərin istifadə etdiyi integrasiya olunmuş informasiya bazası mövcud olduqda verilənlərin tətbiqi proqramlardan asılı olmaması prinsipini tətbiq etməklə və proqramlararası əlaqələri unifikasiya etməklə həll olunur.

Beləliklə, АПРОП və СИПП sistemlərində TPP-nin layihələndirilməsi modul proqramlaşdırma, RTK-komplekslərdə - struktur, və ППОЕКТ sistemində - formal texniki tapşırıqlar metodu üzərində qurulmuşdur.

Qeyd etmək lazımdır ki, sadalanan TPP generatorlarında parametrləşdirmə prinsipindən geniş istifadə edilir ki, bu da yaradılan paketlər haqqında məlumat tapşırığını sadələşdirməyə və fərdi alqoritmləri və altsistemlərini birləşdirməyə və bununla da yaradılan TPP-lər üçün bir növ hazırlıq görməyə, yəni paketin meta-alqoritmini yaratmağa imkan verir.. Paket metaalqoritmın qurulması qaydası parametrik proqramlaşdırma sisteminin fəaliyyətinə oxşardır.

Bir sıra TPP generatorları, yaradılan paketlər haqqında məlumat olaraq domen modellərindən istifadə edirlər, bu da domen anlayışının rəsmiləşdirilməsi ilə əlaqədar olaraq paketin metaalqoritmının alqoritmik bir şəkildə qurulması prosedurunun təsvir etməyə imkan verir. Aşağıdakı sistemlər belə TPP generatorlarına nümunə ola bilər: qraf modellərindən istifadə edən SIRIUS; hesablama modelləri üzərində qurulmuş PRIZE; domen modelinin əlaqəli verilənlər bazası sxemi olaraq təyin olunduğu MARS modelləri.

Bu cür TPP generatorlarının yaradılmasının əsasını predmet sahəsi və həll olunan problemlərin şərtlərinin təmsil olunması üsulları təşkil edir. Bu günə qədər, ayrı-ayrı TPP sinifləri üçün layihələndirmə metodları ilə əlaqədar olaraq bir sıra nəticələr əldə edilmişdir. Bu metodlardan bəziləri, xüsusən də bəzi nəzəriyyələrdəki teoremlərin sübutuna və ya qərar qəbul etmə cədvəlləri vasitəsi ilə

problemlərin həlli alqoritmlərinin təsvirinə əsaslanaraq hələ kifayət qədər inkişaf etdirilməmişdir və buna görə də real TPP generatorlarında praktiki olaraq istifadə edilmir.

Burada baxılan genişləndirilmiş proqram təminatı sistemləri bəzi ümumiləşdirilmiş TPP-lərin generatorları hesab edilə bilər. Konkret halda DISUPPP predmet sahəsinin "tref" modelindən istifadə edən marşrut sistemlərinin generatoru kimi, marşrut sistemlərinin özlərinə isə çoxdilli sistemlər kimi baxılmalıdır.

3.2. DISUPPP kompleksinin tətbiq sahəsinin formal modeli.

Burada, TPP dedikdə, ayrı-ayrı hesablama addımlarını həyata keçirən və xüsusi bir monitor tərəfindən birləşdirilmiş $T_1 \dots T_n$ modullarını nəzərdə tuturuq. Baxılan TPP-ə əlavə tələblər qoyulur: modullar vahid bir instrumental proqramlaşdırma dilində həyata keçirilməlidir və verilənlərin təşkili və idarə edilməsi funksiyaları TPP-nin üzərinə qoyulur.

TPP-nin giriş dili paketin həll etdiyi məsələlərin şərtləri haqqında məlumat yaratmaq üçün bir vasitə kimi istifadə olunur, məsələlərin şərtlərini göstərən parametrlərin, məsələn, cədvəl şəklində, siyahısıdır. Paket alqoritmi sonlu olduğundan parametrlərin sayı T -sabitdir. Həqiqi məsələlər üçün bu ədəd T - çox böyük ola bilər, ona görə də konkret bir məsələdə müəyyən parametrlər dəsti üçün (dövrələrin, budaqlanmaların mövcud olması səbəbindən, eləcə də parametrlərin emalı prosesində istifadəçidən əlavə informasiyanın paketin alqoritmi ilə əldə olunmasının mümkünlüyü) hesablama zəncirində addımların təyin edilməsi parametrləri ilə işləmək çətinlik törədir. Lakin bununla yanaşı fərdi TPP modulları arasındakı əlaqə ardıcılığı və bağlantılar - hesablama addımları – ayrı-ayrı parametrləri, onların qiymətlərinin mümkün diapazonlarını və bu parametrlər arasındakı əlaqələri göstərməklə paket qrafını qurmaq mümkündür.

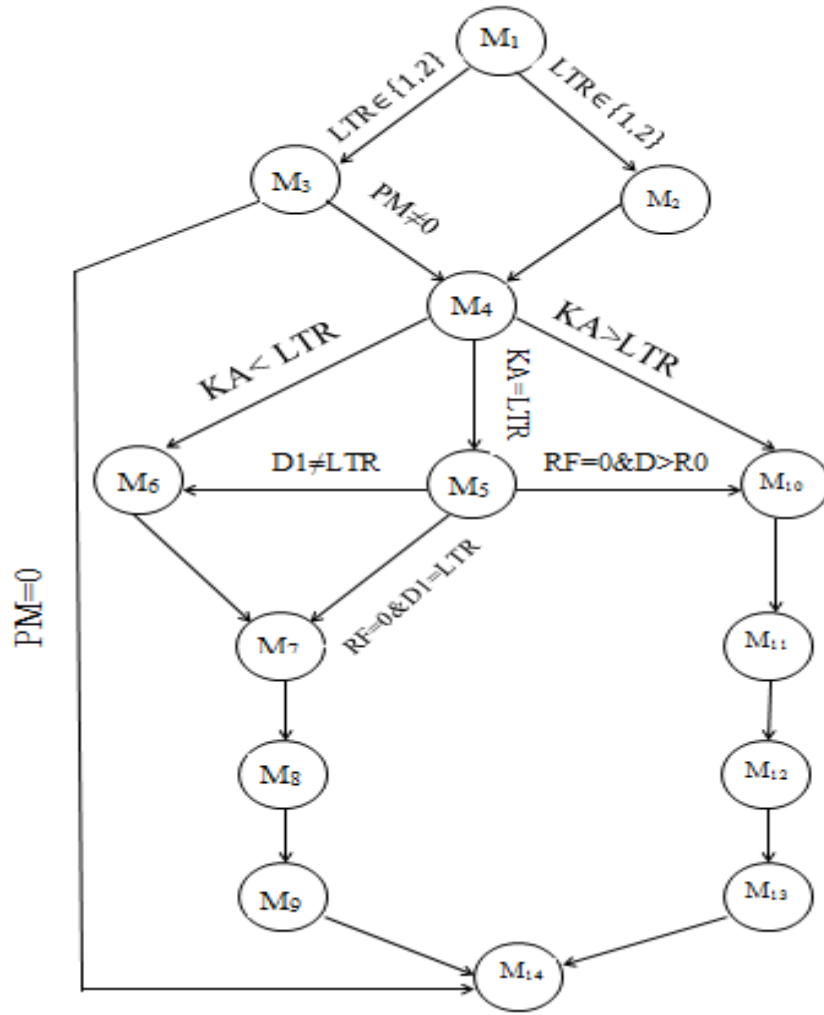
DISUPPP-da predmet sahəsinin modelləri qraf şəklində verilir, onların təpələri parametrlərin transformasiya operatorlarıdır və paketin giriş dillərində verilir. Qrafın qövsləri uclar (təpələr) arasında informasiya axınını göstərir. TPP-də məsələlərin həlli üçün predmet sahəsinə dair sadələşdirilmiş konkret bir model

kifayətdir və bu da informasiyanın çevrilməsi marşrutunu təyin edir ki, onun əsasında hər bir konkret məsələnin həlli üçün fikir söyləmək olar. Marşrut seçimi istifadəçinin giriş proqramlarında ayrı-ayrı parametrlər üçün istifadəçi tərəfindən müəyyən edilmiş qiymətlərə uyğun olaraq TPP tərəfindən avtomatik olaraq həyata keçirilir.

Marşrutlaşdırma qrafını paketdə məsələlərin həllinin ayrı-ayrı addımlarına və R qövslər çoxluğuna uyğun olan $M=(M_1, \dots, M_n)$ təpələrindən ibarət olan istiqamətlənmiş C qrafı kimi müəyyən edək. Əgər M_1 ucundan (təpə) bir neçə qövs çıxırsa, bu cür qövsləri problemin həlli yollarının hər birini seçmək şərtləri kimi qeyd edirik. Seçim şərtləri paketin $P_1, \dots, P_m$ parametrlərinin altçoxluqları arasındakı əlaqəni müəyyənləşdirir: nizam, bərabərlik, parametrlərin müəyyən qiymətlər diapazonuna aid olması və s. hansı ki, bunlardan konyunksiya (birləşmə), dezyunksiya (ayrılma) və inkar əməliyyatlarından istifadə edərək daha mürəkkəb predikatlar qurula bilər.

Şəkildə, layihələndirilmiş marşrut sistemlərindən birinin - САПЛИИПОВ TPP-nin qrafı göstərilir. Bu paketdə həll yolu M_1, M_3, M_4, M_5 pillələrində LTR, PM, KA, D1, RF, D, RO yeddi parametrin qiymətləri ilə təyin olunur.

PT predmet sahəsinin modelini təyin edən və məsələlərin həlli üçün bir qraf anlayışını təyin edək. Qraf, paketin əsas modullarına uyğun olan və $m = (m_1, \dots, m_n)$ uclar çoxluğundan və eyni zamanda ya boş bir simvol ilə, ya həll yolu seçmə şərti ilə, ya da parametr adları dəsti ilə (hansı ki, onlarda düzəliş yalnız terminal vasitəsi ilə aparıla bilər) işarələnmiş R qövslər çoxluğundan ibarət olan istiqamətli və yüklənmiş bir $A/P, K$ psevdografıdır. Son tip qövsləri düzəldilə bilməyən qövslər adlandıracağıq. Düzəldilə bilməyən qövslər məsələnin həllinin gedişatını əks etdirir və bu barədə informasiya istifadəçi tərəfindən düzgün tərtib edilir.



Şəkil 3.1. SAPLINPOV tətbiqi proqramları paketinin marşrut qrafı

Məsələlərin həll qrafı, DISUPPP sistemində bir paketə xidmət məqsədləri üçün marşrutlaşdırma qrafının konkretləşdirilməsidir.

Əgər M_i addımı paketdə M_k modulu tərəfindən yerinə yetirilərsə M_i, M təpəsi ya $m_k > m$ təpələrindən biri ilə üst-üstə düşür, ya da $m_k, m_{k+1}, \dots, m_{k+e}$ modullarının qoşulmasını əks etdirən və həm də M_i addımlarını reallaşdıran $M_i = ([m_k, m_{k+1}, \dots, m_{k+e}]_1 R_I)$ altqrafıdır. Bu halda M_i altqrafı bu təpədən nə qədər qövs çıxarsa bir o qədər də təpələrə malik olacaqdır.

Əgər məsələnin həll qrafının bir təpəsindən kəsişən parametrlər dəsti ilə işarələnmiş bir neçə düzəldici qövs çıxarsa, onda bir ucundan gəlicə, bu dəstlər məsələnin həll yolunun seçilməsi şərtləri ilə eyni qaydalara uyğun qurulmuş predikat şəklində şərtlərlə təmin olunur. Məsələn, M_{12} (m_{12}) təpəsindən kəsişən

parametrlər dəsti olan iki düzəldici qövs çıxırsa, onda bu qövslər LTR şərtinə görə fərqlənəcəklər.

Düzəldici qövslərinin çıxdığı bütün uclarda (təpələrdə) TPP istifadəçiləri ilə dialoq seansları mümkündür. Bu modullarda istifadəçiyə sistem tərəfindən verilən diaqnostik mesajə əsasən yalnız adları göstərilən ucdan (təpədən) çıxan düzəliş qövslərini qeyd edən parametrləri düzəltmək imkanı verilir. Növbəti modula keçid, məsələlərin həll qrafına uyğun olaraq sistem tərəfindən müəyyən edilir. Məsələn, m_8 ucunda H, W, RF, D, RO, D1 parametrləri tənzimləyə bilər. Üstəlik, H parametrində düzəliş aparıldığı halda, həllər m_8 modulundan davam etdiriləcək; W - m_2 modulundan və m_3 modulundan isə RF, D, RO, D1 parametrlərinin hər hansı birində yaxud da onların hər hansı bir birləşməsində düzəliş aparıldıqda, həll davam etdirilir.

Bəzi modulların alt qraflarına daxil olan düzəldici qövslərin uclardan aparılmasına icazə verilir, bununla birlikdə hər bir bu cür qövs ayrıca bir uca (təpəyə) və ya marşrutlaşdırma cədvəlində altqrafın kökünə uyğun olan uca çıxmalıdır. Başqa sözlə, parametrləri tənzimlədikdən sonra, marşrutlaşdırma qrafında hər hansı bir yol seqmentinin təkrarlanması yalnız bəzi ucların (təpələrin) əvvəlindən mümkündür. Məsələn, M_{10} altqrafında RF, D, RO, D1 parametrləri eyni olan iki düzəldici qövs. Bu, yol seqmentinin və $m_3 \sim m_4 \sim m_{10}$ (m_{10} ucundan çıxan qövsün) və başqa bir yolun $m_2 \rightarrow m_{10}$ seqmentinin (M_m altqrafın m_{14} təpəsindən çıxan qövsün) təkrarlanması deməkdir.

Yolun kəsiyinin məsələlərin həlli qrafında (müvafiq olaraq marşrutlaşdırma qrafında) təkrarlanmasına dair sonuncu tələbin yerinə yetirilməsi yalnız TPP modullarının rentabelliği şərti ilə mümkündür.

DİSUPPP sistemində dialoqun təşkili. Giriş informasiyasının obyektlərinin sərt şəkildə qeydə alındığı ixtisaslaşdırılmış TPP - də həll olunan məsələlərin şərtlərini göstərən Paket parametrləri - predmet sahəsinin modeli, (və ya bizim halda marşrutlaşdırma və məsələlərin həlli qrafları) obyektlər arasında, say və nomenklatura arasında olan asılılıqları əks etdirir. Beləliklə, belə bir TPP - də problemlərin həllinin gedişini idarə etmək məqsədilə, obyektlərin aydın şəkildə

müəyyən edilmiş dəsti ilə dialoq dili istifadə edilə bilər, hansı ki, paket parametrləri adlanır.

DISUPPP sisteminin dialoq dili, ekran displayindən istifadəçiyə ünvanlanan bir sıra sorğu və mesajlar və terminal haqqında sistemə cavab verməyə xidmət edən göstərişləri əhatə edir. Təbii ki, konkret paketlərin təsviri onun mümkün cavablarını qurmaq üçün istifadəçiyə diaqnostik mesajların və rekomendasiyaların siyahısı ilə təchiz olunur. Dialoq dilinə aşağıdakı təlimatlar daxil edilmişdir:

- məsələlərin həllinin başlanğıcı və həllin gedişatının bərpası;
- məsələlərin həllinin gedişatında fasilələr;
- məsələlərin hesablanması prosesində TPP parametrlərinin qiymətlərində düzəlişlər;
- ilkin verilənlərin və məsələlərin hesablama nəticələrinin arxivləşdirilməsi ;
- Digər paketin giriş məlumatları kimi tətbiq olunan bir paketin nəticələrinin razılaşdırılması məqsədi ilə paketlərin giriş məlumatlarının tənzimlənməsi.

Paketlərdəki məsələləri həll edərkən, istifadəçi ilə dialoq bir neçə direktivdən ibarət olan sessiyaların marşrutlaşdırma qrafı ilə həyata keçirilir. Sessiya həm istifadəçi, həm də sistem tərəfindən başlanılır, sonuncu halda sistem tərəfindən diaqnostik mesaj verilir və istifadəçiyə müəyyən parametrlərin dəyərlərini düzəltmək (və ya əlavə olaraq göstərmək) imkanı verilir. Bu halda məsələnin həll addımlarının bir neçəsini təkrar etməyə icazə verilir.

İstifadəçinin istəyi ilə məsələnin həlli prosesi marşrutlaşdırma qrafına uyğun olan anda kəsilə bilər; TPP hal vektoru yadda saxlanılır və paket kəsilmə addımına qədər davam edə bilər. İstifadəçi yalnız (bu paketin marşrutlaşdırma qrafının kökünə uyğun olan) başlamaq və ya fasilə addımına qədər yenidən başlamaq üçün dialoq seansını başladır.

3.3. DISUPPP sisteminin arxitekturu.

DISSUPPP tərəfindən yaradılan marşrut sisteminin funksional diaqramı şəkl.3-də verilmişdir. Sxemdə nəzarət əlaqələri tək oxlarla, informasiya əlaqələri isə ikili oxlarla göstərilir.

DİSUPPP sisteminə daxildir:

- verilənlərin idarə edilməsi üzrə sistem razılaşmalarına uyğun olaraq, məsələlərin həlli qrafını və paketin modifikasiyasını əldə etmək üçün marşrutlaşdırma qrafının konkretləşdirilməsi funksiyasına daxil olan paketlərin qoşulması altsistemi;

- paketdə məsələnin yerinə yetirilməsinin alternativ yolları halında əlavə informasiyanın daxil edilməsini və istifadəçi tərəfindən qəbul edilmiş qərarların həyata keçirilməsini həyata keçirən dialogun aparılması altsistemi;

- bir paketdə həll edilmiş tapşırıqların nəticələrini, digər paketlərlə məsələnin həllində sonrakı istifadə üçün redaktə etmək məqsədi ilə tənzimləyən bir altsistem;

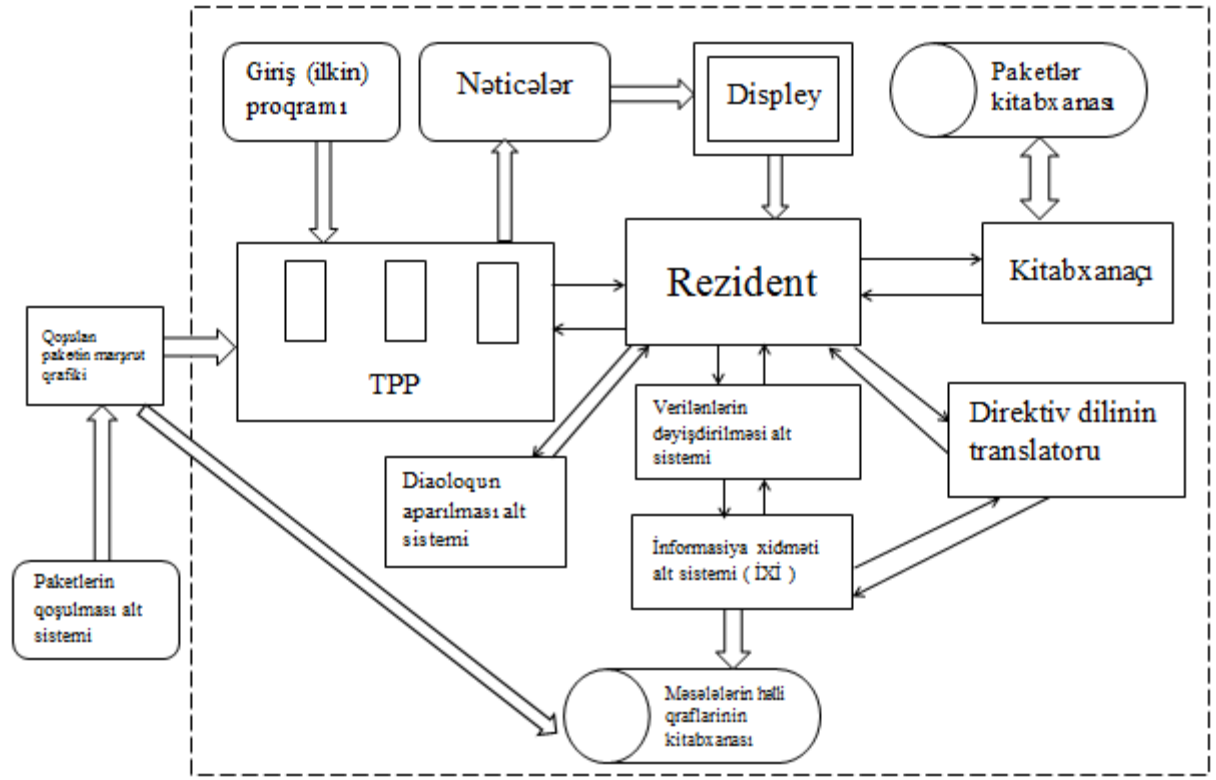
- EC 7066 displeyi ilə rabitə funksiyalarını yerinə yetirən, TPP-nin marşrutlaşdırma sistemini dəstəkləyən və paket verilənlərini idarə edən rezident sistem.

- məcəllələrin proqram versiyalarının, yekun verilənlərin və paketlərin idarəetmə məlumatlarının saxlanması üçün kitabxana, məsələnin diaqnostik sənədlər, parametrlərin adlarından ibarət cədvəllər və s.;

- sistemin dialog dilini həyata keçirən direktiv dilin tərcüməçisi;

- SIO məsələlərini həll edən, istinad funksiyalarını yerinə yetirən və funksional modulları məlumatlardan ayıraraq dəyişdirildikdə qrafların dəyişdirilməsi prosesini asanlaşdırmaq üçün hazırlanmış informasiya xidmətlərinin altsistemi.

Dialogun aparılması alt sistemi. DİSUPPP tərəfindən yaradılan marşrut sisteminin paket idarəedilməsi ayrıca paket modullarında həll ediləcək vəzifələrin idarəetmə vasitələrini dinamik şəkildə birləşdirməklə və müxtəlif paketlərin yerinə yetirdiyi problemlərin həlli üçün fərdi addımlara qoşulmağa imkan verən giriş nəticəsində meydana gələn məlumatların təşkili üçün xüsusi bir sistemin olması ilə təmin edilir.



Şəkil 3.2. DISUPPP sisteminin arxitekturu.

DISUPPP-nin dinamik qurulması aparatına aşağıdakı nəzarət funksiyalarını yerinə yetirməlidir:

- TPP-də modulların icrasının dayandırılması;
- TPP-də modulların icrasının ardıcılığını dəyişdirmək;
- paketlərin ümumi yaddaş bloklarına aid olan və ya modulun içərisində və ya xarici yaddaşda olan bəzi obyektlərin (parametrlərin) qiymətlərinin dəyişdirilməsi.

Dinamik bir quraşdırma aparatının inkişafı marşrut sistemlərində işləyən paketlərə aşağıdakı tələblər qoyulması ilə mümkün oldu:

- bütün modulların vahid instrumental proqramlaşdırma dilində tətbiqi; bu halda modullara və istifadə etdikləri yaddaşa giriş bir instrumental proqramlaşdırma sisteminin köməyi ilə mümkündür;
- modulun özündə modifikasiya edilə bilən hissələrinin olmamasını nəzərdə tutan modulların rentabelliği;

- hər bir paketdə məlumatların təşkili və idarə olunması funksiyalarının yerinə yetirilməsi;

- emal obyektlərinin qeydə alınması – TPP-nin modullarının və parametrlərinin. Modullar arasında real əlaqə (altqraflar və onun bütün modulları arasında əlaqə olduqda) operativ yaddaşa yükləmə səviyyəsində istifadə olunan proqramlaşdırma alətləri paketində qəbul edilmiş sistem konvensiyalarına uyğun olaraq qurula bilər.

Parametrlər haqqında məlumatlar: adlar, formatlar, təqdimetmə üsulları, yaddaşdakı yerləşmə - TPP-də məsələnin həlli üçün istənilən anda mövcud olan cədvəllərdə qruplaşdırılmışdır.

Paketlərin qoşulması üçün altsistemlər. Yeni TPP-lərin öz domen modellərini göstərərək bağlantısı olan marşrutlaşdırma sistemlərinin yaradılması prosesi, marşrutlaşdırma qrafının təfərrüatını və müstəqil altsistemlərin qurulmasını əhatə edir: dialoq dilinin və informasiya xidmətinin altsistemlərinin tərcüməçisini.

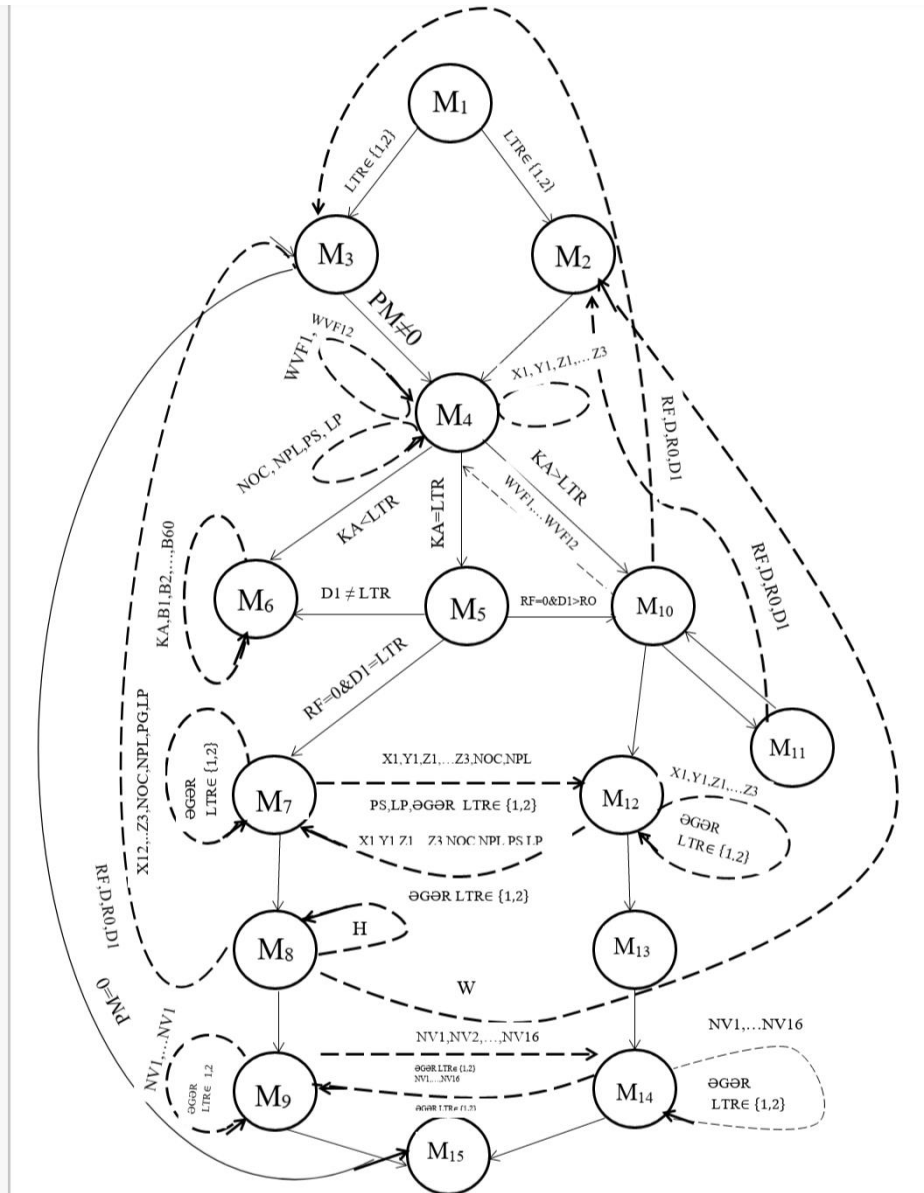
Dialoq dilinin tərcüməçisi istifadəçi təlimatlarını yenidən kodlaşdırır və başladır və sonra lazımlı cədvəlləri (xüsusən paket parametrlərindən ibarət cədvəlləri) çevirən bir sintaktis idarə olunan və xüsusi mövqe qrammatikasının bir sinfinə əsaslanan kompilyator kimi reallaşır və bunun vasitəsi ilə dilin sintaksis Kobol, PL/1, Fortran, yəni, alternativ mötərizədə və standart mötərizədən istifadə edilərək BNF şəklində verilir. Marşrut sisteminin yaradılması prosesində, semantikasi əsasən məsələlərin həllini TPP-yə çevirmək üçün mümkün istifadəçi hərəkətlərinin siyahısı ilə müəyyənləşdirilən direktivlərin sintaksisini dəyişdirmək çərçivəsində dialoq dilini çevirmək mümkündür.

Qeyd etmək lazımdır ki, marşrut sistemlərinin yaradılması prosesinin bir xüsusiyyəti var - tərcüməçi iki növ dialoq dilinə icazə verir: qoşulan paketlərin qeyd olunmuş konkret parametrləri dəstinə, hansı ki, sonlu sayda müəyyən emal obyektlərinə müvafiqdir və ənənəvi adlandırma və formatlaşdırma aparatına malik olan qeyd olunmuş parametrlər dəstinə. Konkret dialoq dilinin tipinə görə emalı iki leksik tərcüməçi analizatordan birini qoşmaqla həyata keçirilir.

Bir marşrutlaşdırma qrafının göstərişinə görə məsələlərin həll qrafının əldə edilməsi, qoşulan paketin tərtibçiləri - yaradılan marşrut sisteminin administratorları ilə dialoqda həyata keçirilir, bu administratorlar, paket ilə işləyərkən paket modullarının yeri və əlaqələri barədə bəzi təfərrüatlara aydınlıq gətirir və düzəliş qövsləri verirlər. Paket parametrləri haqqında informasiya: adlar, formatlar, təqdimat üsulları, yaddaşa yerləşdirmə - paketin instrumental proqramlaşdırma sistemi tərəfindən çevrilməsi prosesində əldə edilən cədvəllərdən seçilir.

Marşrutlayıcı qraf administrator tərəfindən aşağıdakı qaydalara uyğun olaraq təyin edilir. Qrafın bütün qövsləri $M_i \rightarrow M_j (P_{ij})$ üçlüyü şəklində verilmişdir, burada M_i - çıxış ucu, M_j - daxil olan uc, π_i - həll yolunun seçilməsinin şərtidir / fərz edilir ki, $\Pi_{ij} = \emptyset$ / Qövslərin sayılması qrafı kökündən çıxan qövslərin birindən başlayır; qövslərin sıralanma qaydası əhəmiyyətli deyildir. Marşrutlayıcı qrafın uclarının adları paket modullarının adlarına uyğun olmalıdır.

Düzəldici qövslər də i administrator tərəfindən $M_i \rightarrow M_j (C_{ij})$ üçlüyü şəklində verilir, və $C_{ij} = P_1, \dots, P_k (J_{ij}, T_{ij})$. Burada $P_1 \dots P_k$ paketin düzəldilən parametrlərinin adlarıdır, Y_{ij} -də düzəldilən qövslərin bir ucdan çıxması və düzəldilən parametrlərin kəsişən dəstələrinin olması şərtidir. Əgər, şərt yoxdursa, $Y_{ij} = \emptyset$. T_{ij} , Y_{ij} şərtləri ödəndikdə $P_1 \dots P_k$ parametrlərini tənzimləmək üçün sistem tərəfindən verilən diaqnostik mesajın mətnidir.



Şəkil 3.4.

Düzəldici qövslər üzrə çıxış nöqtələrini fərqləndirmək üçün paketlərin müvafiq modullarında dəyişikliklər edilir, nəticədə paketlərdə məsələlərin həlli zamanı parametrləri tənzimləmək üçün modullardan bütün çıxışlar sistemlə birmənalı olaraq identifikasiya edilir ki, bu da istifadəçi ilə əlaqə qurmağa imkan verir: ekrana müvafiq diaqnostik məlumatı çıxarır və istifadəçinin cavabını reallaşdırır. Aparatda çıxış nöqtələrinin fərqləndirilməsi zərurəti məsələnin həll qrafının növü ilə müəyyən olunur ki, bu da bir ucdan bir neçə düzəldici qövsün çıxmasının mümkünlüyünə yol verir.

SIO altsistemi. Müasir proqramlaşdırma sistemlərinə xas olan informasiyanın çevrilməsi proseslərinin mürəkkəbləşməsi, bu məlumatın məntiqi quruluşunu

şərtləndirərək, funksional modulların verilənlərdən ayrılması prinsipinin istifadəsini zəruri edir, hansı ki, yalnız funksional modulların verilənlərdən ayrılmasının modul prinsipinə riayət olunduğu təqdirdə tətbiq olunur və böyük proqram sistemlərində verilənlərin ƏS-də işləmə proqramlarından müstəqillik prinsipi kimi, VBİS-lərində və TPP ilə əlaqəli VBİS kompleksində verilənlərin məntiqi və fiziki müstəqilliyi prinsipi kimi istifadə edilir.

Sistemlərin funksional modullarının verilənlərdən ayrılması prinsipindən istifadə aşağıdakılara imkan verir:

- funksional alqoritmlərin informasiya massivlərinin dövrü emalı ilə əlaqəli fraqmentlərini verilənlərin mürəkkəb yaddaş növlərindən istifadə etməklə mürəkkəb təşkil edilmiş nümunələrinin emal dövrünü təyin edən fraqmentlərlə əvəz etməklə funksional alqoritmləri asanlaşdırmaq: cədvəllər, siyahılar, dükanlar, növbələr və s .;

- funksional alqoritmləri onların strukturlarını deyil verilənlərin görünüşündə mümkün dəyişikliklər aparmaqla unifikasiya etmək olar.

Nəticədə, funksional alqoritmlərin verilənlərdən ayrılması prinsipinin tətbiqi proqram yaratmaq sistemlərində əhəmiyyətli səmərə verir (məsələn, ДИСПОМ və ya ДИФИП sistemləri). Bu prinsip sistemin vahid informasiya bazasına verilənlərin daxil edilməsi və istinad funksiyalarını yerinə yetirən və informasiyanın dəyişdirilməsi zamanı informasiya bazasının çevrilməsi prosesini asanlaşdırmaq üçün hazırlanmış xüsusi, informasiya xidməti alt sisteminin yaradılması ilə həyata keçirilir.

SİO-lər üçün informasiya xidmətlərinin oxşar alt sistemi DESUPPP-də TPP-nin predmet sahələrinin modellərinə xidmət etmək üçün hazırlanmışdır. SİO-nun funksiyalarına marşrutlaşdırma dərəcələrinin təqdim edilməsi və EHM-larının yaddaşında xüsusi bir növ pleks şəklində məsələlərin həlli və paketin parametrlərinin məsələnin həllinin gedişatına dair məlumatlara görə tənzimlənməsi barədə mesajların verilməsi daxildir.

Marşrutlaşdırma qraflarının SİO çərçivəsində məsələlərin həll qraflarına çevrilməsi üzrə hərəkətlərin reallaşdırılması məqsədilə plekslərin çevrilməsi

əməliyyatları (pleksləri qövslər ilə uzatmaq və ucları alt nöqtələrlə əvəz etmək) işlənilib hazırlanmışdır ki, bu da analoji pleks cəbrini əmələ gətirir. Plekslərin qövslərlə uzadılması əməliyyatı düzəldici qövslər qurarkən marşrutlaşdırma qrafının tərkib hissələrini tədqiq etmək üçün istifadə olunur. Bu əməliyyatın yerinə yetirilməsi zamanı paketlərin predmet sahələrinin müfəssəl modellərinin semantik aspektləri həyata keçirilir: tənzimlənən parametrlər dəstlərinin kəsişməməsi və ya bir ucdan çıxan düzəldici qövslərinin seçim şərti olduqda dəstlərin kəsişməməsi tələbi və düzəldici qövsün giriş ucunun marşrutlayıcının bölünməz addımına uyğunluğu. Ucların altqraflarla əvəz edilməsi əməliyyatı məsələlərin həll qraflarında M_i altqraflarının qurulması zamanı vacibdir. Əvəzləmə o vaxt aparılır ki, M_i altqrafın uclarının sayı, marşrutlaşdırma qrafının uclarından çıxan qövslərin sayı ilə eyni olsun.

Marşrut sistemlərinin yaradılması təcrübəsi. Hal-hazırda DISUPPP vasitəsi ilə bir sahədən olan beş TPP-də məsələlərin həllinə operativ nəzarət üçün istifadə olunan СИЕЦКАП marşrut sistemi hazırlanmışdır.

DISUPPP-a qoşulmadan əvvəl, СИЕЦКАП dəzgahlarda hissələrin səthi təmizlənməsi məsələlərini həll etmək üçün fərqli TPP-lər toplusu idi. DISUPPP-nin dialoq dili ilə bu TPP-lərin giriş dilləri birlikdə proqram təminatı sinifinə aid edilə bilər və dəzgahlara nəzarət üçün proqram təminatı dillərinin ilk işləmələrindən biridir.

СИЕЦКАП işləmələrinə sistemli bir yanaşmadan istifadə edilməsi, istehsal olunan hissələr haqqında informasiyanın emalında məsələnin həllinin bütün mərhələlərində texnoloq-istifadəçinin operativ müdaxiləsini təmin etmək üçün yeni bir sxem yaratmağa imkan verdi. Marşrut sisteminin dialoqa yönəldilməsi, onu sazlama məqsədləri üçün istifadə etməyə və mürəkkəb hissələrin emalı üçün istifadə olunan proqramların istehsalında əmək xərclərini azaltmağa imkan verir.

Bu marşrut sistemini inkişaf etdirərkən, TPP-də dialoq yönümünün olmaması və TPP-nin quruluşu və seqmentləşdirilməsi ilə doldurulan məhdud yaddaş miqdarı səbəbindən çətinliklər yarandı. İnkişaf etmiş marşrutlaşdırma sistemi qədər İООК RAM tələb edir

Hazırlanan marşrut sistemlərinin operativ yaddaşının həcmi həddən artıq çoxdur. Marşrut sisteminin ümumi həcmnin əsas hissəsini DİSUPPP altsistemi təşkil edir. Bütün beş TPP-nin işlənilib hazırlanmış marşrut sisteminin instrumental dili Fortran dilidir ki, bu da dinamik qurulma aparatlarının reallaşmasını çox asanlaşdırdı. Gələcəkdə, PL/I, Algol-60 proqramlaşdırma sistemləri ilə, əksər hallarda (Fortran ilə birlikdə) ixtisaslaşdırılmış TPP-lərin həyata keçirilməsi üçün istifadə olunan vahid sistem razılaşmalarının əldə edilməsi üçün bu aparatın genişləndirilməsi planlaşdırılır.

Nəticə və təkliflər

Tətbiqi proqramlar paketləri tətbiq olunan məsələlərin həllində insanın kompüterlə ünsiyyətinin səmərəliliyini artırmaq üçün bir vasitə rolunu oynayır. Hər bir xüsusi paketin hazırlamasının müvəffəqiyyəti avtomatlaşdırılmış tətbiqetmə fəaliyyətinin xüsusiyyətlərinin hərtərəfli nəzərə alınmasına əsaslanır.

Hesablama texnikasından istifadə sahəsində əhəmiyyətli problemlərdən biri müxtəlif tətbiqi məsələlərin həlli zamanı hesablama maşını ilə insanın ünsiyyət problemi olur. Bu cür ünsiyyətin effektivliyini artırmaq üçün müvafiq alqoritmlərin və proqram təminatlarının inkişafı və təkmilləşdirilməsi sahəsində daimi səylər tələb olunur.

Maşınla insanın ünsiyyətinin təşkili vasitələrinin hazırkı inkişaf mərhələsində kompüter sistemlərinin tematik ixtisas səviyyəsinin yüksəldilməsi məsələnin həlli formalarına nisbətən əvvəlkindən daha mükəmməl və xüsusi diqqət yetirilir. Konkret olaraq bu o deməkdir ki, proqramlaşdırmanın vacib məqsədlərindən biri istifadəçilərin maşın tərəfindən təqdim olunan "hesablama xidmətləri" nə daha rahat girişini təmin edən və minimal peşəkar –proqramçı səviyyəsi tələb edən "dost" vasitələrin inkişafı olmuşdur.

Hesablama sistemlərinin tematik ixtisaslaşma səviyyəsinin yüksəldilməsi fərdi əlavələrə yönəldilmiş proqram təminatının ixtisaslaşdırılmış komponentlərini yaratmaqla həyata keçirilir

Hazırkı dövrdə hesablama maşınlarının köməyiylə həll edilən mühüm məsələlər hesablama təcrübəsi məsələləridir. Qurğu və proseslərin riyazi modelinin qurulması və bu modelin hesablamalarla tədqiqi vasitəsilə öyrənilməsi metodu hesablama təcrübəsi adlanır. Riyazi modeli reallaşdıran proqrama təbii təcrübədə istifadə olunan təcrübə qurğusunun analoqu kimi baxmaq olar. Hesablama təcrübəsi analitik tədqiqat və təbii təcrübə arasında aralıq mövqe tutur. Analitik tədqiqatda da riyazi model qurulur, lakin model o qədər sadə olur ki, onun öyrənilməsi üçün analitik hesablama kifayət edir və hesablama texnikasının istifadəsinə əsaslanan ədədi metodların cəlb olunmasına ehtiyac qalmır. Baxmayaraq ki, verilən model üçün analitik yanaşma çox vaxt dəqiq həll verir, bu

sadələşdirilmiş model adətən praktiki hesablamalarda qiymətləndirmələrin tələb olunan dəqiqliyini əldə etməyə imkan vermir. Hesablama təcrübəsində əldə edilən ədədi metodlar tədqiq olunan prosesləri kifayət qədər dolğun əks etdirən və daha mürəkkəb olan modelləri öyrənməyə imkan verir.

Sistem proqram təminatı (SPT) (Riyazi dəstək sistemləri (RDS)) kompüterlərin yaradılması üzrə iş həcminin yüksək artımına səbəb olan kompüter texnologiyalarının sürətlə böyüməsi və onun tətbiqi sahəsinin daim genişlənməsi ilə əlaqədar olaraq müasir kompleks kibernetikada böyük komplekslər, xüsusən də alətlər istehsal etmək məqsədi ilə proqramlaşdırmanın avtomatlaşdırılması metodları intensiv inkişaf edir. Tətbiq olunan proqram təminatı (RT), problem yönümlü sistemlər (proqram sistemləri) və fərdi yönümlü sistemlərlə təyin edilə bilər. Sonuncular, bu sistemin tətbiqi sahəsində çalışan insanların kontingentinin peşə lüğətini əks etdirən bir dildə istifadəçilərlə dialoq vasitələri ilə xarakterizə olunur.

Tətbiq olunan SPT (RDS) vasitələrinin yaranmasının mürəkkəb və çoxölçülü probleminin həllində əsasən instrumental sistemlərin inkişafı ilə əlaqəli ayrı-ayrı istiqamətlər xüsusən tətbiqi proqram paketlərinin generatorları /TPP/ ilə əlaqələndirilmişdir. TPP generatorları, bir meta-alqoritm quraraq, problemlərin müəyyən bir hissəsini həll etmək üçün daha dar bir sinif sinfinə yönəlmiş bir proqram sisteminin müəyyən bir versiyasını seçməyə imkan verir.

Mövcud SPT (RDS)-nin bir sıra komponentlərindən, bu seriyanın ayrı-ayrı komponentləri tərəfindən həll olunan siniflərin böyüdülməsindən ibarət olan problemlər üçün proqram versiyalarını tərtib etməyi mümkün edən genişləndirici proqram sistemlərini çağırırıq.

Problemin belə bir həlli DISUPPP sistemində tətbiq olunan yeni bir riyazi dəstək yönümlü proqram təminatının layihələndirilməsinin texnologiyasının əsasını təşkil edir. Sistem, həll olunan problemlər sinfinin genişlənməsi məqsədi ilə ixtisaslaşmış TPP-lərini birləşdirmək üçün hazırlanmışdır.

İstifadə edilmiş ədəbiyyatın siyahısı

1. .Самарский А. А. Математическое моделирование и вычислительный эксперимент // Вестник АН СССР.- 1979
2. Говорун Н. Н., Иванченко И. М., Нефедьева Л. С.. Диалог в системах автоматизированной обработки данных // Управляющие системы и машины.- 1974
3. Говорун Н.Н., Иванченко И.М. Математическое обеспечение многоабонентной системы для обмера снимков с трековых камер // Программирование.- 1976.- № 4.- С.52-65.
4. Гуляев Ю. П., Петров В. И. Формирование библиотеки программных модулей для множества проблемно-ориентированных задач // Программирование. -1976.
5. Родионов С. Т. Основные концепции модульности и анализ некоторых направлений развития общей технологии программирования // Программирование.-1980.
6. Бабаев И. О., Новиков Ф.А., Петру-шина Т. И. Язык Декарт - входной язык системы СПОРА // Прикладная информатика: Сб. статей. Вып.1. / Под ред. В. М. Савинкова. - М.: Финансы и статистика, 1981.
7. Мартынюк В. В. О методе символических адресов // Проблемы кибернетики: Сб.статей. Вып.6 / Под ред.А.А.Ляпунова.-М.: Наука, 1961.
8. Говорун Н. Н., Иванченко И. М., Нефедьева Л. С.. Диалог в системах автоматизированной обработки данных // Управляющие системы и машины.- 1974.- № 1.-С.8-13.
9. Бабаев И. О., Новиков Ф.А., Петру-шина Т. И. Язык Декарт - входной язык системы СПОРА // Прикладная информатика: Сб. статей. Вып.1. / Под ред. В. М. Савинкова. - М.: Финансы и статистика, 1981.- С. 35-73
10. Мартынюк В. В. О методе символических адресов // Проблемы

- кибернетики: Сб.статей. Вып.6 / Под ред.А.А.Ляпунова.-М.: Наука, 1961.-С. 45-58
11. Жоголев Е. А. Принципы построения многоязыковой системы модульного программирования // Кибернетика,- 1974.- № 4.- С. 1-5.
 12. Воеводин В. В., Гайсарян С. С, Кабанов М. И. Автоматизированная генерация программ // Численный анализ на Фортране: Сб. статей. Вып.1.-М.: Изд-во МГУ, 1973.-С.3-13,
 13. Зизин М. К, Загацкий Б. А., Темноева Т. А., Ярославцева Л.Н. Автоматизация реакторных расчетов.-М.: Атомиздат, 1974.
 14. Столяров Л. Н. Краткий обзор принципов организации пакетов программ // Труды IV Всесоюзного семинара по комплексам программ математической физики.- Новосибирск, 1976.
 15. Тамм Б. Г. Описание языка САП-2 для программирования работы станков с ЧПУ // Алгоритмы и алгоритмические языки: Сб. статей. Вып.5.-М.:, 1971.-С.67-78.
 16. Прууден Ю. И. АПРОКС - специализированный язык с проблемной ориентацией // Алгоритмы и алгоритмические языки: Сб. статей. Вып.5.- М.:, 1971.-С.79-93.
 17. Бухштаб Ю. А., Горлин А. И., Камынин С. С. и др. Об одном методе планирования расчетных цепочек // Программирование.-1981.- №3.- С.34-38.
 18. Корухова Л. С., Любимский Э.З. Система планирования действий, основанная на некоторых естественных принципах - ПЛАР.- Препр. / .-М., 1978.-№53.
 19. Литвинцев Н. И. Планирование вычислений в диалоговой системе долгосрочного планирования //-1980.-№ 4.-С.21-28.
 20. Жук В. И., Малашинин И. И. Системы баз данных, ориентированные на применение при автоматизации физических экспериментов. Состояние проблемы. Обзор.-М.: ЦНИИАтоминформ, 1986.-61 с.
 21. Мальковский М. Г. Диалог с системой искусственного интеллекта.-

М.: Изд-во МГУ, 1985.

22. Корягин Д. А., Кротов В. И., Сигалев В. А. Система ПНФ. Технология разработки пакетов прикладных программ.-Препр. / ИПМ АН СССР.-М., 1986.-№151.-26 с.
23. Горелик А.М., Корягин Д. А., Кротов В. И., Луховицкая Э. С. Пакет ЗАЩИТА. Общее описание языка заданий.- Препр. / ИПМ АН СССР.-М., 1982.-№207.-27 с.
24. Волощенко А.М., Корягин Д. А., Кротов В. И., Луховицкая Э.С. Пакет ЗАЩИТА. Системное и функциональное наполнение. -Препр. / ИПМ АН СССР.-М., 1984.-№16.- 28с
25. Горбунов-Посадов М.М. Изменяемые программы и однородные модули// Программирование.-1988.-№ 4,-С. 38-49.
26. Горбунов-Посадов М. М., Ермаков А. В., Карпов В. Я. и др. КРИТ - пакет прикладных программ для решения многокритериальных задач оптимального проектирования объектов машиностроения.-Препр. /.- М., 1985.-№175.-29 с.
27. Волощенко А. М., Гермогенова Т. А., Исаенко Т. Г. и др. Объединенная система константного обеспечения - ОКС. Версия 3.0.-Препр. / М., 1984.-№ 20.-30 с.
28. Бухштаб Ю.А., Гордин А. И., Камынин С. С. и др. Об одном методе планирования расчетных цепочек // Программирование.- 1981.-№3.- С. 34-38.
29. Лаймен Дж. Новые средства предотвращения аварий на АЭС // Электроника.-1986.-№ 11.-С.3-5.
30. Каталог диалоговых систем: материалы по математическому обеспечению—Киев: Ин-т кибернетики АН УССР, 2006, —232 с.

Р Е З Ю М Е

Пакеты прикладных программ играют роль инструмента для повышения эффективности общения человека с компьютером в решении прикладных задач. Успех разработки каждого конкретного пакета основан на тщательном учете особенностей автоматизированной деятельности приложений.

Одной из важных проблем в области использования вычислительной техники является проблема общения человека с вычислительной машиной при решении различных прикладных задач. Для повышения эффективности такого общения требуются постоянные усилия в области разработки и совершенствования соответствующих алгоритмов и программного обеспечения.

На современном этапе развития средств организации общения человека с машиной повышение уровня тематической специализации компьютерных систем является более совершенным и уделяет особое внимание, чем прежде, формам решения вопроса.

Повышение уровня тематической специализации вычислительных систем осуществляется путем создания специализированных компонентов программного обеспечения, ориентированных на отдельные приложения.

Системное программное обеспечение (системы математической поддержки (RDS)) связано с быстрым ростом компьютерных технологий и постоянным расширением сферы его применения, что приводит к увеличению объема работ по созданию компьютеров. Поэтому в современной комплексной кибернетике интенсивно развиваются методы автоматизации программирования с целью производства крупных комплексов, особенно инструментов. Применяемое программное обеспечение может быть определено проблемно-ориентированными системами и индивидуально ориентированными системами. Последние характеризуются диалогом с пользователями на языке, отражающем профессиональный словарь контингента людей, работающих в области применения данной системы.

S U M M A R Y

Application software packages play the role of a tool for improving the efficiency of human-computer communication in solving applied problems. The success of developing each specific package is based on careful consideration of the features of automated application activity.

One of the important problems in the use of computer technology is the problem of communication between a person and a computer when solving various applications. To improve the effectiveness of such communication, continuous efforts are required in the development and improvement of appropriate algorithms and software.

At the present stage of development of the means of organizing human-machine communication, increasing the level of thematic specialization of computer systems is more advanced and pays special attention than before to the forms of solving the issue.

Increasing the level of thematic specialization of computing systems is carried out by creating specialized software components focused on individual applications.

System software (mathematical support systems (RDS)) this is due to the rapid growth of computer technology and the constant expansion of its scope, which leads to an increase in the volume of work on creating computers. Therefore, modern complex Cybernetics is intensively developing methods for automating programming in order to produce large complexes, especially tools. The software used can be defined by problem-oriented systems and individually oriented systems. The latter are characterized by a dialogue with users in a language that reflects the professional vocabulary of the contingent of people working in the field of application of this system.