

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

**MAGİSTRATURA MƏRKƏZİ**

*Əlyazması hüququnda*

**İslamova Tünzalə Mehman qızı**

**“İnstrumental proqramlaşdırma sistemlərinin yaradılması texnologiyasının bəzi  
problemləri” mövzusunda**

**MAGİSTR DİSSERTASIYASI**

**İstiqamətin şifri və adı: 060509 “Kompüter Elmləri”**

**İxtisasın şifri və adı: “İqtisadi informasiya sistemləri”**

**Elmi rəhbər dos. Musayev M.N.**

**Magistr proqramının rəhbəri dos. Bayramov H.M.**

**Kafedra müdiri dos. Bayramov H. M**

**BAKI – 2018**

## MÜNDƏRİCAT

<b>GİRİŞ.....</b>	<b>3</b>
<b>FƏSİL I. Problem - oriyentasiyalı sistemlər.....</b>	<b>6</b>
1.1 Problem - oriyentasiyalı sistemlərin yaradılması səbəbləri .....	6
1.2 Problem – oriyentasiyalı sistemlərin inkişaf istiqamətləri.....	9
1.2.1. Avtomatlaşdırılmış informasiya sistemləri.....	10
1.2.2. Tətbiqi proqramlar paketləri.....	13
1.2.3. İnformasiya – axtarış sistemləri.....	20
1.2.4. Ekspert sistemlər.....	24
1.3 Problem – oriyentasiyalı sistemlərin generasiyası problemi.....	28
<b>FƏSİL II. Tətbiqi proqram təminatının layihələndirilməsi metodları.....</b>	<b>34</b>
2.1. Proqram təminatının yaradılmasının seçilməsi metodları.....	34
2.2. Proqram təminatının yaradılması metodlarının inkişafı.....	37
2.3. Proqram təminatının hazırlanması metodları.....	44
2.4. Proqramların hazırlanması strategiyaları.....	48
<b>FƏSİL III. İnteraktiv marşrutlu sistemlərin yaradılması texnologiyas....</b>	<b>54</b>
3.1. İnteraktiv marşrutlu sistemlər.....	54
3.2. İnteraktiv marşrutlu sistemlərin tətbiq sahəsinin modeli.....	60
3.3. Marşrut qrafları cəbri.....	69
3.4. Tətbiq oblastı modelinin layihələndirilməsinin modul prinsipi .....	74
 <b>NƏTİCƏ VƏ TƏKLİFLƏR.....</b>	 <b>78</b>
 <b>İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT.....</b>	 <b>80</b>

## GİRİŞ

Son illərdə riyazi metodların insan fəaliyyətinin müxtəlif sahələrində geniş tətbiqi proqram fondlarındakı tətbiqi proqramların sayını nəzərə cərpacaq dərəcədə artırmışdır. Eyni zamanda iqtisadiyyatın, təşkilatı idarəetmənin, elmi tədqiqatların, tibbi və digər sahələrin kompüterləşdirilməsi, son istifadəçilərin, başqa sözlə, müxtəlif ixtisaslı mütəxəssislərin kompüterlə bilavasitə əlaqəsini tələb edir. Bu mütəxəssislərin əksəriyyəti tətbiqi riyaziyyatı və proqramlaşdırmanı bilmirlər və yaxud onların elementləri ilə bu və ya digər dərəcədə tanışdırlar. Bundan başqa cəmiyyətin hesablama texnikası vasitələrindən istifadə etməklə idarəetməyə və informasiyaların işlənməsinə olan sosial və iqtisadi tələbatı durmadan yüksəlir. Bununla əlaqədar olaraq, elə əlverişli və istifadə üçün sadə proqram sistemlərinin yaradılması problemi ortaya çıxır ki, son istifadəçilər öz müxtəlif məsələlərini kompüterdə həll edə bilsinlər. Fərdi kompüterlərin çox geniş yayılması bu problemin aktuallığını daha da artırmış oldu.

**Mövzunun aktuallığı.** Instrumental iribloklu sistemlərin yaradılmasında qarşıya qoyulan əsas məqsəd, tətbiqi məsələlərin həlli üsullarının bildiyimiz mərhələlərini avtomatlaşdırmaqla, müasir kompüterlərin vasitəsi ilə son istifadəçilərin işinin asanlaşdırılmasına xidmət etməkdir.

İnsan fəaliyyətinin bütün sahələrinin kompüterləşdirilməsinin yetmiş illik dövrü ərzində kompüterlərin nəinki sosial funksiyaları və imkanları çox böyük sürətlə genişlənmiş, onların qiymətləri xeyli ucuzlaşmış, həm də kompüterdə məsələlərin həllinin alqoritmləşdirilməklə reallaşdırılması konsepsiyası kökündən dəyişmişdir. Hal hazırda sadə infarmasiya sistemlərinin yerini daha mükəmməl və kefiyyətə yeni qərarların qəbul olunmasının avtomatlaşdırılması sistemləri tutmaqdadır.

Modelləşdirmə mürəkkəb sistemlərin tədqiqatı aləti olmaqla, bu sistemlərin məsələlərin həllinin çox mürəkkəb, qoyuluşlarının qeyri dəqiq olmasına baxmayaraq, idarə etmə kefiyyətinin yüksəldilməsinə və onların əsasında düzgün proqnozların verilməsinə imkan yaradır. Son illərdə tədqiq olunan obyektlərin və

proseslərin modellərinin böyükmodellə üsullarla qurulmaları geniş vüsət almışdır. Belə modellər imkan verir ki, qərarların qəbul olunması prosesində çoxsəviyyəli idarə etmə zamanı informasiyaların natamam olması şərtini aradan qaldırsın.

Tətbiq oblastlarının kompüterdə modelləşdirilməsinin ən effektiv vasitələrindən biri, konkret tətbiq oblastından toplanmış təcrübə və kompüterlə istifadəçini peşə dilinə yaxın olan dildə qarşılıqlı surətdə əlaqələndirən, problem-orientasiyalı sistemlərdir (PO-sistemlər). Bu sistemlərin elm və texnikanın bütün sahələrində müvəffəqiyyətlə tətbiqi, onların yaradılması metodlarının tədqiqinin aktual olduğu aşkar görünür.

**Tədqiqatın məqsədi.** Problem-orientasiyalı sistemlərin, o cümlədən intellektual tətbiqi proqramlar paketlərinin arxitekturu öyrənmək, bu sistemlərdə tətbiq sahəsi modellərinin şərh üçün istifadə olunan üsulları araşdırmaq, təsnifat aparmaq, dialoqlu marşrut sistemlərinin tətbiq sahəsi modelinin yaradılması üçün təlimat hazırlamaq, problem-orientasiyalı sistemlərdə hesablama prosesinin planlaşdırılması metodlarını öyrənmək tədqiqatın məqsədinə daxildir.

**Tədqiqatın obyekt.** Problem - orientasiyalı sistemlər, bu sistemlərin tətbiq sahələrinin modelləri və dialoqlu marşrut sistemlərin tətbiq sahəsinin elementləri, onlar arasındakı əlaqələrin şərh üsulları, informasiya bazasının layihələndirilməsi sxemi, dialoqlu - marşrut sistemlərdə məsələnin həlli alqoritmi tədqiqat obyekt kimi araşdırılmışdır.

**Tədqiqatın nəzəri və praktiki əhəmiyyəti.** Problem - orientasiyalı sistemlərin yaradılması səbəbləri araşdırılmış, bu sistemlərin inkişaf istiqamətləri, avtomatlaşdırılmış informasiya sistemləri, tətbiqi proqramlar paketləri, informasiya - axtarış sistemləri, ekspert sistemlər və onların generasiyası problemi öyrənilmiş, problem-orientasiyalı sistemlərin tətbiq sahəsi modellərinin təsnifatı aparılmış və dialoqlu marşrut sistemlərinin tətbiq sahəsi modelinin layihələndirilməsi üçün modul prinsipi təklif olunmuşdur. Marşrut qraflarının tillərinin uyuşanlığının yoxlanması üçün mövcud olan «cüt-cüt yoxlama» metodu təkmilləşdirilmişdir.

Dissertasiya işinin birinci fəslə – «Problem - orientasiyalı sistemlər»- problem - orientasiyalı sistemlərin yaradılması probleminə həsr olunmuşdur. Bu sistemlərin

yaradılması səbəbləri araşdırılmış, onların inkişaf istiqamətləri öyrənilmişdir. Konkret olaraq avtomatlaşdırılmış informasiya sistemlərinin, tətbiqi proqramlar paketlərinin, informasiya - axtarış sistemlərinin və ekspert sistemlərin arxitekturu öyrənilmiş və problem-orientasiyalı sistemlərin generasiyası probleminə baxılmışdır.

İkinci fəsil - «Tətbiqi proqram təminatının layihələndirilməsi metodları» - instrumental proqramlaşdırma kompleksləri yaradılarkən istifadə olunan tətbiqi proqram təminatının layihələndirilməsi və istifadəsi metodlarının yaradılması probleminə həsr olunmuşdur.

Tətbiqi roqram təminatının layihələndirilməsi və yaradılması metodları ətraflı araşdırılmış, proqram təminatında əsasən səkkiz əsas müddəadan ibarət olduğu qeyd olunmuş, eyni zamanda kompilyasiya prosesində buraxılan səhvlərə, yaxud yerinə yetmə prosesində kəsilmələrə malik olmaması, məsələnin spesifikasiyalarını ödəyən bütün mümkün olan verilənlər topluları üçün düzgün nəticənin alınmasının təmin olunması dəqiqləşdirilmiş, PDL- proqramların layihələndirilməsi dili, həm də tapşırıqların müəyyən olunması dili və tapşırıq analizatoru araşdırılmış və bu dillərin problem-yönlü dillərlə müqayisəsi aparılmışdır.

Üçüncü fəsil - «İnteraktiv marşrutlu sistemlərin yaradılması texnologiyası»- tətbiq sahəsi modelinin marşrut qrafları ilə şərh olunduğu xüsusi dialoqlu problem-orientasiyalı sistemlərin bir sinfinə həsr olunmuşdur. Bu sinif problem-orientasiyalı sistemlər DİSUPPP instrumental kompleksi vasitəsilə generasiya olunurlar və dialoqlu marşrut sistemlər adlanırlar. Bu fəsildə məsələnin həlli marşrutunun seçilməsinin bir qiymətlilik xassəsi araşdırılmış, marşrut qrafları cəbri və tətbiq sahəsi modelinin layihələndirilməsinin modul prinsipi təklif olunmuşdur. Dialoqlu marşrut sistemlərin hesablaşdırıcısı tədqiq olunmuşdur.

Dissertasiya işi girişdən, üç fəsildən, nəticə və təkliflərdən, istifadə edilmiş ədəbiyyat siyahısından ibarətdir. Dissertasiya işində 21 şəkildən istifadə olunmuş, işin ümumi həcmi 82 səhifədə verilmişdir.

## **FƏSİL I. PROBLEM - ORİYENTASIYALI SİSTEMLƏR.**

### **1.1. Problem - oriyentasiyalı sistemlərin yaradılması səbəbləri.**

İnformasiya kompüter texnologiyasının (gələcəkdə sadəcə olaraq informasiya texnologiyası deyəcəyik, çünki, kompütersiz o hec bir məna kəsb etmir) müasir mərhələsini informasiya ehtiyatlarından istifadə olunması proseslərinin asanlaşdırılması məqsədi ilə informasiyaların toplanması, işlənməsi, saxlanması, yayılması və əks olunmasını təmin edən, vahid texnoloji sırada birləşdirilmiş metodlar, informasiya prosesləri və proqram-texniki vasitələr toplusu kimi müəyyən etmək olar.

İdarəetmənin informasiya texnologiyası informasiyaların işlənməsinin və istifadə olunmasının çox mühüm olan ən azı üç mərhələsini reallaşdırmalıdır: uçot, analiz və qərarların qəbul edilməsi. Bu üç komponentin reallaşması aydındır ki, çoxsaylı sənədlərdə əks olunur və bu sənədlər çoxluğu ildən ilə artaraq işi xeyli çətinləşdirirdi.

60-ci illərdə keçmiş Sovetər İttifaqında avtomatlaşdırılmış idarəetmə sistemləri (AİS-lər) geniş yayılmışdı. Lakin AİS-lərin xalq təsərrüfatında tətbiqi həmişə və tam həcmdə informasiya texnologiyası komponentlərinin optimal reallaşdırılmasına və idarəetmənin təkmilləşdirilməsi məsələlərinə tam cavab vermirdi.

«Avtomatlaşdırılmış idarəetmə sistemləri» adı bu sistemlərin yerinə yetirdiyi funksiyaları heç də düzgün əks etdirmirdi. Bu sistemləri «idarəetməni təmin edən avtomatlaşdırılmış sistemlər» adlandırmaq daha yaxşı olardı. Çünki, mövcud olan AİS-lərdə «sistem» anlayışı idarəetmənin həlledici ünsürü olan istifadəçini nəzərə almırdı. Bununla bərabər AİS-lərin aşağıdakı çatışmayan cəhətləri vardır.

- idarə olunan obyektlərin keçmiş vəziyyətini əks etdirən, əsasən, uçot funksiyalarını reallaşdırırdı, onun inkişaf perspektivlərini qiymətləndirməyə imkan vermirdi, başqa sözlə, idarəetmədə lazım olan dinamizmi təmin etmirdi;

- mərkəzi EHM ilə istifadəçinin avtomatlaşdırılmış iş yeri arasında inkişaf etmiş kommunikasiya vasitələri yox idi, EHM yalnız paket rejimində işləyirdi, dialoqun səviyyəsi çox aşağı olduğundan statistik hesabatların nəticələrini yüksək keyfiyyətə analiz etmək mümkün deyildi;

- aşağı səviyyələrdə, başqa sözlə, verilənlərin toplanması vasitələrinin olmadığı bir vaxtda daxil olan informasiyaların həddindən artıq olması səbəbindən informasiya axınlarının əmələ gəldiyi yerlərdə AİS-lərin səmərəliliyi çox aşağı idi.

Bununla əlaqədar olaraq AİS-lərin inkişafı uçot funksiyalarını yerinə yetirsin işçilərin (qeyd edək ki, onlar idarəetmə orqanlarında çalışanların 1/6 hissəsini təşkil edirdilər) sayının xeyli artmasına səbəb olmuşdu.

70-ci illərdən başlayaraq AİS-lərin inkisaf etdirilməsi sahəsində bir sıra yeni ideyalar və metodlar tətbiq olunmağa başladı. Nəzərdə tutulurdu ki, informasiya təminatının keyfiyyəti, tamlığı, düzgünlüyü, vaxtında verilmə səviyyəsi xeyli yüksələcək.

Lakin AİS-lər gözlənilən nəticəni vermədi. Tətbiq olunan iqtisadi-riyazi modellərin praktikada məhdud imkanlara malik olduğu aşkarlandı.

Digər tərəfdən isə son illərdə riyazi metodların insan fəaliyyətinin müxtəlif sahələrində geniş tətbiqi proqram fondlarındakı tətbiqi proqramların sayını nəzərə cəpacaq dərəcədə artırdı. Eyni zamanda iqtisadiyyatın, təşkilatı idarəetmənin, elmi tədqiqatların, tibbin və digər sahələrin kompüter-ləşdirilməsi, son istifadəçilərin, başqa sözlə, müxtəlif ixtisaslı mütəxəssislərin kompüterlə bilavasitə aktiv qarşılıqlı əlaqəsini tələb edir. Bu mütəxəssislərin əksəriyyəti tətbiqi riyaziyyatı və proqramlaşdırmanı tamamilə bilmirlər və yaxud onların elementləri ilə bu və ya digər dərəcədə tanışdırlar. Bununla əlaqədar olaraq, elə əlverişli və istifadə üçün sadə proqram sistemlərinin yaradılması problemi ortaya çıxır ki, son istifadəçilər öz müxtəlif məsələlərini kompüterdə həll edə bilsinlər. Fərdi kompüterlərin çox geniş yayılması bu problemin aktuallığını daha da artırmış olur.

İxtiyari tətbiq sahəsindən olan məsələlərin kompüterdə həlli texnologiyası yaxşı məlum olan dörd mərhələdən ibarətdir. Birinci mərhələdə obyektin (sistemin, yaxud prosesin) modeli qurulur və bu model tətbiqi model adlanır. Tətbiqi model mütəxəssislər tərəfindən uyğun tətbiq sahəsi haqqındakı biliklər əsasında qurulur. Elə bu mərhələdə obyektlərin xassələrini tədqiq etmək məqsədilə tətbiqi məsələlər

qoyulur. Tətbiqi modelin yaradılması və tətbiqi məsələlərin qoyulması mütəxəssisin, yəni son istifadəçinin peşə dilində həyata keçirilir.

İkinci mərhələdə tətbiqi modelə əsasən obyektin riyazi modeli qurulur, tətbiqi məsələlər üçün isə uyğun riyazi məsələlər qoyulur. Belə çevirmənin həyata keçirilməsi, yəni tətbiqi səviyyədə olan şərh dilindən riyazi model və məsələlərin dilinə keçid adətən konkret tətbiq sahəsindən olan mütəxəssislə riyaziyyatçının birgə fəaliyyəti nəticəsində reallaşır və riyaziyyatçıdan həmin tətbiq sahəsinin spesifikasiyasını bilməyi tələb edir. Bu onunla izah olunur ki, problemin, başqa sözlə, obyektin, yaxud məsələnin tətbiqi sərhəndə çox hallarda bir sıra tətbiqi nöqtəyi - nəzərdən sadə görünən faktorlar buraxılır. Belə «susmalar», onların aşkarlığına baxmayaraq, tətbiq sahəsi haqqında biliklərdən istifadə etməyən heç bir formal sistem tərəfindən bərpa oluna bilmir. Eyni zamanda adekvat riyazi modelin qurulması üçün bütün əsas faktorları nəzərə almaqla problemin tam şərhə zəruridir. Ona görə də bu və ya digər tətbiq sahəsində işləyən riyaziyyatçı müəyyən qədər uyğun sahəni bilməlidir ki, buraxılan faktorları bərpa etsin, yaxud o həmin sahənin mütəxəssisinə müraciət etməlidir. Beləliklə, tətbiqi modeldən riyazi modelə keçid kifayət qədər mürəkkəb elmi-tedqiqat məsələsidir və onun həlli zamanı həm tətbiqi, həm də riyazi biliklərdən istifadə olunur.

Üçüncü mərhələ məsələnin həlli alqoritminin seçilməsi və uyğun proqramların yaradılması mərhələsidir. Bu mərhələdə işçi dil kimi yüksək səviyyəli proqramlaşdırma dillərindən, verilənlərin şərhə və manipulyasiyası dillərindən və s. dillərdən istifadə olunur. Alqoritmlər seçilərkən və proqramlar yazılarkən, yəni riyazi modellər və məsələlər proqram obyektlərinə çevrilərkən həm riyazi model və məsələlərin riyazi xüsusiyyətləri, həm də hesablama riyaziyyatının və sistem proqramlaşdırmasının imkanları nəzərə alınır.

Dördüncü, nəhayət sonuncu mərhələ, məsələlərin bilavasitə EHM-də həlli məsələsidir.

Şərh olunan bu texnologiyanın hətta tətbiqi riyaziyyat və proqramlaşdırma nöqtəyi-nəzərindən kifayət qədər öyrənilmiş tətbiqi problemlərin həlli üçün istifadə olunması onunla nəticələnir ki, problemin qoyuluşundan onun EHM-də həllinə



qədər keçən müddət və yüksək ixtisaslı mütəxəssislərin əmək sərfi kifayət qədər böyükdür. Bunun səbəbi qaçılması mümkün olmayan iterasiyalar və məsələnin qoyuluşundan EHM-ə qədər olan «yolun» dəfələrlə təkrarlanmasıdır.

Problem-orientasiyalı sistemlərin yaradılmasında məqsəd tətbiqi problemlərin həlli texnologiyasının yuxarıda sadalanan mərhələlərini avtomatlaşdırmaqla kompüterləri bu və ya digər tətbiq sahələrinin mütəxəssislərinə yaxınlaşdırmaqdır.

## **1.2 Problem - orientasiyalı sistemlərin inkişaf istiqamətləri.**

Birinci problem- orientasiyalı sistemlərin yarandığı vaxtdan keçən müddət ərzində problem orientasiyalı proqram təminatının yaradılması sahəsində artıq müəyyən nəliyyətlər əldə olunmuşdur: bu sistemlərin yeni yaradılma metodları öyrənilmiş və praktikada sınaqdan keçirilmişdir; çox böyük problem – orientasiyalı sistemlər yaradılmışdır. (məsələn, POLE, PRIZ, VECTOR və s.). BU sistemlərin axırıncı onillikdə müvəffəqiyyətlə istismarı seçilmiş konsepsiyanın məqsədə uyğunluğunu bir daha təsdiq etdi. Problem - orientasiyalı proqram təminatının inkişafı miqyasını aşağıdakı faktla daha aydın göstərmək olar: axırıncı illərdə təkcə Ukraynada Milli Elmlər Akademiyasının V. M. Qluskov adına Kibernetika İnstitutunda avtomatlaşdırılmış informasiya sistemlərinin optimallaşdırma məsələlərinin həlli iyirmidən artıq problem–orientasiyalı sistem yaradılmışdır.

Problem - orientasiyalı sistemlərin yaradılması bir neçə nəsillə EHM-in ümumsistem riyazi təlimatını yaradılması ilə sıx əlaqədə həyata keçirilmiş və sistem proqramlaşdırmasının 60-70-ci illərdəki prosedurlar üçün dil prosessorlarının qurulması sahəsindəki nəliyyətlərə əsaslanmışdır.

Sistem proqramlaşdırmasının çox mühüm nəzəri və praktiki nəticələrindən (modul prinsipi, parametrləşdirmə, translyasiyanın sintaktik idarə olunması, proqram komplekslərinin yaradılması və instrumental vasitələri, hesablama prosesinin planlaşdırılması metodlarının tətbiqi və istifadəçilərə xidmətin

qaydaları) bəhrələnərək, problem - oriyentasiyalı sistemlərin yaradılması metodologiyasını öz müstəqil naliyyətləri ilə zənginləşmişdir.

Birinci növbədə bu verilələrin idarə olunmasının başlıca üsullarının müəyyən edilməsinə aiddir, Problem - oriyentasiyalı sistemin vahid informasiya bazasının təskili metodları, baza şəklində təşkil olunmuş verilələrin xüsusi manipulyasiya üsulları, problem - oriyentasiyalı sistemin modulluluq prinsipinin dəqiqləşdirilməsi (tətbiqi proqramlar verilələrdən asılı olmamasını əldə etməklə) problem - oriyentasiyalı proqram təminatı yaradıcılarına çox böyük təsir göstərmişdir. Bunun nəticəsidir ki, müasir problem - oriyentasiyalı sistemlərin əksəriyyətində xüsusi komponent - informasiya xidməti altsistemi - vardır.

### **1.2.1 Avtomatlaşdırılmış informasiya sistemləri**

İnsanın əmək fəaliyyəti daim ətraf mühit haqqında informasiyaların qəbulu və toplanması ilə əlaqədardır. Bu informasiyalar müxtəlif tip məsələlərin həlli üçün seçilir və işlənir. Bir neçə onillik ərzində bu əməliyyatların kompleksi, metodları və onların reallaşdırılması vasitələri informasiya sistemlərinin yaradılması zərurətini ortaya çıxarmışdır. Belə sistemlərin əsas təyinatı istifadəçiləri ona lazım olan informasiyalarla təmin etməkdir, başqa sözlə desək, istifadəçiyə müəyyən bir tətbiq sahəsindən zəruri məlumatları verməkdən ibarətdir. EHM-lərin meydana gəlməsi hesabına avtomatlaşdırılmış informasiya sistemlərini yaratmaq mümkün olmuşdur.

Avtomatlaşdırılmış informasiya sistemlərinin inkişafında iki nəslə bir-birindən fərqləndirirlər:

Birinci nəslə avtonom fayllara əsaslanaraq yaradılan informasiya sistemləri təşkil edir. Bu sistemlər sadəcə arxitektura malik olmaqla, nisbətən məhdud imkanlara malik idilər. Onlar avtonom fayllar toplusundan və bu faylları emal edən tətbiqi proqramlardan ibarət idilər və istifadəçiyə lazım olan sənədi vermək vəzifəsini yerinə yetirirdilər. Bu sistemlər bir sıra mühüm çatışmamazlıqlara malik idilər. Əvvəla bu sistemlərdə informasiya «artıqlığı» mövcud idi. Bundan başqa faylların təşkili və birgə emalı mümkün deyildi, tətbiqi proqramlar verilələrdən

asılı idilər və s. Bu çatışmamazlıqlar belə quruluşa malik informasiya sistemlərinin tətbiq dairəsini xeyli kiçiltmiş olurdu.

İkinci nəsil avtomatlaşdırılmış informasiya sistemlərini verilənlər bankları təşkil edir. Bu sistemlərdə verilənlər artıq yüksək dərəcədə inteqrasiya olunmuş və onların avtomatlaşdırılmış idarə olunması həyata keçirilmişdir. Onlar kollektiv istifadə üçün təyin olunmuşlar və birinci nəsil avtomatlaşdırılmış informasiya sistemlərində olan çatışmamazlıqlar bu sistemlərdə artıq aradan qaldırılmışdır.

Avtomatlaşdırılmış informasiya sistemlərinin fəaliyyəti əsasən informasiyaların toplanması və işlənməsi ilə əlaqədardır. İnformasiya dedikdə, burada faktiki verilənlər haqqında biliklər və onlar arasındakı qarşılıqlı əlaqələr toplusu basa düşülür. Əksər hallarda EHM-lərdə informasiya və verilənlər anlayışları eyniləşdirilirlər. Ancaq, əslində verilənlər daha dəqiq desək, EHM-ə daxil etmək üçün zəruri formaya salınmış, saxlanmaq, işlənmək və istifadəçiyə verilmək üçün olan informasiyalardır.

Avtomatlaşdırılmış informasiya sistemlərinə daxil edilən, həm də istifadəçiyə verilən informasiyalar sənəd şəklində təqdim olunurlar. Sənəd əla material obyektidir ki, burada informasiyalar qəbul olunmuş qaydalara əsasən tərtib olunurlar və müəyyən hüquqi qiymətə malik olmaqla istifadə və otürülmək üçün təyin olunmuşlar. Avtomatlaşdırılmış informasiya sistemlərində informasiya mənbəyi insanlar və ölçü cihazları, istifadəçi isə insanlardır.

İnformasiya sistemlərinin istifadəçilərini üç kateqoriyaya bölmək olar:

1. sistemin administratorları; onlar sistemin istismarına cavabdehdir;
2. tətbiqi proqramçılar; onlar müxtəlif məsələlərin həlli üçün tətbiqi proqramlar yazanlardır;

3. son istifadəçilər; onlar isə böyük bir qrupu təşkil etməklə informasiyalara ehtiyacı olanlardır. Son istifadəçi dedikdə özünə zəruri olan verilənləri almaq üçün sistemə müraciət edən istifadəçi başa düşülür. Son istifadəçi ixtiyari proqramlaşdırmanı bilən, həm də hesablama texnikası və proqramlaşdırmanı bilməyən şəxs ola bilər. Sorğu-sistemin girişinə daxil olan, verilənlərin axtarılması şərtini özündə əks etdirən və tapılmış verilənlərə sonradan nə etmək lazım olduğunu

göstərən formallaşdırılmış məlumatdır.

Daxil olan sorğuların interpretasiyası, onlarda göstərilən əməliyyatların yerinə yetirilməsi, məlumatların və sənədlərin çıxışa, yəni istifadəçiyə verilmək üçün hazırlanması avtomatlaşdırılmış informasiya sisteminin işinin əsas mərhələləridir. Beləliklə, avtomatlaşdırılmış informasiya sistemi dedikdə, verilənlərin istifadəçinin sorğusuna uyğun olaraq verilməsi, toplanması, saxlanması, axtarılması və işlənməsi üçün təyin olunmuş informasiya massivləri toplusu, texniki, proqram və dil vasitələri kompleksi başa düşülür.

Avtomatlaşdırılmış informasiya sistemləri insan fəaliyyətinin demək olar ki, bütün sahələrində tətbiq olunur:

- müəssisənin, təşkilatın, istehsalın idarə olunması;
- elm, tədqiqat, eksperimentlərin aparılması;
- kitabxanaçılıq işi;
- layihələndirmə.

Avtomatlaşdırılmış informasiya sistemlərinin bu və ya digər sahələrdə istifadə olunması aşağıdakı iki üsulla həyata keçirilə bilər:

1. Avtomatlaşdırılmış informasiya sistemi avtonom şəkildə, heç bir sistemin tərkibinə daxil olmadan fəaliyyət göstərir və ayrıca istifadə olunur. Buna misal olaraq sənədli (kitabxanaçılıq) informasiya-axtəriş sistemi göstərilə bilər. Belə sistemlər istifadəçinin sorğusuna cavab olaraq lazım olan elm sahəsi haqqında sənədləri ona təqdim edir. Bundan başqa aviabiletlərin, dəmir yol biletlərinin satışında istifadə olunan «SİRENA» və «EKSPRESS» sistemləri də informasiya - axtəriş sistemləridir. Növbəti hissədə bu sistemlər haqqında daha geniş məlumat veriləcəkdir. Adı çəkilən sistemlər sərnişinlərə tələb olunan istiqamətdə, sənəd şəklində bilet təqdim edir, yaxud biletin olmadığı haqqında, daha doğrusu tələb olunan istiqamətdə qatar, yaxud təyyarədə boş yerin olmadığı haqqında məlumat verilir.

2. Avtomatlaşdırılmış informasiya sistemi digər bir sistemin tərkibində istifadə olunur. Bu halda çıxış verilənləri sistemin digər komponentləri tərəfindən istifadə oluna bilər, yaxud da son istifadəçiyə çatdırıla bilər, məsələn, öyrədən

sistemlərdə avtomatlaşdırılmış informasiya sistemi öyrəniləcək materialı, suallar toplusunu və cavabları, avtomatlaşdırılmış layihələndirmə sistemlərində normativ - arayis informasiyasının, dövlət stan-dartları haqqında məlumatları, yaxud başqa verilənləri, avtomatlaşdırılmış idarəetmə sistemlərində müəssisənin idarə olunması üçün zəruri olan bütün informasiyaları, başqa sozlə, analiz, qiymətləndirmə, proqnozlaşdırma, qərarın qəbul olunması, planlaşdırma, yerinə yetirməyə nəzarət üçün lazım olan informasiyaları saxlaya bilər.

### **1.2.2 Tətbiqi proqramlar paketləri.**

Instrumental proqramlaşdırma sistemi ilə generasiya olunan tətbiqi proqramlar paketi dedikdə, bir birinə verilənləri və informasiyaları ötürə bilən, hər hansı bir tətbiq oblastından olan məsələlərin həlli üçün tərtib olunmuş proqramlar toplusu başa düşülür. Məsələlərin həll olunduğu oblasta paketin tətbiq oblastı deyilir. Yaradılan hər bir paket mütləq yüksək səviyyəli giriş dili ilə təchiz olunur. Məsələn, Min-Priz instrumental proqramlaşdırma sistemində bu dil A-Utopist dilinin genişlənməsi kimi təşkil olunmuşdur.

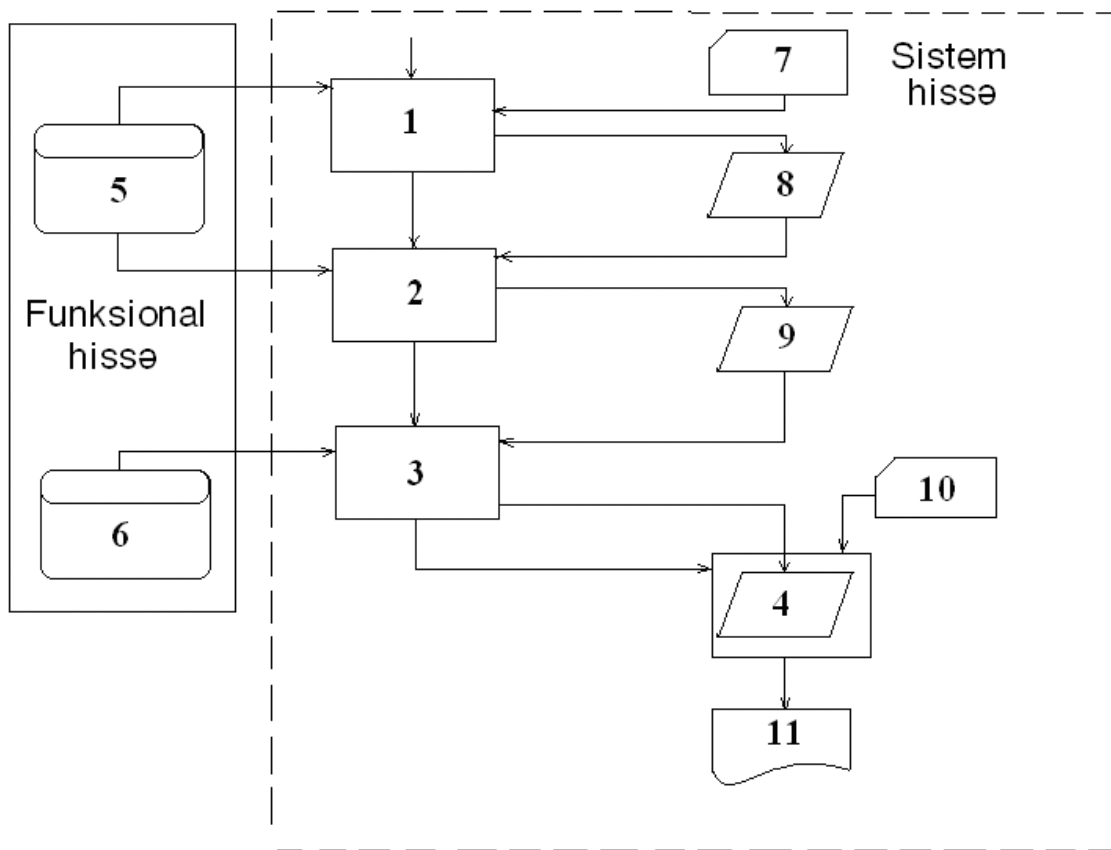
Tətbiqi proqramlar paketi iki hissədən, paketi gövdəsi adlanan hissədən və təşkiledici proqram hissəsindən ibarət olur. Paketin gövdəsi, paketin tətbiq oblastından olan məsələlərin həllinin ayrı ayrı addımlarının doldurulması üçün əvvəlcədən proqramlaşdırılmış bütün tətbiqi proqramların daxil olduğu modullar kitabxanasından, həmçinin paketin tətbiq oblastından olan anlayışların şərhini saxlayan semantik modellər kitabxanasından ibarətdir. Bu semantik modellər birlikdə paketin predmet oblastının modelini əmələ gətirir.

Təşkiledici proqram, paketin giriş dilində şərh olunmuş məsələlərin həlli zamanı paketin bütün işini idarə etmək üçün yaradılmışdır. Min-Priz instrumental proqramlaşdırma sistemində, paketlərə nəzərən təşkiledici proqram rolunu bu instrumental kompleksin sistem proqramları oynayır.

Təşkil edici proqramı başqa adla da, idarə edici proqram da adlandırırlar. Bu proqrama hətta dispetçer, yaxud da monitor adı verilir. Bu proqram;

- əməliyyat sistemi ilə əlaqə yaratmaqla bütün paketin işini idarə edir;
- həll olunacaq məsələnin paketin giriş dilindəki şərhini qəbul edir və onu sistemin daxili dilinə tərcümə edir;
- qoyulmuş məsələnin həlli üçün hansı modullardan ardıcılıqla istifadə olunacağını müəyyən edir;
- məsələnin həlli üçün seçilmiş modulları vahid məsələnin həlli proqramında birləşdirərək bu modullar arasında əlaqə yaradır;

Tətbiqi proqramlar paketinin sxemi aşağıdakı şəkildə verilmişdir (şəkil 3.1).



Şəkil 3.1 Tətbiqi proqramlar paketinin sxemi

Yuxarıda sadalanan bəndləri yerinə yetirən təşkil edici proqramın ayrı ayrı komponentləri verilmişdir. Burada 1- giriş dilinin translyatoru, 2 – sistemin planlaşdırıcısı, 3 – sistemin kompilyatoru, 4 – məsələnin həlli proqramı, 5 – predmet oblastının modeli, 6 – funksional modullar, 7 - həll olunacaq məsələnin mətni, 8 – həll olunan məsələnin paketin daxili dilində şərh, 9 - məsələnin həlli alqoritmi, 10 – başlanğıc verilənlər, 11 - məsələnin həlli proqramıdır. Modullar

arasındakı informasiya əlaqələri qalın xəttlərlə göstərilmişdir. Həll olunacaq məsələnin mətni, giriş dilində şərh olunduqdan sonra sistemin girişinə verilir. Daxil edilən mətn, giriş dilinin translyatorunun vasitəsi ilə, paketin predmet oblastının modelində verilmiş, giriş dilindəki anlayışların semantikasını nəzərə almaqla, sistemin daxili dilinə çevirir. Məsələnin daxili dildə alınan şərhinə əsaslanaraq sistemin planlaşdırıcı komponenti lazım olan modullar ardıcılığını müəyyən edir və sistemin daxili dilində məsələnin həlli alqoritmini verir. Nəhayət sistemin kompilyatoru, yerinə yetirilmək üçün hazır olan, məsələnin həlli proqramını yaradır. Bu proqram həm də işçi proqram adlanır. Belə rejimdə işləyən tətbiqi proqramlar paketləri kompilyasiya tipli tətbiqi proqramlar paketləri adlanırlar. Həmin işçi proqramları kompüterin yaddaşında saxlamaq və dəfələrlə ilkin verilənləri dəyişməklə istifadə etmək olar. Bu proqramlar EXE, yaxud da COM tipli yerinə yetməyə hazır olan proqramlardır. Bu, kompilyasiya rejimində işləyən tətbiqi proqramlar paketlərinin ən üstün cəhətidir. Lakin belə paketlərin strukturu xeyli mürəkkəbdir və yaradılan zaman müəyyən çətinliklər törədə bilər. Elə tətbiqi proqramlar paketləri də mövcuddur ki, onlarda kompilyator komponenti əvəzinə interpretator komponentindən istifadə olunur. Belə olan təqdirdə planlaşdırıcı komponentinin aşkar etdiyi modullar bir birinin ardınca operativ yaddaşa çağrılaraq interpretator tərəfindən yerinə yetirilirlər, və heç bir işçi proqram yaradılmır. Belə rejimdə işləyən tətbiqi proqramlar paketləri interpretasiya tipli tətbiqi proqramlar paketləri adlandırılırlar. Bu tətbiqi proqramlar paketləri sadə quruluşa malik olsalar da, kompilyasiya tipli tətbiqi proqramlar paketlərinə nisbətən xeyli ləng işləyirlər. Bu isə onların ən böyük çatışmayan cəhəti hesab olunur.

Tətbiqi proqramlar paketi hazırlanması prosesində həm sistem proqramçıları, həm də paketin istifadə olunacağı tətbiq oblastı üzrə mütəxəssis iştirak etməlidir. Bu çox çətin və kifayət qədər baha başa gələn bir prosesdir. Çünki müxtəlif tətbiq oblastlarından olan mütəxəssislərin, eyni bir problemi həll edərkən, birgə işinin təşkili xeyli mürəkkəbdir. Son məqsədin özünün mürəkkəbliyi də işi daha da qəlizləşdirir. Tətbiqi proqram hər hansı bir tətbiq oblastından olan məsələnin həlli

üçün hazırlanırdısa, burada böyük bir proqramlar toplusu uyğun tətbiq oblastından olan məsələlər toplusu üçün hazırlanır. Lakin, əgər hətta proqramlaşdırma ilə əlaqədar olan problemlər həll olunmuş olsa belə, həlli tələb olunan məsələlər sinfinin şərhilə bağlı çətinliklər, həm də paketin tətbiq oblastından olan bütün məsələlər üçün düzgün layihə həllinin qəbul olunması ilə əlaqədar zəruri çətinliklər qalmış olurlar.

Tətbiqi proqramlar paketlərinin yaradılması o qədər mürəkkəb prosesdir ki, hətta xeyli iş təcrübəsinə malik olan proqramçı belə çox yaxşı fəaliyyət göstərən paketin hazırlanmasında öz acizliyini boynuna alır.

Tətbiqi proqramlar paketləri hazırlanan zaman tətbiq oblastı haqqında və bu tətbiq oblastının əhatə etdiyi məsələlər çoxluğunun həlli metodları haqqında yaxşı biliklərin olması zərurətdir. Bu biliklərin “mənbəyi”, obrazlı desək, tətbiq oblastı mütəxəssisi hesab olunur. Lakin paketin yaradılması bu biliklərin elə dərəcədə formallaşdırılmasını tələb edir ki, bu formallaşdırma həmin mütəxəssislər üçün vərdiş olunmayıdır, yəni qeyri adidir. Bir çox sistemlərdə, məsələn, elə Min-Priz sisteminin özündə paketlərin giriş dili enən (azalan) metodla yaradılır, belə ki, bir çox ciddi formalizmlər və bir çox kiçik həllərin qəbul olunması yaradılmanın sonrakı mərhələlərinə saxlanılır. Dilin yaradılması əsas anlayışların müəyyən olunması və onların ətraflı şərh olunmasından başlayır. Bundan sonra seçilmiş anlayışlar çoxluğunun tamlığı yoxlanılır. Belə yoxlama seçilmiş anlayışların, yəni operatorların köməyi ilə həmin uyğun oblastdan tipik məsələlərin şərhilə vasitəsi ilə həyata keçirilir. Bu mərhələdə həm də seçilmiş anlayışların semantikasi dəqiqləşdirilir. Yalnız bundan sonra anlayışların formal şərhilə tərtib olunur. Təbiidir ki, bu proses tezliklə sona çatmır. Yəni yeni anlayışların daxil edilməsinə qayıtmaq, yenə onları sınaqdan keçirmək və sonra dəqiq şərh etmək (mümkündür ki, artıq əvvəlcədən qəbul olunmuş anlayışlarda dərişiklik edilsə) lazəm gəlsin. Ona görə də məsləhətdir ki, paket hazırlanan zaman bütün misallar diqqətlə öyrənilsin.

Tətbiqi proqramlar paketlərinin hazırlanması mərhələlərindən ilkinilə olan “texniki tapşırıqın hazırlanması” birqiymətli olaraq TKİ – Təcrüba Konstruktor İşlərinin aparılması mərhələlərindən birinə uyğun gəlir. Qalan işlər isə texniki və



işçi layihələrin həm də bütünlükdə sistemin hazırlanmasının mərhələləri arasında müxtəlif şəkildə paylana bilər.

Texniki tapşırığın hazırlanması mərhələsində hazırlanmanın effektivliyinin qiymətləndirilməsi, hazırlanacaq məhsula, yəni proqramlar paketinə qoyulan tələbləri nəzərə almaqla son məqsədin dəqiq qoyuluşu, konseptual xarakterli tövsiyələrin hazırlanması, sonrakı bütün mərhələlərdə görülməli işlərin qrafikinə tərtibi problemləri öz həllini tapır. Bu mərhələ təcrübə konstruktor işləri və elmi tədqiqat işləri çərçivəsində texniki tapşırığın hazırlanmasına tələblərinə uyğun olaraq yerinə yetirilir.

Sonrakı mərhələdə həll olunacaq məsələlər sinfi şərh olunur. Həlli nəzərdə tutulan məsələlər sinfi uyğun ixtisasın anlayışları ilə ətraflı şərh olunur. Həlli metodları və mürəkkəbliyi dərəcəsi bir birinə yaxın olan məsələlərin təsnifatı aparılır, həm də bir tip məsələlər qrupunda birləşdirilir. Hər bir tip məsələlər üçün mütləq aşağıdakılar verilir:

- misal;
- məsələnin ölçüsünü müəyyən edən ədədi xarakteristikalar;
- verilmiş tip məsələlərə rast gəlinmənin nisbi tezliyi;
- həll metodları;
- giriş-çıxışa qoyulan tələblər.

Yaradılan paket üçün bütünlükdə paketlə iş rejimi müəyyən olunur: paket rejimi, translyasiyanın gedişi zamanı dialoq, hesablama prosesində dialoq.

Növbəti mərhələdə baxılan sinif məsələlərin həllini təmin edəcək alt məsələlər seçilib ayrılır və alt məsələlərin həlli alqoritmləri hazırlanır. Burada əsas məqsəd hazırlanan paketin bütün məsələlərinin həlli üçün tərtib olunacaq əsas proqramın hansı modullar toplusundan ibarət olacağını müəyyən etməkdir.

Tətbiqi proqramlar paketinin giriş dilinin hazırlanması əsas mərhələlərdən biri hesab olunur. Dilin tipi müəyyən olunur; baza dilinin genişlənməsi şəklində asılı olmayan dil; yüksək səviyyəli proqramlaşdırma dilinə, yaxud da Assembler dilinə daxil edilmiş dil. Sonra isə dilin əsas anlayışları müəyyən olunur, onların semantikasi yüksək səviyyəli proqramlaşdırma dilində mətn şəklində şərh olunur.

Paketin giriş dilinin hazırlanması, paketlə həll olunacaq məsələlərin həlli zamanı istifadə ediləcək tətbiq oblastının əsas anlayışlarını saxlayan cədvəllərin hazırlanmasından başlayır. Kəmiyyətlər cədvəlinin hazırlanması zamanı tətbiq oblastının bütün kəmiyyətləri oraya daxil edilir, belə ki, tərtib olunan cədvəllərin ölçüsü paketdə hazırlanan proqramların ölçüsünə təsir etmir. Hər bir anlayış üçün dolğun (məzmunlu) ad, ümumi qəbul olunmuş işarələmə, yəni əlamət, paketdə baxılan anlayışı işarələmək üçün qəbul olunan identifikator, baxılan anlayışa tabe olan anlayışlar, anlayışın tipi, qiymətlərin saxlanması üçün tələb olunan maksimal yaddaşın həcmi göstərilir. Kəmiyyətlərin tipi yüksək səviyyəli proqramlaşdırma dilinin vasitələri ilə şərh olunur və həqiqi, tam, məntiqi, sətir, yaddaş, sıra, mürəkkəb, qeyri müəyyən tipli ola bilərlər. Anlayışların siyahısının tərtibi dilin hazırlanması zamanı ən mürəkkəb və məsuliyyətli mərhələlərdən biridir. Bu zaman tətbiq oblastı haqqında yaxşı biliyə ehtiyac vardır, həmçinin yaradılacaq dilin anlayışlarının baza dilinin anlayışları ilə birlikdə istifadəsi problemi həll olunmalıdır.

Kəmiyyətlər müəyyən olunduqdan sonra onlar arasındakı münasibətlər şərh olunur. Bunun üçün tətbiq oblastının modelinin sxemi tərtib olunur, bu modelə görə baxılan paketin məsələlərinin həlli olduğu yoxlanılır və paketin gövdəsinə daxil ediləcək modulların cədvəli tərtib olunur. Hər bir modul üçün identifikator, tip, rəqəm, parametrlərin siyahısı göstərilir. Parametrlərin siyahısında hər bir parametr üçün onun modulla əlaqəsi tipi göstərilir. Tətbiq oblastı modelinin sxemi anlayışlarının cədvəlinə və modullar cədvəlinə əsasən problem-yönlü dildə tətbiq oblastı modelinin şərhini tərtib olunur.

Hal hazırda elm və texnikanın, o cümlədən də iqtisadiyyatın qarşısında duran məsələlər xeyli mürəkkəbdir və onların həlli üçün bir tətbiqi proqramlar paketi kifayət etmir. Odur ki, belə mürəkkəb məsələlərin həlli zamanı bir neçə paketdən istifadə etmək zərurəti ortaya çıxır. Paketlərin inteqrasiyası problemi müasir proqramlaşdırmanın əsas istiqamətlərindən biridir və bu problemin həlli üçün instrumental proqramlaşdırma sistemləri yaradılır. Min-Priz belə instrumental proqramlaşdırma kompleksidir və bir çox sahələrdə müvəffəqiyyətlə tətbiq olunur.

Bu sistemin köməyi ilə müəyyən sinif məsələlərin həlli üçün birlikdə tətbiq olunan bir neçə paketin daxil olduğu inteqrasiya edilmiş proqramlaşdırma sistemi generasiya etmək mümkündür. Ümumi təyinatlı proqramlar paketləri, məsələn, riyazi statistika, optimallaşdırma, xətti cəbr və sair paketləri bu sistem vasitəsi ilə inteqrasiya etmək mümkündür.

Tətbiqi proqramlar paketlərinin birlikdə istifadəsi üçün bu paketlərin proqram, informasiya və funksional uyuşanlığını təmin etmək lazım gəlir.

Birlikdə istifadə olunan paketlərin modulları, qoyulmuş məsələnin həlli üçün proqram tərtib olunan zaman, əməliyyat sisteminin komponenti olan əlaqə redaktoruna əlçatan olmalıdır, başqa sözlə, əlaqə redaktoru bu modulların hamısına eyni zamanda müraciət etmək imkanına malik olmalıdır. Birlikdə istifadə olunan paketlərin tətbiq oblastlarının modelləri mütləq eyni bir kitabxanada saxlanılmalıdırlar. Min-Priz instrumental proqramlaşdırma sistemi proqramlar paketlərinin proqram uyuşanlığını tam şəkildə təmin edir. Lakin aşağıdakı şərtlərin ödənilməsi mütləqdir:

- müxtəlif paketlərdə modulların adları eyni ola bilməz;
- müxtəlif paketlərdə ümumi oblastların adları üst-üstə düşə bilməz;
- müxtəlif paketlərin modullarında qeyd olunmamış ümumi oblastlardan istifadə etmək olmaz.

Tətbiqi proqramlar paketlərinin informasiya uyuşanlığını təmin etmək üçün, zəruridir ki, müxtəlif paketlərdə bir birlərinə uyğun gələn anlayışlar bir birləri ilə ekvivalentlik münasibəti ilə birləşə bilsinlər və müqayisə edilə bilən, ekvivalentlik münasibəti mümkün olan növlərə malik olsunlar.

Bir neçə tətbiqi proqramlar paketlərindən inteqrallaşdırılmış proqramlaşdırma sistemi yaradılarkən hər bir paket üçün xarici anlayışlar, yəni digər paketlərin anlayışları ilə əlaqə yarda bilən anlayışlar, müəyyən olunur. Xarici anlayışlar üçün uyğunluq cədvəli tərtib olunur. Cədvəldə hər bir xarici anlayışa digər paketlərdən uyğun gələn anlayış göstərilir.

### 1.2.3. İnformasiya –axtarış sistemləri.

Akademik S.I. Vavilovun obrazlı ifadəsi ilə desək, «Müasir oxucu kitabxanalar himalayı qarşısında özünü qızıl axtaranlar kimi hiss edir, belə ki, o böyük qum dağında xırda qızıl dənəcikləri axtarır».

Çap olunmuş bütün əsərləri oxumaq, yaxud tələsik də olsa, nəzərdən keçirmək və onlardan lazım olanını seçmək praktik olaraq mümkün deyildir. Hesablamaq çətin deyildir ki, misal üçün kimya sahəsində 1 il ərzində dərc olunan məqalələri nəzərdən keçirmək üçün 1 saatda 30 məqalə nəzərdən keçirməklə, həftədə 40 saat isləməklə 5 il vaxt itirmək lazımdır. Ona görə də hər sənəd (kitab, məqalə) bu sənədin əsas mövzusu, yaxud predmetin xarakteristikalarını özündə əks etdirən «birka» ilə təmin olunur. İnformasiya axtarışı nəzəriyyəsində bu «birkaların» mətnləri sənədin «axtarış obrazı» adlanırlar. Beləliklə, bizə lazım olan sənədi tapmaq üçün axtarış obrazlarının mətnləri nəzərdən keçirilir ki, bu da axtarışı xeyli sürətləndirir. Lakin sənədin axtarışının tamlığının və dəqiqliyinin azalması hesabına başa gəlir. Digər tərəfdən axtarış prosesin elmi işçi tərəfindən deyil, informasiya xidmət işçiləri tərəfindən aparılır. Ona görə də sənədlərin axtarış obrazları, həm də informasiya tələbi elə dildə təsvir olunmalıdır ki, bu dilin bütün sözləri və cümlələri birqiymətli mənaya malik olsunlar, onların müxtəlif təsvirləri (baş düşülməsi) mümkün olmasın. Belə dil informasiya-axtarış dili (İAD) adlanır. Onda İAD-də yazılmış informasiya tələbini mexaniki olaraq sənədin axtarış obrazı ilə müqayisə etmək olar. Nəzərə alaq ki, axtarış obrazları da həmçinin İAD-də təsvir olunurlar. Əgər bu iki element üst-üstə düşürsə, yaxud 1-ci tamamilə 2-ciyə daxildirsə, yaxud da onun tərkib hissəsidirsə, onda sənəd informasiya tələbinə cavab vermiş hesab olunur və istifadəçiyə verilir.

Eyni bir İAD-də təsvir olunmuş sənədin axtarış obrazı ilə informasiya tələbinin uyğunluğunun zəruri və kafilik dərəcəsini müəyyən edən əlamətlər toplusu verilmə kriteriyası adlanır. Verilmə kriteriyası sadə şəkildə informasiya tələbinin sənədin axtarış obrazı ilə tam üst-üstə düşməsi və yaxud onun tamamı ilə axtarış obrazına daxil olması kimi başa düşülür.

İnformasiya axtarışı dedikdə hər hansı bir sənədlər çoxluğundan infor-masiya tələbində göstərilən mövzuya (fənnə) aid olanların, yaxud istifadəçi-yə lazım olan faktları, yaxud məlumatları özündə saxlayaların axtarılması prosesi basa düşülür.

Sənəd dedikdə hər hansı bir daşıyıcıda yazılmış, müəyyən məntiqi tamlığa malik və onun mənbələri, həm də yaradıcılar haqqında məlumatları saxlayan, basa düşülən mətn nəzərdə tutulur.

Bu tərifə görə təkcə kitab, məqalə, müəlliflik şəhadətnaməsi və patentin şərhli və s. deyil, hər hansı mətnin fraqmenti - fəsil, bölmə, abzas və s. də sənəd ola bilər.

Axtarılan obyektin tipindən asılı olaraq iki növ informasiya axtarışın bir-birindən fərqləndirirlər: sənədlərə əsaslanan (sənədli); faktoqrafik.

Sənədlərə əsaslanan informasiya axtarışını növlərindən biri də biblioqrafik axtarışdır. Sənədlərə əsaslanan axtarışda axtarılan obyekt ya sənədin biblioqrafik şərhli (buna biblioqrafik axtarış deyilir), yaxud bu sənədlərin özləri, yaxud da sürətləri hesab edilir.

Faktoqrafik axtarışda axtarılan obyektlər konkret faktların, yaxud hadisələrin şərhli (məsələn, hər hansı bir maşının texniki xarakteristikaları, onların təsvirləri, maddələrin və materialların xassələri və i.a.) hesab edilir.

İnformasiya axtarışı informasiya - axtarış sistemləri (İAS) vasitəsilə yerinə yetirilir. İAS dedikdə zəruri informasiyaların saxlanması və istifadəçiyə verilməsi proseslərini həyata keçirmək üçün təyin olunmuş dil, alqoritm və texniki vasitələr kompleksi başa düşülür.

İAS informasiyaların avtomatik axtarışını həyata keçirir. İAS-nın girişinə iki növ informasiya daxil olur: hər hansı bir sinif obyektlər haqqında əldə olunmuş biliklərdə əks olunan informasiya (məs., hər hansı bir qurğu, tex-noloji proses, kimyəvi maddə, reaksiya, teorem və s.); İAS abonentlərinin tələbini əks etdirən informasiya.

Birinci tip informasiyalar informasiya massivləri, yaxud axtarış massiv-ləri, ikinci tip informasiyalar isə informasiya sorğuları adlanır. Hər iki növ informasiyalar İAS-nə təbii dildə daxil edilir və sonradan adətən formalaşdırılmış İAD - nə translyasiya olunur.

İAS-nin əsas funksiyası sistemə daxil olan sorğulara cavab verən informasiya massivindəki elementlərin aşkar edilməsidir. İnformasiya axtarışı ya əllə, yaxud da avtomatlaşdırma vasitələri tətbiq etməklə yerinə yetirilir. Konkret sənədli İAS-nə misal olaraq kitabdakı göstərişləri, kitabxana kataloqunu və s. göstərmək olar. Sadə faktoqrafik İAS-nə misal telefon sorğu kitabı, ünvanlar kitabı, nəşrlərin kataloqu və s. ola bilər. Bu iki növ İAS-ləri arasındakı fərq prinsiplial deyildir və yalnız ondan ibarətdir ki, onların axtarış obyektləri müxtəlifdir. Sənədli İAS-lərində axtarış obyektini sənədlər, faktoqrafik İAS-lərində isə müəyyən şəkildə yazılmış faktlar, məlumatlardır.

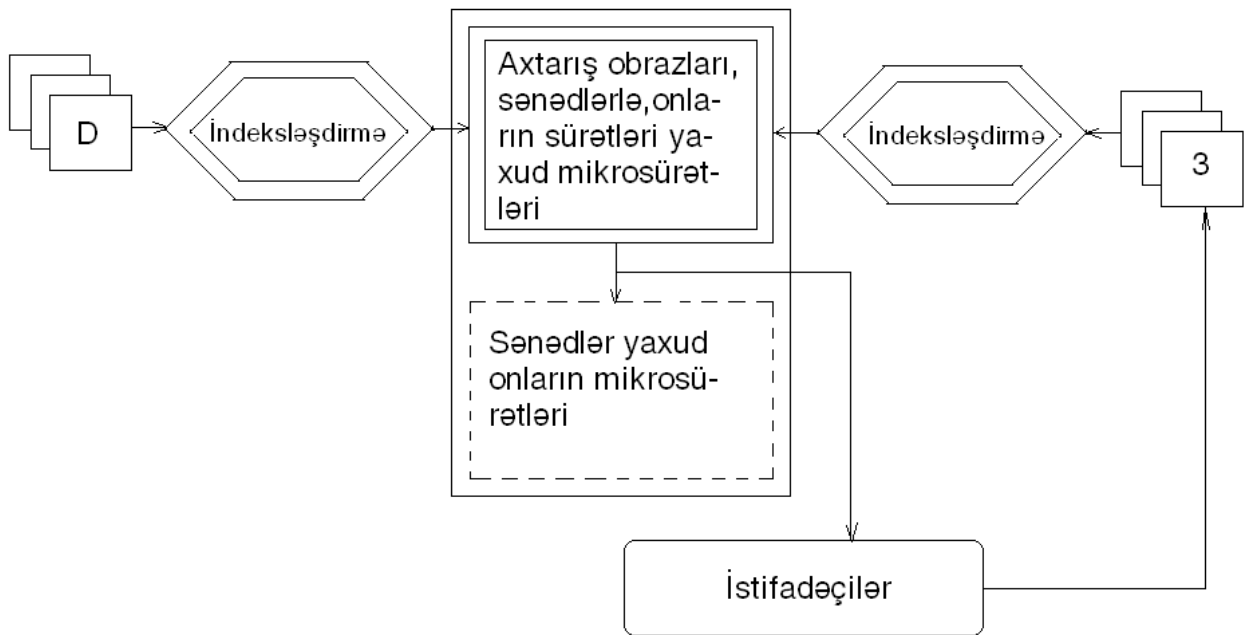
Bəzi müəlliflər faktoqrafik İAS-lərini xüsusi sinfə aid edirlər, belə ki, bu sistemlər sənədli İAS-lərinə nisbətən daha mürəkkəbdirlər. Lakin bu belə deyildir. İxtiyari faktın axtarış obyektini olması üçün, o müəyyən sənəd formasına salınmalıdır. Ola bilsin ki, fakt özü sənəddən kənarında olsun, lakin onun haqqında məlumat mütləq sənədləşdirilməlidir. Faktın sənədləşdirilməsi o deməkdir ki, o hər hansı bir təbii, yaxud süni dildə ifadə, ya da şərh olunsun, belə ki, belə şərhə həmin faktın nə vaxt və harada baş verdiyi (olduğu), məna kəsb etdiyi haqqında mütləq göstəriş olmalıdır.

Beləliklə, faktoqrafik axtarış zamanı, sənədli axtarışda olduğu kimi, biz sənədlərin axtarışı ilə məşğul oluruq.

Bu iki növ İAS-ləri arasındakı fərqi, onların yaradılmasını ümumi prinsipləri və metodlarında deyil, axtarış obyektlərinin, yaxud axtarış məsələlərinin spesifikliyində (özünəməxsusluğunda) axtarmaq lazımdır.

Faktoqrafik axtarışı informasiyaların məntiqi işlənməsində fərqləndirmək lazımdır, belə ki, faktoqrafik axtarışda informasiyaların işlənməsi prosesindən fərqli olaraq verilən suala mütləq cavab verilməlidir. Faktoqrafik axtarış zamanı yalnız və yalnız elə faktlar və məlumatlar axtarılır ki, (onlar tapıla bilərlər) onlar İAS-nə daxil ediləblər, lakin onlar şəkildə həmin sistemə daxil edilməyən informasiyalar tapıla bilməzlər (ola bilər ki, belə faktlar məntiqi nəticə kimi daxil edilən sənədlərdən alınsın). Tələb olunan faktların bir, ya-xud bir necə dildə yazılmış sənədlərin mətnlərində axtarılması üçün, yaxud belə faktların məntiqi nəticə kimi artıq məlum

olan digərlərindən alınması üçün mətdən onların dəqiq mənasını çıxarmaq alqoritminin olması, onun analizini və sintezini aparmaq bacarığının olması zəruridir. Başqa sözlə desək informasiya məntiqi cəhətdən işlənilməlidir. Bu tip məsələlər İAS-lərinin köməyi ilə həll olunur. İAS-nin fəaliyyəti sxemini nəzərdən keçirək (şəkil 1.2).



Şəkil 1.2

Sxemdən görüldüyü kimi hər bir sənəd İAS-nə daxil edilərkən indeksləşdirilir. İndeksləşdirmə dedikdə ikimərhələli proses basa düşülür:

- verilmiş sənədin əsas mövzusu, yaxud predmetinin müəyyən olunması;
- əsas mövzu, yaxud predmetin baxılan İAS-nin İAD-də təsvir olunması və onun axtarış obrazı şəklində yazılması; bu axtarış obrazı indeksləşdirilmiş sənədə aid edilir.

İndeksləşdirmə dərinliyi anlayışı daxil edək. İndeksləşdirmə dərinliyi sənədin axtarış obrazında onun mərkəzi mövzusunun, yaxud predmetinin, həmçinin onunla əlaqədar mövzu və predmetlərin təsvir olunmasının ətraflılıığı dərəcəsi kimi müəyyən olunur.

Sənədlərin tamamilə eyni şəkildə heçnədən asılı olmayaraq indeksləş-dirilməsi indeksləşdirmənin avtomatlaşdırılması yolu ilə həyata keçirilir. Ola bilsin ki, bu indeksləşdirmənin keyfiyyət dərəcəsini əllə indeksləşdirməyə nisbətən aşağı salsın, lakin şübhə yoxdur ki, bu İAS-nin işlənməsi etibarlılığını yüksəltməyə imkan verir.

İnformasiya sorğusu da həmçinin indeksləşdirilir, başqa sözlə desək, İAD-də ifadə olunur. Lakin burada informasiya sorğusunun mərkəzi mövzu-su, yaxud predmetinin müəyyən olunması ilə əlaqədar çətinliklər ortaya çıxmır, belə ki, indeksləşdirilən tələbin məzmunu vahid mövzudan, yaxud predmetdən ibarət olur.

İxtiyari real fəaliyyətdə olan İAS dörd əsas komponentdən ibarətdir:

- məntiqi semantik aparat (başqa sözlə, bir və ya bir neçə informasiya -axtarış dili, indeksləşdirmə qaydaları, verilmə kriteriyası);
- axtarış massivi (axtarış obrazları ilə təchiz olunmuş müəyyən sənəd-lər çoxluğu; lazım olan sənədlər bu çoxluqda axtarılır);
- texniki vasitələr (sənədlərin özlərinin, axtarış obrazlarının və s. yazıl-ması və saxlanması üçün zəruri olan qurğular);
- sistemlə ünsiyyətdə olan insanlar (buraya İAS-indən istifadə edənlər, ona xidmət edənlər, yəni sənədləri və informasiya sorğularını indeksləş-dirənlər, axtarış strategiyasını müəyyən edənlər, həmçinin digər intellektual əməliyyatları həyata keçirənlər aiddirlər).

#### **1.2.4. Ekspert sistemlər.**

Artıq bir neçə ildir ki, süni intellekt sahəsində aparılan nəzəri tətqiqatların nəticələri praktikada tətbiq olunmağa başlamışdır. Son illərdə ekspert sistemlərinin çox böyük müvəffəqiyyətlə tətbiqi və bu istiqamətdə aparılan elmi işlər buna əyani sübutdur. Problem-orientasiyalı sistemlərin müstəqil bir istiqaməti kimi ekspert sistemlər müxtəlif tətbiq sahələrində məsələlərin həlli üçün həmin sahənin ən bilikli mütəxəssis - ekspertlərinin vəziyyəti qiymət-ləndirməsi bacarığından istifadə edirlər.



Ekspert sistemlər intellektual hesablama sistemlərinə aid edilirlər və modelləşdirmə, yaxud hər hansı bir dar sahədən olan məsələlərin həlli zamanı təcrübəli mütəxəssis-ekspertlərin davranışını təqlid etmək məqsədilə yaradılırlar.

Ekspert sistemlər ortaya çıxan problemləri müstəqil həll etmək üçün mütəxəssislərə onların öz bilik, bacarıq və intuisiyaları kifayət etmədikdə köməklik göstərilir. Belə sistemlər böyük proqram kompleksləri olub, məsələləri real şəraitdə ekspertin həll etdiyi kimi həll edir. Bu isə imkan verir ki, müəyyən bir konkret tətbiq sahəsindən məsələləri ən əlverişli üsulla həll edə bilən ekspertlərin bilik və professional təcrübəsi toplansın, sistemləş-dirilsin və saxlanılsın. Birinci növbədə bu sistemlər o sahələrə tətbiq olunurlar ki, həmin sahələrdə məsələlər və onların həlli pis formalaşdırılır və yaxud tamamilə formalasdırılmır. Həmin sahələrə misal olaraq maşınqayırma, hesablama texnikası, informatika, tibb, biologiya, geologiya, nüvə energetikası, nüvə fizikası, iqtisadiyyat, tarix, hərbi işi göstərmək olar.

Ekspert sistemlər keyfiyyət xarakterli bilikləri və insan-ekspertin təcrübəsini modelləşdirməyə cəhd edən proqram sistemidir. Daha doğrusu, ekspert system - süni intellekt sistemi olub, biliklərin ekspertlərin şərh formasında təqdim olunduğu və geniş yayılmış riyazi modellərdən istifadə etməyin mümkün olmadığı və ya cətin olduğu sahələrdə kompüterləşməni səmərəli sürətdə həyata keçirməyə imkan verir.

Ekspert sistemlərin əsasını biliklər bazası təşkil edir. Biliklər bazasında ekspertlərdən və xüsusi ədəbiyyatlardan alınmış faktlar və qaydalar toplusu saxlanılır. Biliklər bazası verilənlər bazasından onunla fərqlənir ki, əgər informasiya vahidləri verilənlər bazasında bir-biri ilə əlaqəli olmayan məlumatlar, düsturlar, teoremlər, aksiomlardan ibarətdirsə, bilik bazasında bu elementlər artıq öz aralarında əlaqəyə malikdirlər, onlar həm də xarici aləmin anlayışları ilə müəyyən münasibətlərlə bağlıdırlar və bu münasibətləri özlərində saxlayırlar.

Ekspert sistemləri istifadə olunduqları tətbiq sahələrinə görə təsnifləş-dirirlər. Bu zaman tətbiq sahələri ekspert sistemlərin metodları ilə səmərəli sürətdə həll olunan əsas məsələlər sinfi ilə müəyyən olunurlar.

Ekspert sistemlər vasitəsilə həll olunan məsələlər altı əsas sinfə bölünürlər: verilənlərin şərhı, diaqnostika, nəzarət, proqnozlaşdırma, planlaşdırma, layihələndirmə.

Təyinatına görə isə serti olaraq ekspert sistemlər konsultasiya, yaxud informasiya, tədqiqat və idarəedici sistemlərə, mürəkkəbliklərinə və biliklər bazalarının həcminə görə isə ekspert sistemlər dərin olmayan və dərin sistemlərə bölünürlər.

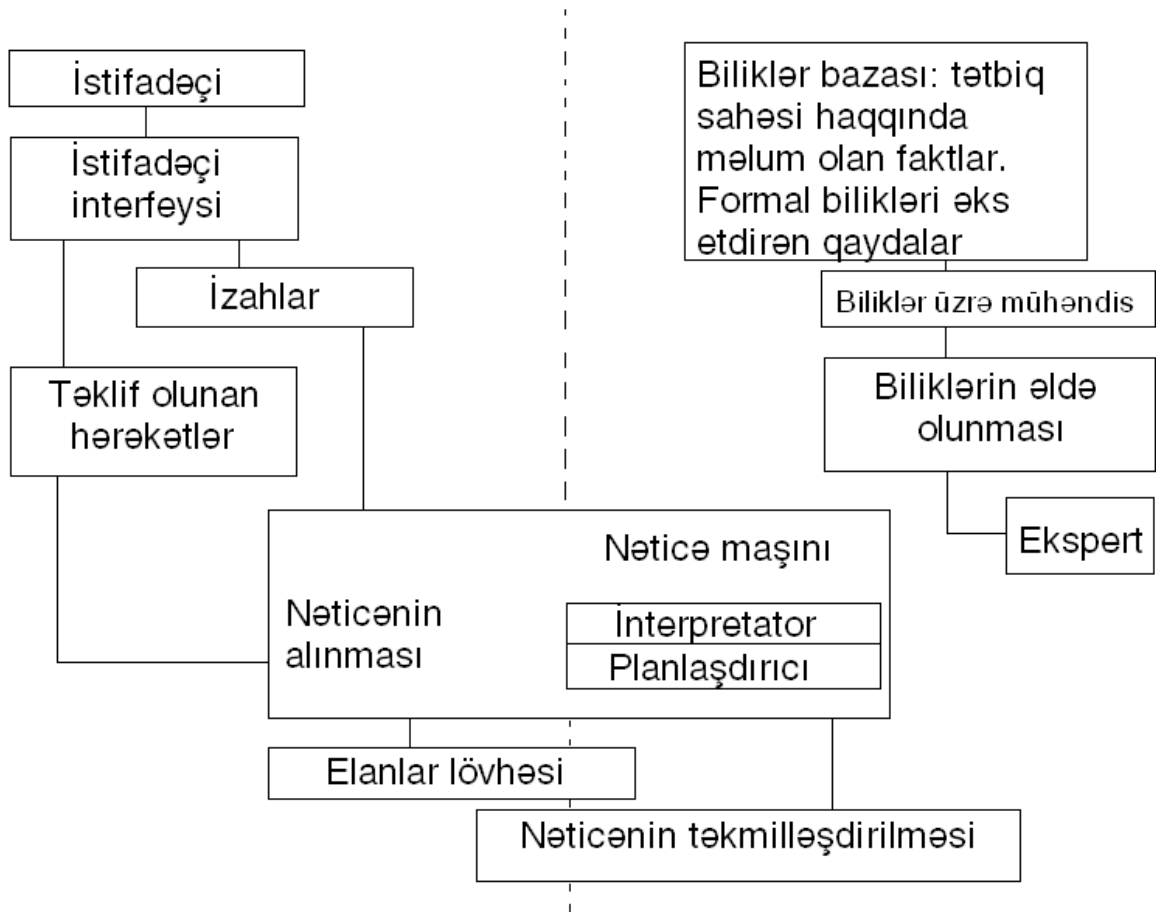
Müasir ekspert sistemlərin tərkibinə, bir qayda olaraq, aşağıdakı komponentlər daxildir (şəkil 1.3).

- biliklərin əldə olunması altsistemi;
- biliklər bazası;
- nəticə maşını;
- elanlar lövhəsi (işçi yaddaş);
- istifadəçi interfeysi;
- izahlar altsistemi;
- nəticələrin təkmilləşdirilməsi və biliklərin «təmizlənməsi» altsistemi.

Biliklərin əldə olunması dedikdə hər hansı bilik mənbəyindən biliklərin toplanması, otürülməsi və müəyyən problemlərin həlli üçün çevrilməsi başa düşülür. Potensial bilik mənbələri ekspert-insanlar, kitablar, verilənlər bazaları, tədqiqat hesabatları, istifadəçilərin öz xüsusi təcrübələri hesab olunur.

Ekspertlərdən biliklərin alınması ekspert sistemlərin yaradılması zamanı həll olunacaq problemlərdən ən çətinidir. Biliklər bazasını yaratmaq üçün ekspertə onun tətbiq sahəsi haqqındakı biliklərini strukturlaşdırmağa kömək edən «biliklər üzrə mühəndis» zəruridir.

Biliklər bazasında informasiyalar problemi başa düşmək, formalaşdırmaq və həll etmək üçün bütün zəruri olanlardır. O, iki əsas elementi özündə saxlayır: tətbiq sahəsindən olan faktlar (verilənlər) və xüsusi evristikalar, yaxud problemin həlli zamanı faktların istifadə olunmasını idarə edən qaydalar.



Şəkil 1.3

Biliklər bazası problemlərin həlli və nəticələrin alınması üçün tez-tez meta-qaydalar (qaydalar haqqında qaydalar) saxlayır. Evristiklər tətbiq sahəsi haqqında formal muhakimələr yürüdürlər. Ekspert sistem üçün ilkin material təmiz faktlar deyil, məhz biliklərdir.

Nəticə maşını ekspert sistemin «beyni» hesab olunur, ona həm də idarəedici struktur, yaxud qaydalar interpretatoru da deyilir. Nəticələr maşının əsas elementləri interpretator və planlaşdırıcıdır.

Elanlar lövhəsi - işçi yaddas sahəsidir və cari problemlərin şərhə üçün istifadə olunur; elanlar lövhəsi həm də aralıq nəticələrin yazılması üçün istifadə olunur. Elanlar lövhəsinə cari hipotezlər və idarəedici informasiya yazılır: plan-problemin həlli üçün hansı strategiya seçilmişdir; gündəlik-yerinə yetiriləcəyini gözləyən potensial hərəkətlər (islər); həll-hipotezalar (fərziyyələr) və sistem tərəfindən generasiya olunan alternativ hərəkət üsulları.

Ekspert sistemlər istifadəçi ilə kompüter arasında «mehriban», problem-orientasiyalı ünsiyyət üçün dil prosessoruna malikdir. Bu ünsiyyət təbii dil vasitəsilə təşkil olunur, qrafika, yaxud çoxpəncərəli menyu ilə müşayət olunur.

Qeyd edək ki, ekspert sistemlərin yaradılması digər problem-orientasiyalı sistemlərə nisbətən «az» avtomatlaşdırılındır. Bu onların yeni tədqiqat sahəsi olması ilə deyil, bəzi tətbiq sahələrindən olan problemlərin «pis» formallaşan olmasıdır. İyirmi illik təcrübə göstərir ki, problem-orientasiyalı sistemlərin bu istiqaməti daha perspektivlidir.

### **§ 1.3 Problem-orientasiyalı sistemlərin generasiyası problemi.**

Problem-orientasiyalı sistem A sinfindən olan məsələlərin həlli zamanı birbirilə qarşılıqlı əlaqədə olan üç komponenti reallaşdırır:

- istifadəçi ilə qeyri-prosedur interfeysin təşkili (problem-orientasiyalı giriş dili və dialoqun aparılması üsulu);
- tətbiq sahəsi modelinə əsasən hesablanmanın planlaşdırılması yolu ilə A sinfindən olan məsələlərin hesablama prosesinin təşkili;
- verilənlərin idarə olunması.

İxtiyari müasir EHM-nin əməliyyat sistemi eynilə bu üç komponenti reallaşdırır.

Problem-orientasiyalı sistem böyük proqram kompleksidir. Bu kompleksi daim aktiv, işlək vəziyyətdə saxlamaq üçün onun yaradıcılarının gündəlik nəzarəti tələb olunur. Aydındır ki, hər bir problem-orientasiyalı sistem hətta bir neçə illik müvəffəqiyyətli istismardan sonra belə tamamlanmış hesab olunmur, çünki sistemin problem vasitələr kitabxanası, giriş dillərinin versiyaları durmadan modifikasiya olunur, yaxud yeniləri yaradılır. Bu elmi-texniki və xalq təsərrüfatı məsələlərinin həlli üçün problem-orientasiyalı sistemlərdən istifadə olunması tələbindən irəli gəlir. Belə ki, problem-orientasiyalı sistemlərin istifadəçiləri daim artır, bu sistemlərin tətbiq dairəsi durmadan genişlənir. Deyilənləri təsdiq etmək üçün POLE və PRİZ komplekslərini misal göstərmək kifayətdir. Hər iki kompleksin yaradıcıları

onilliklər ərzində öz komplekslərində böyük dəyişikliklər etmiş, kompleksin yeni-yeni versiyalarını yaratmışlar. Hər yeni yaranan versiya daha geniş, həm də mürəkkəb, sinif məsələlərin həlli üçün imkanlar yaratmışdır. Buradan belə bir nəticəyə gəlmək olar ki, problem-orientasiyalı proqram təminatının hazırlanmasının sənaye texnologiyasının yaradılması sistem proqramlaşdırılmasının aktual problemlərindəndir. Bu problemin həlli yuxarıda göstərilən üç komponentin instrumental generasiya vasitələrinin yaradılması sayəsində mümkündür.

Instrumental vasitələr sistem proqramçıların rəhbərliyi altında istifadəçi-tətbiqçilər tərəfindən istifadə olunurlar. İstifadəçi-tətbiqçilər A sinfindən məsələləri həll edən, bu sinifdən olan məsələlərin həlli metodlarını mükəmməl bilən və problem-orientasiyalı sistemin problem vasitələri altsisteminə əlavələr etməyi bacaran mütəxəssislərin professional fəaliyyəti ilə yaxından tanış olanlardır.

İstifadə üçün asan və əlverişli problem-orientasiyalı sistem təkcə sistem proqramçılarının gücü ilə yaradıla bilməz. Problem - orientasiyalı sistemin layihəsi istifadəçi - tətbiqçinin A sinfindən olan məsələlərin sistem analizi üzrə apardığı tədqiqatların nəticəsidir.

Beləliklə, universal EHM və problem-orientasiyalı sistemlərin proqramçılarının kateqoriyalarını və onlar arasındakı fərqi müəyyən etmək mümkündür. EHM-lər üçün bu kateqoriya ikidir: sistem proqramçıları və istifadəçi-tətbiqçilər. Problem-orientasiyalı sistemlər üçün istifadə etdikləri vasitələrin səviyyəsinə görə EHM-lərin istifadəçilərinin aşağıdakı ierarxiyası müəyyən olunur:

- sistem proqramçıları - problem-orientasiyalı proqram təminatının təşkilinin sistem vasitələrinin yaradıcıları;
- tətbiqi proqramçılar - problem-orientasiyalı sistemin problem vasitələrinin yaradıcıları;
- istifadəçilər-onların ixtisasi proqramçı deyil, onlar sadəcə olaraq problem-orientasiyalı sistemin istifadəçiləridir.

Problem-orientasiyalı sistemlərin generasiyası məsələsi prosedur dillər üçün sistem proqramlaşdırılmasının generasiyası məsələlərindən daha mürəkkəbdir. Birinci növbədə bu yuxarıda göstərilən üç komponentin daha yüksək səviyyəsi və qarşılıqlı

əlaqəli olması, həm də problem-orientasiyalı sistemin reallaşdırılması zamanı bir-birini əvəz edən kompilyasiya, inter-pretasiya və makrohesablama mərhələlərindən ibarət mürəkkəb sxemin tətbiq olunması ilə izah olunur.

Sadə tip problem-orientasiyalı sistemlər üçün, xüsusi halda hesablama xarakterli problem-orientasiyalı sistemlər üçün (ilk tətbiqi proqramlar paketləri də nəzərdə tutulur) generasiya problemi nisbətən asan həll olunur. Bu məsələnin həlli A sinfindən olan məsələnin həllinin mümkün yolları haqqında məlumatlara əsasən tətbiq sahəsinin modelini quran tətbiqi proqramlar paketlərinin generatorlarının yaradılmasını nəzərdə tutur. Bu modeldə məsələnin konkret həlli metodunun seçilməsi və yanaşı problem vasitələri kitabxanasından hazır modulların qoşulması üsulu da verilir.

Qurulmuş problem-orientasiyalı sistemlərin yaradılması məsələsinin həlli də kifayət qədər sadədir. Lakin, əgər generasiya olunan problem-orientasiyalı sistemlərin giriş dilləri baza prosedur dillər üzərində qurulmamışdırsa, onda belə problem-orientasiyalı sistemlərin generasiya məsələsinin həlli zamanı prosedur dillər üçün metasistemlərin yaradılması təcrübəsindən istifadə etmək lazim gəlir. Bu təcrübənin əsasını proqramlaşdırma dillərinin sintaksis və semantikasının formallaşdırılması sahəsində əldə olunan nailiyyətlər və məsələ, makroprosessorların, kompilyatorların və parametrik sistemlərin, yaradılması təşkil edir.

Vahid informasiya bazası əsasında fəaliyyət göstərən problem-orientasiyalı sistemlərin generasiyası üçün A sinfindən olan məsələlərin bəlli zamanı emal olunacaq informasiyaların əlverişli şəkildə verilməsi üçün verilənlər bazalarının layihələndirilməsinin perspektiv formalarının geniş tətbiq olunması tələb olunur.

Dialoglu problem-orientasiyalı sistemlərin generasiyası problemi nisbətən mürəkkəbdir və az tədqiq olunandır. Bir qayda olaraq, dialog həmişə ixtiyari problem-orientasiyalı sistemdə istifadəçi ilə interfeysin təşkilində köməkçi rol oynamışdır.

Problem-orientasiyalı sistemlərin yaradıcıları diqqəti birinci növbədə problem vasitələrin və problem-orientasiyalı sistemin strukturuna toxunan sistemin təşkili

prinsiplərinin vahid şəklə gətirilməsinə yönəlmişlər. Dialoqun aparılması vasitələrinə yaradılan sistemin müstəqil komponenti kimi baxılmışdır. Bu komponent çox hallarda sistemlər bir neçə il müvəffəqiyyətlə istismar olunduğdan sonra belə reallaşdırılmışdır.

Dialoq xidməti məsələlərinə belə laqeyid münasibətin bir sıra obyektiv səbəbləri vardır. Hər şeydən əvvəl, uzun müddət istifadəçi ilə EHM arasında operativ şəkildə qarşılıqlı əlaqənin təşkili üçün texniki vasitələr olmamışdır. Bu isə onunla nəticələnmişdir ki, dialoq sistemlərin çoxdan (1962-ci il) yaradılmasına baxmayaraq, onların generasiyası problemi axırncı illərdə nəzər diqqəti cəlb etmişdir. Müqayisə üçün qeyd edək ki, proqramlaşdırma dillərinin yaranıb istifadə olunmasından beş il sonra Bröker və Morris tərəfindən 1962-ci ildə metakompilyator yaradılmışdır.

Dialoqlu problem- oriyentasiyalı sistemlərin generasiyası problemi birinci növbədə, dialoq dillərin modelini vermək üçün təyin olunmuş, dillərin sintaksis və semantikasının formallaşdırılması vasitələrinə əsaslanmalıdır. Bu modellır dialoqlu qarşılıqlı əlaqə prosesini və təbii ki, dil konstruksi-yalarının mənalərini adekvat əks etdirməlidirlər (dialoqun strategiyası və üsulu).

Dialoqlu sistemlərin fəaliyyət prosesini şərh etmək üçün avtomat və qrammatik modellər müvəffəqiyyətlə tətbiq oluna bilər. Bu halda dialoq sistemlərdə hesablamanın təşkili dedikdə, funksional modullar toplusu ilə reallaşdırılan müəyyən bir vəziyyətdən dialoq sistemin istifadəçinin cavab-larına reaksiyası ilə əlaqədar digər vəziyyətə keçidlər ardıcılığı başa düşü-lür. Beləliklə, dialoq xidmətin spesifikasiyası tətbiq sahəsi modelində kifayət qədər formallaşdırıla və əks oluna bilər.

Mürəkkəb, lakin istifadə üçün əlverişli problem- oriyentasiyalı sistemlərin generasiyasının instrumental vasitələrinin yaradılması probleminin həlli yolları artıq müəyyənləşdirilmiş və bu istiqamətdə xeyli istər görülmüşdür.

Məlumdur ki, problem- oriyentasiyalı sistemlərin parametrləri aşağıdakılardır:

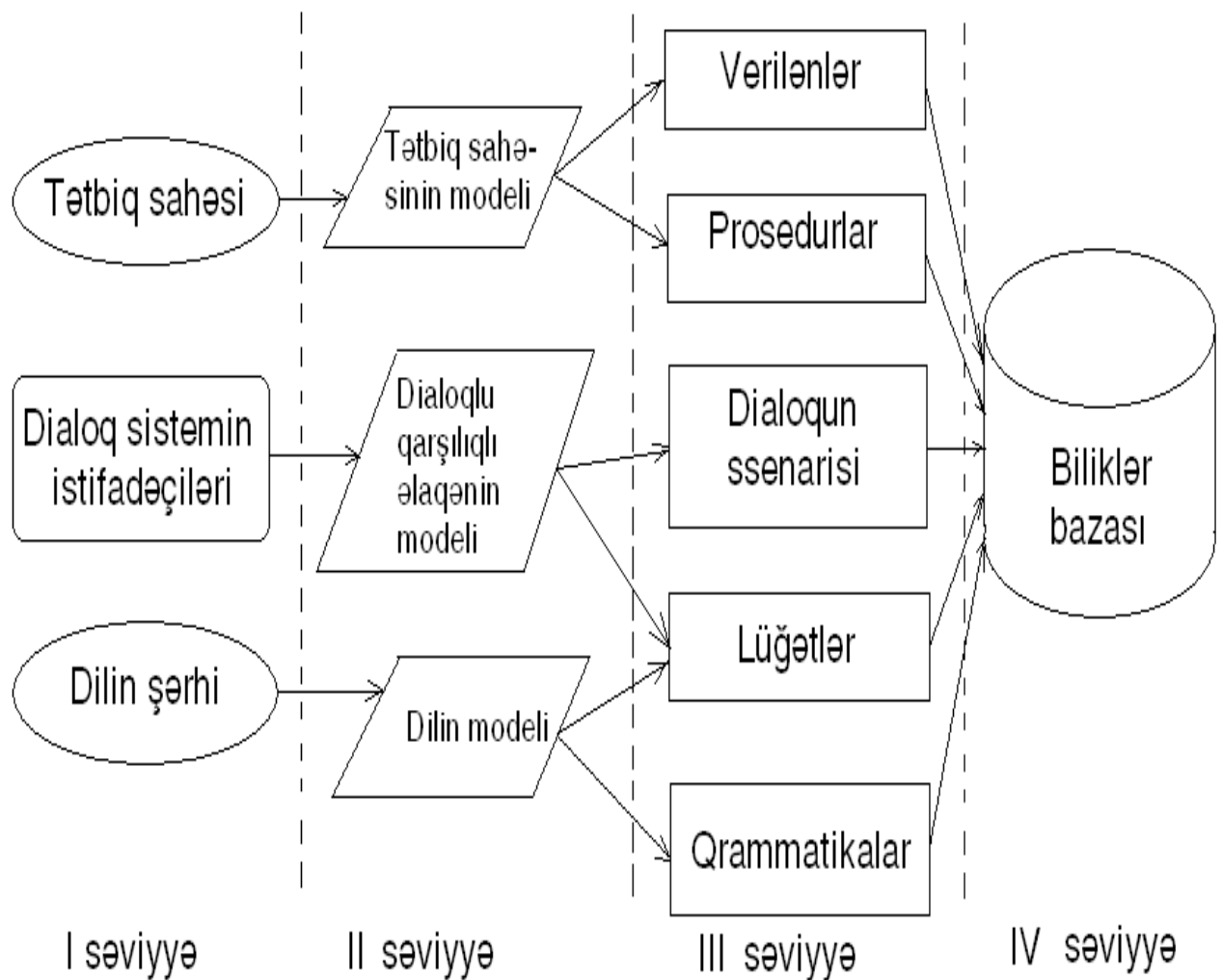
- 1) tətbiq sahəsinin modeli;
- 2) problem vasitələr kitabxanasından olan modullar arasında informa-siya

interfeysinın təşkilı üçün tətbiq sahəsinin modeli ilə bağılı olan verilənlər bazası və onun sxemi;

3) problem vasitələr kitabxanası;

4) giriş dili; bu dil qrammatik (sintaksisi əks etdirən), program (seman-tikanı əks etdirən) və tətbiq sahəsi modelinə əsasən hesablamaların planlaş-dırılması metodu aqreqları toplusu şəklində verilir.

Bu parametrlərin çox sadə münasibətlərlə əlaqələndirildiyi problem-orientasiyalı sistemin modelinin qurulması hal-hazırda çox mürəkkəb bir məsələdir. Lakin, belə modeli qurmadan problem-orientasiyalı sistemlərin generasiyası məsələsinin həlli mümkün deyildir.



Şəkil 1.4 Dialoqlu sistemin strukturu



Problem- oriyentasiyalı dillər prosedur dillərlə müqayisədə yeni tip dillər olduğundan, belə dillərin reallaşdırılmasının yeni formaları zəruridir. Gözlə-mək olar ki, problem- oriyentasiyalı dillərin semantik aspektlərini əks etdirən dialoqun ssenarisi və tətbiq sahəsi modeli arasındakı asılılığın formalaşdırılması əsaslı çətinliklər əmələ gətərəcəkdir. Formallaşdırma məsələsi həm də dialoq sistemlərdə geniş tətbiq olunan makrohesablama ifadələrinə, sistem proqramlaşdırmasında ənənəvi qəbul olunmuş reallaşdırılan dillərin sintaksis və semantikasi arasındakı qarşılıqlı əlaqə formasına aiddir.

Dialoq sistemlərin generatorunda dialoq sistemlərin yaradılması məsələlərinin ardıcıl olaraq dəqiqləşdirilməsinin ümumi sxemi maraq doğurur. Bu sxemdə biliklərin təsvir edilməsini əlverişli şəkildə dörd səviyyəyə ayırmaq olar: xarici səviyyə; model səviyyəsi; məntiqi səviyyə; fiziki səviyyə (biliklər bazası) (şəkil 1.4).

## FƏSİL II TƏTBİQİ PROQRAM TƏMİNATININ LAYİHƏLƏNDİRİLMƏSİ METODLARI

### 2.1 Proqram təminatının yaradılmasının seçilməsi metodları.

Proqram təminatının yaradılmasının keçirilməsi (seçilməsi) metodları xeyli inkişaf etmiş metodlardır, belə ki, onlar proqramçıların professional fəaliyyət sahəsinə aid edilirlər. Onların yerinə yetirilməsi, yəni keçirilməsi vasitələri çoxdan məlumdur və dinamik olaraq inkişaf etdirilirlər. Spesifikasiyaların müəyyən olunması metodları çox yaxşı şəkildə formallaşdırılmışdır. Burada artıq xeyli effektiv metodologiya olmasına baxmayaraq, bəzi texniki problemlər aradan qaldırılmamışdır.

Yoxlama və sınaq. Yoxlama və sınaq demək olar ki, sistemin yaradılması üçün nəzərdə tutulan vaxtın yarısını aparır. Bu proseslərlə bağlı, sistemin yaradılmasına ayrılan xərclərinin azaldılması üçün xeyli sazlanma vasitələri yaradılmışdır. Bu vasitələr sistemin bu və ya digər xarakteristikalarının yoxlanılması üçün yaradılan xüsusi proqramlardır. Bu proqramların ən köhnə və primitivləri, yəni ibtidailəri **damp və köçürülmə** hesab olunur.

Damp – kompüterin yaddaşının məzmununun çap olunmasıdır. Lakin dampda axtarış effektiv deyildir, belə ki, o səhv baş verdikdən sonra keçən vaxtdan xeyli müddət sonra əldə olunur. Odur ki, səhvin baş vermə səbəbini müəyyən etmək xeyli çətin olur.

Köçürülmə - yəni istiqaməti göstərilmə dedikdə proqramdakı hər bir operator yerinə yetirildikdən sonra verilmiş dəyişənlərin qiymətlərinin analizi nəzərdə tutulur.

Proqramların qraflarının analizatorları ya adlandırılmamış dəyişənlərə müraciət olunduğu vəziyyəti, yaxud da bu dəyişənlərə qiymət mənimsədildikdən sonra onlara müraciət olunmadığını müəyyən edə bilirlər.

Proqramların testləşdirilməsi üçün test verilənlərinin **generatorları** istifadə olunur. Verilmiş möqtələrdə müəyyən şərtlərin yoxlanılması razılaştırma

kompilyatorunun köməyi ilə həyata keçirilir. Nisbətən sadə hesab olunan dillər üçün avtomatik yoxlayan sistemlər yaradılmışdır. Spesifikasiyaların müəyyən olunması mərhələlərinin və layihələndirmənin işlənməsi üçün təyin olunmuş vasitələrə misal olaraq **PSL/PSA** sistemini göstərmək olar.

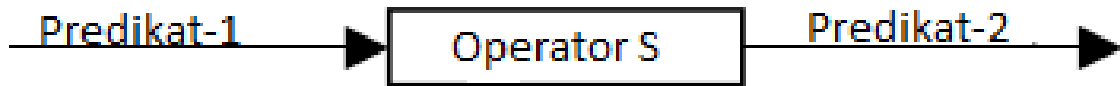
Proqram təminatının yaradılması prosesinə tətbiq edilməklə “düzgün” proqram anlayışı müxtəlif interpretasiyalara malik ola bilər. Proqram təminatında əsasən səkkiz əsas tezisdən (müddəadan) istidadə olunur. Düzgün proqram:

- 1) sintaktik səhvlərə malik deyildir;
- 2) kompilyasiya prosesində buraxılan səhvlərə, yaxud yerinə yetmə prosesində kəsilmələrə malik deyildir;
- 3) müəyyən test verilənləri toplusu üçün düzgün nəticənin alınmasını təmin edir;
- 4) tipik test verilənləri toplusu üçün düzgün nəticənin alınmasını təmin edir;
- 5) mürəkkəbləşdirilmiş test verilənləri toplusu üçün düzgün nəticənin alınmasını təmin edir;
- 6) məsələnin spesifikasiyalarını ödəyən bütün mümkün olan verilənlər topluları üçün düzgün nəticənin alınmasını təmin edir;
- 7) bütün mümkün olan verilənlər topluları və səhv daxil olmanın bütün mümkün şərtləri üçün düzgün nəticənin alınmasını təmin edir;
- 8) bütün mümkün verilənlər topluları üçün düzgün nəticənin alınmasını təmin edir.

Təbiidir ki, düzgünlüyün səkkizinci səviyyəsini həmişə əldə etmək mümkün deyildir. Həm də bu çox zaman zəruru hesab edilmir. Adətən proqram kompleksləri üçün düzgünlüyün altıncı səviyyəsi kifayətdir.

Düzgünlüyün hər bir səviyyəsini əldə etmək, yəni bu səviyyəyə çatmaq xərc tələb edir və proqram kompleksi layihələndirilərkən kompleksi yaradan özü zəruri olan düzgünlük səviyyəsinin əsaslandırılmalıdır. Proqramların “düzgünlük” səviyyəsinin yüksəldilməsi yollarından biri verifikasiyadır (yoxlamadır). Proqram təminatının yaradılması metodlarının inkişafının hazırkı səviyyəsində verifikasiyanın, yəni yoxlamanın tam avtomatlaşdırılması mümkün deyildir.

Müxtəlif verifikasiya (yoxlama) sistemləri üçün ümumi olan proqramın qraf şəkildə təqdim olunmasıdır. Bu qrafın hər bir tilinə müəyyən bir predikat, yəni məntiqi şərt (müddəa) əlaqələndirilir (şəkil 2.1)



Şəkil 2.1. Proqramın hər bir operatorunun  $A_i$  və  $A_j$  müddəaları ilə əlaqəsi

Əgər  $A_i$   $S$  operatoruna daxil olan tillə əlaqədar giriş müddəası,  $A_j$  isə həmin operatorun çıxış tili ilə əlaqədar müddəadırsa, onda operatorun düzgünlüyü isbat olunur «əgər  $A_i$  həqiqidirsə və  $S$  operatoru yerinə yetirilmişdir, onda  $A_j$  müddəası həqiqidir». Uyğun proses proqramın hər bir operatoru üçün təkrarlana bilər. Əgər  $A_1$  qrafın başlanğıc tərə nöqtəsinə uyğun müddəadırsa (başlanğıc müddəa),  $A_n$  – qrafın son tərə nöqtəsinə uyğun müddəadırsa, onda «əgər  $A_1$  həqiqidirsə və proqram yerinə yetirilmişdir, onda  $A_n$  müddəası həqiqidir» (şəkil 2.2).



Şəkil 2.2.  $A_1$  və  $A_n$  predikatlarının köməyi ilə proqramın girişinin və çıxışının müəyyən olunması

Belə yanaşma Xoor tərəfindən formallaşdırılmışdır. O bir sıra aksiomlar vasitəsi ilə proqramlaşdırma dilində hər bir tip operatorun müddəaya təsirini

müəyyən etmişdir. Beləliklə proqramın düzgünlüyünün isbatı predikatların hesablanması teoreminin isbatına gətirilir.

Təbiidir ki, belə riyazi yanaşma zamanı proqramın işinin nəticəsini onun spesifikasiyaları ilə müqayisə etmək zəruridir. Hər bir proqram yalnız müəyyən spesifikasiyalar toplusuna nəzərən düzgündür. Ona görə də aşağıdakı suallar meydana çıxır:

- spesifikasiya proqrama qoyulan tələblərə ekvivalentdirmi ?
- bu spesifikasiyalara nəzərən proqram düzgündürmü ?

Bu suallar proqramların düzgünlüyünün analizi zamanı cavablandırılır.

## **2.2 Proqram təminatının yaradılması metodlarının inkişafı.**

Çox yaxşı məlum olan ənənəvi proqram təminatının yaradılması (layihələndirilməsi) metodlarından başqa hal hazırda avtomatlaşdırılmış yaradılmaya istiqamətlənmiş metodlar geniş tətbiq olunurlar.

### **Tapşırıqların müəyyən olunması dili və tapşırıq analizatoru.**

Bu metodların içərisindən Miçiqan univeristetinin yaratmış olduğu və iki baza hissədən ibarət olan **İSDOS** metodunu xüsusilə qeyd etmək lazımdır:

1) **PSL** tapşırıqların şərh olunması dili funksional tələblərin və ehtiyatlara olan tələblərin inikası üçün təyin olunmuşdur. Bu dil istifadəçiyə sistemin obyektlərini, onların xassələrini müəyyən etməyə, həm də obyektləri qarşılıqlı əlaqə vasitələri ilə birləşdirməyə imkan verən vasitələr toplusuna malikdir.

2) **PSA** tapşırıqların müəyyən olunması analizatoru prosessordur və onun köməyi ilə **PSL** dilində yazılmış cümlələrin yoxlanılması həyata keçirilir. Analizator verilənlər bazası generasiya edir, belə ki, bu baza sistem tələblərini inikas etdirir və verilənlərin tamlığını, həm də ardıcılığının yoxlanılmasını həyata keçirir. O həmçinin bir sıra verilənlərin seçilməsi haqqında məlumatları, həm də

səhvlər haqqında məlumatları, verilənlər bazasında olan obyektləri, onlar arasındakı qarşılıqlı əlaqələri saxlayan çıxış sənədlərini dəstəkləyir.

**PSL** dili sadə sintaksisə malikdir və açar sözlərindən istifadə olunmaya istiqamətləndirilmişdir. Bu dilin əsas operatorları aşağıdakılardır:

<b>PROCESS</b> < ad >	< ad > yeni proses kimi müəyyən olunur;
<b>DESCRIPTION</b> < mətn >	< mətn > ingilis dilinin vasitələri ilə proseslə reallaşdırılan funksiyaların şərhinin təqdimatıdır;
<b>SUBPARTS ARE</b> < ad >	< ad > prosesi cari proseslə əlaqədardır və iyerarxiyalar ağacında ondan a.ağıda yerləşmişdir;
<b>PART OF</b> < ad >	< ad > prosesi cari prosesi çağırır;
<b>DERIVE</b> < fayl >	< fayl > faylı cari proses üçün çıxış faylıdır;
<b>USING</b> < fayl >	< fayl > faylı cari proses üçün giriş faylıdır;
<b>PROCEDURE</b> < mətn >	< mətn > proses vasitəsi ilə reallaşdırılan PDL dilində şərh olunan alqoritmin təqdimatıdır.

Misal: əgər **Y** prosesi **PART OF X** cümləsini saxlayırsa; onda **X** prosesi **SUBPARTS ARE Y** cümləsini daxil etməlidir.

**PSL/PSA** sistemi aşağıdakı xarakteristikalara malikdir:

- 1) istifadəçiyə problemin formallaşdırılmış təqdimatını almağa imkan verir;
- 2) sistem tələblərini vahid şəkllə salınmış formada sənədləşdirilməsini həyata keçirir;
- 3) bəzi növ səhvlərin yaranması şərtlərinin, ilk növbədə informasiyanın qeyri tamlığı və onların daxil edilməsi ardıcılığının pozulması ilə şərtləndirilən səhvlərin aşkarlanmasına imkan verir.

**İSDOS** metodunun üçüncü tərkib hissəsi **PDL** – proqramların layihələndirilməsi dilidir. Bu dil iki kateqoriya strukturundan ibarətdir. Əsas tip operatorlar ( **if – then – else, sequence** və başqaları) üzərində qurulan xarici sintaksisi vardır. Xarici sintaksis ayrı ayrı komponentlərin proqramda birləşdirilməsi

üçün təyin olunmuşdur. Bundan başqa, konkret tətbiq üçün nəzərdə tutulan daxili sintaksis də vardır. Daxili sintaksis təbii dilin vasitələri ilə təsvir olunur.

Misal:

**Max: procedure** (list);

/\* Siyahıda ən böyük elementin axtarılması \*/

**declare** (maxmin, next) tam;

**declare** list tam ədədlər ardıcılığı;

maxmin = list –in birinci elementi;

**do while** (list –də elementlər vardır)

next = list – də növbəti element;

maxmin = next – dən və maxmin – dən ən böyüyü;

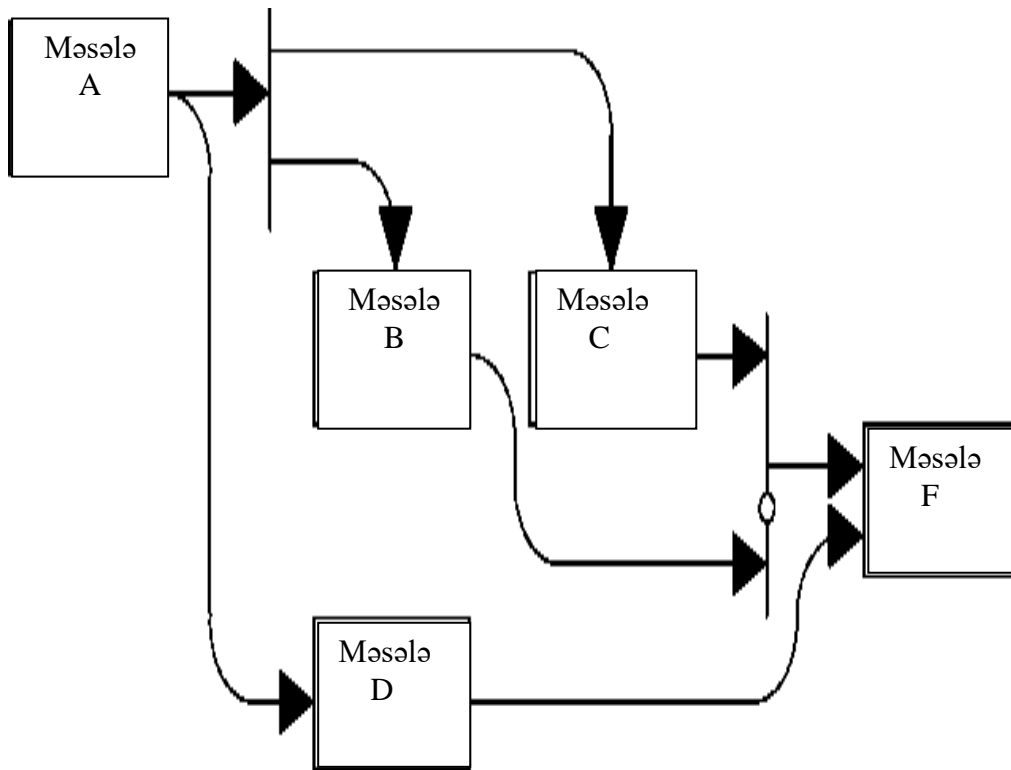
**end;**

**return** (maxmin);

**end** Max;

Təbiidir ki, baxılan sistem çatışmamazlıqlardan xali deyildir. Birinci növbədə bir çox spesifikasiyalar üçün həddindən artıq universal olduğu faktı aşkardır. Lakin, nəzərə alsaq ki, bu sistem avtomatlaşdırılmışdır, sistemin yaradılmasının çox böyük həcimli sənədləşdirilməsini yerinə yetirir, onun tətbiq olunması özünü doğruldur.

**SADT** struktur analizi və layihələndirmə sistemi kimi fəaliyyət göstərir. **SADT** - Software Technology firma markasının abbreviaturasıdır və böyük proqram komplekslərinin layihələndirilməsi üçün təyin olunmuş avtomatlaşdırılmamış qrafiki sistemdir. **SADT** sisteminin qrafiki dili strukturlaşdırılmış iyerarxik diaqramlar toplusudur, özü də diaqramın hər bir bloku digər diaqramın köməyi ilə daha ətraflı şəkildə açılır. Beləliklə, sistemin layihəsinin yaradılması müddətində qurulan modelin strukturu ən kiçik detallarına qədər dəqiqləşdirilir (şəkil 2.3). **SADT** sistemindən istifadə edərkən sistemi yaradanlardan hər birinin ciddi müəyyənləşdirilmiş səlahiyyətlərə malikdirlər.



Şəkil 2.3. SADT sisteminin strukturu (SA – sistemin komponentləri arasında qarşılıqlı əlaqələrin şərhinin formal dilidir)

Sistemin strukturu:

*A məsələsi*

*(B yaxud C məsələsi) və D məsələsi*

*F məsələsi*

Mütləq nəzərə almaq lazəmdır ki, **SADT** sistemindən istifadə edərkən sistemi yaradanlardan hər birinin ciddi müəyyənləşdirilmiş səlahiyyətlərə malikdirlər və bu qaydanı pozmaq olmaz.

Müəlliflər. Sistemə qoyulən məhdudiyyətlərin və tələblərin öyrənilməsi və onların **SADT** sisteminin köməyi ilə şərh etməklə məşğul olan sistemin yaradıcıları.



Şərhcilər (kommentatorlar). Adətən bu qrupa daxil olanlar öz kolleqalarının ( müəlliflərin) işini analiz edən və görülən işlərə irad tutmağa çalışan sistemi layihələndirənlərdir.

Oxucular. Bu qrupa daxil olanlar isə digər mütəxəsislər tərəfindən yaradılan layihələrin analizi ilə məşğul olanlardır və onlar öz işləri haqqında kommentariya verməyə bilərlər.

Texniki komitə. Sistemin şərhinə əssən onun layihəsini ən yuxarı səviyyədə analiz edən təcrübəli texniki mütəxəsislər qrupu.

Layihənin kitabxanaçısı. Layihənin fayllarının aparılmasına cavabdeh olan şəxs.

Layihənin rəhbəri. Layihənin texniki cəhətdən yaradılmasına məsuliyyət daşıyan əsas şəxs.

Baş analitik. **SADT** sisteminin istifadə olunması üzrə əsas məsləhətçi. O **SADT** sisteminin istifadə olunmasının xüsusiyyətlərini çox yaxşı bilir və onun tətbiq olunması haqqında məsləhətlər verir.

Təlimçi. **SADT** sisteminin istifadə olunmasını istifadəçilərə öyrədən şəxs.

**SADT** sisteminin əsas üstünlüklərinə aşağıdakıları aid etmək olar :

1) sistem kollektiv işin təşkilinə, həmçinin layihələndirmənin ilkin mərhələlərində spesifikasiyalara nəzərən sazişlərin qurulmasına imkan verir;

2) texniki komitənin yazılı hesabatları ardıcıl olaraq nəzarət analizinin aparılmasına imkan verir ki, bu da sistemin sınaqdan keçirilməsi zamanı az əhəmiyyət kəsb eləmir.

3) ekspertlərin çıxışları, yəni söylədikləri fikirlər xüsusi informasiyaların alınmasının effektiv vasitəsidir;

4) sistem başlanğıc mərhələlərdə yüksək səviyyədə əsas qərarları qəbul etməyə imkan verir. Bu isə sonrakı aşağı səviyyələrdə növbəti qərarların qəbul edilməsi üçün möhkəm bünövrə yaradır.

5) **SADT** sisteminin istifadə olunması proqram təminatının yaradılması və istifadəsi üzrə mütəxəssis olmayan şəxslərə də yaradılan sistemi dərk etməyə imkan verir;

6) layihələndirmənin gedişinə nəzarət etmək üçün çox asan istifadə oluna bilən vasitələr nəzərdə tutulmuşdur.

Bu sistemin ən böyük çatışmayan cəhəti onun avtomatlaşdırılmamasıdır.

**SREM** sistemi proqram təminatına qoyulan *tələblərin analizi* mərhələsinin avtomatlaşdırılması üçün istifadə olunur. Bu sistemə tələblərin müəyyən olunması dili daxildir. Bu dil **RSL** adlanır və onun vasitəsi ilə sistemin obyektləri arasında əlaqə yaradılır. **RSL** dilində cümlələrin ardıcılığının yoxlanılması **REVS** prosessorunun köməyi ilə həyata keçirilir. **SREM** sisteminin istifadəsi zamanı aşağıdakı addımlar yerinə yetirilir:

- **Translyasiya.** Verilənlərin şərhədiciləri və onların işlənməsi mərhələləri də daxil olmaqla tələblər sistemi yaradılır.

- **Dekompozisiya.** Ətraflı layihələr hazırlanır.

- **Paylanma.** **REVS** prosessorunun köməyi ilə qəbul olunmuş səpmələr nəzərə alınmaqla layihə həllinin ayrı ayrı aspektləri modelləşdirilir. Nəticədə **REVS** prosessoru vasitəsi ilə generasiya olunan sistemin qurulmasına qoyulan tələblər çoxluğu alırıq.

- **Analiz.** İstifadəçi gələcəkdə yaradılacaq, yəni yaradılması nəzərdə tutulan sistemə qoyulan bütün tələbləri yoxlayır.

**Cekson metodikası.** Cekson metodikasına ənənə (azalan) layihələndirmə, struktur proqramlaşdırma və struktur nəzarət analizi daxildir. Bu metodikaya uyğun olaraq struktur proqramlaşdırmada idarəetmə strukturlarına oxşar dörd komponenti olan iyerarxik strukturlaşdırılmış proqramlar qurulur. Bu komponentlər aşağıdakılardır :

Element. Daha sadə funksiyalara bölünən bilməyən funksiya.

Ardıcılıq. Ardıcıl və təkrarlanmadan reallaşdırılan funksiyalar sırası.

Çıxarış (seçmə). Mümkün ardıcılıqlardan biri.

İterasiya. Sıfır da daxil olmaqla, verilmiş sayda yerinə yetən funksiya.

Bu metodun əsas ideyası ondan ibarətdir ki, sistemin strukturu istifadə olunan verilənlərin strukturuna tam oxşar, yəni verilənlərin strukturu ilə eyni olsun.

Beləliklə, sistemin ağacabənzər təşkilı sxemi verilənlərin strukturunu əks etdirməlidir: əks halda layihə düzgün hesab edilməyəcəkdir.

Proqram təminatının layihələndirilməsi metodlarının icmalı onu göstərdi ki, bütün baxılan metodların hamısına ümumi əlamətlər xasdır. Belə metodlar sistemin rəşional strukturunun alınmasına imkan versələr də, sistem yaradılan zaman onun əsas mərhələləri hesab olunan tələblərin analizi, spesifikasiyaların müəyyən olunması, layihələndirmə mərhələlərinə yaradıcılıqla yanaşmaq zərurətini inkar etmir. Bu metodlardan istifadə olunmaq hesabına proqram təminatının yaradılmasına peşəkarcasına yanaşmağa daha çox imkan yaranır.

Müxtəlif metodları vahid bir metodologiyada birləşdirən bit sıra strategiyalar mövcuddur. Ən effektiv strategiya Boem tərəfindən yaradılmışdır. O proqram təminatının yaradılmasının yeddi prinsipini formalaşdırmışdır (tərtib etmişdir):

1) *proqram təminatının həyat siklinin ayrı ayrı mərhələlərinin ardıcılıqla reallaşdırılması əsasında hazırlanmasının idarə olunması.* Bu prinsipin tətbiq olunması hər bir mərhələdə əvvəlki addımlarda əldə olunmuş nəticələr nəzərə alınmaqla əsaslandırılmış həll (qərar) qəbul etmək olar. Bu prinsip həm də imkan verir ki, layihənin hazırlanmasının vəziyyətini qiymətləndirmək üçün nəzarət nöqtələrindən istifadə olunsun.

2) *sınağın yerinə yetirilməsi.* Modulun təkmilləşdirilməsinin hər bir addımında onun attestasiyası həyata keçirilir.

3) *Hazırlanmanın gedişi üzərində ciddi nəzarətin həyata keçirilməsi.* Bütün çıxış sənədləri – proqramların layihələri, ilkin proqramlar, istifadəçi üçün təlimatlar və s. – formal olaraq təsdiq olunmalıdır. Sənədlərdə və proqramlar kitabxanasında ixtiyari dəyişikliyin edilməsinə ciddi nəzarət olunmalıdır və yalnız uyğun məsul şəxsin icazəsi ilə belə dəyişiklik həyata keçirilə bilər.

4) *Struktur proqramlaşdırma vasitələrinin bütün diapazonunun istifadə olunması.* İmkan dairəsində arzu olunandır ki, əlverişli idarəetmə strukturları və verilənləri reallaşdırmağa imkan verən dillər tətbiq olunsun. Prosesləri şərh etmək üçün addımlarla təkmilləşdirmə texnikasının istifadə olunması arzu olunandır.

5) *Ciddi uçot.* Sistemin hazırlanması zamanı əldə olunan tərəqqinin uçotu üçün nəzarət nöqtələrinin istifadə olunması zəruridir. Hər bir icraçının işinin yoxlanması üçün isə sistem jurnalından istifadə olunmalıdır.

6) *Yüksək ixtisaslı mütəxəssislərdən təşkil olunmuş çox da böyük olmayan kollektivin istifadəsi.* Baş proqramçının briqadasının təşkili prinsipinin istifadə olunması işin yaxşı nəticəsini əldə etməyə imkan verir.

7) *Yüksək səviyyə.* Proqram təminatının hazırlanması və texnologiyası sahəsində əldə olunmuş nəticələrdən istifadə etmək zəruridir. Bu zaman proqramların modifikasiyası və etibarlılığı haqqında da düşünmək lazımdır.

Proqram təminatının hazırlanması üzərində idarəetmə və nəzarət vasitələrinin hələ də təkmilləşdirilməyə ehtiyacı vardır, lakin idarəetmə prosesinin öz q bu zaman digər texniki oblaslara xas olan səviyyəyə yaxınlaşır.

### **2.3 Proqram təminatının hazırlanması metodları.**

Aparılan tədqiqat işləri göstərir ki, hal hazırda proqram təminatının hazırlanması metodları intensiv şəkildə inkişaf edir. Lakin hazırlanan metodların hamısı praktiki olaraq geniş tətbiq olunmur.

Proqramların layihələndirilməsi dili.

Proqramlar yaradılan zaman istifadəçi adətən iki esuldan istifadə edir: o ya proqramların struktur sxemini qurur, yaxud da elə birbaşa proqramı yazır. Aydındır ki, ikinci üsulda proqramın strukturu güman ki, qeyri dəqiq olur. Lakin məlumdu ki, çox yaxşı struktur sxemi də bəzən pis struktura malik proqramın yaradılmasına gətirib çıxarır. Elə bir yazılar sistemi zəruridir ki, onun köməyi ilə proqramların formal layihələndirilməsi mümkün olsun.

Belə yazılar sisteminin yaradılması üçün **PDL** – proqramların layihələndirilməsi dili yaradılmışdır. Bu dil universal proqramlaşdırma dillərinin tərzinə görə yaradılmışdır. Lakin qeyd etmək lazımdır ki, proqramlaşdırma və layihələndirmə dilləri müxtəlif məqsədlərlə istifadə olunurlar. Odur ki, onları müqayisə etmək düzgün deyildir. Proqramlaşdırma dilləri adi proqramların

verilənlərin strukturlarının emalı ilə əlaqədar olduğu halda, layihələndirmə dillərində belə “elementar konstruksiyalar”, məsələn, “məsələnin həlli” yaxud “matrisin çevrilməsi” ola bilər yaxud da aşkar şəkildə verilmiş ifadə olmaya bilər. Əgər proqram yazısı proqramlaşdırma dilində məlumdursa, onda layihələndirmə mərhələsi mütləq deyildir.

Layihələndirmə dili adətən iki hissədən ibarət olur.

- proqramlaşdırma dillərinin nümunəsinə görə yaradılan verilmiş operatorlar toplusu;

- verilmiş oblastda məsələlərin şərh üçün yararlı olan ümumi adətən şərtləndirilməyən sintaksis.

Beləliklə proqramların layihələndirilməsi dili, proqramlar layihələndirilən zaman proqramın məntiqini şərh edən, müəyyən xarici sintaksisə malik olur. Bu məntiq bir çox proqramlaşdırma dillərində olan idarəedici strukturların köməyi ilə qurulur.

Digər tərəfdən, **PDL** dili bütün verilənlərin strukturlarını və onların işlənməsi prosedurlarını saxlayan qeyri müəyyən daxili sintaksisə malikdir. Demək olar ki, təbii dildə ixtiyari cümlə verilənlərin çevrilməsinin şərh üçün istifadə oluna bilər.

**if X** mənfi deyildir

**then return** ( **X** – ın kvadrat kökü - həqiqi ədəd);

**else return** ( **X** – ın kvadrat kökü - lal ədəd);

Belə olan halda **PDL** dilində proqramın İF - THEN - ELSE məntiqi qeydə alınır, daxili ifadələr isə (məsələn, “kvadrat kök”) müəyyən olunurlar. Sonra isə **PDL**-proqramın köməyi ilə layihələndirmə həyata keçirilir “ X – dan kvadrat kökün alınması”. Layihələndirmə dilin adətən *pseudokod* adlandırılır.

**PDL** dilinə altı operator daxildir.

**Seçmə operatoru.**

a) İF ifadə;

THEN operator-1;

ELSE operator-2;

Operatorlardan hər biri DO – END qrupuna daxil olan operatorlar ardıcılığı ola bilər.

```
b) DO CASE (ifadə);
    /indeks-1/ operator-1;
    .....
    /indeks-n/ operator-n;
    ELSE operator- n+1;
    END
```

CASE operatörü bir neçə variantdan birinin seçilməsi üçün istifadə olunur. CASE operatoru ifadənin qiymətini hesablayır və indeksi ifadənin qiymətinə bərabər olan operatoru yerinə yetirir. Əgər heç bir indeks ifadənin hesablanmış qiymətinə bərabər deyilsə, onda ELSE –dən sonrakı operator yerinə yetirilir (təbii ki, əgər bu operator varsa). İF operatoru kimi bu operatorlardan hər biri DO – END qrupuna daxil ola bilərlər.

#### **Dövr operatoru.**

```
a) DO WHILE (ifadə);
    Operatorlar toplusu;
    END;
```

Operatorlar toplusu ifadənin qiyməti doğru olana qədər yerinə yetirilirlər.

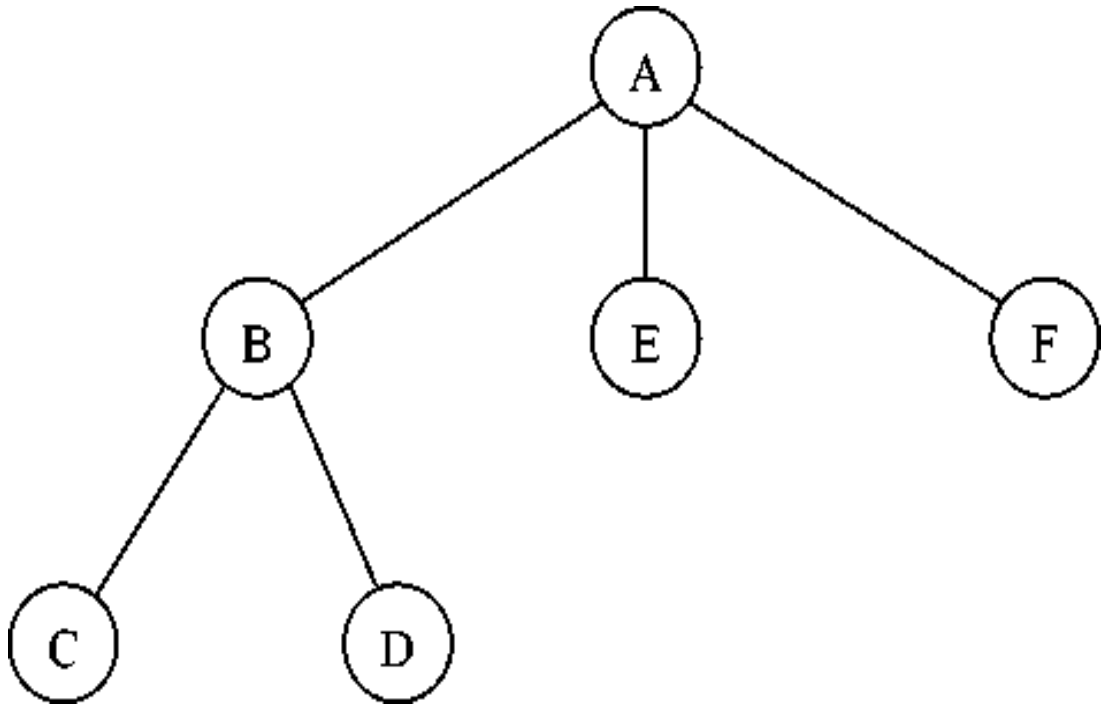
```
b) DO dəyişən = ifadə-1 TO ifadə-2 BY ifadə-3;
    Operatorlar toplusu;
    END;
```

Birinci dəfə dövrə daxil olan zaman ifadə-1, ifadə-2 və ifadə-3 selementlərinin qiymətləri hesablanır. İfadə-3 artım addımıdır və o həm müsbət, həm mənfi qiymət ala bilər, yaxud da buraxıla bilər (belə olan halda susmaya görə addın +1 qəbul edilir və yazılmır). Dövr ixtiyari sayda yerinə yetirilir.

#### **Verilənlərin şərh operatoru.**

```
DECLARE ad atributlar;
```

Adlar dəyişənlər üçün olan atributlar siyahısı üzrə elan olunurlar. Atributlar proqramlaşdırma dilinin standart verilənlər tipi ola bilərlər (real, integer, float və başqaları) yaxud da yüksək səviyyəli verilənlərin strukturu ola bilərlər (şəkil 2.4)



Şəkil 2.4. **PDL** dilində verilənlərin strukturu

Mürəkkəb verilənlər strukturlarını müəyyən etmək üçün aşağıdakı tip strukturlardan istifadə olunur.

```

DECLARE 1 A,
      2 B,
      3 C,
      3 D,
      2 E,
      2 F;
  
```

Belə yazılış yuxarıdakı şəkildə göstərilən (şəkil 2.4) ağacın strukturuna yuğundur.və bu strukturun elementlərinə istinad üçün dəqiqləşdirilmiş adlar sistemindən istifadə olunur. Beləliklə C düyününə elə A, B, C düyünlərinə

müraciət olunan qaydada müraciət oluna bilər, lakin C düyününə, əgər o birqiymətli istifadə olunubsa, bilavasitə də müraciət etmək olar.

### **Digər operatorlar.**

- a) <dəyişən > = <ifadə>;
- b) CALL prosedurun adı(arqumentlərin siyahısı);
- c) RETURN (qiymət);
- d) <ad> PROCEDURE (<parametrlərin siyahısı>);  
     <operatorların siyahısı>;  
     END;
- e) GET <daxil ediləcək verilənlərin siyahısı>;
- f) PUT <çap ediləcək verilənlərin siyahısı>;

Prosedurdakı bütün parametrlər istinadların köməyi ilə çağrılır (başqa sözlə, dəyişənlərin ünvanları prosedura ötürülür). Adların təsir oblastı onların elan olduğu blok hesab olunur, onlar bu blokda şərh olunurlar.

### **LEAVE operatoru.**

**LEAVE** operatoru DO operatoru vasitəsi ilə təşkil olunmuş dövrdən çıxmağı təmin edir. Bu operator idarəedicilik keçid operatoru tipindədir.

### **Təbii dildə cümlələr.**

Yuxarıda baxılan beş tip operatorlardan başqa təbii dildə yazılmış ixtiyari cümləni **PDL** dilinin operatoru kimi istifadə etmək olar.

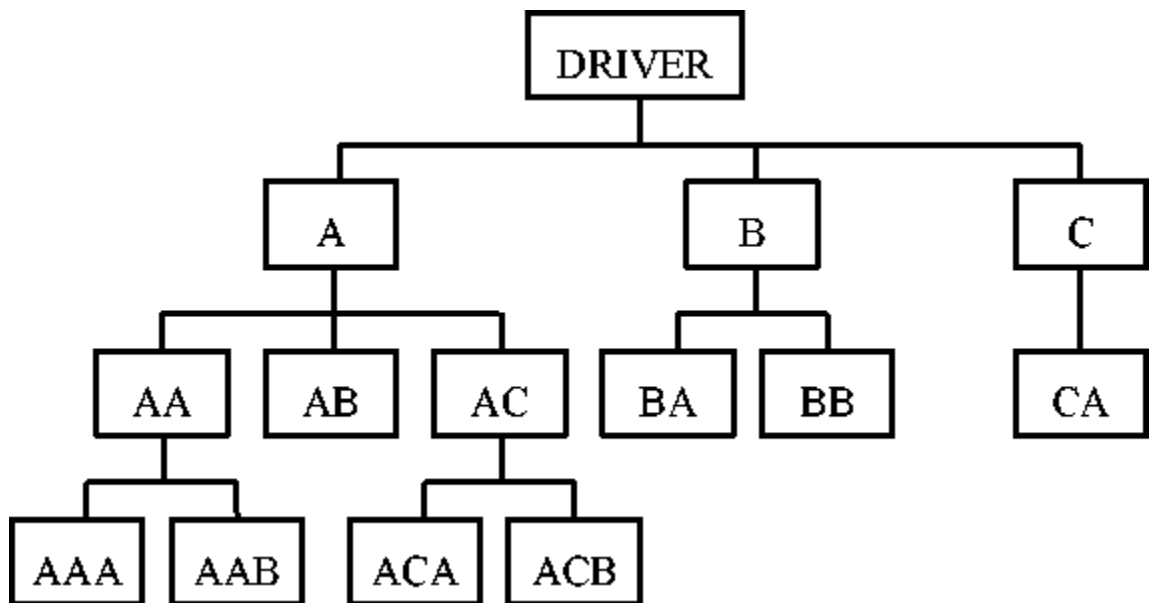
- a) B massivinin ən böyük elementini tapmalı;
- b) {a, b, c} çoxluğundan olan bütün X-lar üçün DO;
- c) C – dən böyük olan A = <B-nin birinci elementi>;

**PDL** dili üçün xüsusi translyator yaradılmışdır. Belə translyatorun köməyi ilə ixtiyari sistemi öyrənmək və müşayət etmək üçün zəruri olan sənədləri almaq mümkündür. Bundan başqa, proqramlaşdırma dilinə oxşar olan dil vasitəsi ilə layihələndirmə, səhvlərin sayının xeyli azaldılmasına imkan verir.

## **2.4 Proqramların hazırlanması strategiyaları.**



Proqramlaşdırma dili sadəcə olaraq layihələrin yalnız hazırlanmasına xidmət edən vasitədir. Düzgün layihələrin qurulması zamanı əsas rolu lahiyələndirmə metodologiyaları oynayır. Layihələndirmə mərhələsində proqramlar hazırlanarkən adətən iki yanaşmadan istifadə olunur: **azalan** və **yüksələn**. Azalan layihələndirmənin mahiyyətini aşağıdakı sxem vasitəsi ilə aşkar izah etmək olar (şəkil 2.5).



Şəkil 2.5 Bazis sxeminə misal

Göstərilən bazis sxemində hər bir blok – sistemin moduludur. Bu zaman hər bir modulun çağırılması nisbətən yüksək səviyyə modulun vasitəsi ilə yerinə yetirilir.

Azalan (enən) layihələndirmə zamanı əvvəlcə idarəedici proqram – drayver layihələndirilir. İdarəedici modul **PDL** dilində təqdim oluna bilər.

**DRIVER: PROCEDURE;**

A məsələsini yerinə yetirməli;

**DO WHILE** (şərt doğrudur);

```

        B məsələsini yerinə yetirməli;
    END;
        C məsələsini yerinə yetirməli;
    END DRIVER;

```

Sonra isə psevdokodun hər bir operatoru ətraflı təsvir olunur və digər modullar hazırlanırlar. Məsələn, A, B, C məsələləri kifayət qədər mürəkkəbdirsə, onda onları ayrıca prosedur kimi təşkil etmək olar. Belə olan halda drayverin layihəsini aşağıdakı kimi təqdim etmək olar.

```

DRIVER: PROCEDURE;
    A məsələsinə başlamalı;
    CALL A;
    DO WHILE (şərt doğrudur);
        B məsələsinə başlamalı;
        CALL B;
    END;
    C məsələsinə başlamalı;
    CALL B;
END DRIVER;

```

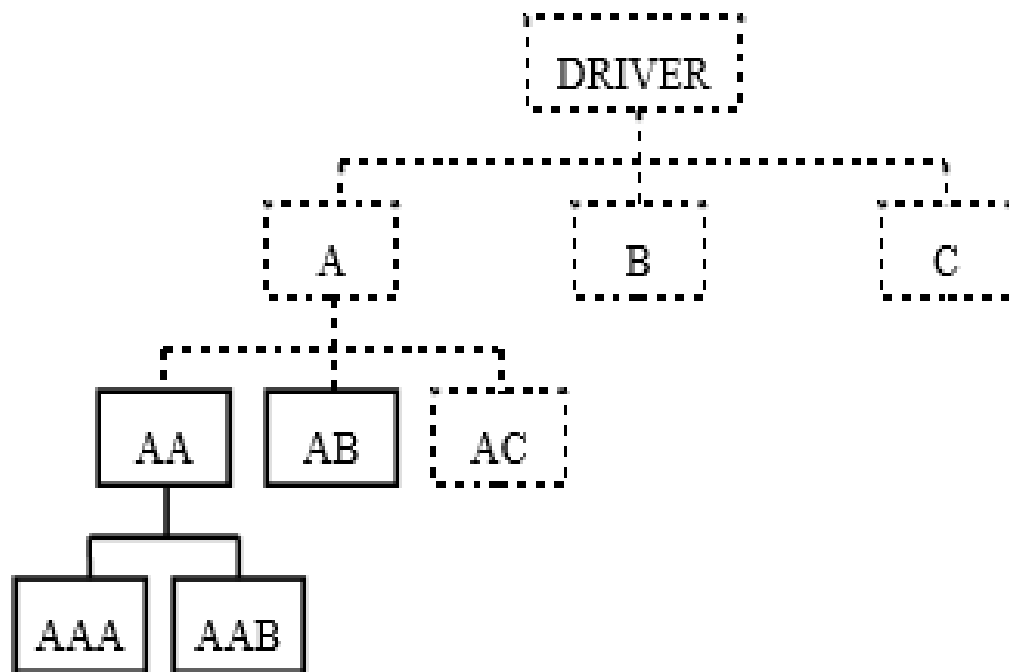
Sonra isə elı bu formada A, B, C prosedurlarını müəyyən etmək olar. Göründüyü kimi PDL dili azalan (enən) layihələndirmə üçün çox əlverişlidir.

Azalan (enən) layihələndirməni həm də addımlarla təkmilləşdirmə adlandırılır: proqramlar iyerarxik strukturlaşdırılırlar və ardıcıl dəqiqləşdirmə yolu ilə bölünürlər. Hər bir addımda modulun fəaliyyəti əvvəlki daha dəqiq addımlara istinadların köməyi ilə şərh olunurlar.

Yüksələn layihələndirmə zamanə əvvəlcə aşağı səviyyənin proqramları layihələndirilirlər. Adətən belə yanaşma əməliyyat sistemləri yaradılarkən istifadə olunur, belə ki, burada iyerarxiyanın ən aşağı səviyyəsi apatat vasitələrindən ibarət

olur (virtual maşınlar texnologiyası). Məsələn, modullardan biri EHM-nın aparat vasitələrinə müraciəti təmin edə bilər və yerdə qalan bütün modullara virtual yaddaş təqdim edə bilər. Bunun nəticəsində bir çox real vaxt rejimində işləyən sistemlər aşağıdan yuxarıya layihələndirilirlər.

Kodlaşdırma və testləşdirmə mərhələlərində vəziyyət təmənilə əksinədir. Buna baxmayaraq, əksər sistemlər yuxarıdan aşağıya layihələndirilirlər, kodlaşdırma və testləşdirməni aşağıdan yuxarıya həyata keçirmək əlverişlidir, AAA və AAB modulları digər komponentləri çağırırlar, onları kodlaşdırırlar və testləşdirirlər (şəkil 2.6).



Şəkil 2.6 Yüksələn kodlaşdırma və testləşdirmə.

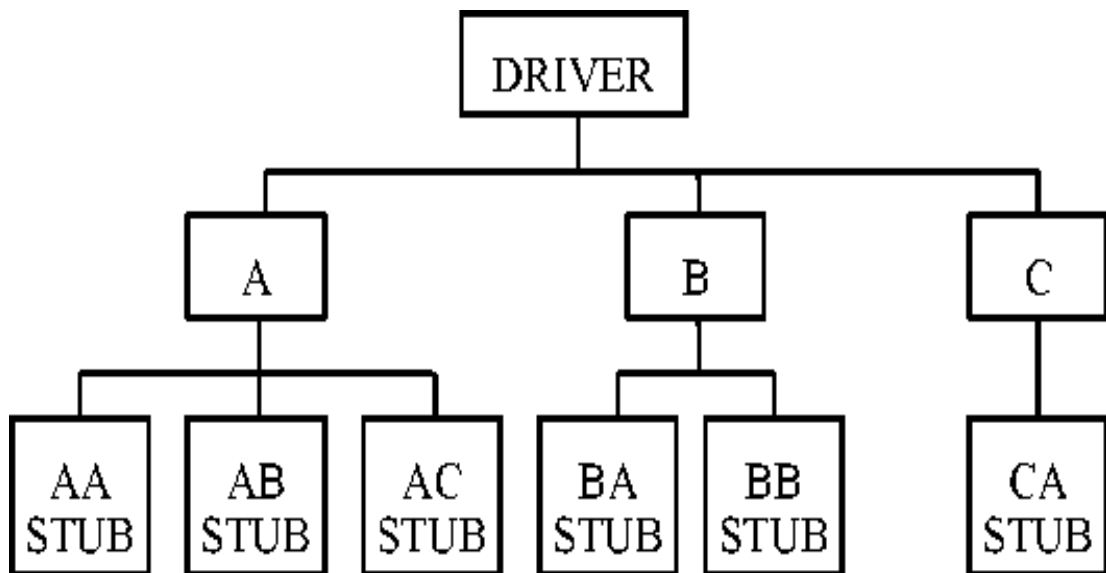
Qoyulmuş məsələ yaxşı müəyyən olunduqda belə yanaşmadan istifadə etmək çox əlverişlidir.

Lakin, əgər həll olunacaq məsələ aydın deyilsə, yaxud detallarına qədər müəyyən olunmayıbsa, onda aşağıdan yuxarıya testləşdirmə ciddi problemlər yarada bilər. Məsələn, istifadəçi yuxarı səviyyə modullar yoxlanılmayana qədər, spesifikasiyalara uyğun olaraq sistemin düzgün fəaliyyət göstərdiyinə əmin ola bilmir. Lakin sistemin bütün iyerarxik strukturu yoxlanılmayana qədər bunu etmək

mümkün deyildir, başqa sözlə, layihə bitənə qədər bunu etmək mümkün deyildir. Bu mərhələdə isə dəyişikliklərin aparılması çox böyük itkilərə səbəb olur və baha başa gəlir.

Belə vəziyyətləri aradan qaldırmaq üçün azalan (enən) kodlaşdırmadan istifadə etmək olar. Bu halda birinci növbədə idarəedici proqramın modullarını, həmçinin A, B, C modullarını yoxlayırlar. Sistemin istifadəçisi hazırlanmanın ilkin mərhələsində yuxarı səviyyənin fəaliyyətini yoxlayır, ona görə də spesifikasiyalarda ixtiyari zəruri olan dəyişiklikləri etmək xeyli asandır.

Kodlaşdırmanın belə metodunun yeganə əlverişsiz olması onunla əlaqədardır ki, A, B, C modullarının yoxlanılması üçün həm də AA, AB, AC, BA, BB və CA modulları yoxlanılmalıdırlar. Bu məqsədə “dəm tutan” – tıxac proqramlar xidmət edirlər. Bu proqramlar çox kiçik proqramlardırlar və məxsusi olaraq hələ yazılmamış modulları modelləşdirmək və idarəedici proqramlara zəruri olan test verilənlərini ötürmək üçün istifadə olunurlar (şəkil 2.7).



Şəkil 2.7 Azalan (enən) testləşdirmə

Qeyd edək ki, azalan layihələndirmə ilk baxışdan göründüyü kimi, o qədər də sadə deyildir. Bu onunla əlaqədardır ki, ixtiyari proqram sistemində üç zirvəsi var:

- 1) işin başlanğıcı;

- 2) idarəedici proqram;
- 3) istifadəçinin sistemlə əlaqəsi proqramı.

Bu anlar, yəni bu üç zirvə arasındakı əsas fərqi kompilyatorun misalında göstərmək olar. Aparat vasitələrini hazırlayan üçün sistemin “zirvəsi” sistemin işinin başlanğıcını gösrərən moduldur. Bu modul - əməliyyat sistemi ilə interfeysin əsas vasitəsi hesab olunur. Onun köməyi ilə diskdən kompilyasiyanın müxtəlif mərhələlərini reallaşdıran proqram fragmentləri oxunur və idarə onların yerinə yetirilməsinə verilir. Sistemin bütün qalan hissələrinə bu modulun alt proqramları kimi baxmaq olar.

Sistem proqramçısı üçün “zirvə” idarəedici proqram hesab olunur. Kompilyatorsa “zirvə” proqramın əsas analizi dövrü hesab olunur, belə ki, bu zaman növməti analiz olunacaq operator (leksemlər) axtarılır. Beləliklə aşağıda göstərilən dövr məntiqi zirvə hesab oluna bilər:

```

DO WHILE (kompilyasiyanın sonuna kimi davam etdirməli);
Növbəti operatorun əvvəlinə qədər oxumalı;
Daxil edilən operatorun analizi;
END;
```

İstifadəçiyə gəldikdə, o belə hesab edir ki, kompilyator əvvəlcə operatoru oxuyur, sonra isə translyasiya edir. Beləliklə onun üçün “zirvə” verilənləri daxil edən proqramdır.

Bütün sistem üçün zirvənin seçilməsi proqramçının məharətindən və yaxşı ixtisasçı olmasından asılıdır. Bu mərhələdə dəyişikliklərin edilməsi xeyli asandır. Ardıcıl modifikasiya metodu da eynilə bu xüsusiyyətlərə malikdir. Bu metoddan istifadə olunarkən əvvəlcə sistemin hər hansı bir variantı layihələndirilir və reallaşdırılır. İstifadəçi çox tez bir zamanda işlək sistem alır.

## FƏSİL III. İNTERAKTİV MARŞRUTLU SİSTEMLƏRİN YARADILMASI TEXNOLOGİYASI

### 3.1. İnteraktiv marşrutlu sistemlər.

Tətbiqi proqramlar paketlərinin idarə olunmasının informasiya instrumental kompleksi, marşrutlu sistemlər adlanan (İMS-sistemlər), bir sinif dialoqlu problem-yönümlü sistemlərin generasiyası üçün yaradılmışdır. Marşrutlu sistemlər ayrı-ayrı tətbiqi proqramlar paketi ilə həll olunan  $B_k(k=1, \dots, n)$  məsələlər sinfini genişləndirən ( $k=1, \dots, n$ )  $D_k$  giriş dillərinə malikdirlər. Onlar müəyyən tətbiqi proqramlar paketlərinin inteqrasiyası vasitəsi ilə yaradılırlar. Hər hansı bir oblastdan olan  $B$  məsələlər sinfindən olan məsələni həll etmək üçün o ayrı-ayrı paketlərlə həll olunan alt məsələlərə bölünürlər.

İnteraktiv marşrutlu sistemlərin generasiyası mərhələsində tətbiqi proqramlar paketlərinin idarə olunmasının informasiya instrumental kompleksi, bilikləri toplayan instrumental sistemlərə aid edilə bilər. Onlar qeyri prosedur proqramlaşdırma dilləri sinfinin semantikasını haqqında biliklərin təqdim olunmasında dünyada ilk cəhdlərdən biri hesab olunur. İnteraktiv marşrutlu sistemlərin generasiyası prosesi iterasiyalı və çoxmərhələli olur. Generasiya mərhələlərinin ardıcılığı tətbiqi proqramlar paketlərinin idarə olunmasının informasiya instrumental kompleksi vasitəsilə nizamlana bilər.

İnteraktiv marşrutlu sistemlərin digər sistemlərdən fərqləndirici xüsusiyyəti kimi, onlarda interpretasiya rejiminin geniş tətbiq olunmasını göstərmək bilərik. Bu xüsusiyyət paketin funksional modullarının istifadə olunan verilənlərdən asılı olmamasını daha da gücləndirə bilər, dialoq rejimində addımlarla hesablamağa geniş şərait yaradır. Bu zaman generasiya prosesinin özü də asanlaşır. Hər bir interaktiv marşrutlu sistem malik olduğu funksional imkanlarına görə çoxdilli interaktiv, yəni dialoq rejimində işləyən sistem olur. İnteraktiv marşrutlu sistemlər bir sıra əsas elementlərə malik olurlar. Bu elementlərdən birincisi həll olunacaq məsələlərin şərhini üçün təyin olunmuş giriş

dillərinin, yəni paketlərinin giriş dillərinin tətbiqi proqramlar paketlərinin idarə olunmasının informasiya instrumental kompleksinin interaktiv dili ilə birləşməsindən yaranan komponentdir. İkinci element, sistemin rezidenti (inteqrasiya olunmuş hissəsi) – interaktiv, addım addım hesablama aparən alt system hesab olunur. Kompleksin problem hissəsi, yəni verilənlərin idarə olunması haqqında razılaşmaların vahid şəkllə salınması, daha doğrusu unifikasiya olunması ilə bağlı bəzi dəyişikliklər aparılmış funksional modullar çoxluğu üçüncü elementdir.

İnteraktiv marşrutlu sistemin k- cı tətbiqi proqramlar paketinin giriş dili aşağıdakı şəkildə göstətilə bilər:

$$D_k = (\{ L_{1k}, \dots, L_{mk} \}, \{ L_{1k}, \dots, L_{rk} \}, Q_k(U_k, V_k)), k=1, \dots, n;$$

Burada  $\{ L_{1k}, \dots, L_{mk} \}$  – birləşdiriləcək tətbiqi proqramlar paketinin ilkin parametrləri toplusu,  $\{ L_{1k}, \dots, L_{rk} \}$  – birləşdiriləcək tətbiqi proqramlar paketinin işi nəticəsində alınan parametrlər toplusudur.  $Q_k$  – k – cı paketinin marşrutlu qrafıdır və  $Q_k(U_k, V_k)$  istiqamətlənmiş yüklü qrafı şəklində inikasını əks etdirir.

$$Q_k: \{ L_{1k}, \dots, L_{mk} \} \rightarrow \{ L_{1k}, \dots, L_{rk} \}$$

$Q_k$  qrafının  $U_k = \{ U_{1k}, \dots, U_{rk} \}$  təpə nöqtələri toplusu paketdə təpə nöqtələri arasında idarəedici əlaqələrə və informasiya axınlarını uyğun gəlir.  $Q_k$  qrafına əsasən, k- cı paketdə hesablama prosesinin planlaşdırılması zamanı interaktiv rejimin ssenarisi yaradılır.

İnteraktiv marşrutlu sistemlərdə tətbiqi proqramlar paketlərinə xidmət dialoq rejimində vahid  $D_k$  dilində aparılır. Baxılan dil sistemin istifadəçisinə iş prosesində cəld şəkildə bəzi lazımi informasiyaları daxil etməyə, yaxud da paketdə məsələnin digər uyğun yollardan birini seçməklə qərarın qəbul edilməsinə şərait yaradır.  $D_k$  dilinə aşağıdakı direktiv-operatorlar daxildir:

- işin başlanğıcı və məsələnin həllinin bərpa olunması operatoru;
- sistemdə olan kəsilmələr və baxılan məsələnin həlli addımlarının icrasının ardıcılığının dəyişdirilməsi direktiv-operatoru;

- proqramlar paketinin hər bir parametrlərinin ayrılıqda qiymətlərinin monitorun ekranına verilməsi və məsələlərin həlli gedişində bu qiymətlər üzərində düzəlişlərin aparılması direktiv-operatoru;

- başlanğıc verilənlərin versiyalarının, yəni variantlarının və hesablamaların son nəticələrinin arxivə yerləşdirilməsi direktiv-operatoru;

- proqram paketlərinin başlanğıc verilənləri üzərində düzəlişlərin aparılması direktiv-operatoru.

Tətbiq oblastından olan məsələnin həlli zamanı tətbiqi proqramlar paketinin daxilində istifadəçi ilə interaktiv rejim marşrutlu qraflar üzrə direktiv-operatorlardan ibarət seanslarla reallaşdırılır. Bu seanslara həm sistemin istifadəçi, həm də sistemin özü tərəfindən start verilir. Əgər seans sistem tərəfindən başlanılıbsa, onda o, yəni sistem, diaqnostik məlumatları istifadəçiyə ötürür və istifadəçi bəzi parametrlərin qiymətlərində dəyişiklik etməyə (yaxud əlavə olaraq göstərməyə) imkan qazanır.

Seans zamanı həllin bir neçə mümkün addımlarının təkrarlanması halı baş verə bilər. Sistemin istifadəçisinin arzusu ilə məsələnin həlli prosesini hər bir zaman anında dayandırmaq mümkündür; belə olan halda tətbiqi proqramlar paketinin vəziyyətləri vektoru xüsusi cədvəldə qeydə alınır və onun işi yenidən dayandırıldığı nöqtədən davam etdirilir.

Sistemin dialoq dilini  $D_K=(G, \mathcal{X}, \delta)$  şəklində vermək olar, burada  $G$ -bütün sistemin ümumi marşrutlu qrafları,  $\mathcal{X}$  və  $\delta$  isə uyğun olaraq tətbiqi proqramlar paketlərinin parametrləri üzərində dəyişiklik aparmaq üçün ssenarilərin qurulduğu və  $A$ -dan olan məsələlərin həlli gedişində alt məsələlərə parçalanmasının əks olunmalarıdır.

Sistemdə  $\mathcal{X} = \{ \mathcal{X}_k / k=1, \dots, n \}$  – multi əks olunması məsələnin həllinin mümkün hər bir addımında tətbiqi proqramlar paketlərinin parametrləri üzərində dəyişikliklərin aparılması qaydalarıdır.

$$\lambda_k = \begin{cases} \{ \{ P_{1k}, \dots, P_{mk} \} \rightarrow \{ P_{1k}, \dots, P_{mk} \}, \\ M_{ik} \rightarrow M_{jk} (i, j = 1, \dots, S_k). \end{cases}$$

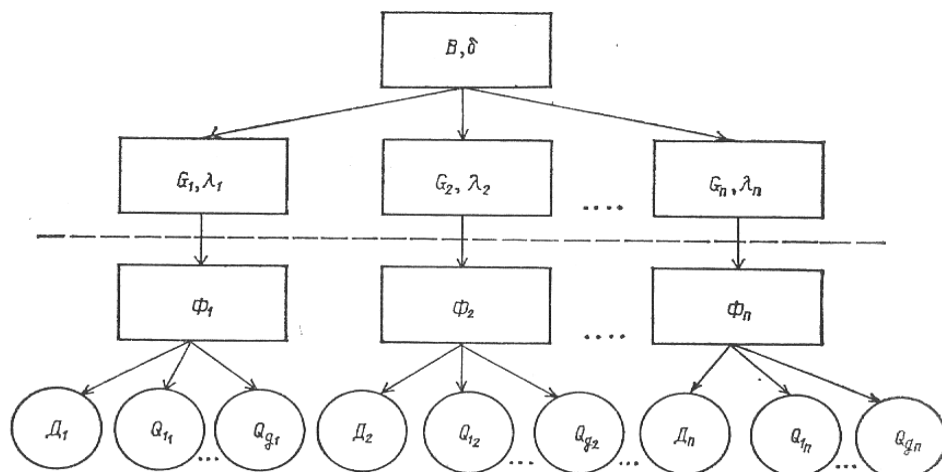


Sistemdə  $\mathcal{X}_k$  inikasının birinci hissəsi tətbiqi proqramlar paketinin həlli addımlarında parametrlərin qiymətləri üzərində dəyişikliklərin aparılması qaydalarıdır. Həmin qaydalar istiqamətin seçilməsi şərtləri və  $Z_{ik}$  addımından  $Z_{jk}$  addımına keçərkən,  $\{Z_{1k}, \dots, Z_{mk}\}$ -da olmaqla, üzərlərində dəyişikliklərin aparılmasına icazə verilən parametrlərin adları toplusundan ibarətdir. Marşrutlu qrafın  $Z_k$  təpə nöqtələri çoxluğunda verilən  $Z_{ik} \rightarrow Z_{jk}$  münasibəti  $Z_{ik}$  addımından  $Z_{jk}$  addımına mümkün keçidləri müəyyənləşdirir.

$$\delta = \begin{cases} \{\bar{P}_{1k}, \dots, \bar{P}_{rk}\} \rightarrow \{P_{1l}, \dots, P_{ml}\} (k \neq l; k, l = 1, \dots, n), \\ \tilde{M}_{pk} \rightarrow M_{1l}, \end{cases}$$

təpə nöqtəsində alınan  $\{D_{1k}, \dots, D_{rk}\}$  nəticələrinin  $l$ -ci paketin  $Z_{1l} \in Z_l (l=1, \dots, n)$  kök təpə nöqtəsində  $\{D_{1l}, \dots, D_{ml}\}$  başlanğıc verilənlər şəklində tətbiq olunmasının razılaşdırılması ssenarisini təqdim edir.

A - şəbəkə marşrutlu qrafları toplusu kimi İMS sisteminin informasiya bazasının strukturunu əks etdirir (şəkil 3.1). Aşağıdakı şəkildə yuxarıdakı sahə, şərti olaraq iki səviyyəli iyerarxik ağac şəklində göstərilmişdir, A - şəbəkəsinə uyğun gəlir. Həmin ağacın kökü  $D_1[Q_1, \mathcal{X}_1], \dots, D_n[Q_n, \mathcal{X}_n]$  psevdorqraflarının birləşdirilməsi qaydasının inikasidir. Bu şəkildə ağacın aşağı sahəsi isə ayrı-ayrı tətbiqi proqramlar paketlərinin informasiya və bilik bazalarının strukturlarını əks etdirir.



Şəkil 3.1. İnteraktiv sistemin bilik bazasının strukturu

Baxılan sistemdə  $Z_k$  ( $k=1,\dots,n$ ) tətbiqi proqramlar paketlərinin  $\{L_{1k},\dots,L_{mk}\}$  öz parametrlər toplusunun strukturunu əks etdirir;  $Q_{1k},\dots,Q_{gk}$  – paketlərin aralıq fayllarının strukturunu göstərir və həmin fayllarda hesablama addımlarının nəticələri qorunur;  $T_1,\dots,T_n$  – hesablamanın mərhələlərinin sistem identifikatorlarını və tətbiqi proqramlar paketlərinin parametrləri toplusunda münasibətlərinin inikası olur. Beləliklə tətbiqi proqramlar paketlərinin parametrləri siyahısı və marşrutlu qraflar arasındakı münasibətlər qeydə alınır. Bunlar tətbiqi proqramlar paketlərinin adlar teki adını almışlar. Yaradılan tekdə bir sıra informasiyalar, o cümlədən  $Q_{1k},\dots,Q_{gk}$  aralıq faylların adları və parametrlər çoxluğu (adlar, şablonlar,  $B_k$  məsələsinin şərtində tutduğu yer) haqqında məlumatlar da qorunurlar.

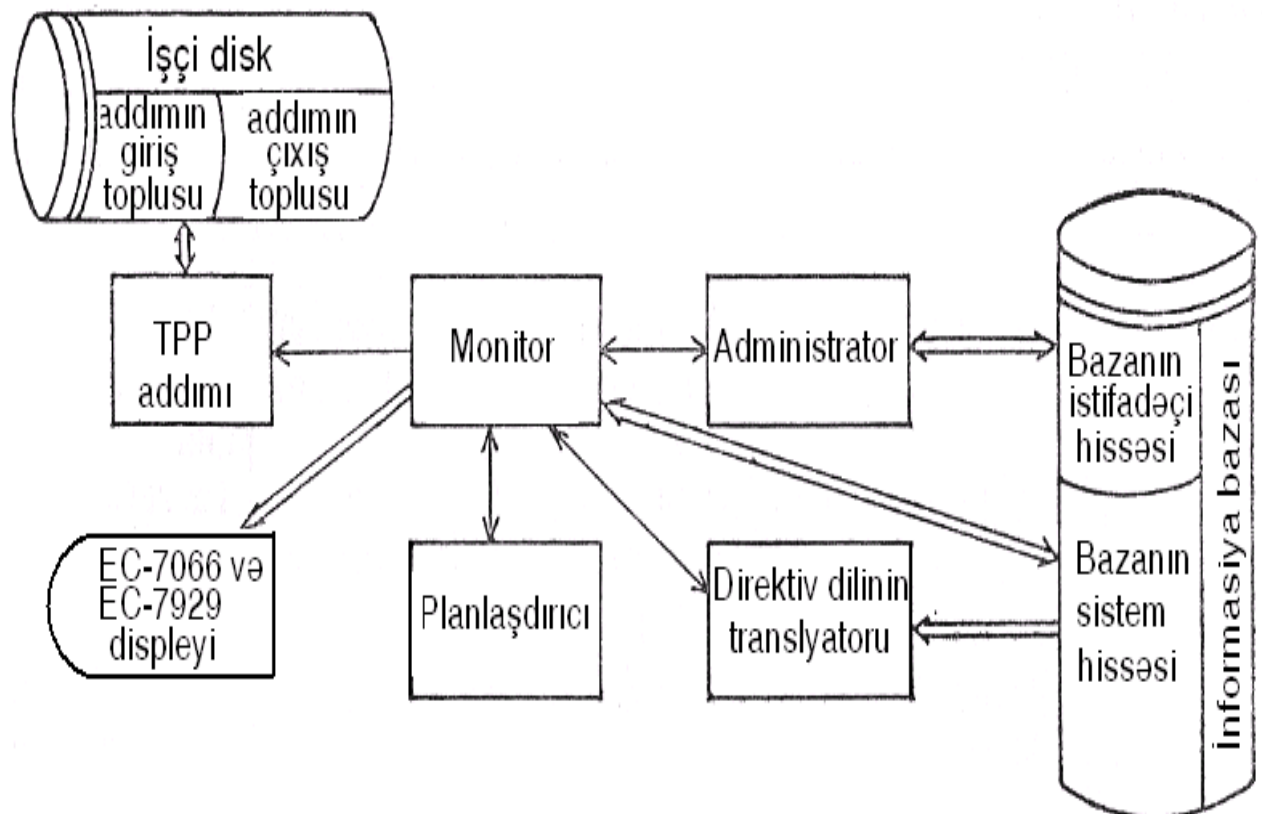
İMS – interaktiv marşrutlu sistemdə həll olunan hər bir məsələyə uyğun  $D_k^h, Q_{1k}^h, \dots, Q_{gk}^h$  ( $gk \leq S_k; h=1,\dots,t$ ) hər bir parametrlər çoxluğu həmin məsələlərin həll olunduğu dövrdə məsələlər kitabxanasında olurlar (şəkil 3.2). Mümkün aralıq faylların bəziləri eyni (tam oxşar) ola bildiklərindən  $gk \leq S_k$  münasibəti doğrudur. Həll olunacaq məsələlər kitabxanasının  $t$  bölmələrinin sayı bu kitabxananın həcmi müəyyənləşdirir. Beləliklə də İMS – sistemi vasitəsi ilə həll olunan mürəkkəb məsələlərin hansı miqdarda olduğu müəyyən edilir.

İMS sistemin informasiya bazası saxladığı informasiyaların tipinə görə iki bölməyə ayrılır. Əvvəldə gələn bölmədə İMS – sisteminin generasiya olunması mərhələsində alınan sistem məlumatları qeydə alınır. Baxılan bölmə tətbiqi proqramlar paketlərinin idarə olunmasının dialoqlu informasiya kompleksi tərəfindən yaradılır. İnformasiya bazasının ikinci bölməsi isə İMS – sistemi fəaliyyət göstərən zamanı bu sistemin administratoru vasitəsi ilə yaradılır və lazım gəldikdə dəyişiklik aparılır.

Məlumdur ki,  $D_K$  dilində sistemin informasiya bazasının fraqmentləri ilə düzəlişlərin edilməsi vasitələri yaradılmamışdır. Ona görə də sistemin istifadəçi marşrutlu sistemin informasiya bazasının aparılmasına heç bir təsir göstərmək

iqtidarında deyildir. Bununla da sistemin informasiya bazasının tamlığı təmin edilmiş olur. Amma baxılan anda İMS - sistemin obyektlərinin (başqa sözlə desək, sistemin informasiya bazasının fraqmentlərinin) identifikasiyası metoduna qoyulan tələblər xeyli mürəkkəbləşmiş olur (şəkil 3.2.).

İMS - sistemin istifadəçisi lazım olan məsələsini interaktiv rejimdə xüsusi direktiv-operatorlar vasitəsi ilə əvvəlcədən müəyyən olunmuş hesablama sxeminə görə reallaşdırır: istifadəçi, tətbiqi proqramlar paketinin daxilindəki həll olunan məsələnin nə dərəcədə mürəkkəb addımlarla həll olunduğunu görə bilmir. Sistemin istifadəçi bilavasitə həll olunacaq məsələni identifikasiya edir və bu məsələ ilə işləməyə başlayır. Sistemin istifadəçi məsələlərinə xidmətin təşkili, başlangıç verilənlərin variantlarının aralıq və son nəticələrinin yaddaşda saxlanması və digər qalan bütün funksiyalar İMS – sistemin idarə edici proqramı tərəfindən reallaşdırılır. İdarə edici proqram bu funksiyaları sistemin informasiya bazasının administratorunun vasitəsi ilə realizə edir.



Şəkil 3.2. İnteraktiv marşrutlu sistemin struktur sxemi

### 3.2 İnteraktiv marşrutlu sistemlərin tətbiq oblastının modeli.

İnteraktiv marşrutlu sistemlərin sistemin rezidentini, əksər proqram generasiya edən problem–yönlü sistemlərdə olduğu qaydada, tətbiq oblastı modelinin hər hansı təsviri və bu təsvir qaydasına əsaslanan prosesin hesablama planının qurulması alqoritmindən ibarətdir. İnteraktiv marşrutlu sistemlərdə əməliyyat obyektləri kimi həll olunan məsələlər və alt məsələlər (bir birindən fərqli tətbiqi proqramlar paketlərinin köməyi ilə həll olunan məsələnin müəyyən hissəsi); həll olunacaq məsələlər və altməsələləri şərh etməkdən ötrü istifadə olunan tətbiqi proqramlar paketlərinin parametrləri çoxluğu; tətbiqi proqramlar paketində həll olunan məsələlər və altməsələlərin həlli marşrutunu əmələ gətirən hesablama addımları qəbul edilmişdir.

Tətbiqi proqramlar paketlərinin idarə olunmasının interaktiv indormasiya kompleksində tətbiqi proqramlar paketləri birləşdirilən zaman «Ş-şəbəkə» adlanan şəbəkə yaradılır:

a) Ş - şəbəkəsinin uyğun təpə nöqtələrinə müəyyən bir tətbiqi proqramlar paketinə uyğun hesab edilən tətbiq oblastı modelinin alt modeli adlanan model qarşılaşdırılır;

b) Ş - şəbəkəsinin təpə nöqtələri çoxluğu yüklənmiş tillərlə əlaqələndirilir. Şəbəkənin bütün təpə nöqtələrinin lokal çıxma yarım dərəcəsi və local daxilolma yarım dərəcələri istifadəçinin arzusu ilə müəyyən olunur;

v) Ş – şəbəkəsində tətbiqi proqramlar paketlərinin müəyyən ardıcılıqla qoşulması şəbəkə üzərində marşrut adlanır; bu marşrutun başlaşgıcına və sonuna heç bir məhdudiyyət tətbiq olunmur;

q) Ş – şəbəkələri bir birinə qoşularaq ümumi Ş – şəbəkəsi əmələ gətirirlər. Yaradılan ümumi Ş – şəbəkəsi adi Ş – şəbəkəsinin malik olduğu xassələri özündə saxlayır.

Tətbiqi proqramlar paketlərinin tətbiq oblastının modelləri xüsusi marşrutlu qraflar vasitəsi ilə verilə bilirlər. Marşrutlu qrafi yükləri istiqamətə malik və tilləri yük daşıyan  $G_i (M_i, R_i)$ ,  $i=1, \bar{n}$  psevdografa (təkrarlanan tillər ilə ilgək saxlayan

istiqaətlanmış qraf) deyilir. Belə qrafın  $M_i = \{M_{i1}, \dots, M_{ir}\}$  təpə nöqtələri toplusu tətbiqi proqramlar paketinin daxilində məhəl olunan məsələnin həlli addımlarına,  $R_i = \{R_{i1}, \dots, R_{ik}\}$  tilləri toplusu isə təpə nöqtələri (modullar) arasındakı idarəetmə əlaqələrinə uyğun olmaqla informasiya da daşıyır. Grafın tilləri toplusu iki alt çoxluğa bölünə bilər:  $R_i = R_{i1} \cup R_{i2}$ , burada  $R_i$ - marşrut tilləri toplusu adlanırlar və onlar həll olunan məsələnin özünəməxsus xüsusiyyətləri bütünlükdə şərh edildikdə modulların birləşdirilməsi (kommuntasiyası) imkanına malik olduğunu bildirirlər.  $R_{i2}$  – tilləri toplusu isə baxılan məsələ səhv qoyulduqda və yaxud yaxud da baxılan məsələnin şərti tam verilmədikdə paketin modullarının yenidən birləşdirilə bilmək imkanını aşkarlayırlar. Belə topluya daxil olan tillər «yükündə dəyişiklik edilə bilən» tillər adlandırılırlar. Burada bu tilləri sadəcə olaraq «düzəldilən» tillər adlandıracağıq.

Sistem vasitəsi ilə həll olunan məsələlərin şərti bir sıra əlamətlərlə, o cümlədən paketin dəyişənlərinin qiymətlərinin ətraflə şəkildə verilməsi ilə təqdim olunurlar. “Nizamlayıcı” dəyişənlər dedikdə tətbiqi proqramlar paketlərinin işi prosesində qraf üzərində marşrutların seçilməsinə təsir edə bilən dəyişənlər başa düşülür.

Tətbiqi proqramlar paketlərinin idarə olunmasının dialoqlu informasiya kompleksinin “planlaşdırıcı” adlanan komponenti vasitəsi ilə baxılan məsələnin alqoritmi qurulur və bu zaman həm də həmin məsələnin özünə məxsus xarakteristikaları verilir. Bu alqoritm həll zamanı hesablama addımlarından ibarət olur ki, bu addımlar da marşrutlu qraf üzərində marşrutu təyin etməklə müəyyənləşdirilir. Graf üzərində marşrutların seçilməsi mürəkkəb prosesdir və addım addım reallaşdırılır: hər hansı bir addım yerinə yetirilir; həmin addım reallaşdırıldıqdan sonra qrafın uyğun təpə nöqtəsindən çıxan və elə bu addımda hesablanan “nizamlayıcı” dəyişənlərin qiymətlərində yükünün məntiqi şərtinin qiyməti həqiqi olan til müəyyən edilir.

Bütün hallarda marşrutlu qrafın üzərində istənilən marşrut həmin qrafın kök təpə nöqtəsindən başlayaraq onun son təpə nöqtəsində sona yetir. Belə qəbul olunub ki, sistemdə hər bir marşrutlu qrafının təkrarlanmayan son təpə nöqtəsi

mövcuddür. Həll olunacaq məsələ tam və düzgün qoyulubsa, onda sistemin planlaşdırıcı komponentinin yaratdığı marşrut ancaq marşrut tillərindən ibarət olur. Lakin həll ediləcək məsələnin şərti səhv şərh olunubsa və yaxud da tam şəkildə qoyulmayıbsa (aydın məsələdir ki, bu halda hesablama aparmaq mümkün deyildir), onda istifadəçiyə bu haqda məlumat verilir və həmin istifadəçi ya məsələnin şərtini təşkil edən müəyyən dəyişənlərinin qiymətlərini digərləri ilə əvəzləyir, yaxud həmin dəyişənlər əvvəlcədən nəzərə alınmayıbsa, onların qiymətlərini sistemə ötürür. Mümkündür ki, belə olan halda hesablamanın istiqaməti təmamilə dəyişdirilsin və bir qrup dəyişənlərin qiymətləri təzədən hesablanmalı olsun. Bəzi hallarda isə alt məsələ dəyişə bilər, yaxud da hesablama istiqamətinin dəyişməsi yükündə dəyişiklik edilən tillərin yükünü nəzərə almaq şərti ilə reallaşdırıla bilər.

Sistemin fəaliyyəti dövründə qrafın üzərində hesablamanın marşrutunun dəyişməsi halını zəruri edən bir neçə səbəb ola bilər. Odur ki, sistemin planlaşdırıcı komponenti bu səbəblərin hər birini nəzərdə tutmalıdır. Ümumiyyətlə götürsək, prosesin hesablama addımı hesab olunan nöqtədən, yəni marşrutlu qrafın hər hansı təpə nöqtəsindən hər biri müəyyən yüklə yüklənmiş bir neçə yükündə dəyişiklik edilə bilən til marşrutlu qrafın digər təpə nöqtələri ilə birləşə bilər. Yükündə dəyişiklik edilə bilən tilin yükünə aşağıda göstərilən elementlər daxil edilə bilər:

$Z$  tənzimləyici dəyişənlər toplusunun köməyi ilə yaradılan  $P_{ij}(Z)$  məntiqi ifadəsi olan predikat; əvvəlcədən qiymətləri məlum olan dəyişənlərin siyahısı (toplusu); baxılan sistemin özünün durumunu əks etdirən dəyişənlər. Yükündə düzəlişlər edilə bilən tillərdən fərqli olaraq, marşrutlu tilinin yükü sadəcə  $P_{ij}(Z)$  predikat-məntiqi ifadəsindən ibarətdir (bu məntiqi ifadə eyniliklə həqiqi olduqdayazılmaya bilər, yəni yükün üzərində göstərilməyə bilər).

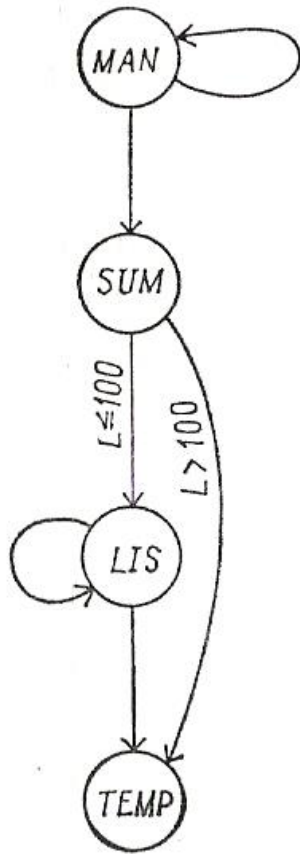
İnteraktiv marşrutlu sistemin inteqrasiya edilmiş tətbiq oblastının modeli onun qomeomorf obrazı olan  $\mathcal{S}$ - şəbəkə əsasında əmələ gətirilir.  $\mathcal{S}$ - şəbəkədən inteqrasiya edilmiş tətbiq oblastının modelinə keçərkən, tətbiqi proqramlar paketlərinin tətbiq oblastının modelinə uyğun gələn təpələri onların marşrutlu

qrafları ilə əvəz dəyişdirilir, qrafa daxil olan bütün tillər kök təpə nöqtəsinə, qrafdan çıxan tillər isə son təpə nöqtəsinə əlavə olunur.

Ən yuxarı məntiqi səviyyədə Ş- şəbəkə ilə marşrutlu qraf arasında yeganə fərq mövcuddur: marşrutlu qrafın bir kökü təpə nöqtəsi və vahid son təpə nöqtəsi olur, amma Ş- şəbəkənin kökləri və son təpə nöqtələri toplusu onun təpə nöqtələri toplusu ilə təmamilə üst-üstə gəlir. Bunun əsas səbəbi sadəcə olaraq interaktiv marşrutlu sistemlərdə baxılan məsələlərin “paket üstü” və “paket daxili” səviyyələrində həllinin modul prinsipindəki ayrılımlarındadır. Sistemin fəaliyyəti prosesində, yəni texnoloji səviyyədə hesablama prosesinin planı tərtib edilərkən Ş – şəbəkəyə fərqləndirici ilkin addım (təpə nöqtəsi) əlavə edilir və bu təpə nöqtəsi bir sıra sistem funksiyalarını yerinə yetirməklə bərabər Ş- şəbəkənin yerdə qalan bütün təpə nöqtələri ilə əlaqələndirilir. Belə vəziyyət Ş-şəbəkə və marşrutlu qraf üzərində planın tərtib olunmasının eyni formada aparılmasına imkan yaradır.

Kompleksin tətbiq oblastı haqqında məlumatların eyni növ, daha dəqiq desək, yeganə şəkilli formalizmin təsviri, interaktiv marşrutlu sistemin ayrı ayrı tətbiqi proqramlar paketlərinin inteqrasiyasından alındığı və yeganə inteqrasiya edilmiş tətbiq oblastına malik olduğununəzərə alaraq, həmin paketləri əlaqələndirməyə şərait əldə olunur. Nəhayət, baxılan məsələnin həllinin planının tərtib olunması həm paketin tətbiq oblastının modelinə görə, həm də baxılan böyük məsələni müəyyən kiçik məsələlərə ayırmaqlıyolu ilə müəyyən sayda paketin tətbiq oblastının modelinə görə aparıla bilər (paket daxili və paket üstü səviyyələrdə hesablama planının qurulması).

MAN1, LİST1, SUM1, TEMP1 tətbiqi proqram paketlərinin birləşdirilməsi variantlarını reallaşdıran Ş – şəbəkələr şəkil 3.3 a və 3.3 b – də verilmişdir.



Şəkil 3.3 a.



Şəkil 3.3 b.

Tələb olunur ki, tətbiq oblastının modeli hesablamaların planlaşdırılması prosesinin determinizm xassəsini ödəməlidir, yəni  $\mathcal{S}$  – şəbəkənin yaxud marşrutlu qrafın bütün mövcud olan təpə nöqtələrindən marşrutun davamının aparılmasının birqiymətliliyi təmin olunmalıdır. Sistemdə həlli tələb olunan məsələlər toplusundan asılı olaraq, marşrutun birqiymətli seçilməsi problemini digər başqa yollarla da həll etmək mümkündür. Marşrutlu qrafın üzərinə dəqiq məhdudlaşdırıcı şərtlər qoymaqla və həll olunacaq məsələlər qrupunu daraltmaqla (kiçiltməklə) marşrutun birqiymətli seçilməsini əldə etmək mümkündür. Ancaq belə olan zaman sistemin planlaşdırıcı komponentinin tətbiq etdiyi alqoritmin intellektuallığı xeyli azalır. Nəzərə alsaq ki, sistemin fəaliyyəti zamanə planlaşdırıcı komponentinin intellekt səviyyəsinin qaldırılması, həll variantlarının sayını xeyli artırmaqla seçim variantlarının sayını daha da artırmış olur, bu isə interaktiv marşrutlu sistemin fəaliyyətgöstəricilərini xeyli yüksəltmiş olmaqla sistemin



imkanlarını da artırmış olur. Sistemə effektivliyin qaldırılması probleminin müxtəlif həlli yolları mövcuddur. Belə həll yollarından biri də həlli tələb olunan bir qrup məsələlər sinfini ayırmaq, onların marşrutlu qrafları çoxluğuna minimal məhdudiyyət şərtləri qoymaq və qeyd olunmuş siniflər üçün uyğun planlaşdırıcı komponentləri yaratmaqdır. Marşrutlu qraflarda marşrutun davam etdirilməsinin birqiymətli təyin olunması tilin üzərindəki yükün vasitəsilə təmin edilir, yəni eyni bir təpə nöqtəsindən çıxan tillərin yükünə (nizamlayıcı dəyişənlərin qiymətləri və İMS– sistemin vəziyyətinin şərhinə) görə növbəti təpəyə keçmək üçün tili birqiymətli seçmək mümkündür.

Dəyişənlərinin ixtiyari mümkün olan qiymətlərində daşdıqları yüklərinin qiyməti eyni zamanda həqiqi olmayan tilləri qoşalaşan tillər adlandırmaq qəbul olunmuşdur. Məsələlərin interaktiv rejimdə həlli tillərin cüt-cüt qoşalaşanlıq şərtini götürməyə imkan yaradır. Əgər bir neçə tilin yükünü təşkil edən şərtlər eyni zamanda həqiqi olarsa, onda sistemin istifadəçi bəzi dəyişənlərin qiymətlərini dəqiqləşdirə bilər və yaxud da digər addıma keçməkdən ötrü bir başa tili təyin edə bilər.

Baxılan iki tilin qoşalaşan olmasını aşağıdakı kimi yoxlayırlar:

1. Əgər tillərdən biri marşrut tildirsə, onda

$$\neg \exists X(P_{ij}(Z) \& P_{ik}(Z))$$

məntiqi ifadəsinin həqiqiliyi yoxlanılır, əgər məntiqi ifadə həqiqidirsə, onda tillər qoşalaşan hesab olunurlar.

2. Əgər hər iki til yükündə dəyişiklik edilə bilən tildirsə, onda

a)  $P_i$  və  $P_j$  məntiqi ifadələrinin eyni olduğu;

b)  $S_{ij}$  və  $S_{ik}$  dəyişənlər ardıcılığının uyğun gəlib gəlmədiyi;

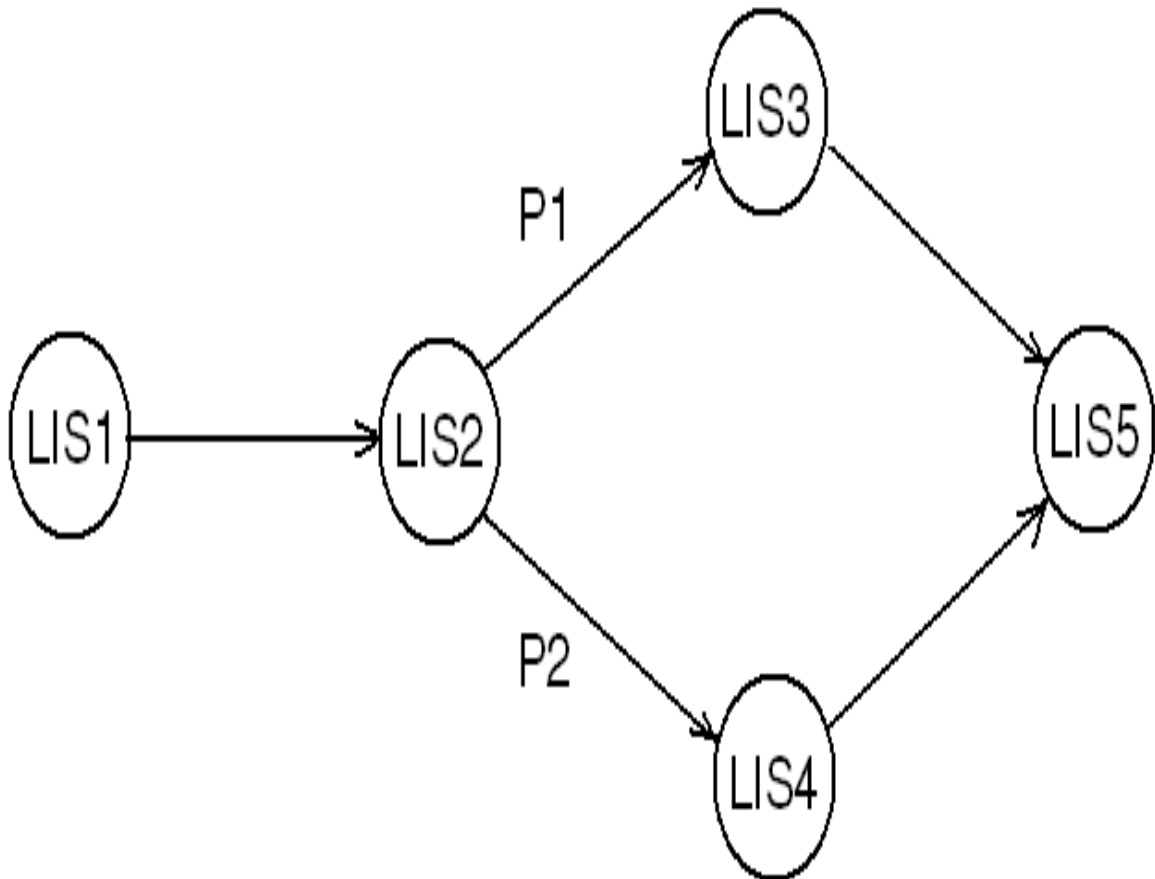
v)  $\neg \exists Z(P_{ij}(Z) \& P_{ik}(Z))$  məntiqi ifadəsinin həqiqi olduğu;

yoxlanılır.

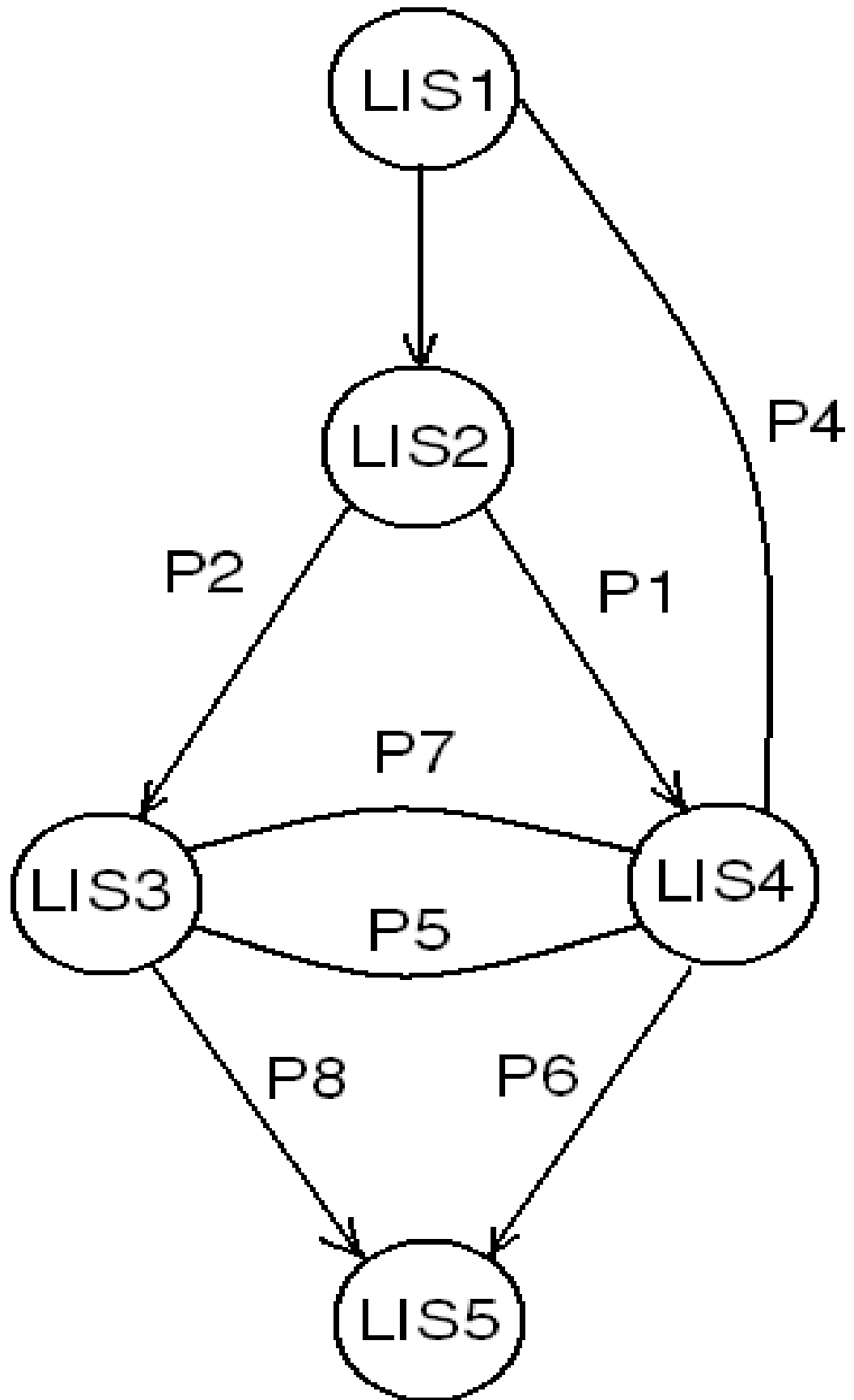
Əgər bu şərtlərdən heç olmasa biri ödənilmirsə, onda tillər qoşalaşan tillərdirlər.

$P_i$  məntiqi ifadəsi  $P_i = K_i$  şəklində göstərilir, burada  $K_i$  yükündə dəyişiklik edilə bilən tildən keçərkən İMS - sistemin displeyin ekranına verdiyi diaqnostik məlumatın mətninin sistemdəki identifikatoru hesab olunur.

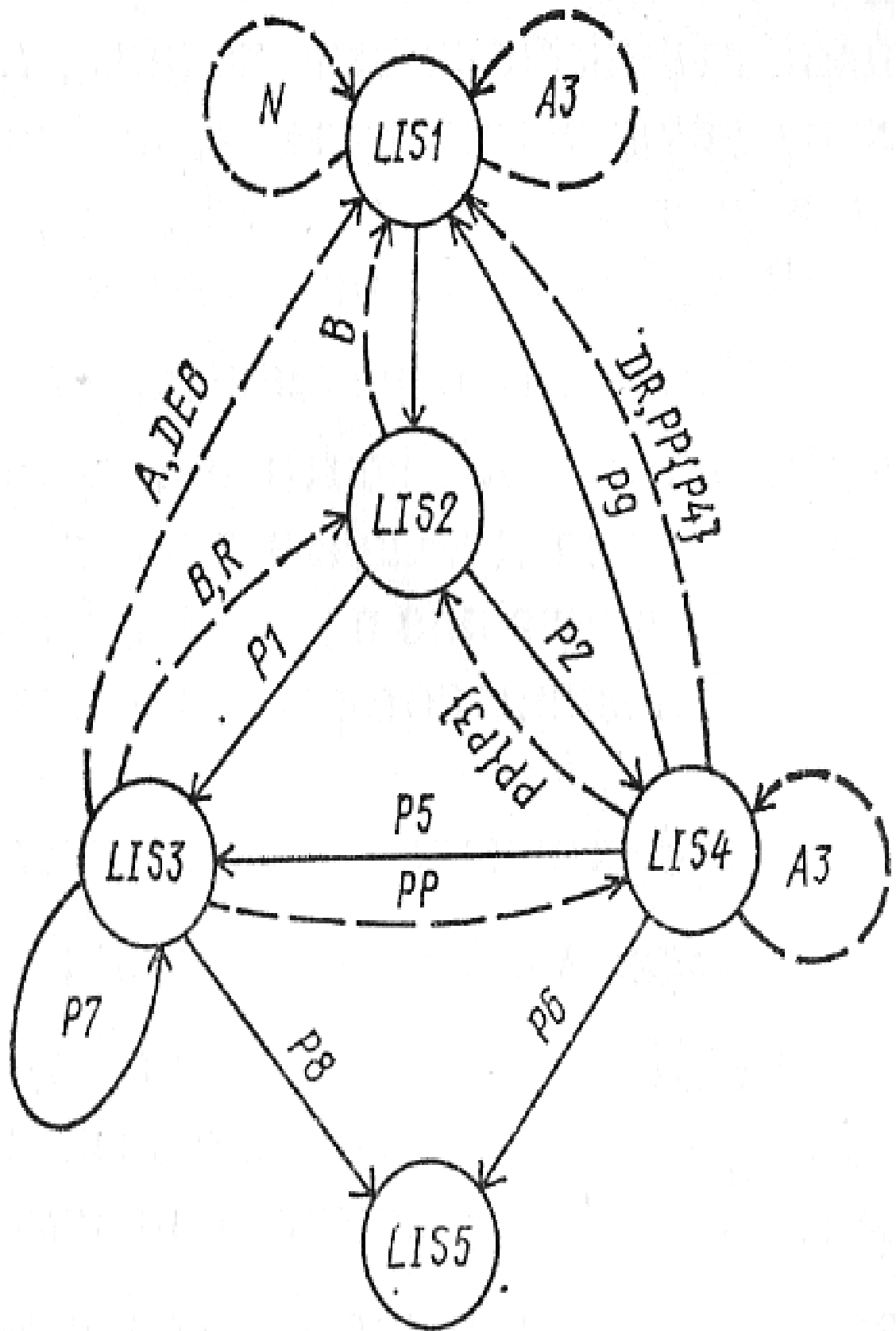
Şəkil 3.4, 3.5, və 3.6 – da marşrutlu qraflarına misallar göstərilmişdir.



Şəkil 3.4.



Şekil 3.5.



Şekil 3.6.

Yükündə dəyişiklik edilə bilən tillər qırıq xətlərlə, marşrutlu tilləri isə adi xətlərlə çəkilməmişdir. Aşağıda tillərin yükündəki məntiqi ifadələrin siyahısı göstərilmişdir:

$$P1: B > CV((A1+A3/2)/KK-C*(A4/3-A6)) \leq PK+PP$$

$$P2: P2 = P1$$

$$P3: DEB > B \& DR \in \{0, 1, 2, 7\};$$

$$P4: SM \in \{0, 1, 2, 7\};$$

$$P5: A1+ A2 \geq C;$$

$$P6: P6 = P5$$

$$P7: DR > 0;$$

$$P8: P8 = P7;$$

$$P9: DR > 0 \& A1+ C < T$$

### 3.3. Marşrut qrafları cəbri.

Məlumdur ki, marşrutlu qraflar tillərin qoşalaşma xassəsinə nəzərən additivlik xassəsinə malik olurlar, yəni qrafa yeni til əlavə edilən zaman da qoşalaşan olaraq qalırlar. Bu xassə imkan verir ki, marşrut qrafları layihələndirilərkən, yeni tillərin əlavə olunması prosesində yalnız qoşulan tili saxlayan cütlük-lərin uyuşan olduğu yoxlanılsın.

Tətbiqi proqramlar paketlərinin idarə olunmasının dialoqlu informasiya kompleksinin əsas komponentlərindən biri də  $\langle G, S \rangle$  marşrut qrafları cəbridir. Burada G- marşrut qrafları çoxluğu, S- siqnaturası isə F0, F1, F2, F3 əməllərindən ibarətdir.

$F0(G,X)$  əməli G qrafına X marşrut tilinin əlavə olunması əməli adlanır. X tili  $X = \langle V_i, V_j, Z \rangle$  şəklində verilir. Burada  $V_i$  – tilin çıxdığı təpə nöqtəsinin adı,  $V_j$  – tilin daxil olduğu təpə nöqtəsinin adı, Z – isə tilin yükü-dür. Marşrut tili əlavə olunan zaman qrafda  $V_i$  və  $V_j$  adlı təpə nöqtələrinin olduğu və əlavə olunan tilin  $V_i$  təpə nöqtəsindən çıxan bütün tillərlə yükünə görə uyuşan olduğu yoxlanılır.

Fraqmentin mətnində bu əməl F0, F1, F2, F3 əməlləri içərisində çox tez-tez təkrarlanır və o aşağıdakı formada verilir:

F0:<tilinin çıxdığı təpə nöqtəsinin adı >#< tilin daxil olduğu təpə nöqtəsinin adı >[< tilin yükü - 1].

F1(G, X) əməli G qrafına X düzəldilən tilin əlavə olunması əməli adlanır. X tili burada da  $X = \langle V_i, V_j, Z \rangle$  üçlüyü ilə verilir. Burada  $V_i$  – tilin çıxdığı təpə nöqtəsinin adı,  $V_j$  – tilin daxil olduğu təpə nöqtəsinin adı,  $Z$  – tilin yüküdür. Til əlavə olunarkən qrafda  $V_i, V_j$  adlı təpə nöqtələrinin olduğu və əlavə olunan tilin  $V_i$  təpə nöqtəsindən çıxan bütün tillərlə yükünə görə uyuşan olduğu yoxlanılır. Bu əməl aşağıdakı formatda verilir:

F1:< tilinin çıxdığı təpə nöqtəsinin adı >#< tilin daxil olduğu təpə nöqtəsinin adı >< tilin yükü-2>.

F2 (G,X) əməli G qrafında X tilinin ayrılması əməli adlanır. Bu əməl aşağıdakı formatda verilir:

F2: < tilin çıxdığı təpə nöqtəsinin adı >#< tilin daxil olduğu təpə nöqtəsinin adı >[<tilin yükü>].

Qeyd edək ki, F0, F1, F2 əməllərində tilin yükü yalnız o zaman göstərilir ki,  $V_i$  və  $V_j$  təpə nöqtələri bir neçə tilə birləşdirilmiş olsun.

F3(G,X,Y) əməli G qrafının X təpə nöqtəsinin Y əltqrafı ilə əvəz olunması əməli adlanır. Bu əməl yerinə yetirilən zaman G qrafına X adlı təpə nöqtəsinin olduğu, alt qrafın bir təpə nöqtəsindən çıxan tillərin yüklərinə görə uyuhən olduğu (yəni Y əltqrafın marşrut qrafı olduğu) və G və Y qraflarında adları üst-üstə düşən təpə nöqtələrinin olmadığı yoxlanılır. Bu əməl aşağıdakı formatda verilir:

F3: < əvəz olunan təpə nöqtəsinin adı > @ Z < alt qrafın tillərinin şərh ardıcılığı > Z @

Əgər əltqraf əvvəlcədən şərh olunubsa və kompüterin yaddaşında fayl şəklində saxlanılıbsa, onda F3 əməli aşağıdakı şəkildə də verilə bilər:

F3: < əvəz olunan təpə nöqtəsinin adı >(<əltqrafın olduğu əltqrafın adı>).

Əməllər marşrut qrafları üzərində müəyyən olunduğundan və tillərin uyuşanlığı xassəsini pozmadığından aşağıdakı teorem doğrudur.

Teorem: Marşrut qrafları çoxluğu tillərin əlavə olunması, silinməsi, əvəz olunması əməllərinə nəzərən qapalı çoxluqdur.

Beləliklə sadalanan bu əməlləri tətbiq sahəsinin modelini qurarkən onun ayrı-ayrı hissələrinə tətbiq etməklə biz bu hissələrin əsas xassəsini, daha dəqiq desək, marşrutun birqiymətli seçilməsi xassəsini pozmururuq.

Dialoqlu marşrut sistemlərin generasiyası zamanı ortaya çıxan əsas problemlərdən biri də tətbiq sahəsi modelinin yaradılması təşkili problemdir. Məlum olduğu kimi tətbiq sahəsinin modeli sistemin informasiya bazasının sxemini verir. İnformasiya bazası interpretasiya dövrünün sistem informasiyalarını, həll olunan məsələlərin ilkin, aralıq və nəticə verilənlərini saxlayır. İnformasiya bazasının tamlığı hər bir verilənin bazada yeganə olması yolu ilə təmin olunur.

### **§3.4. Tətbiq oblastı modelinin layihələndirilməsinin modul prinsipi.**

Dialoqlu marşrut sistemlərin tətbiq sahəsinin modeli (yaxud informasiya bazasının sxemi) aşağıdakı komponentlərdən ibarətdir:

a) marşrut sistemlərdə  $n$  tətbiqi proqramlar paketi inteqrasiya olunan zaman əlaqələrin növünü reqlamentləşdirən  $\mathcal{S}$  – şəbəkə;

b)  $n$  tətbiqi proqramlar paketinin parametrlərinin tipii və bu parametrlərin paketlərin hesablama addımlarında saxlanan verilənlər topluslarına daxil olması üsulu, bundan başqa məsələlər və altməsələlərin hər birini hesablama addımlarının aralıq və son nəticə informasiyaları haqqında məlumatları saxlayan  $E_1, E_2, \dots, E_n$  adlı  $E$  – şəbəkələr;

v)  $n$  tətbiqi proqramlar paketində məsələlərin bütün mümkün həlli marşrutları haqqında məlumatları saxlayan  $P_1, \dots, P_n$  pleksləri;

Tətbiqi proqramlar paketlərinin idarə olunmasının dialoqlu informasiya kompleksində dialoqlu – marşrut sistemlər iki mərhələdə yaradılır. Birinci mərhələdə  $G_1, \dots, G_n$  qrafları generasiya olunur və  $E_1, \dots, E_n$  şəbəkələrində əlaqələrin strukturu müəyyənləşdirilir. İkinci mərhələdə  $\mathcal{S}$  - şəbəkə haqqında

toplanmış informasiya əsasında qurulan xüsusi ssenari üzrə dialoqlu marşrut sistemin yaradıcısı ilə  $E_1, \dots, E_n$  şəbəkələrini doldurmaq məqsədilə sorğu aparılır.

Dialoqlu marşrut sistemin generasiyasının birinci mərhələsində tətbiq sahəsi modelinin layihələndirilməsi proseduruna baxaq. Tətbiq sahəsi modelinin xeyli mürəkkəb olduğunu nəzərə alaraq prosedura tətbiq sahəsi modelinin addımlarla ardıcıl olaraq yaradılması, dəqiqləşdirilməsi və seçilməsi prosesi həyata keçirilir.

Dialoqlu marşrut sistemin yaradıcısı tətbiq sahəsi modelinin addımlarla, hər bir addımda müəyyən hissə informasiya – bir və ya bir neçə tətbiq sahəsi modelinin fraqmentini daxil etməklə, layihələndirmək imkanına malikdir. Belə ki, istifadəçi eyni fraqment daxil edərkən ondan əvvəlki fraqmentin adını da göstərir və bununla da o fraqmentlərin əlaqəsi ardıcılığını birqiymətli müəyyən edir. Hər bir fraqmentin özündən əvvəlki bir fraqmenti, özündən sonra isə bir neçə fraqmenti vardır.

Marşrut sisteminə yeni tətbiqi proqramlar paketi əlavə olunan zaman onun yaradıcısı paralel olaraq paketin tətbiq sahəsi modelinin bir neçə versiyasını (variantını) yarada və analiz edə bilər (Tətbiqi proqramlar paketlərinin idarə olunmasının dialoqlu informasiya kompleksinin vasitələri ilə testləşdirilə bilər).

Ümumiyyətlə, fraqmentlərin qarşılıqlı əlaqəsinin strukturu, kökkü başlanğıc fraqment olmaqla  $K$  – ağac əmələ gətirir. Bu ağacın əsas xüsusiyyəti ondan ibarətdir ki, başlanğıc fraqmentdən başqa bütün fraqmentlərin hər biri, kökdən yarpağa qədər, özündən əvvəlki fraqmentə əla-vələr edir və yaxud onu dəqiqləşdirir.

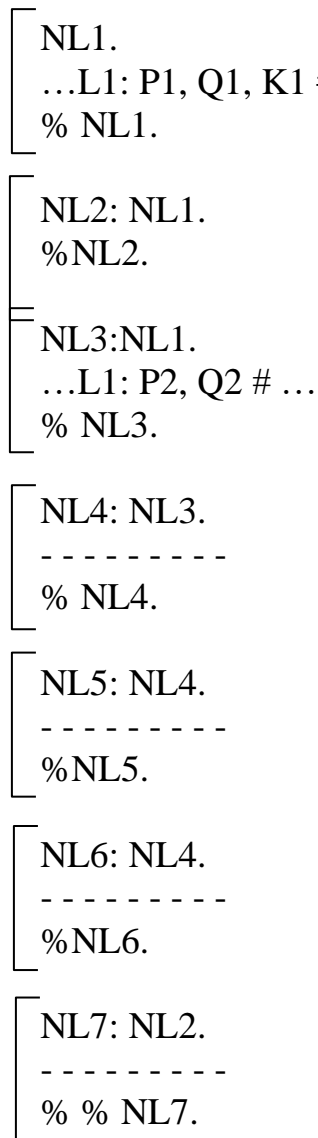
Tətbiq sahəsi modelinin versiyası dedikdə  $K$  – ağacın budağında yerləşən, başqa sbzlə, kökdən təpəyə qədər olan yolda yerləşən fraqmentlər toplusu başa düşülür. Təpə nqtəsini göstərməklə, versiya birqiy-mətli təyin olunur, çünki kökdən təpəyə yeganə yol vardır. Uyğun olaraq, demək olar ki,  $K$  – ağacın versiyalarının sayı bu ağacı versiyalarının sayı qədərdir.

Fraqmentin mətni dedikdə müəyyən ada malik  $F_0, F_1, F_2, F_3$  əməlləri toplusu başa düşülür. Fraqmentin başlığı aşağıdakı şəkildə yazılır:

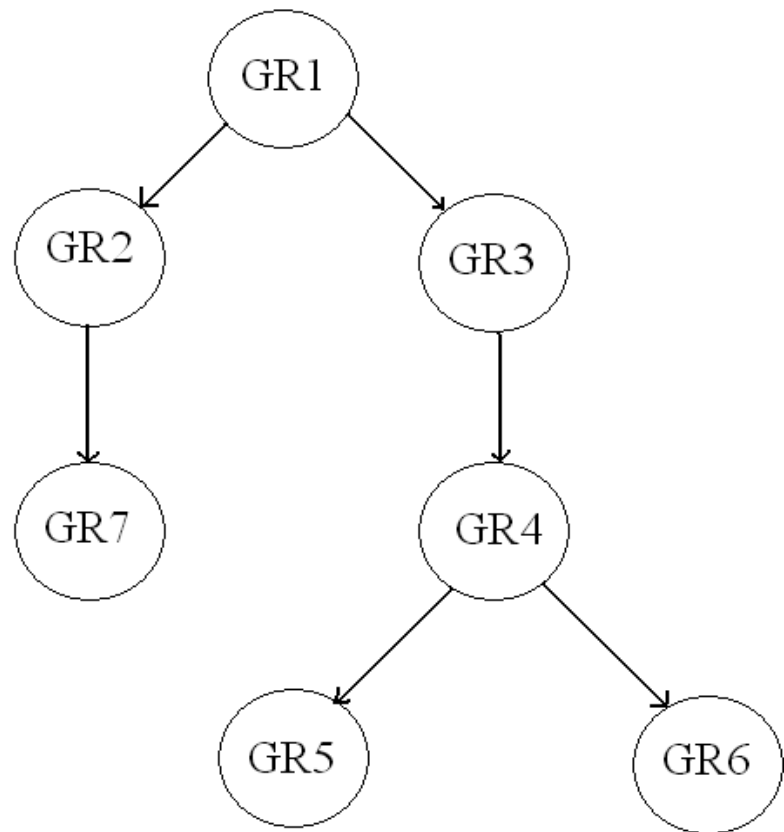
<fraqmentin adı>:[<əvvəlki fraqmentin adı>].



Dialoqlu marşrut sistemin yaradıcısı ixtiyari budaqda yerləşən tətbiq sahəsi modelinin versiyalarını müəyyən bir əlamətə yaxud əlamətlərə görə müqayisə edib, son nəticə kimi onlardan ixtiyarisini seçə bilər. Tətbiq sahəsinin modeli ilə işi asanlaşdırmaq üçün növbəti addım layihələndirildikdən sonra (tətbiq sahəsi modelinin versiyasının növbəti fraqmenti daxil edildikdən sonra) tətbiq sahəsi modelinin bütün versiyalarının fraqmentlərinin mətnləri çap olunur. Bundan başqa



Şəkil 3.6.



Şəkil 3.7.

tətbiq sahəsi modelinin versiyası formatlaşdıqdan sonra, yaxud layihələndirmənin növbəti addımı başa çatdıqdan sonra istifadəçi sonuncu fraqmentin axırında “P%<fraq-mentin adı>” konstruksiyasından istifadə etməklə paketin alınmış G qrafını qrafik formada çap edə bilər. Şəkil 3.6 – da tətbiq sahəst modelinin üç

versiyasının yeddi fraqmentinin mətnləri, şəkil 3.7-də isə bu versiyaları əks etdirən K – ağac göstərilmişdir.

NL5 və NL6 addımlarında tətbiq sahəsi modelinin uyğun versiyalarının qrafları alınmışdır, lakin onlar hansı səbəbdənsə istifadəçini Qane etməmişdir. Ona görə də axırncı qəbul olunan versiya NL1, NL2 və NL7 fraqmentlərindən ibarət versiya olmuşdur. Bu NL7 fraqmentinin «% % <fraqmentin adı>» konstruksiyası ilə təsdiq olunmuşdur. Tətbiq sahəsinin modeli layihələndirilərkən, K – ağacda müəyyən bir budaqdan altağacın ayrılıb başqa bir budaqda hər hansı bir təpəyə birləşdirilməsi ilə istifadəçini Qane edən versiya alınmışdır. Bunun üçün sadəcə olaraq fraqmentin başlığında əvvəlki fraqmentin adının əvəzinə ayrılan altağacın kök təpə nöqtəsinin adını göstərmək kifayətdir. Məsələn, şəkil 3.6-dakı fraqmentlər toplusuna

$$\left[ \begin{array}{l} \text{NL2: NL3.} \\ \text{-----} \\ \% \text{ NL2.} \end{array} \right.$$

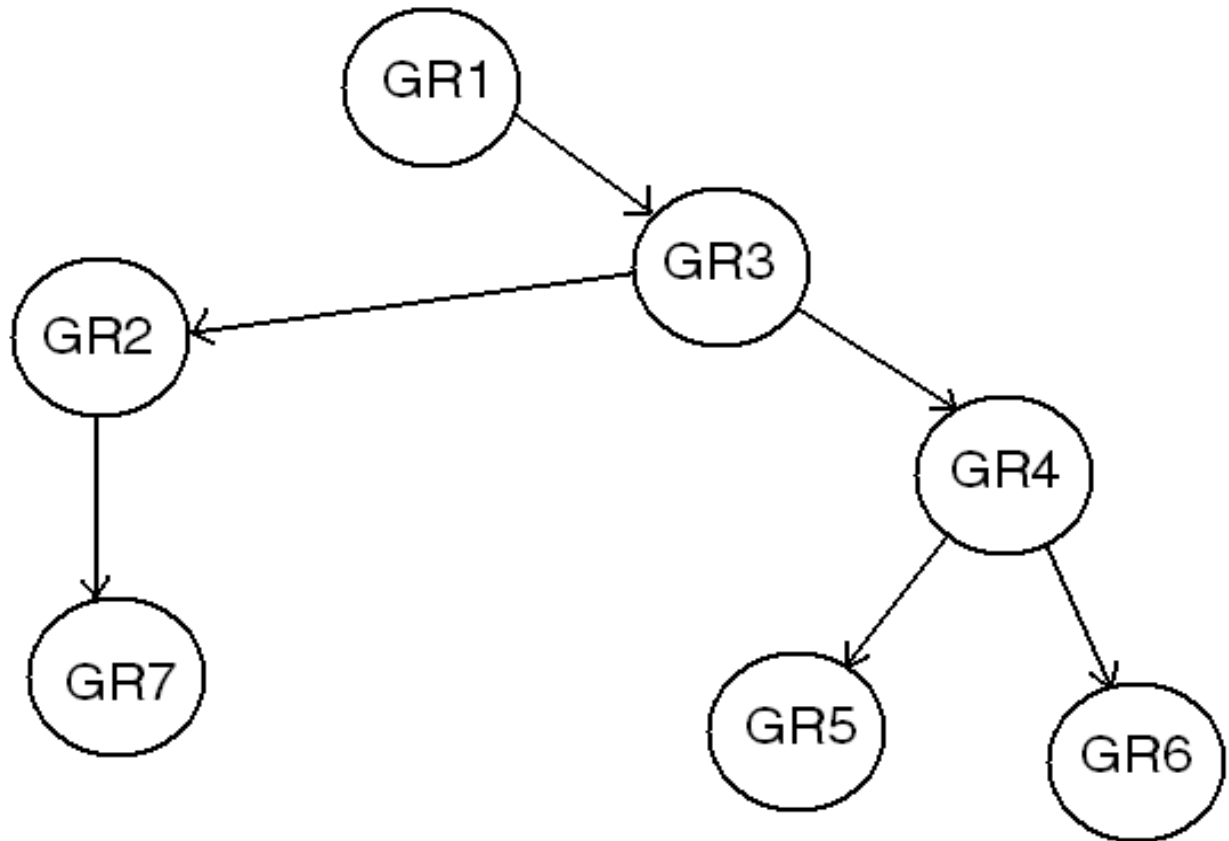
fraqmentinin əlavə etsək, onda tətbiq sahəsi modelinin versiyalarının K – ağacı şəklini dəyişəcək (şəkil 3.8) və son nəticə kimi qəbul olunan NL1, NL3, NL2, NL7 fraqmentlərindən ibarət versiya olacaqdır.

Marşrut qraflarının tillərinin yükündə bəzən eyni konstruksiyalara, məsələn, ümumi qrupp parametrlərə, yaxud bir neçə dəfə təkrarlanan predikatlara, yaxud da digər til yüklərinə rast gəlmək olar. İşni yüngülləşdirmək məqsədi ilə belə konstruksiyalar qeyd olunmuş xüsusi qruplar şəklində təşkil olunurlar.

Məsələn parametrlər qrupunun əvvəli “<nişan>” , sonu isə “ # ” konstruksiyaları ilə qeyd olunur. Ola bilər ki, bir qrupp parametr başqa bir qrupun tərkibində təkrarlansın. Tətbiq sahəsi modelinin fraqmentlərində təkrarlanan konstruksiyaların əvəzində həmin qrupun adı – nişanı istifadə olunur.

Parametrlər qrupunu adlandıran nişanın adından elə həmin fraqmentdə onun şərhindən əvvəl də istifadə etməyə icazə verilir. Hər hansı bir NLK fraqmentinin

mətnədə qrup (parametrlər qrupu) şərh olunduqdan sonra, budaqda NLK – dan sonra gələn fraqmentlərdə həmin nişandan istifadə etmək olar.



Şəkil 3.8

Bir budaqda iki nişanın şərhinə rast gəldikdə onlar üçün təsir oblastı aşağıdakı kimi müəyyən olunur: budaqda nişanın təsir oblastı onun elan olduğu fraqmentdən, bu elanın təzələndiyi fraqmentə qədərdir. Sonrakı fraqmentlərdə nişanın yeni elan olunmuş denotatı qüvvədə olur, köhnəsi isə yadda saxlanılır. Əgər hər hansı bir fraqmentdə nişanın köhnə denotatından istifadə etmək zərurəti ortaya çıxarsa, onda nişanı göstərər-kən onu n elan olduğu fraqmentin adını da göstərmək lazımdır.

Şəkil 3.6-da verilmiş tətbiq sahəsi modelinin fraqmentlərinin şərhinə baxaq. NL1, NL2, NL7 fraqmentlərindən ibarət budaqda M1 nişanın yeganə denotatı P1, Q1, K1 parametrlər qrupudur. Digər budaqlarda NL3, NL4, NL5 və NL7 fraqmentləri üçün L1 nişanın P2, Q2 parametrlərindən ibarət denotatı həqiqi hesab olunur.

L1 nişanı bu denotatla NL4, NL5 və NL7 fraqmentlərində dəqiqləşdirilmədən istifadə oluna bilər. Əgər bu fraqmentlərdə L1 nişanının əvvəlki denotatından, yəni P1, Q1, K1 parametrlərindən, istifadə etmək zərurəti yaranırsa, onda tələb olunan denotatın təsir oblastı dəqiqləşdirilməklə, NL1.L1 nişanından istifadə etmək lazımdır.

Altağacın kökündə elan olunmuş denotatı onun bütün təpə nöqtələrində istifadə oluna bilər. Əgər alt ağac denotatın elan olunduğu alt ağacın alt ağacı deyilsə, onda onun təpə nöqtələrində bu denotatdan istifadə etmək olmaz.

Qeyd edək ki, tətbiq sahəsi modelinin müəyyən versiyasındakı fraqmentlərin şərhində bir-birinin daxilində olan qruplar iştirak edərsə və sonradan daxiləki denotat dəyişərsə, bu denotatın dəyişdiyi fraqmentdən başlayaraq sonra gələn bütün fraqmentlərdə yeni dəyişmə baş verənə qədər dəyişən denotatıözündə saxlayan denotatdan istifadə olunur.

Məsələn şəkil 3.9-da göstərilən fraqmentlər üçün NL2 fraqmentindən başlayaraq NL1 fraqmentində elan olunmuş L1 nişanının yeni denotatından istifadə olunur.

Nişanın denotatlarının təsir oblastının şərhli blok strukturuna malik Paskal tipli (daha doğrusu Alqol tipli) dillərdə dəyişənlərin təsir oblastının şərhli tələbi ilə üst-üstə düşür, belə ki, K – ağacın alt ağacları bir - birinin daxilindədirlər və iki

```

NL1.
...L1:...L2:...# #.
% NL1.

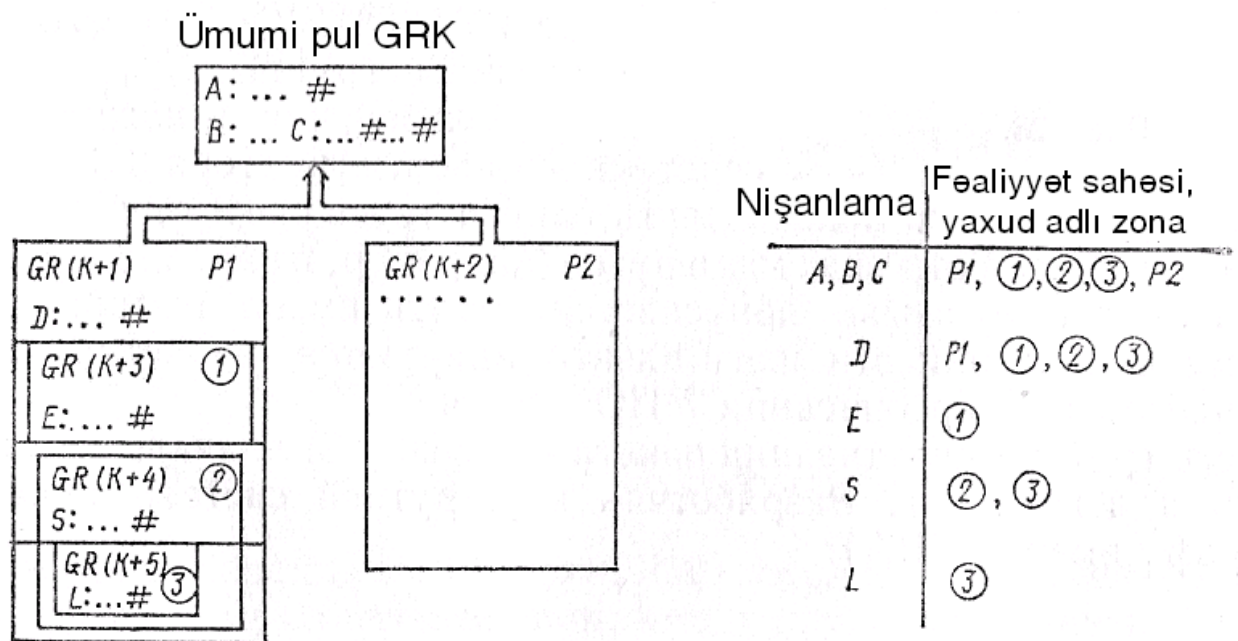
NL2: NL1.
...L2:... #.
% NL2.

NL3: NL2.
...L1.
% NL3.

```

Şəkil 3.9

Alt ağac hissə-hissə kəşişə bilməzlər (yəni ya biri digərinin daxilindədir, yaxud da onların ümumi təpə nöqtələri yoxdur). Alt ağacın kökündə, bu alt ağacda istifadə olunan nişanların şərh olunması tələbi, proqramlaşdırma dillərində blokun əvvəlində dəyişənlərin şərh olunması tələbi ilə üst-üstə düşür. Bu proses şəkil 3.10-da əyani şəkildə təsvir olunmuşdur. Nişanların adları və onların təsir oblastları sağdakı cədvəldə verilmişdir.



Şəkil 3.10

## NƏTİCƏ VƏ TƏKLİFLƏR

İnteraktiv marşrutlu sistemlər tətbiqi proqramlar paketlərinin idarə olunmasının dialoqlu informasiya kompleksinin köməyi ilə tətbiqi məsələlərin kompüterdə həlli texnologiyası mərhələlərini avtomatlaşdırmaqla generasiya olunurlar.

İdarəetmənin informasiya texnologiyası informasiyaların işlənməsinin və istifadə olunmasının çox mühüm olan uçot, analiz və qərarların qəbul edilməsi mərhələlərini reallaşdırmalıdır.

Problem –oriyentasiyalı sistemlər, əsasən dörd istiqamətdə, avtomatlaşdırılmış informasiya sistemləri, informasiya - axtarış sistemləri, tətbiqi proqramlar paketləri, ekspert sistemləri istiqamətlərində inkişaf etdirilirlər.

Avtomatlaşdırılmış informasiya sistemlərinin bu və ya digər sahələrdə istifadə olunması iki üsulla həyata keçirilə bilər. Bu sistemlər ya avtonom şəkildə, heç bir digər sistemin tərkibinə daxil olmadan və yaxud digər bir sistemin tərkibində istifadə olunur.

Tətbiqi proqramlar paketləri praktikada ən çox istifadə olunan proqramm kompleksləridir. Bu komplekslərin yaradılması sahəsində bu gün də gərgin iş gedir. İndii elə bir təşkilat, müəssisə, firma tapılmaz ki, Microsoft Office 2000 paketindən istifadə etməsin.

Ekspert sistemlər vasitəsi ilə həllolunan məsələlər altı əsas sinfə, verilərin şərhı, diaqnostika, nəzarət, proqnozlaşdırma, planlaşdırma, layihələndirmə siniflərinə bölünürlər.

Ekspert sistemlərin yaradılması digər problem – oriyentasiyalı sistemlərə nisbətən «az» avtomatlaşdırılındırlar. bu onların yeni tədqiqat sahəsi olması ilə deyil, bəzi tətbiq sahələrindən olan problemlərin «pis» formallaşan olmasıdır. Çoxillik təcrübə göstərir ki, problem –oriyentasiyalı sistemlərin bu istiqaməti daha perspektivlidir.

Problem –oriyentasiyalı sistemin modeli onun əsas parametrləri hesab olunan tətbiq sahəsinin modeli, verilənlər bazası və onun sxemi, problem vasitələr

kitabxanası və giriş dili parametrlərini çox sadə münasibətlərlə əlaqələndirir. Lakin belə modeli qurmadan problem –oriyentasiyalı sistemin generasiyası məsələsinin həlli mümkün deyildir.

Problem –oriyentasiyalı sistemlərin tətbiq sahəsinin modeli hesablama modelləri, marşrut qrafları, həll cədvəlləri, semantik şəbəkələr, aksiomatik modellərlə şərh olunur. Aparılan tədqiqatlar belə nəticəyə gəlməyə imkan verir ki, hesablama modelləri, aksiomatik modellər və semantik şəbəkələr çevirmələr yolu ilə asanlıqla marşrut qrafına gətirilə bilər.

Problem –oriyentasiyalı sistemlərdə hesablama planı iki metodla reallaşdırılır. Hesablama modelləri, semantik şəbəkələr, aksiomatik modellər tətbiq olunan problem –oriyentasiyalı sistemlərdə statik planlaşdırma, marşrut qrafları və həll cədvəlləri tətbiq olunan sistemlərdə dinamik planlaşdırma həyata keçirilir.

Tətbiq sahəsinin modeli dialoqlu marşrut sistemin informasiya bazasının sxemini verir. İnformasiya bazası interpretasiya dövrünün sistem informasiyalarını, həll olunan məsələlərin ilkin, aralıq və nəticə verilənlərini saxlayır. İnformasiya bazasının tamlığı hər bir verilənin bazada yeganə olması yolu ilə təmin olunur.

Dialoqlu marşrut sistemin inteqrasiya olunmuş tətbiq sahəsinin modeli  $\mathcal{S}$  – şəbəkəyə əsasən qurulur.  $\mathcal{S}$  – şəbəkədən inteqrasiya olunmuş tətbiq sahəsinin modelinə keçərkən paketin tətbiq sahəsi modelinə uyğun təpə nöqtələri onların marşrut qrafları ilə əvəz olunur. Məntiqi səviyyədə  $\mathcal{S}$  – şəbəkədə marşrut qrafları arasında yeganə fərq vardır: marşrut qraflarının bir kökü və yeganə son təpə nöqtəsi vardır, lakin  $\mathcal{S}$  – şəbəkənin kökləri və son təpə nöqtələri çoxluğu onun təpə nöqtələri çoxluğu ilə üst-üstə düşür. Səbəb sadəcə olaraq dialoqlu marşrut sistemlərdə məsələlərin paket üstü və paket daxili səviyyələrdə həllinin modul konsepsiyasındakı fərqdədir.

Dialoqlu marşrut sistemin tətbiq sahəsinin modeli sistemin generasiyasının birinci mərhələsində addımlarla, hər bir addımda müəyyən hissə informasiya – bir və ya bir neçə tətbiq sahəsi modelinin fraqmentini daxil etməklə layihələndirilir.

**İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT**

1. Карминский А. М., Черников Б. В. Применение информационных систем в экономике. Учебное пособие, изд-во Форум Инфра-М, 2012. - 320 стр.
2. Информационные системы в экономике. Под ред. проф. Романова А. Н., Одинцов Б. Е. Учебное пособие, 2-е издание, изд-во Вузовский учебник, 2010, - 410 стр.
3. Терехов А. Н. Технология программирования. Учебное пособие, изд-во Бином, 2011. – 148 стр.
4. Анализ опыта реализации диалоговых систем. /Л.В. Кокарева, И.И. Малашинин, О.Л. Перевозчикова, Е.Л. Ющенко // Упр. системы и машины. —1987.—№4. -с. 63-69.
5. Аннотированный библиографический указатель по диалоговым системам. - Пушкино: НИВЦ АН СССР, 1986.—220 с.
6. Бабаев И.О., Лавров С.С. Развитие автоматизированной системы решения задач СПОРА // Пакеты прикладных программ. Инструментальные системы. —М.: Наука, 2007.—с. 5-18.
7. Диалоговая информационная система управления пакетами прикладных программ ДИСУППП / О.Л. Перевозчикова, И.В. Криштопа, Г,Э. Микаилов, М.Н. Мусаев и др.; Ин-т кибернетики им. В.М.Глушкова АН УССР—Киев, 1985 —290с. —деп. В Гос ФАП СССР. 24.12.85, №50860000937.
8. Ещенко В.Г. О реализации на языке высокого уровня математического обеспечения пакета программ РАЗМЕЩЕНИЕ.— Программирование, 1983, № 2, с.12—16.
9. Каталог диалоговых систем: материалы по математическому обеспечению—Киев: Ин-т кибернетики АН УССР, 2006, —232 с.



10. Кахро М.М., Калья Л.П., Тыугу Э.Х. Инструментальная система программирования ЕС ЭВМ (ПРИЗ).-М.: Финансы и статистика, 1988.- 181 с.
11. Компьютерные технологии обработки информации: Учебное пособие /С.В. Назаров, В.И. Першков, В.А. Гафинцев и др.; Пол. ред. С.В. Назарова.—М.: Финансы и статистика, 1995—248 с.
12. Крижановский В.В., Мусаев М.Н., Первозчикова О.Л. Создание схем вычислений маршрутных систем.—Киев, 1984.—30с.— (Препринт/АН УССР, Институт Кибернетики; 84-27).
13. Мусаев М.Н. Об одной проблеме проектирования проблемно-ориентированных систем. Труды конф. «Новые информационные технологии и проблемы прикладной математики», Баку, 1997.
14. Мусаев М.Н. Об одном способе построения моделей предметных областей.—Изв. АН Азербайджана, сер. Физ.-техн. и матем. Наук, XVI том, № 5-6, 1995.
15. Пакеты программ офисного назначения: Учеб. Пособие /С. В. Назаров. Л.П. Смольников, В.А. Тафинцев и др.;Под, ред. проф. С.В. Назарова.-М.: Финансы и статистика, 2007—320 с.
16. Первозчикова О.Л. Модели общения при решении задач на ЭВМ. //Упр. системы и машины.—1987.—№5.—с.61-68.
17. Первозчикова О.Л., Ющенко Е.Л. Диалоговые системы.—Киев: Наук. думка, 1990.—184 с.
19. Первозчикова О.Л., Ющенко Е.Л. Системы диалогового решения задач на ЭВМ—Киев: Наук. думка, 1986.—264 с.
20. Чаплинскас А.А., Матулис. В.А. Система «Вильнюс»: концепции, структура и технологии ее использования. — Вильнюс: Ин-т математики и кибернетики АН Лит. ССР, 1981 .—146 с.
21. Петрушин В.А. Экспертно-обучающие системы.-Киев: Наук думка, 1992.-с.196.

22. Петрушин В.А., Третьяк В.А., Лисюк Т.А. Инструментальные средства для создания экспертно- обучающих систем. / Интеллектуальные системы в задачах проектирования, планирования и управления в условиях неполноты информации. Матер. всесоюз. науч.- техн. совещ. (январь 1990 г.)-Казань: НПО "Волга", 1990.-с.112-115.
23. Ездаков А. Л. Функциональное и логическое программирование. Учебное пособие. Изд-во Бином, 2011. – 119 стр.
24. Фуфаев Э. В., Фуфаева Л. И. Пакеты прикладных программ. Учебное пособие, изд-во Academia, 2012. – 352 стр.