

## ნინო თოფურია

### განაწილებული ენერგეტიკული ოფის-სისტემების საქმიანი პროცესების კომპიუტერული წარმოება

სპეციალობა 05.13.16 – გამოთვლითი ტექნიკის, მათემატიკური მოდელირებისა და  
მათემატიკური მეთოდების გამოყენება სამეცნიერო კვლევებში

ტექნიკის მეცნიერებათა კანდიდატის სამეცნიერო  
ხარისხის მოსაპოვებლად წარმოდგენილი დისერტაციის

### დისერტაცია

სამეცნიერო ხელმძღვანელი:

ტექნიკის მეცნიერებათა დოქტორი,  
პროფესორი გ. სურგულაძე

შესავალი -----	3
<b>თავი I. განაწილებული ენერგეტიკული ოფის-სისტემების მართვის ამოცანების და მათი მოდელირების ინსტრუმენტების ანალიზი</b>	
1.1 ენერგეტიკული ოფის-სისტემების კლასიფიკაცია, მათი მართვისა და მოდელირების პრობლემები და ამოცანები -----	6
1.2 ოფის-დოკუმენტების კლასიფიკაცია და დოკუმენტბრუნვა -----	10
1.3 საოფისე სისტემების პროგრამული პაკეტის ინფრასტრუქტურა VBA ენის თვალთახედვით -----	13
1.4 UML-უნიფიცირებული მოდელირების ენა – ოფის სისტემების დაპროექტების ინსტრუმენტი -----	15
1.5 რელაციური ბაზების სემანტიკური პრობლემები, მათი გაფართოების ლოგიკურ-აღგებრული მეთოდები და გრაფული მოდელები -----	18
1.6 კატეგორიული მიდგომა ობიექტ-ორიენტირებული სქემის ასაგებად მონაცემთა ბაზაში -----	20
1.7 ობიექტ-როლური მოდელირება (ORM) -----	28
1.8 ამოცანის დასმა -----	30
- I თავის დასკვნები -----	33
<b>თავი 2. ბიზნეს-პროცესების მოდელირების და დაპროექტების ტექნოლოგია განაწილებულ ოფის-სისტემაში</b>	
2.1 დოკუმენტების ავტომატიზებული ფორმირების ანალიტიკური მოდელი -----	34
2.2 ბიზნეს-პროცესების ანალიზი და UseCase და Activity-დიაგრამების აგება -----	35
2.3 ბიზნეს-პროცესების საპრობლემო სფეროს კონცეპტუალური მოდელის აგება კატეგორიული მიდგომით -----	43
2.4 ORM-დიაგრამების აგება განაწილებული ოფის-სისტემებისთვის -----	49
2.5 ბიზნეს-პროცესების ობიექტ-ორიენტირებული დიაგრამის კლასთაშორისი კავშირების ნორმალიზაცია -----	59
- II თავის დასკვნები -----	66
<b>თავი 3. განაწილებული ოფის-ობიექტების ინფორმაციული და პროგრამული უზრუნველყოფა.</b>	
3.1 ოფის-სისტემების ობიექტ-ორიენტირებული მოდელირება და დაპროექტება UML-ტექნოლოგიის გამოყენებით -----	67
3.2 ოფის-სისტემების ინფორმაციული ბაზების აგება და SQL Server პაკეტით -----	84
3.3 SQL-ბაზისათვის .DDL ფაილების მომზადება და ინსტრუქციები XML დოკუმენტის მისაღებად -----	88
3.4 ინტერფეისის მომზადება ADO.NET პროგრამული პაკეტის საშუალებით -----	94
3.5 VBA ენის ინტეგრირებული გამოყენება განაწილებული ოფის-სისტემების აპლიკაციებში -----	100
3.6. საქმიანი პროცესების ობიექტ-ორიენტირებული მოდელირება და ანალიზი პეტრის ქსელებით -----	107
- III თავის დასკვნები -----	115
<b>თავი 4. განაწილებული ენერგეტიკული ოფის-სისტემების ამოცანების კომპიუტერული რეალიზაცია</b>	
4.1 ენერგოსისტემის ობიექტ-როლური მოდელი -----	117
4.2 ენერგოსისტემის მართვის ავტომატიზებული სისტემის სტრუქტურა -----	118
4.3 გამომუშავებული ელექტროენერჯის აღრიცხვის მათემატიკური მოდელი და მომხმარებლის ინტერფეისი -----	124
4.4 მონაცემთა ბაზების ავტომატიზებული დაპროექტება ენერგეტიკის სფეროში მომუშავე პერსონალისათვის -----	133
4.5 ელექტრომომარაგების აღრიცხვის საშუალებების კონტროლის სისტემა კლიენტ-სერვერული არქიტექტურის ბაზაზე -----	
- IV თავის დასკვნები -----	137
დასკვნები -----	142
ლიტერატურა -----	145
დანართი -----	150

## შუსაკალი

თანამედროვე ენერგეტიკული ობიექტების ქსელი უდავოდ მიეკუთვნება რთული და დიდი სისტემების კლასს, რომლისთვისაც დამახასიათებელია ძნელადფორმალიზებადი მართვის ამოცანების, დიდი მოცულობის საინფორმაციო ბაზების და გადაწყვეტილების მიღების რთული ალგორითმების არსებობა [20]. ყოველივე ეს კი აუცილებლად მოითხოვს ენერგეტიკაში მართვის ახალი სისტემების გამოყენებას.

ენერგოსისტემის საწარმოთა და ორგანიზაციათა მართვის ავტომატიზებული სისტემების ქსელის აგების კონცეფციამ, ერთის მხრივ, მცირე და საშუალო კერძო ბიზნესის სექტორის გაფართოების გამო, და მეორეს მხრივ, მკაცრი კონკურენციის პირობებში მათი მოქმედების არეალის საერთაშორისო ბიზნესის მოთხოვნების ჩარჩოებში მოქცევამ, მეტად მნიშვნელოვანი ფუნქცია შეიძინა [11,12,21]. განსაკუთრებით საყურადღებოა საქმიანი პროცესების ტექნოლოგიური ციკლების თანმხლები საინფორმაციო ნაკადების მართვის მექანიზმების დახვეწა, უნიფიცირებული დოკუმენტებისა და დოკუმენტბრუნვის პრობლემათა გათვალისწინებით, ახალი საკანონმდებლო აქტებისა და ინტერნეტ-ინტრანეტის პირობებში. ოფის-სისტემა, ზოგადად წარმოადგენს ფიზიკურად განაწილებულ ოფის-ობიექტებისა და მათი ლოგიკურად ერთიანი მონაცემთა ბაზების და ადმინისტრაციული მართვის კლასიკური მოდელის ინტეგრირებულ სუბიექტს, უახლესი ინფორმაციული, მოდელურ-პროგრამული, ტექნიკურ-ტექნოლოგიური, ლინგვისტურ-ინტერფეისული, იურიდიული და ორგანიზაციულ-მეთოდური უზრუნველყოფათა ერთობლიობით.

ნაშრომის მიზანია ორგანიზაციული და ბიზნესის მართვის განაწილებული ენერგოსისტემის ოფის-ობიექტებისათვის საქმიანი პროცესების წარმოების უნიფიცირებული დოკუმენტბრუნვის

ტექნოლოგიის შემუშავება და პროგრამული რეალიზაცია, თანამედროვე ობიექტ-ორიენტირებული დაპროგრამების მეთოდებისა და კლიენტ-სერვერული არქიტექტურის საფუძველზე.

**პირველ თავში** წარმოდგენილია ოფის-სისტემების კლასიფიკაცია, მათი მართვისა და მოდელირების პრობლემები და ამოცანები, ოფის-დოკუმენტების კლასიფიკაცია, აღწერილია საქმისწარმოების ძირითადი პროცესები და დოკუმენტბრუნვა. განიხილება განაწილებული ენერგეტიკული (და სხვა დარგის) ოფის-ობიექტების მონაცემთა ბაზების ობიექტ-ორიენტირებული მოდელირებისა და დაპროექტების ავტომატიზებული პროცესების დამუშავების ამოცანა, რაც საგრძნობლად ამცირებს სისტემების ინფორმაციული და პროგრამული პაკეტების აგების დროს და, ამასთანავე ორიენტირებულია გამოყენებითი სფეროს მომხმარებელზე. გადმოცემულია რელაციური ბაზების სემანტიკური პრობლემები და კატეგორიული მიდგომა ობიექტ-ორიენტირებული სქემის ასაგებად მონაცემთა ბაზებში.

**მეორე თავში** შემოთავაზებულია განაწილებულ ოფის-სისტემებში დოკუმენტების ავტომატიზებული ფორმირების ანალიტიკური მოდელი და მოთხოვნილებათა ანალიზი. წარმოდგენილია UseCase და Activity-დიაგრამები კონკრეტული შემთხვევებისათვის. აგებულია კონცეპტუალური სქემა კატეგორიალური მიდგომით და ამ მიდგომის საფუძველზე აგებულია ORM-დიაგრამები განაწილებული ენერგეტიკული ოფის-სისტემებისათვის. მოცემულია კლასებზე გაფართოებული ნორმალურ ფორმათა თეორია.

**მესამე თავში** განიხილება განაწილებული ოფის-ობიექტების ბიზნეს პროცესების მართვის კომპიუტერული სისტემის ინფორმაციული და პროგრამული უზრუნველყოფის დამუშავების ამოცანები. მონაცემთა განაწილებული ბაზა აგებულია MsSQL Server

და ADO.NET პროგრამული პაკეტებით კლიენტ-სერვერ არქიტექტურით. დამუშავებულია მომხმარებელთა ინტერფეისები ბაზებთან სამუშაოთ. პროგრამული აპლიკაციები შესრულებულია თანამედროვე კლასიკური ინსტრუმენტული საშუალებებით, კერძოდ ჯგუფური CASE-ტექნოლოგიებით და UML-მეთოდოლოგიით. მისი საშუალებით ხორციელდება ეფექტური მრავალდონიანი გამოყენებითი სისტემების დამუშავება მონაცემთა ბაზებთან სამუშაოდ ინტრაქსელსა და ინტერნეტში. ნაშრომში შემოთავაზებულია განაწილებული, საერთო გამოყენების რესურსების (მონაცემთა ბაზები, პროგრამები, საბეჭდი მოწყობილობანი, გადაცემის არხები და სხვ.) მართვის პროცესების მოდელირებისა და ანალიზის ეფექტური ალგორითმები პეტრის ქსელების ბაზაზე.

მეოთხე თავში მოცემულია განაწილებული ენერგეტიკული ობიექტების ბიზნეს-პროცესების ობიექტ-ორიენტირებული მოდელირების, ანალიზისა და დაპროექტების ამოცანების კომპიუტერული რეალიზაცია. კერძოდ, აგებულია საქართველოს ენერგოსისტემის ობიექტ-როლური მოდელი, მისი არსთა-დამოკიდებულების მოდელი და მონაცემთა განაწილებული ბაზების რელაციური მოდელი, რომელიც პროგრამულად რეალიზებულია კლიენტ-სერვერ არქიტექტურით MsSQL\_Server პაკეტის გამოყენებით. კლიენტ-მომხმარებელთათვის დამუშავებულია ინტერფეისები MsAccess, Excel, VBA და Visual Studio.NET ინტეგრირებული პროგრამული პაკეტებით, ASP.NET და ADO.NET გამოყენებით.

სადემონსტრაციო პროგრამული პაკეტი დამუშავებულია ენგურჰესის ჰიდროელექტროსადგურზე ელ-ენერჯის ყოველ-დღიური გამომუშავების ავტომატიზებული ადრიცხვის ამოცანისათვის.

დანართში მოტანილია ზოგიერთი რეალიზებული კომპიუტერული ამოცანის აღწერა, რომელიც ერთგვარ კომენტარს უკეთებს დისერტაციის ძირითად ტექსტში მიღებულ შედეგებს.

# I თავი

## განაწილებული ენერგეტიკული ოზის-სისტემების საქმიანი პროცესების ანალიზი, კლასიფიკაცია და ამოცანები

### 1.1. ენერგეტიკული ოზის-სისტემების კლასიფიკაცია, მათი მართვისა და მოდელირების პრობლემები და ამოცანები

ენერგეტიკა ქვეყნის ეკონომიკის წარმმართველი ძალაა. ენერგოსექტორის გამართული და ეფექტური ფუნქციონირება ყოველი ქვეყნის მდგრადი განვითარების საფუძველია, ბუნებრივია, ენერგოსექტორს უდიდესი პოლიტიკური, ეკონომიკური, სოციალური და გარემოს დაცვითი დატვირთვა აქვს.

საქართველოს ენერგეტიკა ქვეყნის საბაზრო ეკონომიკის პოტენციურად მაღალრენტაბელურ დარგს წარმოადგენს. ელექტროენერჯის ძირითადი წყაროებია ჰიდრო – (58,47%) და თბო – (41,53%) ელექტროსადგურები. [21]

ჰიდროელექტროსადგურებს მიეკუთვნება : ენგურჰესი, ვარციხეჰესი, ვარდნილჰესი, ჟინვალჰესი, რიონჰესი, ხრამი-1, ხრამი-2, ლაჯანურჰესი, გუმათჰესი, ზაჰესი, ძვერულჰესი, შაორჰესი, ხადორჰესი, ნამახვანჰესი, ჟონეთიჰესი, ტვიშიჰესი, ფარავანჰესი. თბოელექტროსადგურებს მიეკუთვნება : მტკვარი, თბილსრესი, გაზოტურბინა, გაზოტურბინა (მეორე).

საქართველოს ელექტრო ენერჯით მომარაგება დაყოფილია ოთხ ძირითად ნაწილად : თელასი, კახეთი, აჭარა,

ქვეყნის ელექტროენერგეტიკაში დღეისათვის არსებულ დონეს განაპირობებს მისი შემადგენელი სამი ძირითადი ნაწილის – გენერაცია, გადაცემისა და განაწილების მდგომარეობა. თავის მხრივ, ელექტროენერჯის მაგენერირებელ წყაროებში და საერთოდ, გენერაციაში მდგომარეობის განმსაზღვრელია ისეთი ძირითადი მახასიათებელი სიდიდეები, როგორცაა : დადგმული სიმძლავრე ;

საპროექტო და ფაქტიური გამომუშავება ; მოხმარება ; დადგმული სიმძლავრის გამოყენების კოეფიციენტი ; პირობითი სათბობის კუთრი ხარჯი ; ექსპორტ-იმპორტი ; გამომუშავებული ელექტროენერჯის თვითღირებულება და ტარიფი.

გადაცემა-განაწილება ის უმნიშვნელოვანესი რგოლია, რომელიც აკავშირებს გენერაციის ობიექტებს მომხმარებლებთან. სწორედ მის გამართულ მუშაობაზეა დამოკიდებული ელექტრო და თბომომარაგების სისტემის საიმედო ფუნქციონირება. ძალიან მნიშვნელოვანია არასწორი ქმედებების ანალიზი და მათი აღმოფხვრა სისტემური მიდგომის პრინციპების საფუძველზე.

არამართებული და არასწორი ქმედებების კლასიფიკაცია მოცემულია ქვემოთ :

- აღრიცხვიანობის მოუწესრიგებლობა ე.წ. „კომერციული დანაკარგები“;
- მოხმარებული ელექტროენერჯის საფასურის ამოუღებლობა ;
- არასრულფასოვნად გატარებული რეფორმები ;
- ელექტროენერგეტიკულ სექტორში ვალების წარმოქმნა ;
- ელექტროენერჯის ტრანზიტთან და ექსპორტ-იმპორტთან დაკავშირებული პრობლემები ;
- შესყიდვები. (სათბობის, მასალების, მოწყობილობების, ზეთების, ქიმიკატების და ა.შ. შექმნა შუამავალი ფირმებისა და პიროვნებების საშუალებით ; რემონტისა და სარეაბილიტაციო სამუშაოების განხორციელების შემთხვევები ტენდერის გამოცხადების გარეშე.)

ასევე აღსანიშნავია, რომ ენერგეტიკულ ოფის ახასიათებს ყველა ის თვისება, რაც ნებისმიერ ოფის-ობიექტს, სადაც ხდება საქმის წარმოება და საჭირო სხვადასხვა ორგანიზაციული სამუშაოს ჩატარება. ეს შეიძლება იყოს საბუხჰალტრო აღრიცხვა, კადრების განყოფილება, დაგეგმვისა და გასაღების განყოფილება და სხვ. თითოეულ განყოფილებაში წყდება მართვის შესაბამისი

ამოცანები, რაც გულისხმობს ინფორმაციის მიღებას, მის გადამუშავებას და მის გაცემას დოკუმენტირებული სახით. მაგ: უნივერსიტეტები, საწარმოო ფირმები, სავაჭრო ფირმები, მომსახურე ფირმები, სამინისტროები, საავადმყოფოები, სამშენებლო ფირმები და ა.შ.

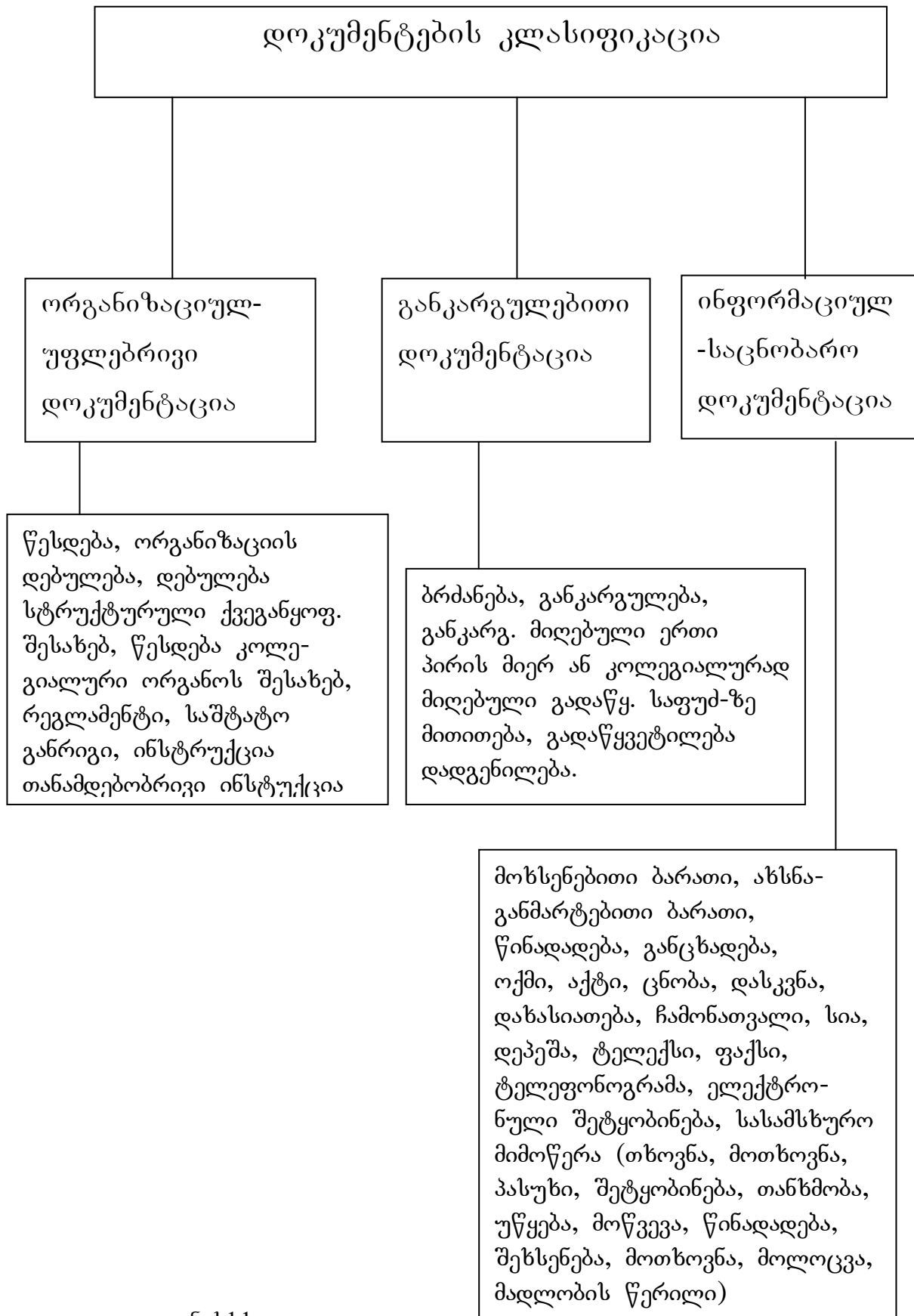
ადმინისტრაციული მართვის სფეროს მარტო გადაწყვეტილების მიღება არ მიეკუთვნება. მასში მნიშვნელოვანი ადგილი უჭირავს ახალი დოკუმენტების: ცნობების, მართვის ობიექტების მიმდინარე და მოსალოდნელი მდგომარეობის ანგარიშების ფორმირებასთან დაკავშირებულ საკანტორო საქმიანობას. ასეთი საქმიანობის ავტომატიზაციამ წარმოშვა ელექტრონული ოფისის კონცეფცია, რომელშიც მონაცემთა დამუშავება, გაცემა, შენახვა და მოძიება პერსონალური კომპიუტერის საშ-ით ხორციელდება. ელექტრონული ოფისის კონცეფცია განსაკუთრებით ნაყოფიერი გამოდგა, რადგან იგი კადრების მნიშვნელოვნად შემცირების, საწარმოს ქვედანაყოფებს შორის კავშირების გაუმჯობესებისა და მართვის ოპერატიულობის ამაღლების შესაძლებლობებს იძლევა.

საშუალო დონის სამმართველო კადრების ძირითად საქმიანობას ტექსტების დამუშავება, შენახვა და დოკუმენტების გამოტანა შეადგენს. დოკუმენტები მრავალი სახის არსებობს იხილეთ ნახ 1.1.

ჩვენთან ყველაზე მეტად გავრცელებულია ფირმა Microsoft-ის პროგრამული პაკეტი. MS Windows და Microsoft office. დოკუმენტების მომზადება ხდება ძირითადად MS Word-ში, საბუჰვალტრო აღრიცხვა Ms Excel-ში, მონაცემთა ბაზებთან მუშაობა Accses-ში, პრეზენტაციების მომზადება Power Point-ში. ელექტრონული ოფისის პრაქტიკაში გამოიყენება ისეთი



საშუალებები, რომლებიც საწარმოს საქმიანობის კონტროლისა და კოორდინაციას ახორციელებენ.



მათი დახმარებით მთელი მმართველობითი საქმიანობა წარმოგვიდგება, როგორც ისეთი პროცესების ერთობლიობა, რომლებიც ხასიათდება შესრულებული სამუშაოების თავისებურებებით, მათი შესრულების ვადითა და შემსრულებლებით. საქმიანობაში დიდი მნიშვნელობა ენიჭება ორგანიზაციის შიგნით მონაცემთა გადაცემის საშუალებებით მონაცემთა ნაკადის ოპერატიულ განაწილებას. ამიტომ მეტად მნიშვნელოვანია ელექტრონული ფოსტის, ტელეკონფერენციების და ვიდეოკონფერენციების, პრეზენტაციების გამოყენება..

მართვის პროცესში მმართველი აპარატის წინაშე შეიძლება წარმოიშვას სხვა სახის სიძნელეებიც. მაგალითად, გადასამუშავებელ მონაცემთა მოცულობა განუწყვეტლივ იზრდება, შესასრულებელი გამოთვლები სულ უფრო რთულდება, მონაცემთა დამუშავების ვადები იზღუდება, მომხმარებელთა რაოდენობა გამომავალ ინფორმაციაზე იზრდება, არქივების მოცულობა იზრდება, ხდება მმართველი პერსონალის გადატვირთვა.

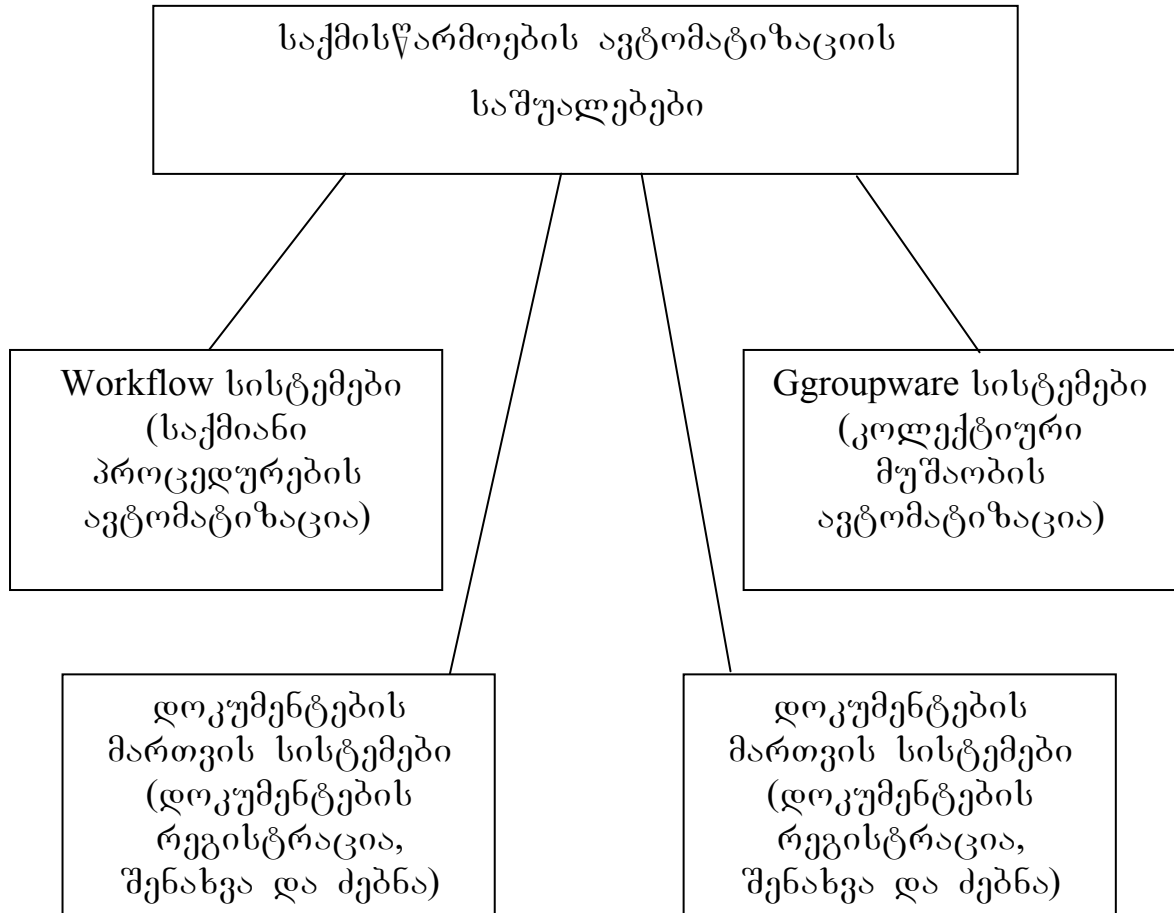
## 12. ოფის-დოკუმენტების კლასიფიკაცია და დოკუმენტბრუნვა

დოკუმენტბრუნვა – ესაა ორგანიზაციაში დოკუმენტების მოძრაობა მათი შექმნის მომენტიდან შესრულების დამთავრებამდე ან გაგზავნამდე. აქ აქცენტი გაკეთებულია “დოკუმენტის მოძრაობაზე”, ანუ ესაა გზა ერთი განყოფილებიდან მეორეში ან ერთი თანამშრომლიდან მეორისაკენ.[37]

საქმიანი პროცედურა – გარკვეული ოპერაციების (სამუშაოების, დავალებების) თანმიმდევრობა, რომლებსაც ასრულებენ ორგანიზაციის თანამშრომლები რაიმე ამოცანის გადასაწყვეტად. დოკუმენტბრუნვის მახასიათებელს წარმოადგენს მისი მოცულობა, რომელშიაც იგულისხმება იმ დოკუმენტების რაოდენობა, რომლებიც

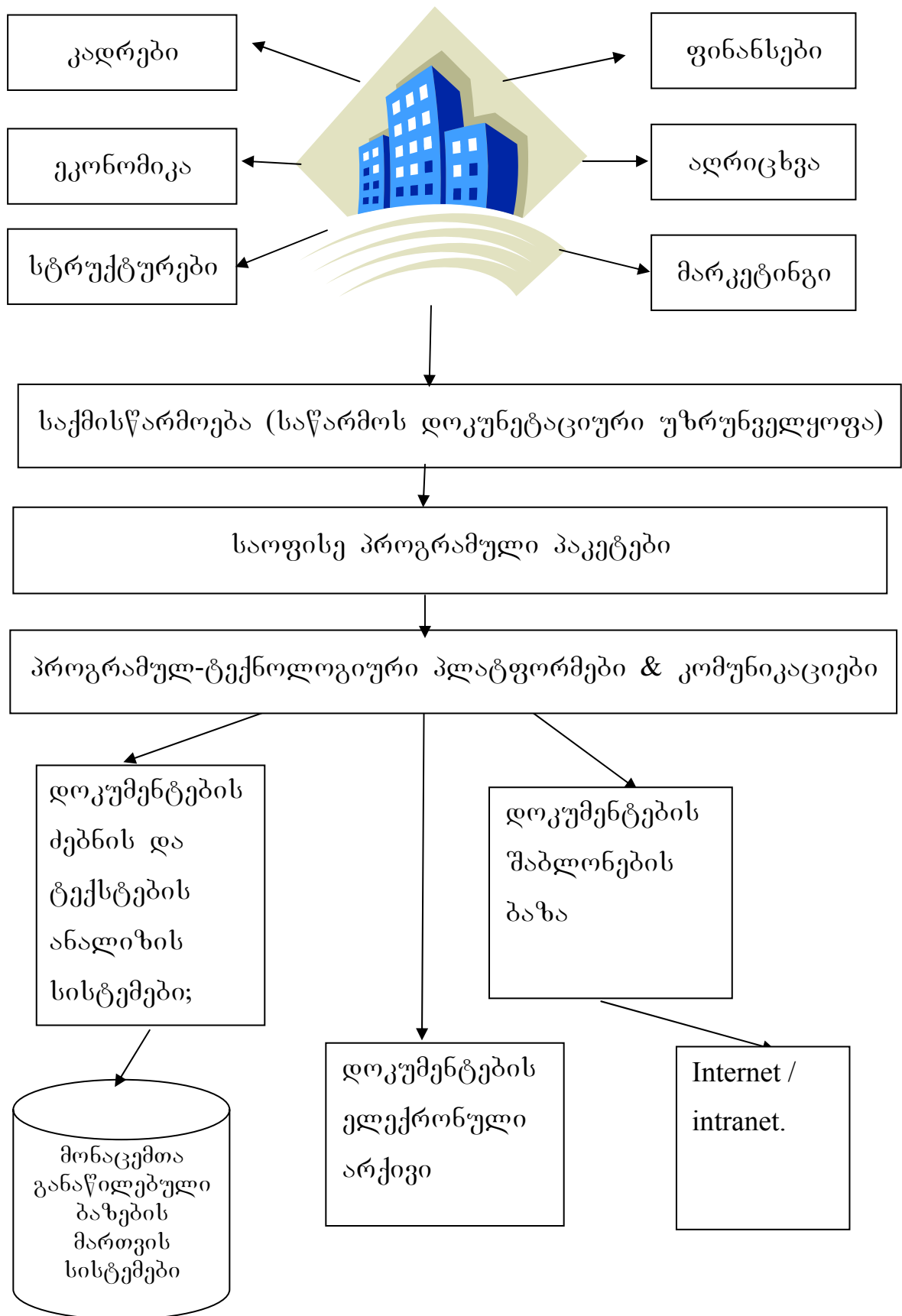
შემოვიდა ორგანიზაციაში და შექმნილია მის მიერ გარკვეული დროის განმავლობაში.

საქმისწარმოების ავტომატიზაციის საშუალებები შეიძლება დაიყოს ოთხ კატეგორიად. იხ. ნახ. ნახ.1.2



ნახ.1.2.

საქმიანი პროცედურების და კოლექტიური მუშაობის ავტომატიზება პრინციპში საკმაოდ რთულია. ბიზნესის წარმოებაში უნივერსალური წესების ჯერ არ გამოუგონიათ. თითოეული ორგანიზაცია აბსოლუტურად უნიკალურია. ამას გარდა, ავტომატიზებული პროცედურის ჩარჩოებში ყოველთვის არაა შესაძლებელი გარე სიტუაციებზე ადექვატური რეაგირება, რომელთაგანაც, ბუნებრივი არავინ არაა დაზღვეული. ამიტომ ავტომატიზაცია ხანდახან ვნებს ბიზნესს.



ნახ.13.

ამჟამად, საქმიანი პროცედურების ავტომატიზაციის ყველაზე რაციონალურ ხერხად უნპრიანია მივიჩნიოთ შემდეგი :” არ უნდა ვეცადოთ დავაპროგრამოთ თანამშრომლების ქცევა ყველა შესაძლო სიტუაციისათვის, ამის ნაცვლად შევქმნათ მათთვის საერთო ინფორმაციული გარემო, რომლის ჩარჩოებშიც ისინი შეძლებენ თანამშრომლობას (ე.ი. ერთად ამოხსნან ბიზნეს-ამოცანები) და გაცვალონ ერთმანეთში შეტყობინებები”.

ისმის კითხვა, რატომ აქვს ორგანიზაციებისათვის სასიცოცხლო მნიშვნელობა საწარმოს დოკუმენტაციური უზრუნველყოფის გაუმჯობესებას. ორგანიზაციის დოკუმენტაციური უზრუნველყოფა უშუალო გავლენას ახდენს მართვის ამოცანების მიღების ხარისხზე. ნახაზზე მოცემულის საწარმოს მართვის კომპონენტები. იხ. ნახ.1.3

### **1.3 საოფისო სისტემის პროგრამული პაკეტის ინფრასტრუქტურა VBA ( Visual Basic Application) ენის თვალთახედვით**

**VB-**ვიზუალ ბეისიკი აპლიკაციებისთვის (გამოყენებითი სფეროსთვის), წარმოადგენს სტანდარტულ მაკრო-ენას (მაკროსების ასაგებად), რომელიც ჩადგმულია Microsoft Office-ს პაკეტებში (Word, Excel, Access და ა.შ.) [14].

VBA-ს ძირითადი დანიშნულებანი: აგებული მაკროსების საშუალებით გააძლიეროს ოფისის პროგრამული აპლიკაციის ფუნქციები; შეცვალოს ოფისის მენიუს სტრუქტურა მომხმარებლის სურვილისამებრ; ოფისის სხვადასხვა პროგრამული პაკეტების აპლიკაციების მონაცემთა ინტეგრირება და გადამუშავება; ინტერნეტის Web-გვერდების ავტომატური შექმნა და განახლება.

VBA – გამოიყენებს ობიექტ-ორიენტირებული დაპროგრამების კონცეფციას. რედაქტორის დანიშნულებაა VBA - პროექტის შექმნა,

ოფისის ნებისმიერი პროგრამიდან (Word, Excel, Access და ა.შ.) შეიძლება რედაქტორის გამოძახება და მასში მუშაობა. VBA – პროექტის ასაგებად საჭიროა დაიწეროს მაკროსი.

VBA უზრუნველყოფს Excel-ში დაპროგრამების პროცესს ანუ Excel-ში იქნება პროგრამული კოდები (მაკროსები), რომლებიც საგრძნობლად ამაღლებენ Excel-პროგრამის სიმძლავრეს.

დღეისათვის საწარმოებსა და ორგანიზაციებში დიდი გამოყენება აქვს მაიკროსოფტის ფირმის საოფისო სისტემებს, როგორცაა Word, Excel, Access, Power Pont, Outlook და სხვ. განსაკუთრებით მნიშვნელოვანია ამ პროგრამების ინტეგრირებული გამოყენება, ასევე მათი კავშირები მონაცემთა ბაზების მართვის სისტემა Ms SQL Server-თან და Internet გლობალურ ქსელთან.

**MS Word - ტექსტური რედაქტორი** გარდა ძირითადი ზოგადი შესაძლებლობებისა, რომლებსაც იყენებენ Word-ის მომხმარებლები დოკუმენტების ასაწეობად, შესაძლებელია MS Word-ის კავშირი მონაცემთა ბაზებთან ან ელექტრონულ ცხრილებთან და მაკროსების დამუშავება Visual Basic Application (VBA) გამოყენებით.

პირველი აუცილებლობა გამოწვეულია მართვის ორგანიზაციულ სისტემებში გასაგზავნი კორესპოდენციის დიდი რაოდენობით, რომლის შესაბამის დოკუმენტთა სტრუქტურა (ფორმა) იდენტურია, ხოლო შინაარსი (მონაცემები) განსხვავებული. ასეთ შემთხვევაში Word-ის ფუნქცია Mail-Merge კარგ საშუალებას გვაძლევს. ერთის მხრივ ავაგოთ რთული ფორმის დოკუმენტი, რომლის ცალკეული ველების მნიშვნელობები შემდგომ დაკავშირებულ იქნება Excel-ის ან Ms SQL Server-ის ბაზის ფაილებთან. ამგვარად, მეორე მხრივ, შევძლებთ შინაარსობრივი მონაცემების ვარირებას რთული დოკუმენტის ფარგლებში.

**Microsoft Excel ელექტრონული ცხრილი** ელექტრონული ცხრილის გამოყენება მოსახერხებელია საინფორმაციო სისტემის აგების დროს. ინფორმაციის დამუშავების თანამედროვე ტექნოლოგია, რომელსაც იყენებს Microsoft ფირმა, ძალზე პრაქტიკულია. მაგალითად, Excel პროგრამის კავშირი სხვა ტიპის სისტემებთან, მაგალითად, Ms SQL Server-თან, Visual Foxpro-თან, Access-თან და ა.შ. Excel-ში შესაძლებელია აგრეთვე (Object Linking and Embedding) ტექნოლოგიით. მისი საშუალებით Windows-თან თავსებადი აპლიკაციები – ელექტრონული ცხრილები, ტექსტური პროცესორები და მონაცემთა ბაზები მუშაობენ ერთობლივად ანუ შეუძლიათ გადასცენ ერთმანეთს თავისი ობიექტები. ამის კარგ მაგალითს წარმოადგენს Word -ის დოკუმენტში უშუალოდ Excel ცხრილის ჩასმა და მუშაობა.

**Microsoft Access – მონაცემთა რელაციური ბაზების მართვის სისტემა** Microsoft-ოფისის პროგრამულ პაკეტში ერთ-ერთი მძლავრი ინსტრუმენტული საშუალებაა ლოგიკურ და გლობალურ ქსელებში სამუშაოდ). Access სისტემა დამუშავდა სპეციალურად Windows ოპერაციული გარემოსთვის და მისი საშუალებით წყდება შემდეგი ამოცანები: მონაცემების ორგანიზება მანქანის მექსიერებაში და მათი მართვა; ინფორმაციის მოძებნა და მათი მოდიფიკაცია; მონაცემთა ამორჩევა წინასწარ მითითებული კრიტერიუმებით; ფორმებისა და ანგარიშების აგება; სპეციალური მაკროსების დამუშავება ბაზებთან სამუშაოდ და ა.შ.

#### **14. UML-შენიშვნითი მოდელირების ენა – ოვის სისტემების დაპროექტების ინსტრუმენტი**

**UML-ტექნოლოგია** ობიექტ-ორიენტირებული მოდე-ლირებისა და სტრუქტურული დაპროგრამების იდეოლოგიის მატარებელია, რომელიც თეორიული და პრაქტიკული ინფორმაციის სამი

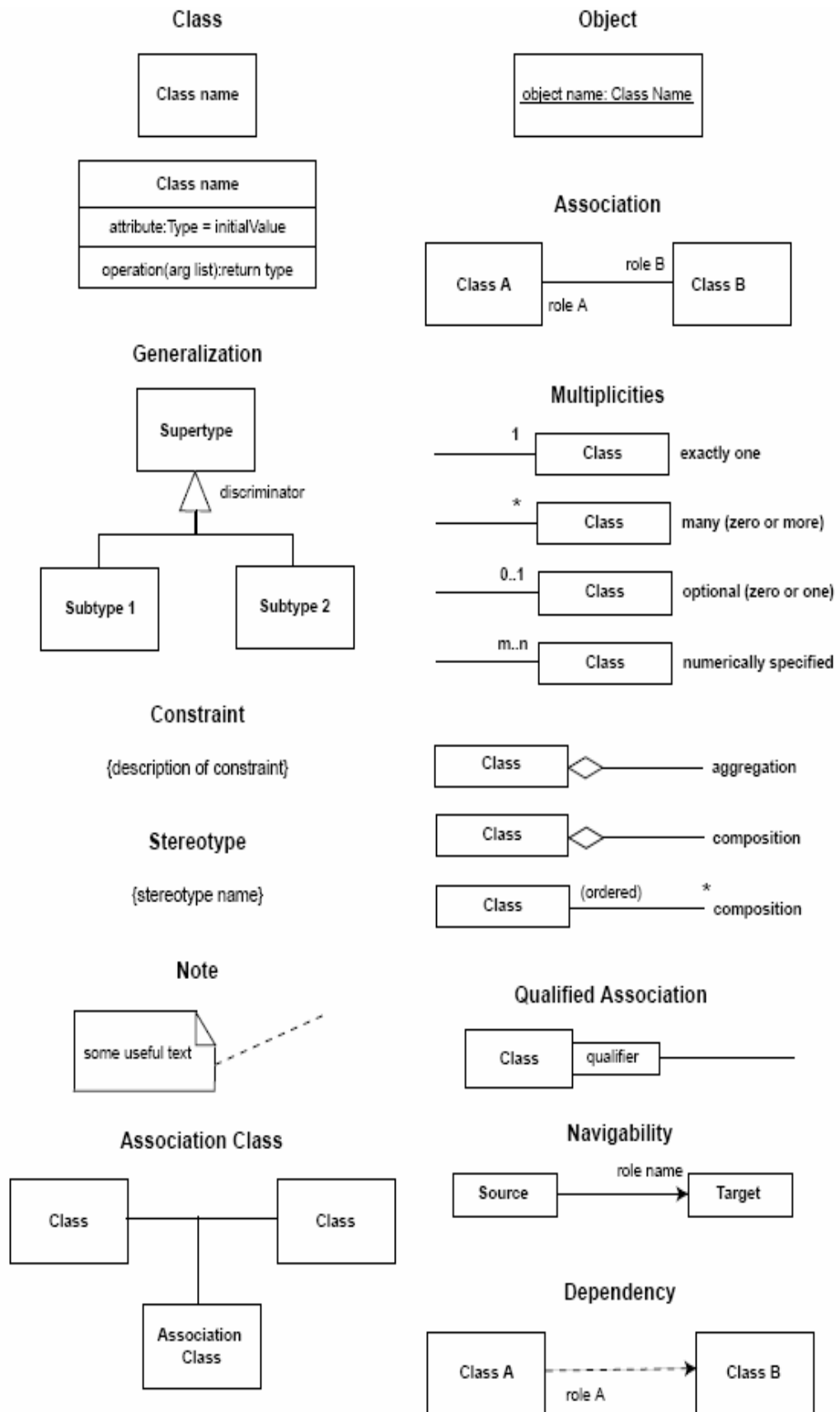
ძირითადი მიმართულების (დაპროექტების ავტომატიზაცია, დაპროგრამების ავტომატიზაცია, მონაცემთა ბაზების დამუშავების ავტომატიზაცია) „გენეტიკური“ მემკვიდრეა. [17]

მართვის კომპიუტერული სისტემების პროგრამული უზრუნველყოფის აგების პროცესების ასეთი სრულფასოვანი ავტომატიზაცია ვიზუალური მოდელირების სახელწოდებით დამკვიდრდა და იგი მოდულების გრაფიკულ წარმოდგენას ეყრდნობა. ასეთი ინსტრუმენტები ფლობს როგორც პირდაპირ (გრაფიკიდან პროგრამული კოდისაკენ), ასევე რევერსიულ (კოდიდან გრაფიკისაკენ) ტექნოლოგიას. ობიექტ-ორიენტირებული ანალიზის, დაპროექტებისა და პროგრამირების სწრაფმა განვითარებამ ხელი შეუწყო მძლავრი ინტეგრირებული, ვიზუალური მეთოდების განხორციელებას და ინსტრუმენტების შექმნას. ახალი კომპიუტერული ტექნოლოგიების ასეთ მაგალითს მიეკუთვნება პროგრამული პაკეტი Paradigm Plus, რომელიც ცნობილია Use Case („გამოიყენე შემთხვევა“ ან „აირჩიე მეთოდი“) ტიპის ადამიან-მანქანური სისტემების შესაქმნელად. Case -Computer aided software engineering სისტემებს საფუძვლად უძევს Unified Modeling Language (UML) ტექნოლოგია.

UML არის პროგრამული მოდულების სპეციფიკაციის, კონსტრუირების, ვიზუალიზებისა და დოკუმენტირების ენა და აღნიშნავს სისტემას. 1997 წლიდან Object Management Group-ს (OMG) მიერ იგი გამოცხადდა როგორც პროგრამული სისტემების აგების სტანდარტი. UML არაა კონკრეტული მეთოდი. იგი არის ბაზა სხვადასხვა მეთოდებისათვის, ვინაიდან მასში შემუშავებულია მოდელირების კონსტრუქციების განსაზღვრული სიმრავლე აღწერის ერთიანი სისტემითა და სემანტიკით. მოდელირების ელემენტები იყოფა სხვადასხვა ტიპის დიაგრამებად:

UML-ის გრაფიკული კონსტრუქციები ნაჩვენებია ნახაზზე 1.4.





6sb.14.

**15. რელაციური ბაზების სემანტიკური პრობლემები და მათი  
გაფართოებული მოდელები**

მონაცემთა რელაციური მოდელი ფორმალური მათემატიკური ობიექტია და მისი საშუალებით საგნობრივი სფეროს არაფორმალურად აღწერილი თვისებების ასახვას მიეცემა რთულ სემანტიკურ პრობლემამდე. იგულისხმება სემანტიკის (შინაარსის) მათემატიკური მოდელირების სპეციფიკური პრობლემები. მათი სირთულე ძირითადად განისაზღვრება მონაცემთა ბაზის რეორგანიზაციის (განახლების) პროცედურების სირთულით, სტატიკური და დინამიკური შეზღუდვების (პრედიკატების) სისწორის შემოწმებით. [19,45]

შეზღუდვები, გამოსატყვის მონაცემთა ბაზის მდგომარეობის (რელაციები) ზოგად, აბსტრაქტულ თვისებებს ანუ მონაცემთა ბაზის სემანტიკას. შეზღუდვების გამოსატყვის ყველაზე ბუნებრივი ხერხია გამონათქვამები I-რიგის პრედიკატების ენაზე, რომელიც განსაზღვრავს მონაცემთა ბაზის დასაშვებ მდგომარეობათა სიმრავლეს.

მთლიანობის პრედიკატების ცოდნის საფუძველზე, რომელიც ერთ–ერთ მნიშვნელოვან ნაწილს ფუნქციონალური დამოკიდებულებები შეადგენს, ხდება სინამდვილის (საგნობრივი სფეროს) მოდელირება ნორმალურ ფორმათა თეორიის საშუალებით. ფორმირდება მონაცემთა ბაზის სქემა, რომელიც ამ დამოკიდებულებათა სასრულ ერთობლიობას შეადგენს. მაგრამ ფუნქციონალური დამოკიდებულებები ასახავს სინამდვილის შინაარსობრივი მხარის მხოლოდ ნაწილს; მონაცემთა ბაზის მოდიფიკაცია კი სემანტიკური ბუნების მატარებელია. აქედან გამომდინარე, არაა ცხადად წარმოდგენილი, თუ როგორ შეიძლება დამოკიდებუ-

ლებათა ნორმალიზაციის სინტაქსური პროცედურებით გადაწყდეს სემანტიკური პრობლემები?!

საგნობრივი არის სტრუქტურისა და კანონზომიერებების მათემატიკური მოდელირებისას რელაციურ მოდელში გამოიყენება სიმრავლეთა თეორიის ელემენტები. ამიტომ ისმის კითხვა, რამდენად შეესაბამება მონაცემთა ბაზაში ფორმალურად ასახული ცნებები სიმრავლეთა თეორიის ცნებებს (ტერმინოლოგიის მათემატიკური სიზუსტის თვალსაზრისით).

რელაციური ბაზა გამოიხატება დამოკიდებულებათა ქვესქემების ერთობლიობის საშუალებით, ე.ი. გვაქვს რელაციების (ცხრილების) ერთობლიობა, რომლებშიც ასახულია ობიექტი ( $R_i$ ), მათი თვისებები ( $A_j$ ), კონკრეტული ურთიერთკავშირებით ( $Z_k$ ). კონკრეტული საგნობრივი სფეროდან გამომდინარე, რელაციური მოდელის აგების დროს ობიექტები, თვისებები და მნიშვნელობები შინაარსობრივად ურთიერთშენაცვლებადია, ე.ი. არ არსებობს მოდელირების საბოლოო ცალსახა ალგორთმი ამ საკითხის გადასაწყვეტად. ამასთანავე, რელაციური ბაზის მოდელში არაა ასახული ობიექტებს შორის კავშირების სემანტიკა. ხშირად კი, შინაარსობრივად ერთი და იგივე თვისება შეიძლება წარმოდგენილ იქნეს სინტაქსურად განსხვავებული ფორმებით (სინონიმები). სისტემას კი სჭირდება დამატებითი ინფორმაციული საშუალებები, რათა ეს მომენტები გაითვალისწინოს.

ჩვენ შემოვიფარგლეთ ზოგიერთი ძირითადი პრობლემური საკითხების ანალიზით, რომლებიც დამახასიათებელია რელაციული მოდელისათვის. შეიძლება აღინიშნოს, რომ ასეთი პრობლემების გადაჭრა შესაძლებელია როგორც რელაციური მოდელის შედარებით გართულებით (დამატებითი ინფორმაციის შემოტანით), ასევე პროგრამული და აპრაქტული საშუალებების სრულყოფით.

რელაციური მოდელის სემანტიკური ანალიზი

ფუნდამენტურად ჩაატარა ეკოდმა და თავის ნაშრომში გადმოსცა ის ძირითადი მოთხოვნები, რომლებიც დამატებული უნდა იყოს რელაციურ ბაზაში. მისი აზრით სისტემა უნდა გაფართოვდეს ახალი ცოდნის დედუქციური გამოყვანის საშუალებებით არსებული ფაქტების საფუძველზე. იგი მნიშვნელოვნად თვლის აგრეთვე მონაცემთა ბაზის დამოკიდებულებებისა და მათი ძირითადი სტრუქტურების ეკრანზე გამოტანის გრაფიკული საშუალებების არსებობას, რაც მომხმარებლისათვის ერთგვარ კომფორტს ქმნის.

რელაციური მოდელის სემანტიკური გაფართოების საშუალებით კორტეჟი ხდება შედარებით ინფორმატიული, ვინაიდან იგი ჩვეულებრივ სალაპარაკო ენის სტრუქტურას მოიცავს. ე.ი. ამ შემთხვევაში წინადადების თითოეულ სიტყვას (არსებით სახელს, ატრიბუტს და მის მნიშვნელობას) აქვს თავისი როლი: სუბიექტი, ობიექტი, დამატება და ა. შ., გამოკვეთილია აგრეთვე მოქმედება (შემასმენელი), რომელიც ერთგვარად ხსნის დამოკიდებულების ატრიბუტთა ურთიერთმიმართებას.

#### **1.6. კატეგორიული მიღზომა ობიექტ-ორიენტირებული სქემის ასაბეზალ მონაცემთა ბაზებში**

კატეგორიები რომლებსაც ვიხილავთ, არის ლოგიკური გრამატიკის შემადგენელი ნაწილი, რომელიც, როგორც ზოგადი ენის საძირკველი, ბევრად მდიდარია, ვიდრე ტრადიციული პრედიკატების ლოგიკა და რომელიც ცხადია დამოუკიდებელი უნდა იყოს ბუნებრივი ენის ემპირიული გრამატიკისგან [71].

პრედიკატების ლოგიკა არის ლოგიკური გრამატიკა დანართების შინაარსის ასაგებად, მაგრამ ძალზე ღარიბია და ნაკლებად განვითარებული - იგი არ ფლობს სინტაქსურ კატეგორიებს,

რათა განსხვავოს ერთმანეთისგან მთავარი და არამთავარი პრედიკატები.

სალაპარაკო ენის წინადადება „ფილო არის ყავისფერი ძაღლი“ პრედიკატების ენაზე ჩაიწერება ასე:

**ფილო  $\varepsilon$  ძაღლი  $\wedge$  ფილო  $\varepsilon$  ყავისფერი**

$\varepsilon$ -აღნიშვნა პრედიკატების ლოგიკაშიც და ზოგად ლოგიკურ ენაშიც არის კოპულა (კავშირი, СВЯЗКА), ანუ დამხმარე საშუალება ენაში ასახვისათვის, რათა პრედიკატი (შემასმენელი) შეუსაბამოს არსს (საგანს, ობიექტს).

პრედიკატიზაცია არის ასიმეტრიული. კოპულას ( $\varepsilon$ ) მარცხნივ ზის საკუთარი სახელი (ფილო) ან მახასიათებელი (აღნიშვნა). მარჯვნივ ზის პრედიკატორი (არსის დასახელება – განსხვავებული გამოსახულება).

საკუთარი სახელები და მახასიათებლები არის ნომინატორები. ამგვარად, ნომინატორები, კოპულა და პრედიკატორები არის სამი კატეგორია, რომლებიც დასაწყისში შემოვიტანეთ.

განსხვავება მთავარ და დამხმარე პრედიკატორებს შორის პრედიკატულ ლოგიკაში წაშლილია, ვინაიდან პრედიკატულ-ლოგიკური გამოსახულება შეიძლება გულისხმობდეს წინადადებას „ფილო არის ძაღლური ყავისფერი“

„ფილო  $\varepsilon$  (ყავისფერი) (ძაღლი)“, სადაც ყავისფერი დამხმარე და ძაღლი მთავარი პრედიკატორია.

ელემენტარულ წინადადებაში (მარტივ გამონათქვამში) შეიძლება იყოს ერთი მთავარი და რამდენიმე დამხმარე პრედიკატორი, აქ ლოგიკური კავშირი არაა. სინტაქსური კატეგორიები, არსებისა (ობიექტების) და მოქმედებებისგან (ოპერაციების) განსხვავებით პრედიკატულ ლოგიკას არ გააჩნია. გამონათქვამიდან:

„ფილო დარბის“ გვექნება „ფილო  $\varepsilon$  მორბენალი“, ან დამხმარე პრედიკატორით: „ფილო დარბის სწრაფად“ ვლებულობთ:

„ფილო  $\varepsilon$  სწრაფი  $\wedge$  ფილო  $\varepsilon$  მორბენალი“.

მნიშვნელოვანია, რომ აქტიური კომპონენტები განვასხვავოთ სუფთა ობიექტებისაგან (არსებისგან).

გამოსახულებაში: „ფილო  $\pi$  სირბილი“ ან „ფილო  $\pi$  (სწრაფი)(სირბილი)“ მოქმედების კოპულით  $\pi$  (კეთება-tun). ნახვენებია, რომ მთავარი პრედიკატორის სახით გამოდის მოქმედების პრედიკატორი და არა ობიექტის პრედიკატორი.

ლოგიკურ გრამატიკაში, რომელსაც ჩვენ განვიხილავთ, არსებობს  $\pi$ -კოპულას შემდეგ ასევე მხოლოდ ერთი მოქმედების პრედიკატორი, ვინაიდან რამდენიმე მოქმედება ერთდროულად ელემენტარულად ვერ შესრულდება.

რთული წინადადება, მაგ., რამდენიმე მოქმედების პრედიკატორით  $\pi$ -ს შემდეგ, მიიღება მარტივი წინადადებების ლოგიკური ნაწილაკით შეერთებით. პრედიკატული ლოგიკა უნდა განვიხილოთ, როგორც ლოგიკური გრამატიკის გაფართოება.

კითხვა, თუ როგორ ამუშავებენ ელემენტარულ წინადადებებში რამდენიმე პრედიკატორ-არსს ერთი  $\varepsilon$  ან  $\pi$  კოპულას საშუალებით, შემდგომი მსჯელობის საგანია. ლოგიკურ გრამატიკაში ჩვენ ვიყენებთ ლორენცის [1] მოსაზრებას.

მაგ., წინადადება: „მიუღერი იღებს დეტალს საწყობიდან“.

დეტალი - პირდაპირი ობიექტია, რომელზეც მიუთითებს მოქმედების პრედიკატორი „იღებს“. პირდაპირ ობიექტს შეიძლება მოჰყვეს ირიბი ობიექტი, მაგ., „საწყობი“, რომელიც ადგილის გარემოებაა. იგი აღნიშვნა ანუ მახასიათებელია, რომლისთვისაც შეიძლება ჩაისვას საკუთარი სახელი (მაგ., ავრორა). დაიწერება

აგრორა =  $i$  საწყობი“, სადაც = არის განსაზღვრების ნიშანი, ხოლო Jota ( $i$ ) არის მახასიათებლის ნიშანი.

ელემენტარული წინადადება ჩაიწერება ასე:

მიუღერი	$\pi$	(აღება)	(დეტალი)	( $i$ საწყობი)
[ნომინატორი]		[მოქმედების პრედიკატორი]	[ობიექტის პრედიკატორი]	[მახასიათებელი]

აქმდე განხილულ კატეგორიებს:

- ნომინატორი – პრედიკატორი;
- მთავარი პრედიკატორი – დამხმარე პრედიკატორი;
- ობიექტის პრედიკატორი – მოქმედების პრედიკატორი;
- არის კოპულა ( $\epsilon$ ) – კეთების კოპულა ( $\pi$ ).

ლოგიკურ გრამატიკაში ემატება შემდეგი მნიშვნელოვანი კატეგორიები, რათა შესაძლებელი იყოს კონცეპტუალური სქემის მეთოდურად აგება. ასეთ კატეგორიებს მიეკუთვნება უპირველეს ყოვლისა მსჯელობა ლოგიკური ენის დონეებზე, ანუ დიფერენცირება კონკრეტულ და აბსტრაქტულ დონეებზე და მათ ობიექტებზე.

შემოიტანება აგრეთვე კატეგორია „მთელი-ნაწილი“;

სქემების აგებისას დრო არის ჩვეულებრივი პრედიკატორი, რომელიც „მანამ“, „შემდეგ“ და ა.შ. მიეთითება ორადგილიან პრედიკატორში „ $x, y \in$  before“ ან „ $x, y \in$  after“.

ამგვარად, შემოიტანება კიდევ შემდეგი კატეგორიები:

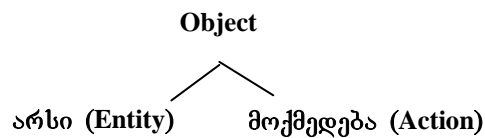
- აბსტარქცია – კონკრეტიზაციის;
- კომპოზიცია – დეკომპოზიციის: „მთელი - ნაწილი“;
- ობიექტების ენა – მეტაენა: ობიექტების ენა ერთი ენაა (I დონე, მაგ., ინგლისური) რომლის შესახებაც საუბრობენ მეტაენაზე (II დონე, მაგ., ქართული);

• სქემის დრო - მიმართვის (ათვის) დრო.

სქემის დროსთვის იყენებენ ტერმინს „ინტენსიონალი“ (შინა-არსი) და მიმართვის დროსთვის „ექსტენსიონალი“ (მოცულობა).

„ობიექტი“ არის ყველაზე მეტად განზოგადებული პრედიკატორი და არ გამოდგება განსხვავებულობის შესადარებლად.

ესაა ერთი მთავარი სახელის ქვეშ მოქცეული მნიშვნელობები. მიღებულია ობიექტის სინონიმად „არსი“-ს გამოყენება (thing – entity – сущность). ამგვარად, ობიექტი ხასიათდება არსით და მოქმედებით (ქცევით – action - действие).



ნახ.1.5.

წინადადება: „მოქმედებები არის ობიექტები“ არაბუნებრივია, ვინაიდან ხშირად არსი აღიქმება ობიექტად და არა მოქმედება.

პრედიკატორ „ობიექტის“ უნივერსალობა დგება ეჭვის ქვეშ. თუ „ობიექტის“ მაგივრად გამოიყენება „არსი“, მაშინ გამონათქვამი “Actions are entities” – ნორმალურად აღიქმება (ინგლისელ-ამერიკელების მიერ). რჩება შეკითხვა: ტერმინი „ობიექტ-ორიენტულობა“ ნიშნავს მხოლოდ „არს-ორიენტულობას“, თუ ასევე „მოქმედება-ორიენტულობასაც“?. დღეისათვის „მოქმედების“ (action) ნაცვლად იყენებენ ტერმინს მოვლენა (event). განმეორებად მოვლენათა ერთობლიობას უწოდებენ პროცესს (process).

ტერმინი „პროცედურა“, „მეთოდი“ არის ოპერაციათა სხვა ტიპები. ინფორმატიკაში მისთვის მიღებულია ტერმინი „ყოფაქცევა“ (behavior). ამგვარად, მე-9 კატეგორიაა:

**არსთა სქემა - ყოფაქცევის სქემა** (ერთად აღებული არის ობიექტის სქემა, ანუ არსთა დამოკიდებულების მოდელი).



მონაცემთა და პროცესთა გამოცხადება (declaration) მათი განმეორებადი გამოყენების მიზნით ამის ტიპური მაგალითია.

ლოგიკური გრამატიკა, რომლის ნაწილიცაა პრედიკატების ლოგიკა, არის ზოგადი სპეციფიკაციების მედიუმი (საკომუნიკაციო საშუალება).

ამოსავალი წერტილია – ამოცანის დასმის წარმოდგენა, პროექტი ტექნიკურ დავალებათა შესახებ (ან მოვალეობათა რვეული). საბოლოო წერტილი (მიზანი) – მოქნილ ანგარიშთა სისტემა (დანახარჯების) მაღალი ეკონომიურობით. მათ შორის გზის აღწერა არაა მარტივი.

ამ საკითხების გადაწყვეტა ფილოსოფიურ სფეროს ეხება, რომელიც ენასთან და შემეცნებასთანაა დაკავშირებული. ამ სფეროში ნაშრომები ეკუთვნის კანტს, რომელმაც პირველმა შემოიტანა განსხვავება სქემასა და მის გამოსახულებას შორის. ფრეგემ შემოიტანა აბსტრაქტული თეორია და პრედიკატების ლოგიკა. რასელმა და ფრეგემ შექმნეს აღნიშვნათა თეორია. კარნაპმა განავითარა „ინტენსინალ-ექსტენსიონალობის“ თემა.

ჩვენი ამოცანაა აღნიშნული თემის გამოყენება ინფორმატიკის ობიექტ-ორიენტირებული მოდელირების სფეროში, კერძოდ მონაცემთა ბაზებსა და დაპროგრამებაში.

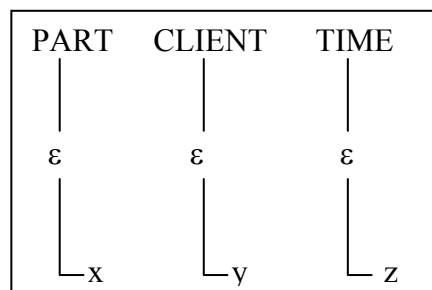
ახალი დანართის (აპლიკაციის) აგება ხორციელდება ცნობილი, უკვე არსებული საშუალებებით, რათა შესაძლებელი გახდეს ახალი ინსტრუმენტის კონსტრუირება. როცა პროგრამულ უზრუნველყოფაზეა დებატები, ამ შემთხვევაში ყოველთვის ენის საშუალებებს გულისხმობენ. მაგალითად, არცერთი პროგრამისტი დღეს არ წერს პროგრამას უკვე არსებულის გამოყენების გარეშე.

ტექნიკური დავალება არის არაფორმალურიზებულად გადმოცემული ტექსტი. კატეგორიული განხილვა, ანუ ტექნიკური

დავალების ენის-კრიტიკული (ანალიზი) რეკონსტრუირება (სპეციფიცირება) მოიცავს კომპლექსურ პრობლემას.

I-ბიჯის ამოცანაა ტექნიკური დავალებიდან მთავარი გამონათქვამების გამოყოფა. რა არის მთავარი გამონათქვამი ? ეს დამოკიდებულია მიზნის დასმაზე ამოცანისათვის. ჩვენ გვინტერესებს გარკვევა, მაგ., შეკვეთების დამუშავების ამოცანაში „საზღვარგარეთელი კლიენტი“ და არა „კლიენტური საზღვარგარეთელი“. აქ „საზღვარგარეთელი“ არის არა-მთავარი (!) პრედიკატორი, ვინაიდან ჩვენ არ ვიხილავთ მოქალაქეთა ცნობათა ბიუროს ამოცანას.

მთავარი პრედიკატები (ნახ.1.6), რომლებიც ჩვენ გვინტერესებს, არის: დეტალი, კლიენტი, შიკვითის-დრო (PART, CLIENT, TIME) და ა.შ. (ისინი ჩაიწერება ინგლისური დიდი ასოებით PART. . .) მათ გარეშე პრობლემა არ გადაწყდება. მთავარ პრედიკატორებს საწყის ფაზაზე აქვს ბირთვის (საწყისის) მნიშვნელობა.

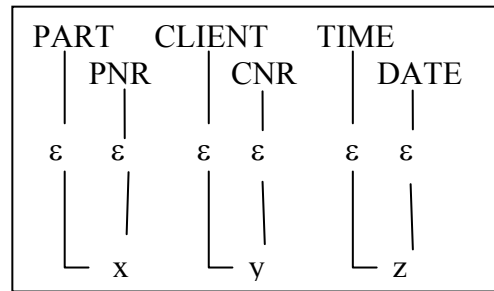


ნახ.1.6

კატეგორიალური თვალსაზრისით უნდა უყოს მოფიქრებული, რომ ჩვენ ვიმყოფებით სქემის-დროში (schma-time) და არა მიმართვის-დროში (referent-time).

მაგ. „მიუღერი ε კლიენტი“ - ეს არაა განზოგადება, პირიქით „y ε კლიენტი“ არის y-ნომინატორული ცვლადით ყველა შემთხვევისთვის განზოგადებული.

მიუღერი მხოლოდ კონკრეტულ მომენტში გააქტიურდება. ე. ი.  $y$  – ხდება მიუღერი. აქ  $x$ ,  $y$  და  $z$  აღიწერება როგორც ცვლადები, ზოგადად და მიმდინარე დროში (გააქტიურებისას) მიიღებს კონკრეტულ მნიშვნელობას. სისტემაში მთავარი პრედიკატები მიიღება ჩანაწერის დამატებით აღნიშვნით, რომლებიც სქემაზე გამოისახება PNR, CNR, DATE და ა.შ.



ნახ.1.7

ამ შემთხვევაში შეისწავლება ენის ლოგიკური პროცესის აბსტრაქცია. მაგალითად, ენაში სინონიმების (ერთნაირი მნიშვნელობების) არსებობა [CLIENT და CNR].

შემოიტანება „ $\Rightarrow$ “ წესის ისარი, რომელიც ნიშნავს, რომ თუ მოცემულია მარცხენა მხარე (გამონათქვამი), მაშინ მარჯვენაც მართებულია:

$$y \in \text{CLIENT} \Rightarrow y \in \text{CNR}$$

$$y \in \text{CNR} \Rightarrow y \in \text{CLIENT}.$$

მეორე მაგალითი: „ $s$  ურეკავს კლიენტს“, ჩაიწერება

$$s \pi \text{ (დარეკვა)(CLIENT)}.$$

მაგრამ ცოტა უხერხულია თქმა  $s \pi \text{ (დარეკვა) (KNR)}$ .

თუ  $A_R(X)$  გამონათქვამის ფორმაა, სადაც  $R$  დასაშვებია და  $X$  პრედიკატორთა ცვლადებია, მაშინ შეიძლება  $A_R(\text{CLIENT})$  ან  $A_R(\text{CNR})$  ჩაწერის ფორმის შემოღება. ორადგილიანი რელაციის  $\rho$ -ს შემოტანით  $A_R$ -ის მაგიერ:

$$(\text{CLIENT}) \rho (\text{CNR}) \rightarrow (A_R(\text{CLIENT}) \leftrightarrow A_R(\text{CNR})).$$

CLIENT და CNR ურთიერთჩანაცვლებადია (არ იცვლება ჭეშმარიტება გამონათქვამისა  $A_R(X)$ ).

$\rho$ -აბსტრაქციის თვალსაზრისით არის ეკვივალენტობის რელაცია, რომლისთვისაც სამართლიანია რეფლექსურობის, სიმეტრიულობის და ტრანზიტულობის თვისებები.

აღნიშნული თეორიის პრაქტიკული გამოყენების საკითხს ჩვენ მეორე თავში დაგუბრუნდებით.

### 1.7 ობიექტ-როლური მოდელირება (ORM)

ცნობილია, რომ მონაცემთა ბაზის აპლიკაციის ხარისხი კრიტიკულადაა დამოკიდებული მის დაპროექტებაზე. ინფორმაციული სისტემების განსაზღვრა ყველაზე უკეთ კონცეპტუალურ დონეზე ხდება, სადაც გამოიყენება ისეთი კონცეფციები და ენა რომელიც ადვილი გასაგებია ადამიანისათვის. კონცეპტუალური დაპროექტება შეიძლება შეიცავდეს მონაცემებს, პროცესებს და ქცევით პერსპექტივებს, ხოლო მონაცემთა ბაზების მართვის სისტემა (DBMS) დაპროექტებისას გამოიყენებდა ცნობილი ლოგიკური მოდელებიდან (რელაციური, იერარქიული, ქსელური, ობიექტ-ორიენტირებული და ა.შ.) ერთ-ერთს. [558-60]

მონაცემთა ბაზის დაპროექტება მოიცავს საპრობლემო არის ფორმალური მოდელის აგებას ანუ მისი აღწერის (საუბრის) ფორმირებას (universe of discourse Uod). სათანადოდ ამის გაკეთება, კი დამოკიდებულია Uod-ის კარგ ცოდნაზე. ობიექტ-როლური მოდელირება (ORM) ამარტივებს დაპროექტების პროცესს, იყენებს რა ბუნებრივ, სალაპარაკო ენას, ასევე ინტუიციურ დიაგრამებს, რომელთა შევსებაც შეიძლება მაგალითების საშუალებით და შესაძლებელია ინფორმაციის შემოწმება მარტივ, ელემენტარულ ფაქტებზე დაყრდნობით. ვინაიდან მოდელი გამოსახულია ისეთ

ბუნებრივ ტერმინებში, როგორცაა ობიექტი და როლი, იგი უზრუნველყოფს მოდელირების კონცეპტუალურ მიდგომას.

ობიექტ-როლური მოდელირების ადრეული ვერსია 1970-იან წლებში გამოჩნდა ევროპაში. აქ აღწერილი მეთოდი დამუშავებულია ავსტრალიასა და ა.შ.შ.-ში. ასოციაციური ენა FORML (Formal Object-Role Modeling Language) თანმხლება Microsoft Visio for Enterprise Architect (VEA)-ის, რომელიც Visual Studio.NET-ის შემადგენელი ნაწილია.

კონცეპტუალური მოდელირება მიიღწევა არსთა დამოკიდებულების (ER) მოდელირებას, თუმცა (ER) მოდელი შეიძლება გამოვიყენოთ მას შემდეგ, რაც დაპროექტების პროცესი დამთავრებულია. იგი ნაკლებადაა შესაფერისი ფორმულირებისათვის, განალებასა და პროექტის შემდგომი გაფართოებისათვის. ER-დიაგრამა შორსაა ბუნებრივი ენისაგან, ვერ ხერხდება შევსება ამა თუ იმ მოვლენის ფაქტით, დამალულია ინფორმაცია იმ სემანტიკური დომენების შესახებ, რომლებიც ქმნიან მოდელს.

ამრიგად, კონცეპტუალური მოდელირების განვითარებულ ტექნიკას წარმოადგენს ობიექტ-როლური მოდელირება. სწორედ ORM-ს შეუძლია უზრუნველყოს სხვადასხვა პროფესიის ადამიანების შეთანხმებული მუშაობა, რომელთა მომზადების დონე ინფორმაციული სისტემების დაპროექტების სფეროში შეიძლება მნიშვნელოვად განსხვავდებოდეს.

ზემოთ აღწერილი მიზეზებიდან გამომდინარე კონცეპტუალური მოდელირებისათვის ჩვენ ვირჩევთ ORM-ს საინფორმაციო სისტემების ცხოვრების ციკლი მოიცავს რამოდენიმე სტადიას: ტექნიკურ-ეკონომიკური დასაბუთება, მოთხოვნათა ანალიზი, მონაცემებისა და ოპერაციების კონცეპტუალური დაპროექტება; ლოგიკური დაპროექტება; გარე დაპროექტება;

მაკეტირება; შიდა დაპროექტება და შესრულება; ტესტირება და შესწორების შეტანა; მომსახურება (თანხლება).

ORM-ის კონცეპტუალური მოდელირების სქემის პროცედურა (CSDP) ყურადღებას ამახვილებს მონაცემების ანალიზზე და დაპროექტებაზე. კონცეპტუალური სქემა აღწერს აპლიკაციის ინფორმაციულ სტრუქტურას: ფაქტების ტიპები, რომლებიც წარმოადგენენ ინტერესის სფეროს; მასზე არსებული შეზღუდვები და შესაძლოა წარმოქმნის წესები, რათა მივიღოთ ესა თუ ის ფაქტი სხვა ფაქტებიდან.

### 1.8. ამოცანის დასმა

კომპიუტერული და საინფორმაციო ტექნოლოგიების განვითარების თანამედროვე ეტაპზე განსაკუთრებით აქტუალურია სამეცნიერო და საინჟინრო სამუშაოების შესრულება განაწილებული მართვის ავტომატიზებული სისტემების დასაპროექტებლად და სარეალიზაციოდ ობიექტ-ორიენტირებული მიდგომის საფუძველზე. უნიფიცირებული მოდელირების ენის (UML) ინსტრუმენტების გამოყენებით მიიღწევა დაპროექტებისა და დაპროგრამების ურთულეს პროცესთა ავტომატიზაცია, რაც საგრძნობლად ამცირებს საპროექტო-საინჟინრო ეტაპების შესრულების დროს და ამაღლებს საბოლოო პროდუქტის, მართვის სისტემის ხარისხს. განაწილებული ოფის-სისტემები კომპლექსური და მეტად მნიშვნელოვანი ობიექტებია მართვის პროცესების შემდგომი სრულყოფის თვალსაზრისით ახალი საინფორმაციო ტექნოლოგიების ბაზაზე [2 3].

მაგალითად, ჩვენი ქვეყნის ბიზნესის სხვადასხვა სფეროს ობიექტების მათ შორის ენერგო ობიექტების მართვის ან ერთიანი განათლების სისტემის შექმნის (ეროვნული გამოცდები, უნივერსიტეტები და ა.შ.) პროცესების ავტომატიზაცია, თავისთავად

მოითხოვს ცენტრალიზებული მონაცემთა საცავის შექმნას, რომელიც ლოგიკურად გააერთიანებს ფიზიკურად განაწილებულ სტრუქტურულ ქვედანაყოფთა მონაცემთა ბაზებს.

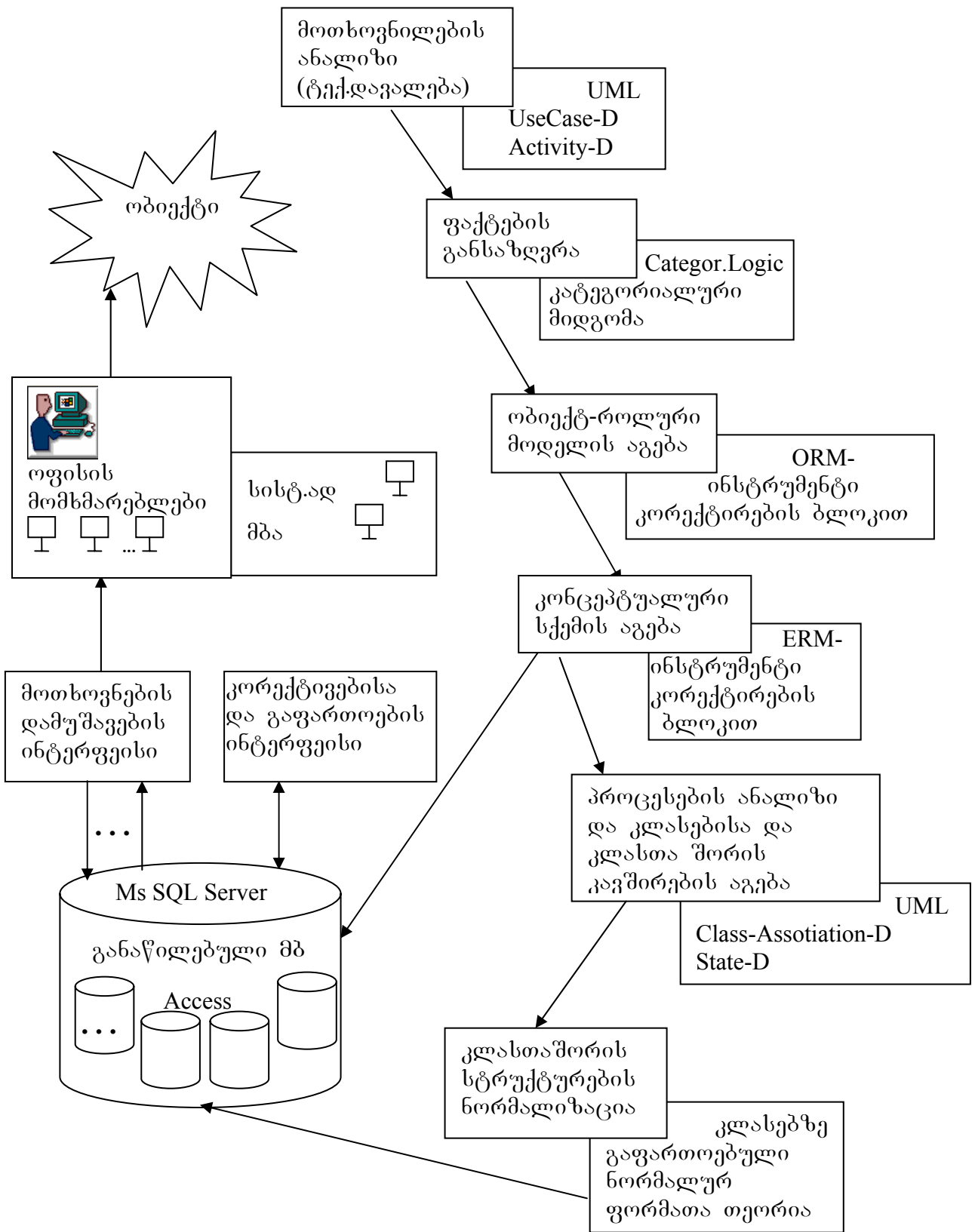
განაწილებული მართვის სისტემის საპრობლემო სფეროს კონტექსტში საჭიროა მონაცემთა რელაციური ბაზის დაპროექტება და აგება, რომელიც შეასრულებს ინფორმაციული უზრუნველყოფის ქვესისტემის ფუნქციას, საქმის წარმოების ავტომატიზაციისა და ელექტრონული დოკუმენტბრუნვის სრულყოფის მიზნით ინტერნეტ/ინტრანეტის პირობებში.

ინსტრუმენტის სახით გამოვიყენებთ უნიფიცირებულ მოდელირების ენის (UML) ერთ-ერთ გავრცელებულ, შედარებით ახალ პროგრამულ პაკეტს Ms Visio [16]. წინამდებარე ნაშრომის მიზანია ამ ინსტრუმენტების გამოყენებით მონაცემთა ბაზის კონცეპტუალური და ლოგიკური სტრუქტურების დამუშავება, შემდგომ კი მისი ფიზიკური ორგანიზაცია, რეალიზაცია მაგალითად, SQL Server მონაცემთა ბაზების მართვის სისტემაში. [10, 32, 53].

1.8 ნახაზზე მოცემულია იმ სამუშაო ეტაპების თანმიმდევრობა, რომელთა შესრულებაც აუცილებელია განაწილებულ ოფის-სისტემებში საინფორმაციო-ტექნოლოგიური პროცესების მოდელირებისათვის.

თავდაპირველად ხდება საპრობლემო არის მოთხოვნილებათა ანალიზი ანუ ტექნიკური დავალების განსაზღვრა, რომლის საფუძველზეც კატეგორიალური მიდგომის გამოყენებით ჩამოყალიბდება ფაქტები, არაფორმალიზებული ტექსტის სახით. ფაქტების საფუძველზე ობიექტ-როლური მოდელირების ინსტრუმენტის საშუალებით აიგება ORM-დიაგრამა. Microsoft Studio.Net საშუალებას გვაძლევს ORM-დიაგრამიდან ავტომატიზებულ რეჟიმში ავაგოთ არსთა-დამოკიდებულების მოდელი, ანუ ER-დიაგრამა (Entity Relationship Model). [51]

განაწილებულ ოვის-სისტემის საინფორმაციო-ტექნოლოგიური პროცესების მოდელირება



ნახ1.8



შემდეგ ეტაპზე დ-მოდელის საფუძველზე ავტომატიზებურად გენერირდება რელაციურ მონაცემთა ბაზების ლოგიკური სტრუქტურა, ანუ .DDL ფაილები (Data Definition Language). ამ ფაილებში პროგრამულად აღიწერება (ფორმალურ ენაზე) ობიექტები და მათი კავშირები. შემდეგ ეს .DDL ფაილები მიუერთდება (Attachment-ით) SQL-სერვერს [1]. მეორე მნიშვნელოვანი ამოცანა, რომელიც უნდა გადაწყდეს დისერტაციაში, არის კლასებზე გაფართოებული ნორმალურ ფორმათა დაპროექტების ალგორითმული სქემის შემუშავება და მისი პროგრამული რეალიზაცია.

### I თავის დასკვნები

1. პირველ თავში გაანალიზებულია ოფის-ობიექტების დოკუმენტების წარმოებისა და დოკუმენტბრუნვის საკითხები მათი შემდგომი სრულყოფის თვალსაზრისით. მიზანშეწონილია თანამედროვე საინფორმაციო ტექნოლოგიების გამოყენება.
2. ჩატარებულია ენერგეტიკული ოფის-სისტემების კლასიფიკაცია მართვის ამოცანებისა და ინფორმაციული უზრუნველყოფის აგების საკითხების გადაწყვეტის თვალსაზრისით.
3. დასმულია ამოცანა განაწილებული ოფის-ობიექტის საპრობლემო სფეროს ობიექტ-როლური მოდელის ასაგებად კატეგორიული მიდგომის გამოყენებით, რომელიც უზრუნ-ველყოფს სემანტიკურად გაფართოებული რელაციური სტრუქტურის შექმნას. ORM-მოდელის საფუძველზე განხორციელებულია არსთა დამოკიდებულების სქემის (ER-მოდელის) აგების პროცესების ავტომატიზაცია.
4. ჩამოყალიბებულია ამოცანა კლასთა ასოციაციების სტრუქტურული სქემის ნორმალიზაციისათვის, რომელიც ეფუძნება მონაცემთა ბაზების ნორმალურ ფორმათა თეორიას ობიექტ-ორიენტირებული კონცეფციის საფუძველზე;

## II ტავი

### ბიზნეს-პროცესების მოდელირების და დაპროექტების ტექნოლოგია განაწილებულ ოფის-სისტემებში

#### 2.1. ღოკუმენტების ავტომატიზებული ფორმირების ანალიტიკური მოდელი

ანალიზის შედეგად ოფის-ობიექტებში გამოიკვეთა შემდეგი ძირითადი ობიექტები: პერსონალი, თანამდებობა, ორგანიზაციული ერთეული, სტრატეგია, აპლიკაცია, ელექტრონული ფორმის შაბლონი, ელექტრონული ფორმის ეგზემპლარი, მოქმედების ფორმირება ნახ 2.1.

სტრატეგიის (Policy) ობიექტი წარმოადგენს მოდელის ძირითად რგოლს, მის ატრიბუტებს მიეკუთვნება დასახელება, რეალიზაციის თარიღი, მიზანი, პასუხისმგებელი, აღწერა და ტიპი. სტრატეგიის ობიექტები შეადგენს ორგანიზაციულ ერთეულს.[14]

თითოეული თანამდებობა მიმაგრებულია კონკრეტულ პირზე, თუმცა ზოგიერთმა თანამშრომელმა შეიძლება შეითავსოს რამდენიმე თანამდებობა. თითოეულ ორგანიზაციულ ერთეულს მართავს განსაზღვრული თანამშრომელი. ზოგმა თანამშრომელმა შეიძლება მართოს რამდენიმე ქვეგანყოფილება. ყველა თანამდებობა პრეზიდენტის გარდა, პასუხს აგებს მეორის წინაშე. ორგანიზაციული ერთეულები, ზოგიერთი გამონაკლისის გარდა, შედის სხვა ორგანიზაციულ ერთეულებში.

სტრატეგია შეიძლება უზრუნველყოფილ იყოს ერთი ან რამდენიმე ელექტრონული ფორმის შაბლონით. ელექტრონული ფორმები გამოიყენება ერთი ან რამდენიმე ელექტრონული ფორმის ეგზემპლარის მისაღებად. სისტემა შეიძლება შეიცავდეს ისეთ რთულ ელექტრონულ ფორმებს, რომლებიც განკუთვნილია პერსონალის სწავლებისათვის. ელექტრონული ფორმა შეიძლება

შეიცავდეს ერთ ან რამდენიმე მოქმედებას, რომელიც უნდა შეასრულოს კონკრეტულმა თანამშრომელმა. ერთნაირი მოქმედებები შეიძლება ვრცელდებოდეს სხვადასხვა თანამდებობებზე ორგანიზაციის შიგნით (მაგალითად, ხარჯების უწყისის დამტკიცება).

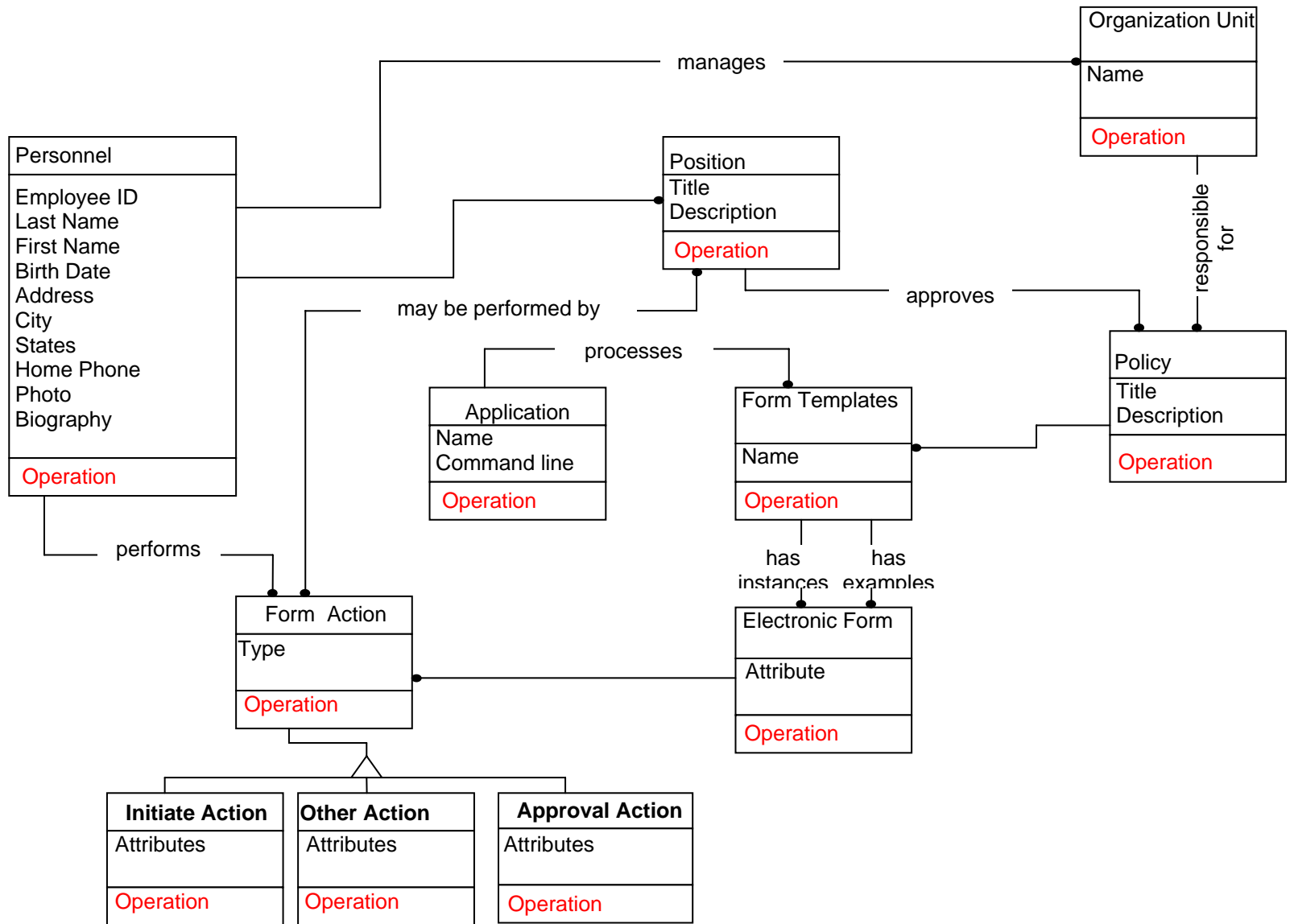
მაგალითის სახით განვიხილოთ სტრატეგიის ის ნაწილი, რომელიც ეხება ფინანსურ ანგარიშს. ხარჯების ანგარიში საბუქლალტრო დეპარტამენტის მოვალეობაა, რომელსაც მართავს მთავარი ბუქლალტერი. ფინანსური ანგარიშის ელექტრონული ფორმის მისაღებად აუცილებელია Excel-ის ელექტრონული ცხრილის გამოყენება. იხ.ნახ.2.2.

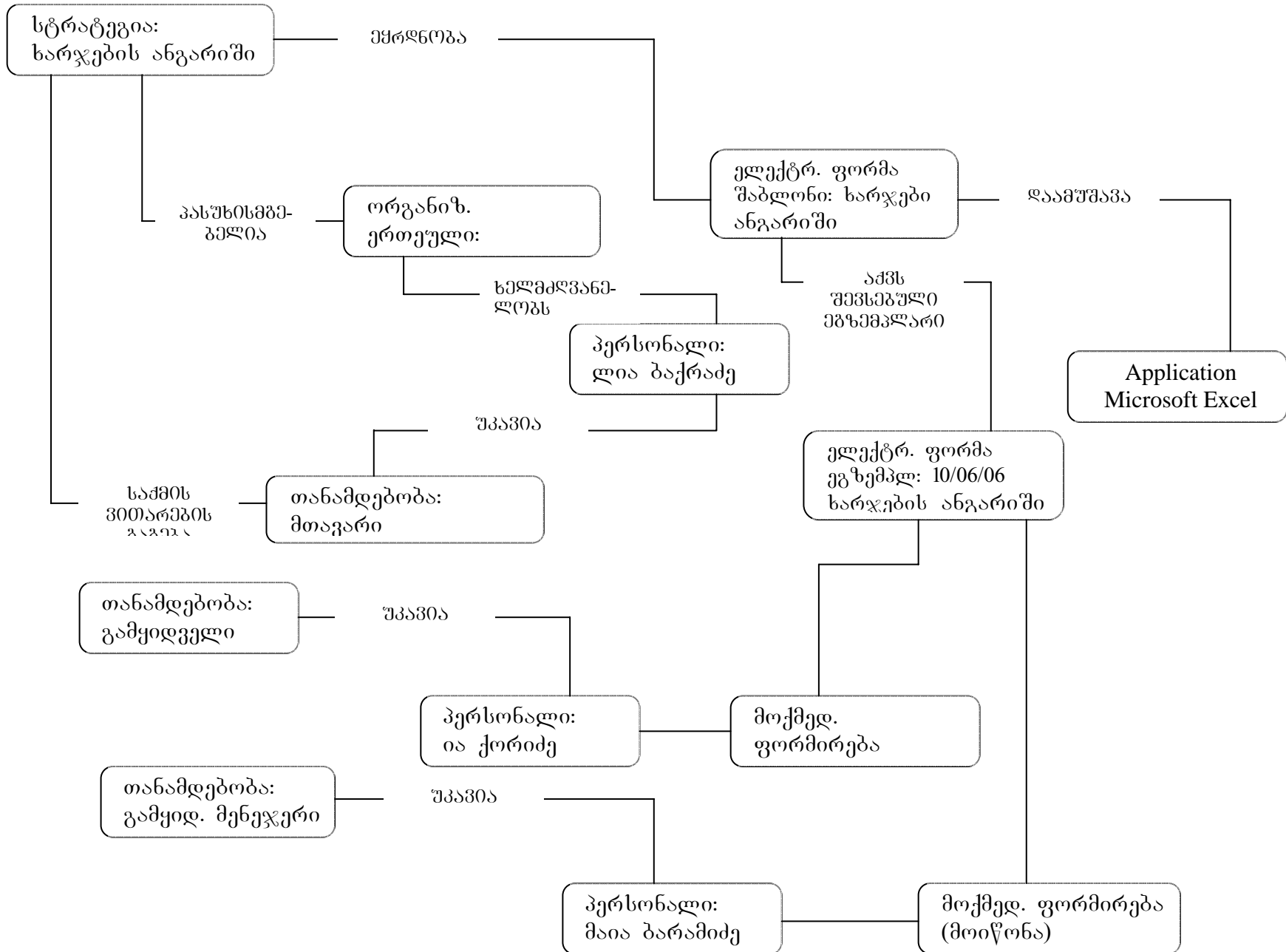
## 2.2. ბიზნეს-პროცესების ანალიზი და USE CASE და ACTIVITY დიაგრამების აბეზა

ორგანიზაციებში გარკვეული სამუშაოს ჩატარების მიზნით ტარდება ტენდერი. ტენდერს აცხადებს ორგანიზაცია, რომელსაც სურს ამა თუ იმ სამუშაოს შესრულება. ხოლო ის ორგანიზაცია, რომელსაც სურს ამ სამუშაოს შესრულება მონაწილეობას ღებულობს ტენდერში. ამიტომ, დღეს ძალიან აქტუალური გახდა ტენდერის ჩატარება ნებისმიერ სფეროში მშენებლობა იქნება ეს, სარემონტო სამუშაოები, მომარაგება, ამა თუ იმ პროდუქციის შესყიდვა და ა.შ.

ყველაზე მეტად, ასეთი საქმიანობით, სხვადასხვა ტიპის ოფისები არიან დაინტერესებულნი. ტენდერში მონაწილეობის მიღების ის პროცესი ასე შეიძლება აღვწეროთ:

„ოფისის ხელმძღვანელი ან ხელმძღვანელთა საბჭო, გაეცნობა რა ტენდერის მოთხოვნებს, ანალიზისა და მსჯელობის საფუძველზე გადაწყვიტავს, მიიღოს თუ არა ტენდერში მონაწილეობა. დადებითი





გადაწყვეტილების შემთხვევაში ხდება წინადადებების შემუშავება და ბრძანების გაცემა, რათა მომზადდეს მასალები ტენდერში მონაწილეობისთვის. ამ მომენტიდან ტარდება შემდეგი სამუშაოები: ოპერატორი გაეცნობა რა ტენდერის მოთხოვნებს, ნახავს ხომ არ აქვს მზა (რომელიც შეეფერება ტენდერის მოთხოვნებს) ფორმა (შაბლონი) შენახული მონაცემთა ბაზაში, თუ აქვს გამოიძახებს მას და უბრალოდ შეავსებს. თუ არ აქვს, იწყებს ახალი ფორმის მომზადებას და ტექსტის შეტანს. როგორც წესი, ტენდერის მოთხოვნებში ძირითადად თხოულობენ შემდეგი სახის ინფორმაციას:

- ორგანიზაციის სტრუქტურა;
- ფირმის რეკვიზიტები;
- კომპანიის კორპორაციული სტრუქტურა;
- თანამშრომელთა CV;
- საქმიანობა ბოლო 3 წლის განმავლობაში;
- ამონაწერი სამეწარმეო რეესტრიდან ფირმის რეგისტრაციის შესახებ;
- ამონაწერი სასამართლოდა, რომ იურიდიული პირის მიმართ არ მიმდინარეობს გაკოტრების ან სანაციის საქმის წარმოება;
- რომ არ ხორციელდება იურიდიული პირის რეორგანიზაცია ან ლიკვიდაცია;
- ცნობა საჯარო რეესტრიდან, რომ ორგანიზაციის ქონებაზე არ არის დადებული ყადაღა;
- ცნობა საგადასახადო ინსპექციიდან, რომ ბიუჯეტის წინაშე არ გააჩნია დავალიანება.

ამ ცნობების შეგროვების მიზნით კანცელარიაში მზადდება წერილები სათანადო ორგანოებში, საიდანაც შემდგომ მიიღება პასუხები.

ტენდერის გამომცხადებელი ორგანიზაცია ყოველთვის თხოულობს სატენდერო მოსაკრებლის გადახდას. ამიტომ ბუღალტერი გადარიცხავს მოთხოვნილ თანხას ტენდერის მოთხოვნებში მითითებულ ბანკის ანგარიშზე და იღებს ქვითარს გადარიცხვის შესახებ.

კანცელარიაში წერილების მოსვლის შემდეგ, ოპერატორი (ან სხვა პირი) მოახდენს ამ მასალების დაკომპლექტებას და გადასცემს დირექტორს. დირექტორი, თავის მხრივ, შეასწორებს მათ და

შეცდომების აღმოჩენის შემთხვევაში, გადასცემს მათ კორექტურაზე. შეცდომების არარსებობის შემთხვევაში აწერს ხელს და გადასცემს კანცელარიას, სადაც არტყამენ ბეჭედს.

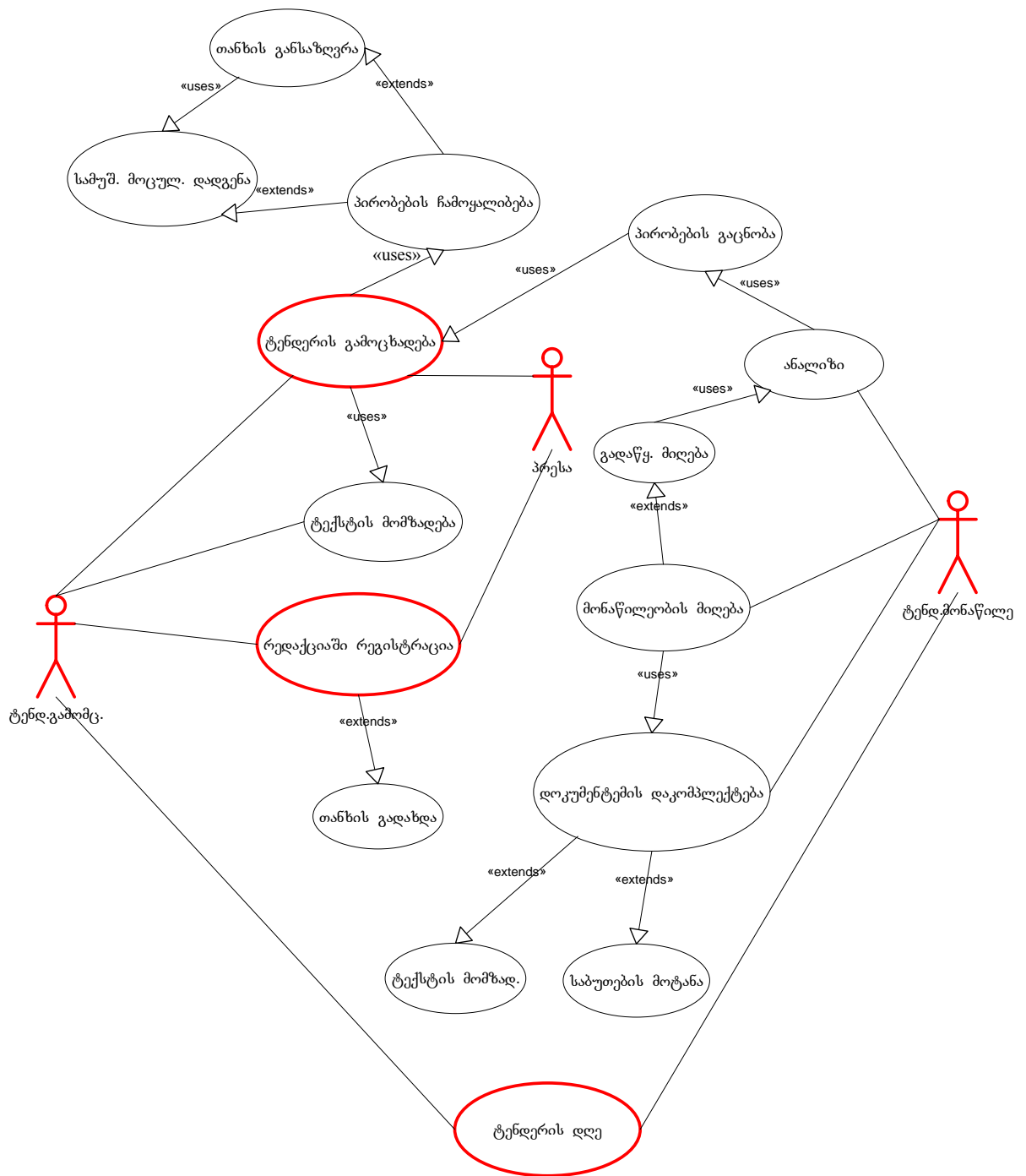
შემდგომ ზემოჩამოთვლილი დოკუმენტები თავსდება კონვერტში, ილუქება და ხდება გადატანა ტენდერის მოთხოვნებში მითითებულ მისამართზე.

ბოლოს მონაცემთა ბაზების ადმინისტრატორი ახდენს დოკუმენტების სკანირებას, ფაილების დაარქივებას და სერვერ-მანქანაში ჩაწერას.

ტენდერის მოთხოვნებში მითითებულია თარიღი და დრო, როდესაც გაიმართება ტენდერი. ამ დღეს, ტენდერში მონაწილეობის მიღების მსურველი ყველა ორგანიზაციის წარმომადგენელი, ვაღდებულება გამოცხადდეს მითითებულ მისამართზე. ყველას თვალწინ, სატენდერო კომისიის წევრები, გახსნიან დალუქულ კონვერტებს, გაეცნობიან შემოთავაზებულ წინადადებებს, გადაამოწმებენ საბუთებს, მსჯელობის შემდეგ ამოირჩევენ საუკეთესო წინადადებას და შემდგომ დაასახელებენ ტენდერში გამარჯვებულ ორგანიზაციას.

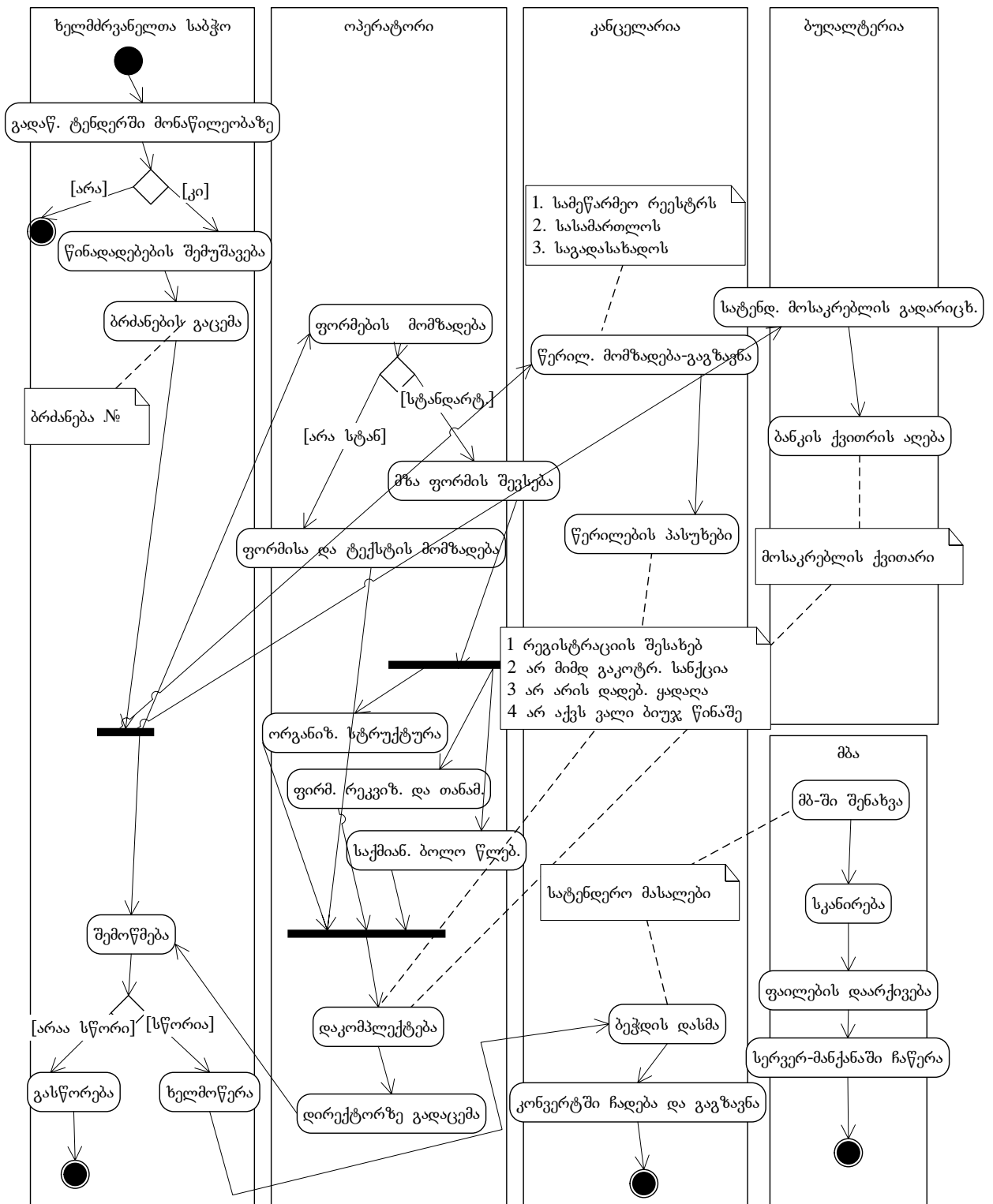
შემდგომ ხდება კონტრაქტის დადება, სადაც დაწვრილებით ჩაიწერება შესასრულებელი სამუშაოს ყველა დეტალი. მაგალითად, თუ სამუშაო ეხება ამა თუ იმ პროდუქციის შესყიდვას, საჭიროა მოდელის ნომრის ამორჩევა პროდუქციის კატალოგიდან, მიწოდების მოცულობის და ვადების დადგენა. აუცილებლობის შემთხვევაში დაკავშირება დამამზადებელთან და მომმარაგებელთან, კონტრაქტის დამოწმება ხელმოწერითა და ბეჭდით.

ნახ 2.3 და ნახ 2.4 გამოსახულია ტენდერში მონაწილეობის მიღების UseCase და Activity დიაგრამები.



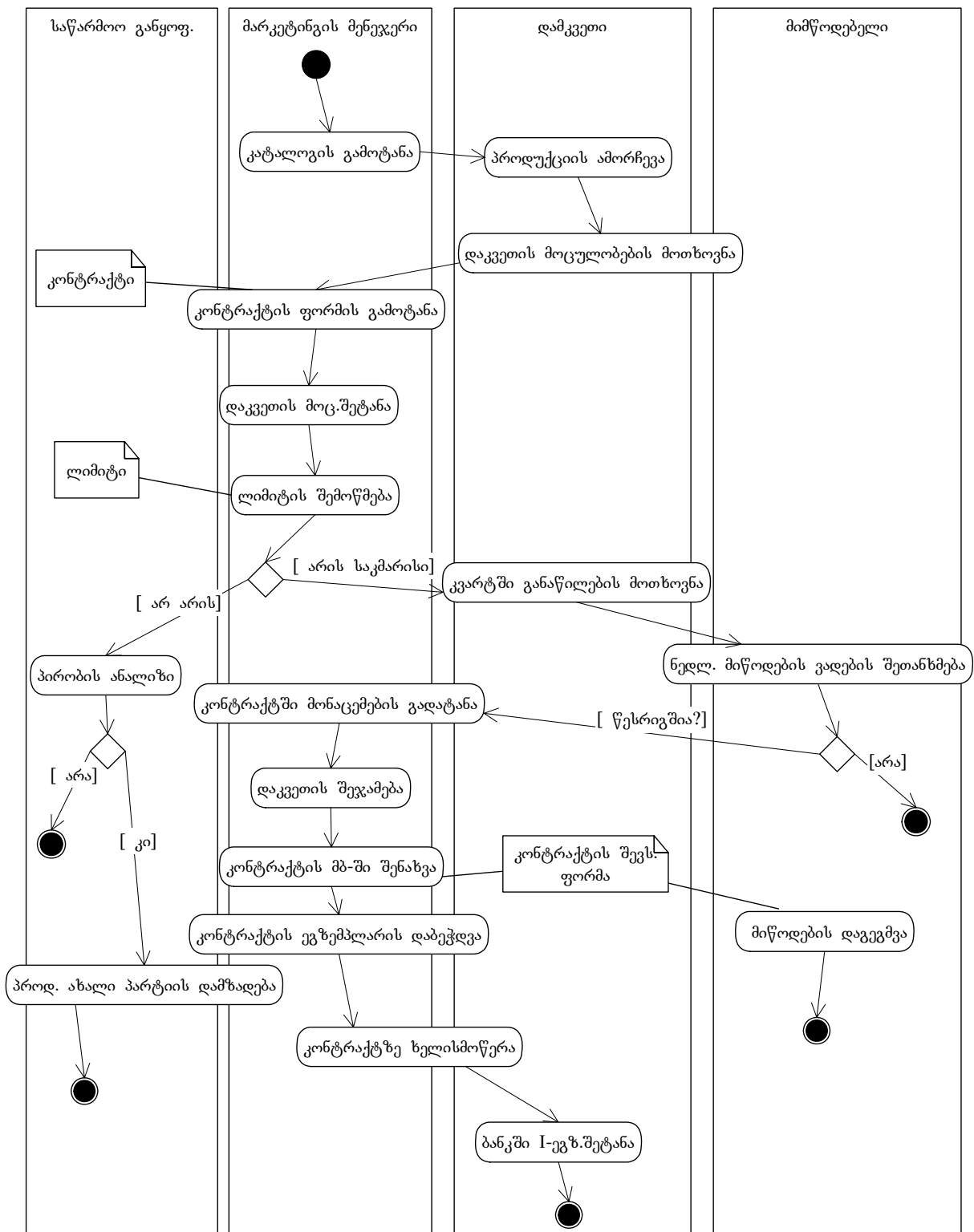
ნახ. 2.3 UseCase-დიაგრამა „ტენდერის ჩატარება“





ნახ. 2.4 Activity-დიაგრამა „ტენდერის მასალების მომზადება“

ნახ 2.5 მოცემულია კონტრაქტის გაფორმების Activity-დიაგრამები დაწვრილებით აღწერს ამ პროცესს.



ნახ. 2.5 კონტრაქტის გაფორმების Activity -დიაგრამა

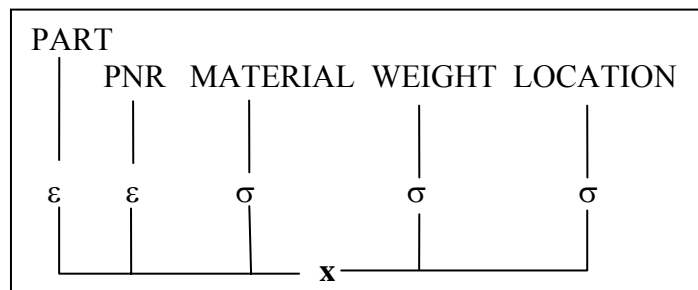
2.3. ბიზნეს-პროცესების საპრობლემო სფეროს კონცეპტუალური მოდელის აგება კატეგორიული მიდგომით

საგნობრივი სფეროს მოდელირებისას კატეგორიული მიდგომის საფუძველზე საჭიროა ძირითადი და დამხმარე პრედიკატორების გამოვლენა. მათი ასეთი დაყოფა დამოკიდებულია ამოცანის დასმაზე ტექნიკურ დავალებაში. ზოგადად, მეთოდულაში ჯერ მთავარ, შემდეგ კი დამხმარე პრედიკატორებს გამოყოფენ. ეს უკანასკნელი მოდელში აღწერს ობიექტის არამთავარ, ანუ დამატებით მახასიათებლებს.

მაგალითად, წინადადებაში „დეტალს აქვს ხის-ნაწილი“, შემოიტანება ახალი  $\sigma$ -კოპულა, რომელიც „აქვს“ ოპერაციას ასახავს. ამგვარად, გვექნება:

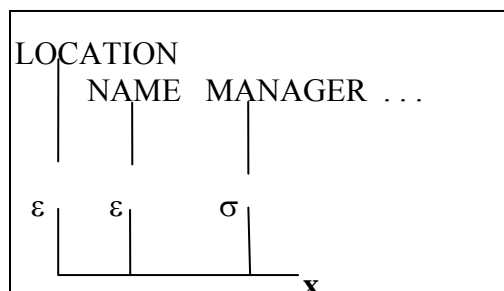
**„დეტალს  $\sigma$  ხის-ნაწილი“.**

პარაგრაფში განხილული სქემა (ნახ2.6) დამატებითი პრედიკატორებით გაფართოების საფუძველზე შეიძლება ასე წარმოვადგინოთ:



ნახ.2.6

სადაც დამხმარე პრედიკატორებია: MATERIAL-მასალა, WEIGHT წონა და LOCATION –საწყობის მდებარეობა. თუ საწყობს განვიხილავთ როგორც მთავარ პრედიკატორს, მაშინ მისთვის სხვა დამხმარე პრედიკატორები გაჩნდება, მაგ., ნახაზი 2.7.



ნახ.2.7

დამხმარე პრედიკატორების საშუალებით პრედიკატული სქემის გაფართოების პროცესს ლიტერატურაში მოიხსენიებენ აგრეთვე რელაციური სქემის სახით. ესაა მონაცემთა რელაციური ბაზის კონცეპტუალური სქემის საფუძველი. დამხმარე პრედიკატორების გამოყენებას კავშირი აქვს მონაცემთა ნორმალიზაციის თეორიასთან, კერძოდ ესაა წესების სისტემა დამხმარე პრედიკატორების შედგენილობის განსაზღვრისათვის.

რელაცია, ჩვეულებისამებრ თეორიაში განიხილება როგორც ატრიბუტთა (დამხმარე კატეგორიების) დეკარტული ნამრავლის ქვესიმრავლე:

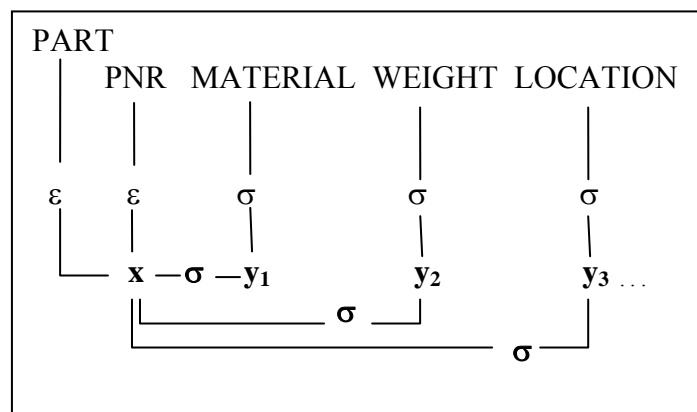
$$PART \subseteq PNR \times MATERIAL \times WEIGHT \times LOCATION$$

რელაცია – მრავალადგილიანი სიმრავლეა, რომელიც აბსტრაქციის საფუძველზე ობიექტის მდგომარეობათა მოწესრიგებულ-კორტეჟთა ერთობლიობით გამოისახება (ნახ.2.8).

PART

PNR	MATERIAL	WEIGHT	LOCATION
123	ფოლადი	24,5	თბილისი
124	ხე	10,25	ბათუმი
135	პლასტმასი	27,55	თბილისი
...	...	...	...

შეიძლება ჩაიწეროს: „ხე  $\in$  MATERIAL“, „10,25  $\in$  WEIGHT“ და ა.შ. დამხმარე პრედიკატორების შიგნით, იერარქიული თვალსაზრისით, ჩნდება მთავარი პრედიკატორები (ნახ.2.9):



ნახ.2.9

ასეთი სქემა აღარაა ელემენტარული. ზოგიერთი დამხმარე პრედიკატორი ლოგიკურად შეიძლება კონიუნქციის სახით ჩაიწეროს:

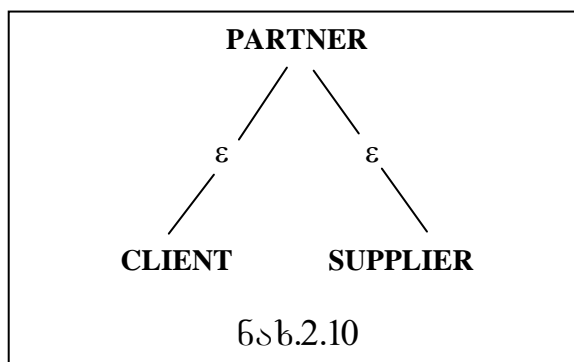
$$x \in (y_1)(\text{PART}) \wedge x \in (y_2)(\text{PART}).$$

საგნობრივი სფეროს აღწერისას ხდება მისი ობიექტების კლასიფიკაცია, მათი დაყოფა და მოწესრიგება (კლასი, ქვეკლასი და ა.შ.). ანუ ხდება გარკვეული სუბორდინაცია, კონკრეტულიდან აბსტრაქტულზე ასვლა გარკვეული მემკვიდრეობითი თვისებების საფუძველზე. ამ თვალსაზრისით განიხილება სახეობათა (ქვეკლასის) პრედიკატორები და კლასის (სუპერკლასის) პრედიკატორები.

მაგალითად, საპრობლემო სფეროსთვის „პროდუქციის წარმოება“ არსები კლიენტი (პროდუქციის მყიდველი - CLIENT) და მიმწოდებელი (ნედლეულის მიმწოდებელი- SUPPLIER) შეიძლება შეესაბამებოდეს სახეობათა პრედიკატორებს, მაშინ მათთვის უფრო განზოგადოებული ცნება წარმოების-პარტნიორი (PARTNER) იქნება კლასის პრედიკატორი, რომელიც ორივეს მოიცავს (ნახ.2.10).

$$\wedge_x x \in \text{CLIENT} \rightarrow x \in \text{PARTNER}$$

$$\wedge_x x \in \text{SUPPLIER} \rightarrow x \in \text{PARTNER}$$



ახალი ცნებების შემოტანა ასევე ლოგიკური კავშირების გამოყენებითაა შესაძლებელი. მაგალითად, კლიენტი ანუ დამკვეთი არის ამავე დროს მიმწოდებელიც. ეს ნიშნავს, რომ პროდუქციის

მწარმოებელი ფირმა, ერთის მხრივ უკვეთავს ნედლეულს სხვა ფირმას (არის დამკვეთი) და მეორეს მხრივ, ის ამზადებს პროდუქციას და მიაწვდის კლიენტს (არის მიმწოდებელი). ასეთი წესები შეიძლება ლოგიკურად ჩავწეროთ შემდეგნაირად:

$$\wedge_x x \in C \vee x \in P \rightarrow x \in S,$$

$$\wedge_x x \in C \wedge x \in P \rightarrow x \in S,$$

სადაც  $\vee$  - ლოგიკური დიზიუნქცია, ხოლო  $\wedge$  - ლოგიკური კონიუნქციაა.

ამგვარად, აბსტრაქცია ახალი ობიექტების შემოტანის ერთ-ერთი გზაა კლასიფიკაციის სისტემაში, რომლის საფუძველსაც მემკვიდრეობითობის თვისებები წარმოადგენს კლასებს შორის.

გარდა ამისა არსებობს ობიექტების და კლასთაშორის კავშირების ასახვის მეორე გზაც, რომელსაც მერეოლოგიურს, ანუ „მთელი-ნაწილი“-დამოკიდებულების ლოგიკას უწოდებენ. აქ ყურადღება მახვილდება სტრუქტურულ შედგენილობაზე, ანუ აგრეგაციაზე. მაგალითად, პროდუქცია შედგება სხვადასხვა ნაწილისგან.

„მთელი-ნაწილი“ - დამოკიდებულებათა ლოგიკის ფუძემდებელია პოლონელი მათემატიკოსი ლეზნიევსკი (1916 წ.). პრედიკატულ-აბსტრაქტული ლოგიკისგან განსხვავებით, რომელიც მაგალითად, რასელის „მათემატიკის პრინციპებიდანაა“ ცნობილი, მერეოლოგიური ენა „მთელი-ნაწილი“-რელაციასთან ( $\leq$ ) ერთად ცნობს მხოლოდ კატეგორიებს: „სახელი“ და „გამონათქვამი“.

მთელი/ნაწილი - რელაცია ( $x \leq y$ ,  $x$  არის  $y$ -ის ნაწილი) სამართლიანია მხოლოდ ლოგიკური ტიპისათვის, როგორც ნაწილისთვის და განსხვავდება ელემენტი/სიმრავლე - რელაციისაგან  $\in$ , რომელიც პრედიკაციასა და აბსტრაქციას

ითვალისწინებს. ობიექტებისათვის არსებობს სახელები, რომლებიც როგორც კონსტანტები ან ცვლადებია შედგენილ ობიექტებში.

მაგალითად, გამონათქვამი „123 ∈ CLIENT“ (123 არის კლიენტი) შეიცვლება მერეოლოგიური თვალსაზრისით გამონათქვამით „123 ∈ CUSTOMERS“, „123 არის ნაწილი კლიენტურისა“. აქ „კლიენტურა“ შედგენილი ნაწილების სახელია.

$$\text{CUSTOMERS} = t_z \wedge_x x \leq z.$$

ყველა  $x$ , რომლებიც ერთტიპური  $z$ -ის ნაწილია, არის CUSTOMERS.

ჩვენ განვიხილეთ ორი ლოგიკა: პედიკატულ-აბსტრაქტული და მერეოლოგიური.

პედიკატულ-აბსტრაქტული ლოგიკა ცნობს ორი სახის ობიექტებს, კონკრეტულებს და აბსტრაქტულებს. მას ასევე რეალისტურ ლოგიკასაც უწოდებენ.

მერეოლოგია არის ნომინალიზმის ლოგიკა (ფილოსოფიური მიმართულება, რომელიც უარყოფს ზოგადი ცნებების არსებობას). მისი კონცეფციით არსებობს მხოლოდ კონკრეტული ობიექტები და მათი სტრუქტურები. მათ შეესაბამება სახელები და გამონათქვამები.

ინფორმატიკოსებისათვის ელემენტარული გამოსახულებანი:

$$x \in P \text{ და } x \leq P$$

უნდა ჩავთვალოთ დუალურად.

ნომინალისტ-ინფორმატიკოსები სააზროვნოდ იყენებენ ხისმაგვარ სტრუქტურას, რომელიც  $x$ -შვილი- და  $y$ -მშობელი კვანძებისგან შედგება, სადაც სამართლიანია:  $x \leq y$ .

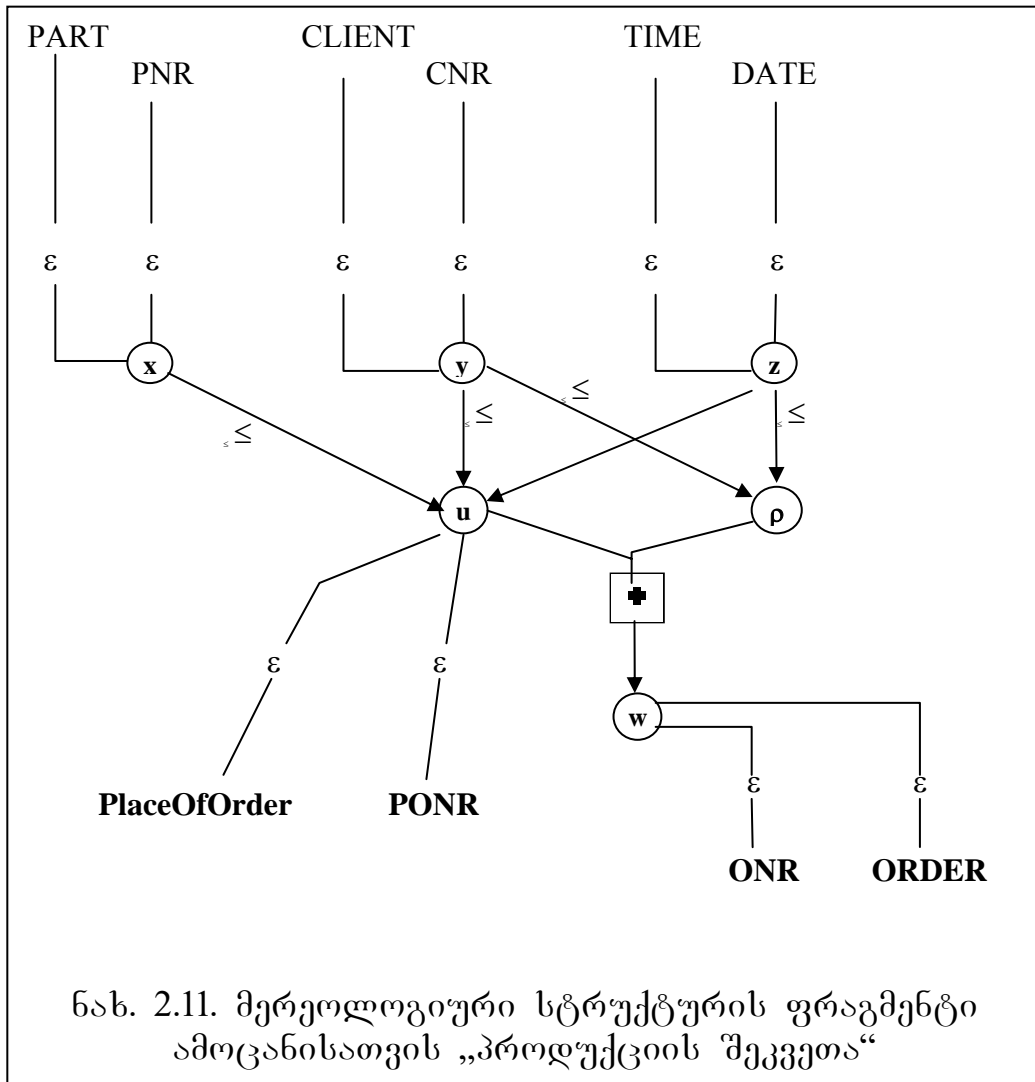
$\leq$ -თვის ასევე სამართლიანია შემდეგი წესები:

- რეფლექსურობის:  $\wedge_x x \leq x$ .
- ტრანზიტულობის:  $\wedge_{xy} (x \leq y \wedge y \leq z \rightarrow x \leq z)$

- <საკუთარი სახელი> =  $k_x A(x)$ . მთელი  $x$ , რომელიც აკმაყოფილებს  $A(x)$ -გამონათქვამს.

2.11 ნახაზზე ნაჩვენებია მერეოლოგიური სტრუქტურის მაგალითი პროდუქციის შეკვეთების ამოცანისათვის.

ორი  $u$  და  $\rho$  ობიექტის წარმოებული ( $u * \rho$ ) არის ახალი ობიექტი  $w$ , რომელსაც იგივე შედგენილობა აქვს, რაც  $u$ -ს და  $\rho$ -ს ერთად.



R-შეკვეთისათვის ლოგიკურად შეიძლება დაიწეროს:

$$R = t_p \vee_{y,z} y \leq \rho \wedge z \leq \rho.$$

ფაქტიური შეკვეთისათვის  $u * \rho$ - წარმოებულისთვის იქნება:

$$u * \rho = t_w \wedge_r (r \leq w) \leftrightarrow (r \leq u \wedge r \leq \rho).$$



დამხმარე ცვლადი  $\rho$  ჩვენს შემთხვევაში აირჩევა ისე, რომ  $r$ -თავისუფალ ცვლადებს შეეძლოს ობიექტების მიღება მხოლოდ  $y$  და  $z$ -სგან.  $w$ -შეკვეთას და  $u$ -შეკვეთის ფოსტას აქვს ერთნაირი CNR ( $y$ ) და DATE ( $z$ ). საკუთარი სახელი  $W$  შეკვეთისათვის მაშინ მოიცემა, როდესაც  $u * \rho$  პროდუქცია არსებობს.

$$W = V_{u,\rho} u * \rho.$$

#### 2.4 ORM-დიაგრამების აბეზა ბანაწილებული ოფის-სისტემებისთვის.

ზემოთ აღწერილი მიზეზებიდან გამომდინარე კონცეპტუალური მოდელირებისათვის ჩვენ ვირჩევთ ORM-ს. საინფორმაციო სისტემების ცხოვრების ციკლი მოიცავს რამდენიმე სტადიას: ტექნიკურ-ეკონომიკური დასაბუთება, მოთხოვნათა ანალიზი, მონაცემებისა და ოპერაციების კონცეპტუალური დაპროექტება; ლოგიკური დაპროექტება; გარე დაპროექტება; მაკეტირება; შიდა დაპროექტება და შესრულება; ტესტირება და შესწორების შეტანა; მომსახურება (თანხლება) [17].

ORM-ის კონცეპტუალური მოდელირების სქემის პროცედურა ანუ CSDP ყურადღებას ამახვილებს მონაცემების ანალიზზე და დაპროექტებაზე [18]. კონცეპტუალური სქემა აღწერს აპლიკაციის ინფორმაციულ სტრუქტურას: ფაქტების ტიპები, რომლებიც წარმოადგეს ინტერესის სფეროს; მასზე არსებული შეზღუდვები და შესაძლოა წარმოქმნის წესები, რათა მივიღოთ ესა თუ ის ფაქტი სხვა ფაქტებიდან.

მრავალმასშტაბიანი აპლიკაციისათვის Uod-ი ყოფს მას მოხერხებულ მოდულებად, CSDP-ს მიმართავს თითოეული მათგანი და შემდეგ ხდება მიღებული კონცეპტუალური ქვესქემების გაერთიანება ერთ გლობალურ კონცეპტუალურ სქემად. თვითონ CSDP-ი შედგება შვიდი ბიჯისაგან. იხ.ცხრილი 2.1

1. ელემენტარული ფაქტების ფორმირება და მათი ადექვატურობის შემოწმება;
2. ფაქტების ტიპებისათვის დიაგრამის აგება და სისრულის შემოწმება;
3. იმ ობიექტთა ტიპების შემოწმება, რომლებიც უნდა გაერთანდეს და მათი მათემატიკური წარმომავლობის დაფიქსირება;
4. დაემატოს უნიკალურობის შეზღუდვა და შემოწმდეს ფაქტების ტიპების ოპერანდების რაოდენობა;
5. დაემატოს როლების იძულებითი შეზღუდვები და შემოწმდეს მათი ლოგიკური წარმომავლობა;
6. დაემატოს ელემენტები, სიმრავლეთა შედარება და ქვეტიპის შეზღუდვები;
7. დაემატოს სხვა შეზღუდვები და მოხდეს საბოლოო შემოწმება.

ბიჯი 1 – CSDP-ის ყველაზე მნიშვნელოვანი სტადიაა, სადაც ხდება სხვადასხვა სახის ინფორმაციის შეგროვება, ბუნებრივ სალაპარაკო ენაზე. ასეთი ინფორმაცია ხშირად არის ხოლმე შემაგალი და გამომავალი ფორმები, შეიძლება იყოს ხელნაწერი. წინააღმდეგ შემთხვევაში მოდელის დამპროექტებელი მუშაობს უშუალოდ კლიენტთან, რათა ზუსტად ჩამოყალიბდეს, თუ რა მოეთხოვება სისტემას იმისათვის, რომ ადგილი არ ჰქონდეს გაუგებრობას. აუცილებელია UoD ექსპერტის (ადამიანი, რომელიც იცნობს აპლიკაციას) არსებობა.

ეს ფაქტი ასე შეიძლება ჩაიწეროს:

f1 თანამშრომელს ნომრით 35 აქვს თანამშ.გვარი ‘პეტრიაშილი’

f2 თანამშრომელი ნომრით 7 მუშაობს კონტრაქტით თარიღამდე ‘12.31.06’

თითოეული ფაქტი არის ბინარული დამოკიდებულება ორ ობიექტს შორის. მუქი შრიფტით გამოყოფილია ლოგიკური პრედიკატი, რომლებიც ახდენს ობიექტების იდენტიფიცირებას ნაჩვენებია კურსივით. იმ შემთხვევაში თუ განისაზღვრება ობიექტის

მხოლოდ ერთი თვისება, საქმე გვაქვს ერთადგილიან პრედიკატთან (unary fact). პრედიკატს შეიძლება ჰქონდეს (1,2,3,..) ოპერანდი, თუმცა რადგან პრედიკატი ელემენტარულია 3-4 ოპერანდზე მეტი იშვიათად გვხვდება. უმრავლეს შემთხვევაში

პრედიკატი არის ორობითი. ასეთი პრედიკატებისათვის არსებობს ინვერსული პრედიკატი. ისე, რომ ფაქტი შეიძლება წავიკითხოთ ორივე მიმართულებით.

ბიჯი 2 – აქ ხდება ფაქტების ტიპებისათვის დიაგრამის აგება. ობიექტები გამოისახება ელიფსებით, პრედიკატები მართკუთხედებით, მნიშვნელობის ტიპი გამოისახება წყვეტილი ელიფსით. პრედიკატი იკითხება მარცხნიდან-მარჯვნივ და ზემოდან-ქვემოთ მანამ, სანამ არ შეხვდება ნიშანი “<<”, რომელიც ცვლის წაკითხვის მიმართულებას საწინააღმდეგო მიმართულებით.

შემდეგ ბიჯებზე ხდება შეზღუდვების დაწესება.

### ORM დიაგრამაში გამოყენებული შეზღუდვები:

#### იძულების შეზღუდვები:

- ობიექტი ასრულებს ზუსტად გარკვეულ როლს.
- ⊙ როლების დიზუნქცია არის იძულებითი. თითოეული ორიექტი ობიექტების ტიპების ნაკრებიდან უნდა ასრულებდეს მხოლოდ ერთ როლს.

#### უნიკალურობის შეზღუდვები:

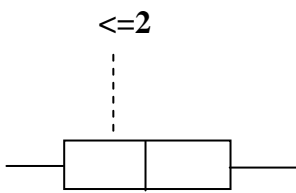
↔ ერთ ან მეტ როლში მონაწილეობა ხდება არა უმეტეს ერთხელ.

- ⊗ როლების გარე უნიკალურობის შეზღუდვა.  
წყვილის გამორიცხვის შეზღუდვა.

სიმრავლების შედარების შეზღუდვები:

- ⊆ პირველი ობიექტის სიმრავლე ყოველთვის უნდა იყოს მეორის ქვესიმრავლე.
- = პირველი ობიექტის სიმრავლე ყოველთვის უნდა იყოს მეორის ტოლი.
- ⊄ პირველი ობიექტის სიმრავლე არ შედის მეორეში.

სისშირის შეზღუდვა.



ობიექტმა რამდენჯერ შეიძლება შეასრულოს ეს როლი

ბუდის ტიპის ობიექტი.



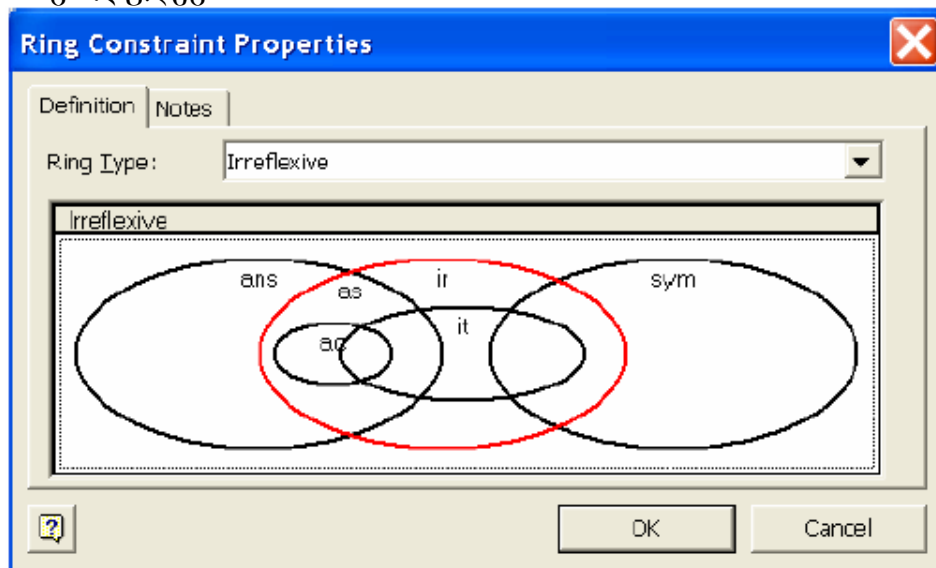
ობიექტი თამაშობს მხოლოდ ერთ როლს და ეს როლი არ არის სავალდებულო.



ქვეტიპი.

ერთი ობიექტი არის მეორის ქვეტიპი

წრიული შეზღუდვები



ნახ.2.12

ანტირეფლექსურობა	<i>Oir</i>	iff for all $x, \sim xRx$
სიმეტრიულობა	<i>Osym</i>	iff for all $x, y, xRy \rightarrow yRx$
ასიმეტრიულობა	<i>Oas</i>	iff for all $x, y, xRy \rightarrow \sim yRx$
ანტისიმეტრიულობა	<i>Oans</i>	iff for all $x, y, x \neq y \& xRy \rightarrow \sim yRx$
ანტიტრანზიტულობა	<i>Oit</i>	iff for all $x, y, z, xRy \& yRz \rightarrow \sim xRz$
აციკლურობა	<i>ac</i>	iff for all $x, y, z, xRy \& yRz \rightarrow \sim zRx$

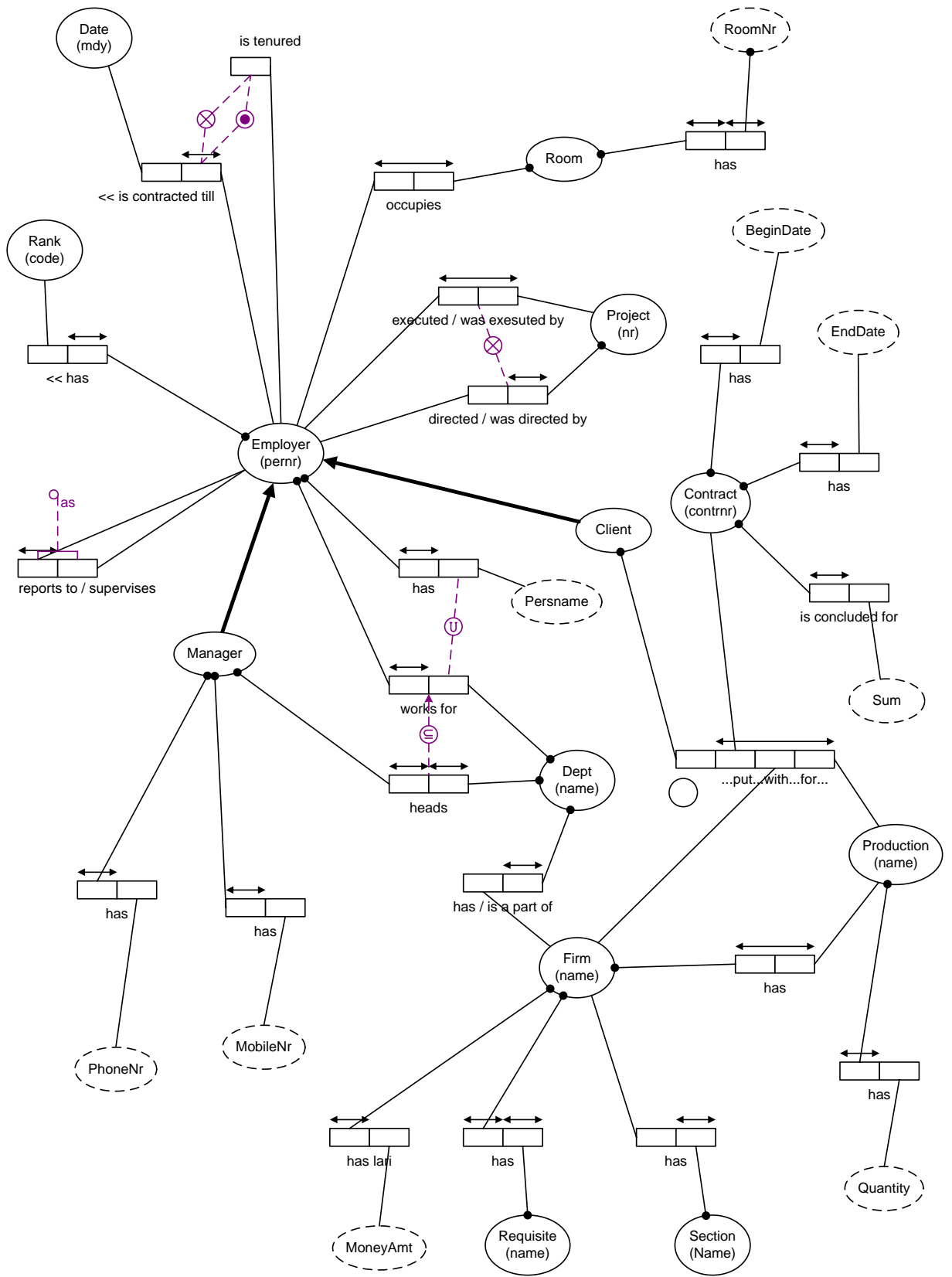
მას შემდეგ, რაც მოხდება ORM-დიაგრამის ყველა წესის დაცვით აგება, ავტომატიზებულად აიგება ER-მოდელი.

მაგალითის სახით, განვიხილოთ ზოგადად ოფის-ობიექტი.

ჩამოვწეროთ ფაქტები:

- f1 თანამშრომელს აქვს სახელი და გვარი.
- f2 თანამშრომელი მუშაობს განყოფილებაში.
- f3 თანამშრომელს დაკავებული აქვს ოთახი.
- f4 თითოეულ თანამშრომელს შეიძლება ჰქონდეს არა უმეტეს ერთი გვარი.
- f5 თითოეულ პროექტს ხელმძღვანელობს არა უმეტეს ერთი თანამშრომელი.
- f6 მენეჯერი ხელმძღვანელობს განყოფილებას.
- f7 მენეჯერს აქვს ქალაქის ტელეფონი და მობილური ტელეფონი.
- f8 ფირმა შედგება განყოფილებებისაგან.
- f9 ფირმას აქვს რეკვიზიტები.
- f10 ფირმას აქვს გახსნილი ანგარიში ბანკში.
- f11 ფირმას აქვს გარკვეული რაოდენობის პროდუქცია.
- f12 კლიენტს შეუძლია დადოს კონტრაქტი გარკვეული რაოდენობის პროდუქციის შესყიდვაზე.
- f13 კონტრაქტი იდება გარკვეულ თანხაზე.
- f14 კონტრაქტში გათვალისწინებულია პროდუქციის მიწოდების ვადები.
- f15 ფირმას შეიძლება ჰქონდეს ფილიალები.
- f16 თანამშრომელს აქვს წოდება.
- f17 თანამშრომელს დაკავებული აქვს გარკვეული თანამდებობა ან აყვანილია კონტრაქტით.
- f18 თანამშრომელი ასრულებს სამუშაოს.
- f19 თანამშრომელი აბარებს ანგარიშს თანამშრომელს.
- f20 თანამშრომელი, რომელიც ასრულებს სამუშაოს ვერ შეამოწმებს მას.

ზემოთ აღწერილი წესების თანახმად დავხაზოთ დიაგრამა და შეზღუდვები ჩვენი მაგალითისათვის. იხ. ნახ. 2.13



ნახ. 2.13

გამყოფენ შიგა და გარე შეზღუდვებს. შიგა შეზღუდვა მიეკუთვნება ერთი ობიექტის ტიპს, ხოლო გარე ობიექტების

ტიპების ნაკრებს. როლი, რომელიც არ არის იძულებითი არის არასავალდებულო. იძულების დიზუნქცია ნიშნავს, რომ თითოეული ობიექტი ობიექტის ტიპების ნაკრებიდან უნდა ასრულებდეს მხოლოდ ერთ როლს.

დიაგრამაზე ასახულია შემდეგი სახის შეზღუდვები:

- იძულების შეზღუდვა – თითოეულ თანამშრომელს აქვს გვარი

- ⊙ იძულების დიზუნქცია – როლი მუშაობს კონტრაქტით და დაკავებული აქვს თანამდებობა არიან არასავალდებულო შეზღუდვები, მაგრამ მათი დიზუნქცია არის იძულებითი.

- ⊗ წყვილის გამორიცხვის შეზღუდვა გვიჩვენებს, რომ თანამშრომელი რომელიც ასრულებს პროექტს, ვერ შეამოწმებს მას.

→ შიდა უნიკალურობის შეზღუდვა ბინარული ფაქტის ტიპზე გვიჩვენებს, რომ თანამშრომელს აქვს მხოლოდ ერთი გვარი, მუშაობს მხოლოდ ერთ დეპარტამენტზე.

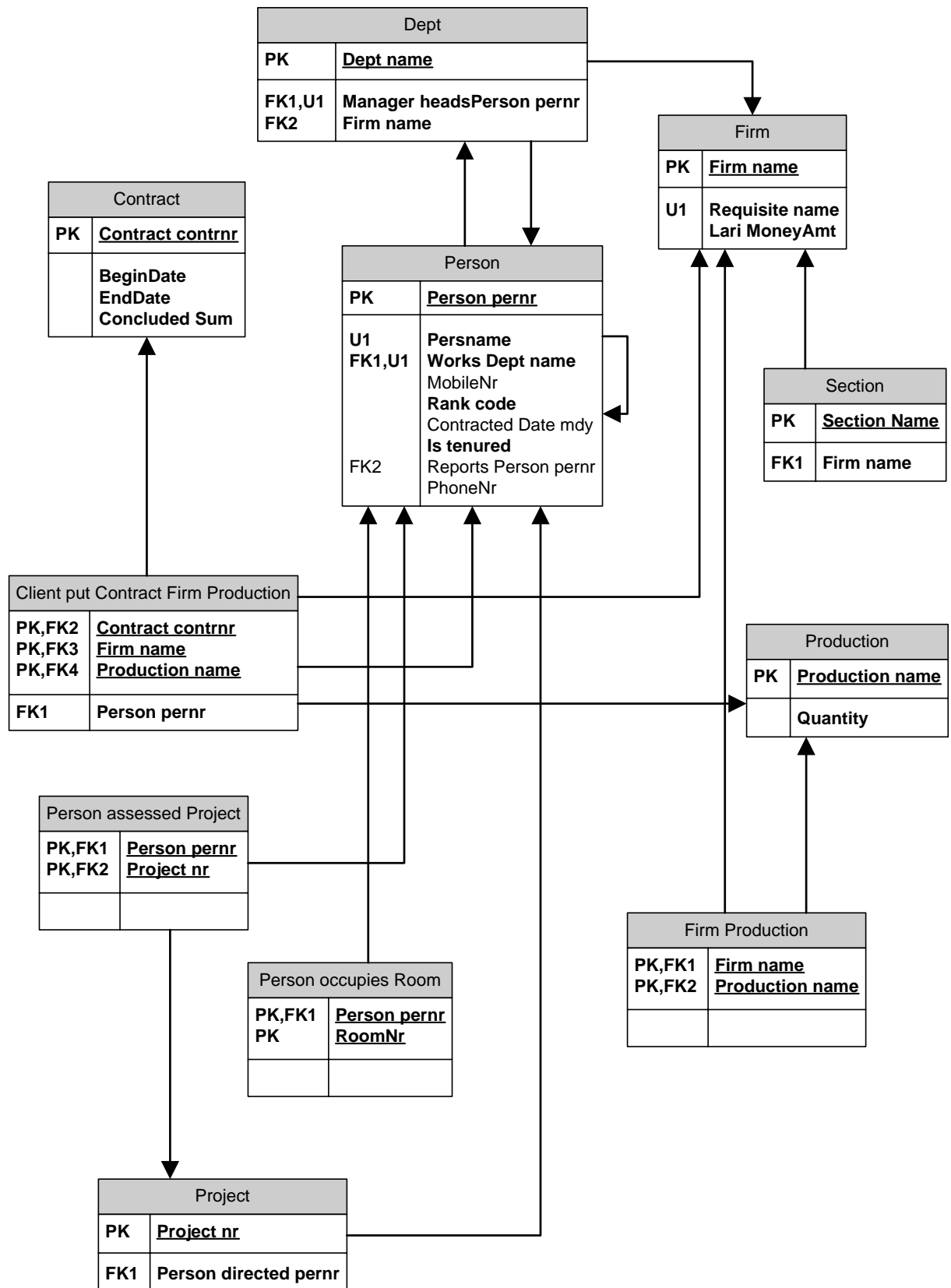
- Ⓢ გარე უნიკალურობის შეზღუდვა, გვიჩვენებს, რომ თითოეული კომბინაცია (დეპარტამენტი, თანამშრომელი) არსებობს არა უმეტეს ერთი თანამშრომლისათვის.

- ⊆ წყვილი ქვესიმრავლის შეზღუდვა გვიჩვენებს, რომ თითოეული მენეჯერი, რომელიც ხელმძღვანელობს განყოფილებას ასევე მუშაობს ამ განყოფილებაში.

↑ მენეჯერი და კლიენტი არის თანამშრომელის ქვეტიპი.

Oas ასიმეტრიულობის შეზღუდვა გვიჩვენებს, რომ საკუთარ თავს ვერ ჩააბარებ ანგარიშს ანუ თუ თანამშრომელი, რომელიც ასრულებს პროექტს ვერ შეამოწმებს მას.

ORM-დიაგრამიდან ავტომატურად მიიღება ER-დიაგრამა, რომელიც ნახვენებია ნახ.2.14-ზე.



ნახ. 2.14



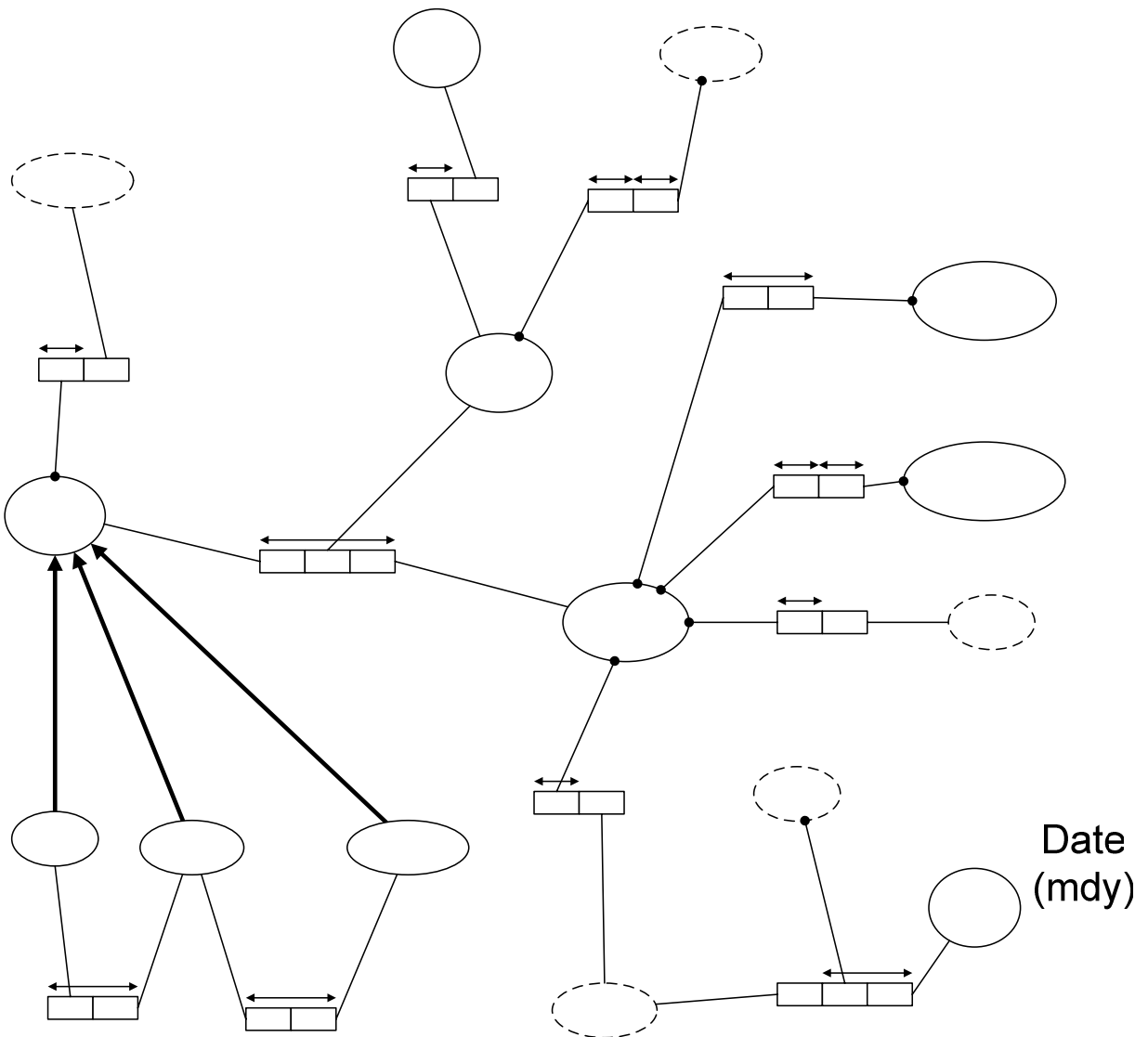
მაბალოთი 2.

განვიხილოთ ORM-დიაგრამა ოფის-ობიექტზე კონტრაქტის დადების მაგალითზე. აღვწეროთ ფაქტები:

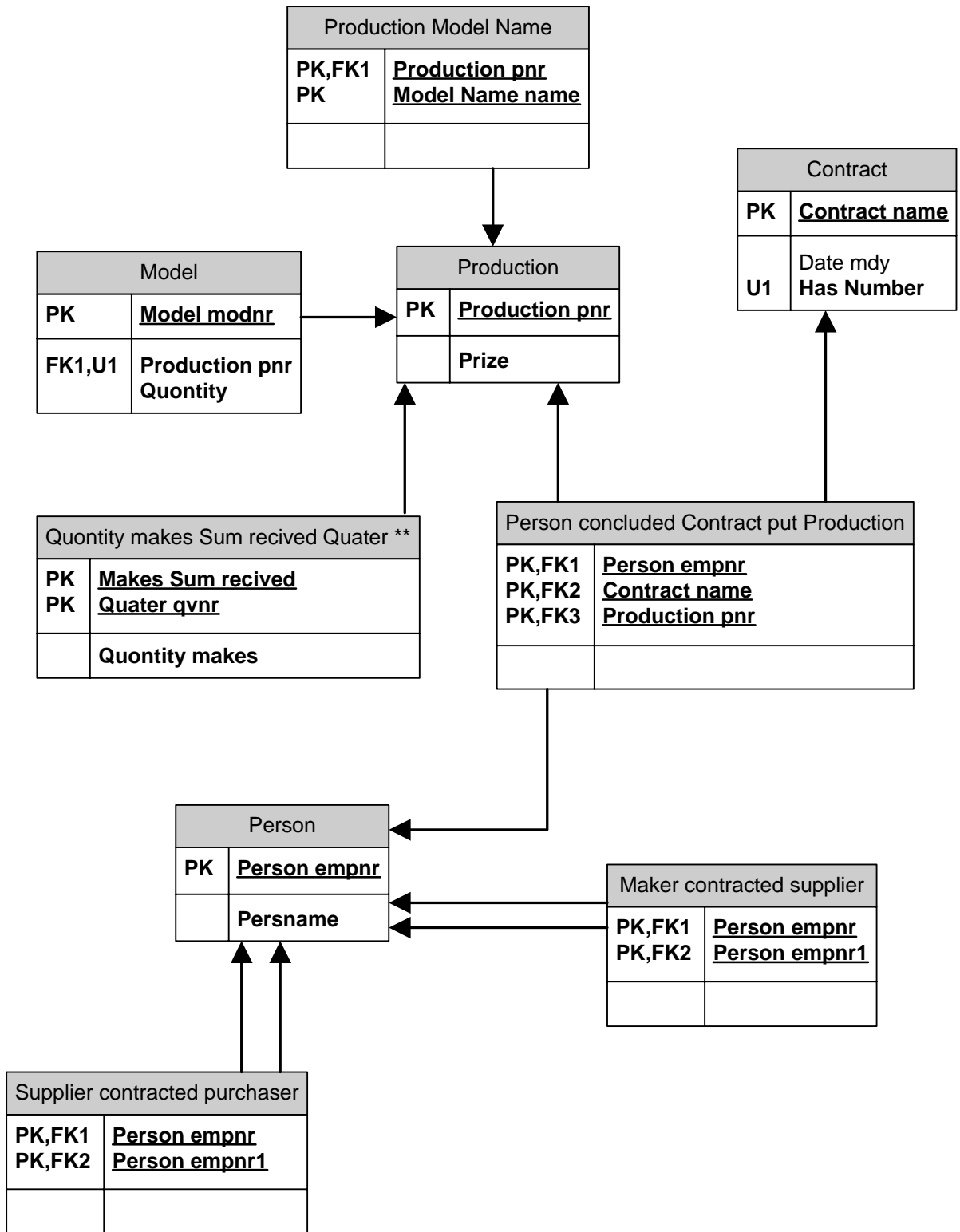
- f1 პერსონას აქვს სახელი
- f2 დამამზადებელი, მიმწოდებელი და მყიდველი არიან პერსონაები.
- f3 პერსონამ დადო კონტრაქტი პროდუქციის შესყიდვაზე.
- f4 კონტრაქტს აქვს ნომერი, .სახელი და თარიღი.
- f5 პროდუქციას აქვს მოდელის დასახელება და ფასი.
- f6 პროდუქციას აქვს მოდელის ნომერი და .რაოდენობა.
- f7 პროდუქცია გარკვეული რაოდენობით და ფასით გაიყიდა კვარტალში.
- f8 თანხა არის ფასი გამრავლებული რაოდენობაზე.

ORM-დიაგრამაზე გამოსათვლელი ფაქტის ტიპი აღინიშნება “\*”

ოფის-ობიექტზე კონტრაქტის დადების ORM-დიაგრამა და მისი შესაბამისი ER-დიაგრამა ნაჩვენებია ნახ.2.15 და ნახ.2.16



ნახ. 2.15 ORM-დიაგრამა „კონტრაქტის დადება“  
Persname



ნახ. 2.16 ER-მოდელი „კონტრაქტის დადება“

## 2.5. ბიზნეს-პროცესების ობიექტ-ორიენტირებული დიაგრამის კლასთაშორისი კავშირების ნორმალიზაცია

ORM-ობიექტ-როლური მოდელის აგების შემდეგ, როგორც აღვნიშნეთ საჭიროა საპრობლემო სფეროს არსთადამოკიდებულების, ER-მოდელის აგება. ეს ამოცანა შესაძლებელია MsVisio-პაკეტის მიხედვით ავტომატიზებულად გადავწყვიტოთ, რასაც მოჰყვება .DDL ფაილების გენერირება და MsSQL\_Server-ზე მიბმა [15].

მნიშვნელოვან ამოცანას ამ ეტაპზე, როგორც აღვნიშნეთ, კლასებისა და კლასთაშორისი კავშირების ნორმალიზაცია წარმოადგენს, რაც უშუალო კავშირშია ER-მოდელის შესაბამისი კლასების დიაგრამის „სტრუქტურის ოპტიმიზაციის“ საკითხთან. მონაცემთა ბაზების დაპროექტების თეორიის ტერმინებით – ესაა კონცეპტუალური სქემის ნორმალიზაციის ამოცანა, ობიექტ-ორიენტირებული მოდელირების თვალთახედვით.

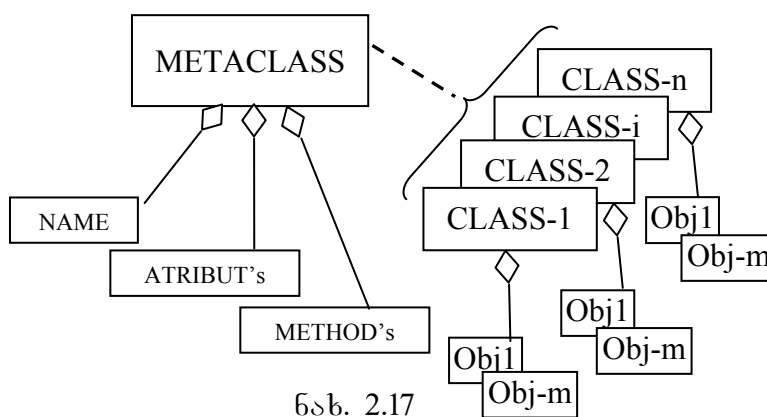
აქ იგულისხმება შემდეგი: უნდა განისაზღვროს თითოეული კლასის მონაცემთა ოპტიმალური სტრუქტურა (მონაცემთა მოდელი), კლასთა შორის კავშირების გათვალისწინებით და დამოკიდებულებათა რელაციების ნორმალიზაციის მეთოდების გამოყენებით.

ოპტიმიზაციის კრიტერიუმად შეიძლება განვიხილოთ რელაციური ცხრილების ინფორმაციული და ინდექსური მონაცემების მოცულობათა მინიმიზაცია, და მათთან მიმართვისა და განახლების დროის შემცირება. მონაცემთა ერთიანი ლოგიკური სტრუქტურების მთლიანობის უზრუნველყოფით (დაცვით). ამისათვის შევიმუშავეთ სპეციალური ალგორითმი [54].

ობიექტ-ორიენტირებული თეორიის ფუნდამენტური განსაზღვრებით, როგორც ცნობილია, მონაცემთა (ატრიბუტების) და

ფუნქციათა (მეთოდების) ინკაპსულაცია არის კლასი. სწორედ კლასებისა და კლასთაშორის კავშირების სიმრავლეთა ცნებები დაედო საფუძვლად ჩვენს კვლევებს და განხორციელდა თეორიული მეთოდებისა და ალგორითმების რეალიზაცია დასმული ამოცანის გადასაწყვეტად. ასეთი მიდგომა დასმული ამოცანის გადასაწყვეტად უზრუნველყოფს, ერთის მხრივ, მონაცემთა სტრუქტურების ოპტიმიზაციას ნორმალურ ფორმათა თეორიის საფუძველზე და, მეორეს მხრივ, დაპროექტების ობიექტ-ორიენტირებული პრინციპების მხარდაჭერას ტექნოლოგიების Case-გამოყენებისას.

კვლევის ობიექტი ამ შემთხვევაში საპრობლემო სფეროს კლასთა ერთობლიობაა, რომლის ზოგადი სტრუქტურა ( ეტაცლასს) მოცემულია 2.17 ნახაზზე.



ნახ. 2.17

აგრეგატული კავშირების საფუძველზე მეტაკლასი შეიძლება დაგახასიათოთ შემდეგი თვისებებით: სახელი (კლასის იდენტიფიკატორი), ატრიბუტები (კლასის მდგომარეობა) და მეთოდები (კლასის ყოფაქცევა). მეტაკლასის ეგზემპლარები კონკრეტული კლასებია, რომელთაც აღნიშნულ თვისებათა განსაზღვრული მნიშვნელობები აქვს და აგრეგატულად შედგება ერთტიპური ობიექტების ერთობლიობისგან.

როგორც ცნობილია, კლასის რომელიმე მეთოდის გააქტიურება ამ კლასის მონაცემთა განსაზღვრული დამუშავების მიზნით,

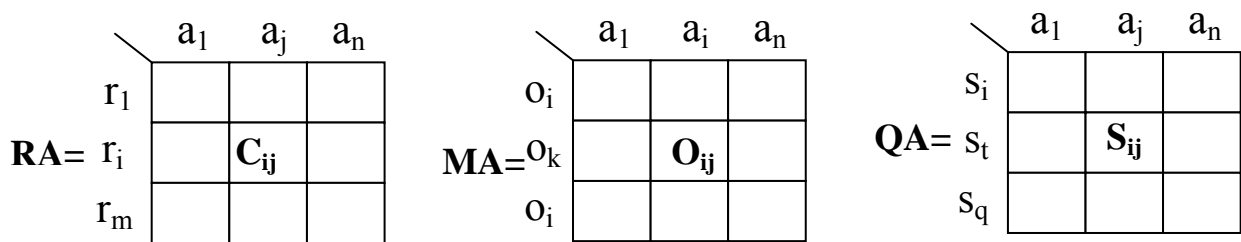
ხორციელდება კლასის გარედან მოსული შეტყობინების ( ესსაგე) საფუძველზე. ესაა ინფორმაცია მართვის ავტომატიზებული სისტემის გარე მომხმარებელთა მოთხოვნების შესახებ.

ამგვარად, მომხმარებელთა მოთხოვნის სტრუქტურის დინამიკურად ფორმირების საფუძველზე, ოპერატიულად, ავტომატურად განისაზღვრება მეთოდი (ან მეთოდები) და მისთვის საჭირო კლასის მონაცემები.

დასმული ამოცანის გადაწყვეტის ეტაპები შემდეგია:

- კლასთა საწყისი სიმრავლის ანალიზის საფუძველზე (ან პირიბითად-უნივერსალური, ერთიანი კლასის საფუძველზე) განისაზღვრება ატრიბუტთა  $(A_j, j = \overline{1, n})$ , დამოკიდებულებათა  $(R_i, i = \overline{1, m})$ , მეთოდებისა  $(M_k, k = \overline{1, p})$  და მოთხოვნების  $(Q_l, l = \overline{1, q})$  სიმრავლეები;

ლოგიკური, სემანტიკური და მიზეზ-შედეგობრივი დამოკიდებულებების საფუძველზე ზემოაღნიშნულ  $(A, R, M, Q)$ , სიმრავლეთა ელემენტებს შორის ვაგებთ სამ მატრიცას  $RA, MA, QA$  ატრიბუტების ერთი და იგივე სიმრავლეზე (იხ. ნახ. 2.18).



ნახ. 2.18

$U = \{a_i\} \quad i = \overline{1, n}$  ატრიბუტები

$M = \{O_k\} \quad k = \overline{1, p}$  მეთოდები

$R = \{r_j\} \quad j = \overline{1, m}$  რელაციები

$Q = \{S_l\} \quad l = \overline{1, q}$  მოთხოვნები

- პირველი  $RA$ -მატრიცის ფორმირების მიზნით, რომელიც წარმოადგენს რი დამოკიდებულებების სიმრავლეს დომენების დეკარტული ნამრავლის ქვესიმრავლეზე  $(R \subseteq a_1 \in A_1 \times a_2 \in A_2 \times \dots$

$am \in Am$  ), საჭიროა გამოვიყენოთ  $n$ -არულ დამოკიდებულებათა განსაზღვრის ლოგიკურ-აღგებრული მეთოდები. კორტეჯის ელემენტი

$$C_{ij} = \{(1, \text{if } a_j \in r_i) \text{ OR } (0, \text{if } a_j \notin r_i) \text{ OR } (2, \text{if } a_j \notin r_i \text{ AND } a_j \text{ is Key})\}$$

აღგორითმი ხუთი ბიჯისგან შედგება:

1)  $F^0$  - ფუნქციონალურ დამოკიდებულებათა საწყისი სიმრავლის დეკომპოზიცია  $F^1$  - სიმრავლეში, სადაც

$$F^0(\bar{A}) = \{r_k\} = \{ \langle \{A_i\} \rangle \rightarrow \{A_j\} \} \quad (1)$$

$$F^1(\bar{A}) = \{r_{\varepsilon_1}^1\} = \{ \langle \{A_i\} \rangle \rightarrow A_j \} \quad (2)$$

და  $i, j \in \gamma; \gamma = \overline{1, n}; i \neq j$ .  $F^0$ -სიმრავლის სიმძლავრე აღვნიშნოთ  $m$ -ით, ე.ი.  $k = (\overline{1, m})$ , მაშინ  $F^1$  სიმრავლის სიმძლავრე იქნება  $m_0 \geq m$ , ხოლო ატრიბუტთა სიმრავლის სიმძლავრე იქნება  $n_0 \leq n$ .

$$\text{აქ } \varepsilon_1 = \overline{1, m_0}; i, j \in r(\{1, 2, \dots, n\}, i \neq j \text{ და } (\varepsilon_1)_{\max} = m_0.$$

2)  $F^1$  - ფუნქციონალურ დამოკიდებულებათა სიმრავლის გამოკვლევა გაფართოების (შთანთქმის) თვისებაზე და  $F^2$  - სიმრავლის ფორმირება, სადაც

$$F^2(\bar{A}) = \{r_{\varepsilon_2}^2 \mid r_{\varepsilon_2}^2 \in \cup \{ (r_{\varepsilon_1}^1) \tilde{\rho} [F^1(\bar{A}) \setminus \cup (r_k)] \} \} \quad (3)$$

$$\varepsilon_1 = 1 \quad k = 1$$

სადაც  $(\varepsilon_2)_{\max} \leq (\varepsilon_1)_{\max}$ .

$\tilde{\rho}$ -ოპერაციას (ნახ.3.11) აქვს შემდეგი სემანტიკა:

$$(\langle \{A_\sigma\} \rangle \rightarrow A_\lambda) \tilde{\rho} (\langle \{A_\delta\} \rangle \rightarrow A_\nu) =$$

$$= \begin{cases} \langle \{A_\sigma\} \rangle \rightarrow A_\lambda, & \text{if } (A_\lambda = A_\nu) \ \& \ (\{A_\sigma\} \subseteq \{A_\delta\}); \\ \langle \{A_\delta\} \rangle \rightarrow A_\nu, & \text{if } (A_\lambda = A_\nu) \ \& \ (\{A_\sigma\} \supseteq \{A_\delta\}); \\ (\langle \{A_\sigma\} \rangle \rightarrow A_\lambda) \ \& \ (\langle \{A_\delta\} \rangle \rightarrow A_\nu) & \text{otherwise.} \end{cases}$$

$\tilde{\rho}$ -ოპერაციის ჩატარების შემდეგ საგრძნობლად მცირდება  $F^2(\bar{A})$  სიმრავლის სიმძლავრე. ე.ი.  $(\varepsilon_2)_{\max} < (\varepsilon_1)_{\max}$ , რაც უზრუნველყოფს ალგორითმის მომდევნო ბიჯზე სტრიქონების გადარჩევის რაოდენობის შემცირებას.

3)  $F^2$ - ფუნქციონალურ დამოკიდებულებათა სიმრავლის გამოკვლევა პროექციის თვისებაზე და  $F^3$  - სიმრავლის ფორმირება, სადაც

$$F^3(\bar{A}) = \{r_{\varepsilon_3}^3 \mid r_{\varepsilon_3}^3 \in \cup_{k=1}^{\varepsilon_2} \{r_{\varepsilon_2}^2\} \tilde{\eta}[F^2(\bar{A}) \setminus \cup(r_k^2)]\} \quad (4)$$

$\varepsilon_2 = 1 \qquad k = 1$

სადაც  $(\varepsilon_3)_{\max} = (\varepsilon_2)_{\max}$  და

$$\begin{aligned} & (\langle \{A_\sigma\} \rangle \rightarrow A_\lambda) \tilde{\eta} (\langle \{A_\rho\} \rangle \rightarrow A_\nu) = \\ & = \begin{cases} (\langle \{A_\sigma\} \rangle \rightarrow A_\lambda) \& (\langle \{A_\mu\} \rangle \rightarrow A_\nu), & \text{if } \{A_\sigma\} \subset \{A_\rho\} \& A_\lambda \in \{A_\rho\}, \\ & \text{სადაც } \{A_\mu\} \subset \{A_\sigma\}, A_\lambda \notin \{A_\mu\}; \\ (\langle \{A_\mu\} \rangle \rightarrow A_\lambda) \& (\langle \{A_\rho\} \rangle \rightarrow A_\nu), & \text{if } \{A_\rho\} \subset \{A_\sigma\} \& A_\nu \in \{A_\sigma\}, \\ & \text{სადაც } \{A_\mu\} \subset \{A_\rho\} \wedge A_\nu \notin \{A_\mu\}; \\ (\langle \{A_\sigma\} \rangle \rightarrow A_\lambda) \& (\langle \{A_\rho\} \rangle \rightarrow A_\nu) & \text{otherwise.} \end{cases} \end{aligned}$$

$\tilde{\eta}$ -ოპერაციის შესრულების შემდეგ სიმრავლის სიმძლავრე არ იცვლება, ვინაიდან არ ხდება ამ სიმრავლიდან დამოკიდებულებების ამოშლა (ან სიმრავლეში ჩამატება). ამ შემთხვევაში ზუსტდება გასაღებურ ატრიბუტთა შემადგენლობა ცალკეული დამოკიდებულებებისთვის.

თუ შედგენილ გასაღებური ატრიბუტში მოხდება ერთი ატრიბუტის ამოშლა მაინც, მაშინ პროცესი ბრუნდება ალგორითმის მე-2 ბიჯზე და ხელახლა სრულდება  $\tilde{\rho}$ -ოპერაცია. თუ ასეთს არ აქვს ადგილი, მაშინ გადავდივართ მე-4 ბიჯზე;

4)  $F^3$  - ფუნქციონალურ დამოკიდებულებათა სიმრავლის გამოკვლევა ტრანზიტულობის და ფსევდოტრან-ზიტულობის თვისებებზე. დაეუშვათ მოცემულია  $F^3(\bar{A})$  - სიმრავლის ორი დამოკიდებულება:

$$\langle \{A_\sigma\} \rangle \rightarrow A_\lambda \quad \text{და} \quad \langle \{A_\sigma\} \rangle \rightarrow A_\nu,$$

აგრეთვე თვისებათა პირობების ცხრილი:

ტრანზიტულობის პირობები	
$\left[ \begin{array}{l} A_\lambda = \langle \{A_\sigma\} \rangle, \\ \langle \{A_\sigma\} \rangle \rightarrow A_\nu, \\ A_\lambda \not\rightarrow \langle \{A_\sigma\} \rangle \end{array} \right.$	$\left[ \begin{array}{l} A_\nu = \langle \{A_\sigma\} \rangle, \\ \langle \{A_\sigma\} \rangle \rightarrow A_\lambda, \\ A_\nu \not\rightarrow \langle \{A_\sigma\} \rangle \end{array} \right.$

ფსევდოტრანზიტულობის პირობები	
$\left[ \begin{array}{l} A_\lambda \subset \langle \{A_\sigma\} \rangle, \\ \langle \{A_\sigma\} \rangle \rightarrow A_\nu, \\ A_\lambda \not\rightarrow \langle \{A_\sigma\} \rangle \end{array} \right.$	$\left[ \begin{array}{l} A_\nu \subset \langle \{A_\sigma\} \rangle, \\ \langle \{A_\sigma\} \rangle \rightarrow A_\lambda, \\ A_\nu \not\rightarrow \langle \{A_\sigma\} \rangle \end{array} \right.$

ფუნქციონალურ დამოკიდებულებათა სიმრავლის გამოკვლევა ტრანზიტულობის და ფსევდოტრან-ზიტულობის თვისებებზე ხორციელდება შემდეგი ალგორითმით:

$$(\varepsilon_3)_{\max-1} \quad \varepsilon_3$$

$$F^4(\bar{A}) = \{r_{\varepsilon_4}^4 \mid r_{\varepsilon_4}^4 \in \cup\{(r_{\varepsilon_3}^3) \tilde{\Psi}[F^3(\bar{A}) \setminus \cup(r_k^3)]\}\}, \quad (5)$$

$$\varepsilon_3 = 1 \quad k=1$$

სადაც  $(\varepsilon_4)_{\max} < (\varepsilon_3)_{\max}$  და

$$\langle \{A_\sigma\} \rangle \rightarrow A_\lambda \quad \bar{\Psi} \langle \{A_\sigma\} \rangle \rightarrow A_\nu = \quad (6)$$

$$= \begin{cases} \langle \{A_\lambda\} \rangle \rightarrow A_\nu, & \text{if } (A_\lambda \in \{A_\sigma\}) \& (A_\nu \bar{\in} \{A_\sigma\}), (A_\lambda) = (\{A_\sigma\} \cup \{A_\sigma\} \setminus A_\lambda); \\ & \text{when } \{A_\sigma\} = A_\lambda \Rightarrow \{A_\lambda\} = \{A_\sigma\}, \text{ because of } \{(\{A_\sigma\} \setminus A_\lambda) = \emptyset; \\ \langle \{A_\lambda\} \rangle \rightarrow A_\lambda, & \text{if } (A_\lambda \in \{A_\sigma\}) \& (A_\lambda \bar{\in} \{A_\sigma\}), (A_\lambda) = (\{A_\sigma\} \cup \{A_\sigma\} \setminus A_\lambda); \\ & \text{when } A_\nu = \{A_\sigma\} \Rightarrow \{A_\lambda\} = \{A_\sigma\}, \text{ because of } \{(\{A_\sigma\} \setminus A_\lambda) = \emptyset; \\ (\langle \{A_\sigma\} \rangle \rightarrow A_\lambda) \& (\langle \{A_\sigma\} \rangle \rightarrow A_\nu), & \text{otherwise.} \end{cases}$$

$\tilde{\Psi}$ -ოპერაციის შესრულების შემდეგ შესაძლებელია ახალი დამოკიდებულებების გაჩენა, რომლებიც დროებით დაემატება  $F^4$ -ს. ამ სიმრავლეზე კვლავ შესრულდება  $\tilde{\rho}$  და  $\tilde{\eta}$ ოპერაციები. ამ პროცედურების შემდეგ სიმრავლიდან ამოიშლება დროებით ჩამატებული დამოკიდებულებანი. შედეგად მცირდება  $F^4$ - სიმრავლის სიმძლავრე.



5)  $F^k$  საბოლოო სიმრავლის კომპოზიცია  $F^4$ -დან. ამ ბიჯზე ფორმირდება  $A/F^k$  ფაქტორ-სიმრავლე. დამოკიდებულებათა საშედეგო სიმრავლის სიმძლავრე განისაზღვრება გასაღებურ ატრიბუტთა განსხვავებულ ერთობლიობათა რაოდენობით.

იმისათვის, რომ მიღებული ახალი სტრუქტურა ვაქციოთ კლასთა დიაგრამად, საჭიროა თითოეულ მათგანში განვსაზღვროთ მეთოდების ერთობლიობა.

აქ ვიყენებთ მეორე მატრიცას, რომლის საფუძველზე კლასის ატრიბუტების სიმრავლეს შეესაბამება ესა თუ ის მეთოდი მეთოდების ბიბლიოთეკიდან. შესაძლებელია მეთოდების დუბლირება, რომელიც რეალიზებულ იქნება მეთოდების ინდექსების საშუალებით.

მეორე, ინფორმაციული MA-მატრიცის ფორმირების მიზნით, რომელიც წარმოადგენს დამოკიდებულებათა მეთოდების სიმრავლეს, რომელიც განისაზღვრება დომენების დეკარტული ნამრავლების ქვესიმრავლეზე ( $M \subseteq o_1 \in O_1 \times o_2 \in O_2 \times \dots \times o_m \in O_m$ ), საჭიროა გავანალიზოთ  $CL_U$  უნივერსალური კლასის თავდაპირველი შედგენილობა, რომელიც შეიცავს ველა მეთოდის სიმრავლეს, რომელიც მოიცავს მოცემული კლასის ატრიბუტებს. ამ ალგორითმის შედეგად შეივსება MA-მატრიცა. ელემენტის კორტეჟით

$$O_{kj} = \{1, \text{if } a_j \in o_k \text{ OR } (0, \text{if } a_j \notin o_k)\}$$

მესამე, ინფორმაციული QA-მატრიცის ფორმირების მიზნით, რომელიც წარმოადგენს დამოკიდებულებათა შეტყობინებების სიმრავლეს, რომელიც განსაზღვრულია შეტყობინებათა დომენების დეკარტული ნამრავლის ქვესიმრავლეზე ( $Q \subseteq s_1 \in S_1 \times s_2 \in S_2 \times \dots \times s_m \in S_m$ ), საჭიროა გავანალიზოთ ყველა შესაძლო შეტყობინება.

QA მატრიცის ელემენტის კორტეჟია:

$$S_{ij} = (1, \text{if } a_j \in s_t \text{ OR } (0, \text{if } a_j \notin s_t).$$

ამ ცხრილის საფუძველზე განისაზღვრება მოთხოვნებისა და მეთოდების ურთიერთშეთანხმების შესაბამისობა.

### მეორე თავის დასკვნები

მეორე თავში შემოთავაზებულია განაწილებულ ოფის-სისტემებში დოკუმენტების ავტომატიზებული ფორმირების ანალიტიკური მოდელი, რომლიდანაც ნათლად ჩანს, რომ ნებისმიერ საქმიან პროცედურას თან სდევს შესაბამისი დოკუმენტაციური უზრუნველყოფა. წარმოდგენილია UseCase და Activity-დიაგრამები ოფისებში ყველაზე მეტად გავრცელებული საქმიანი პროცედურებისათვის, კერძოდ კონტრაქტის დადება სხვადასხვა ოფისებთან, ახალი თანამშრომლის მიღება კონტრაქტით, ფირმის მონაწილეობა ტენდერში.

გადმოცემულია მონაცემთა ბაზების დაპროექტებისათვის აუცილებელი კონცეპტუალური სქემის აგების წესები კატეგორიალური მიდგომით. დაპროექტების ინსტრუმენტის სახით, რომელსაც საფუძველად უდევს კატეგორიალური მიდგომა, გამოყენებულია ობიექტ-როლური მოდელირება (ORM). ობიექტ-როლური მოდელირება, რომელიც კონცეპტუალური მოდელირების განვითარებულ ტექნიკას წარმოადგენს, მიახლოებულია ბუნებრივ სალაპარაკო ენასთან. ამ მიდგომის საფუძველზე აგებულია ORM-და ER-დიაგრამები განაწილებული ენერგეტიკული ოფის-სისტემებისათვის.

წარმოდგენილია ალგორითმი, კლასებზე გაფართოებული ნორმალურ ფორმათა თეორიისათვის, რომელიც ეფუძნება მონაცემთა ბაზების ნორმალიზაციის თეორიას ობიექტ-ორიენტირებული კონცეფციის საფუძველზე.

### III ტაპი

#### ობიექტ-ორიენტირებული დიაგრამების დაპროექტება და რეალიზაცია სპრობლემო სფეროსათვის „მარკეტინგი“

##### 3.1. ოფის-სისტემების ობიექტ-ორიენტირებული მოდელირება და დაპროექტება UML-ტექნოლოგიით

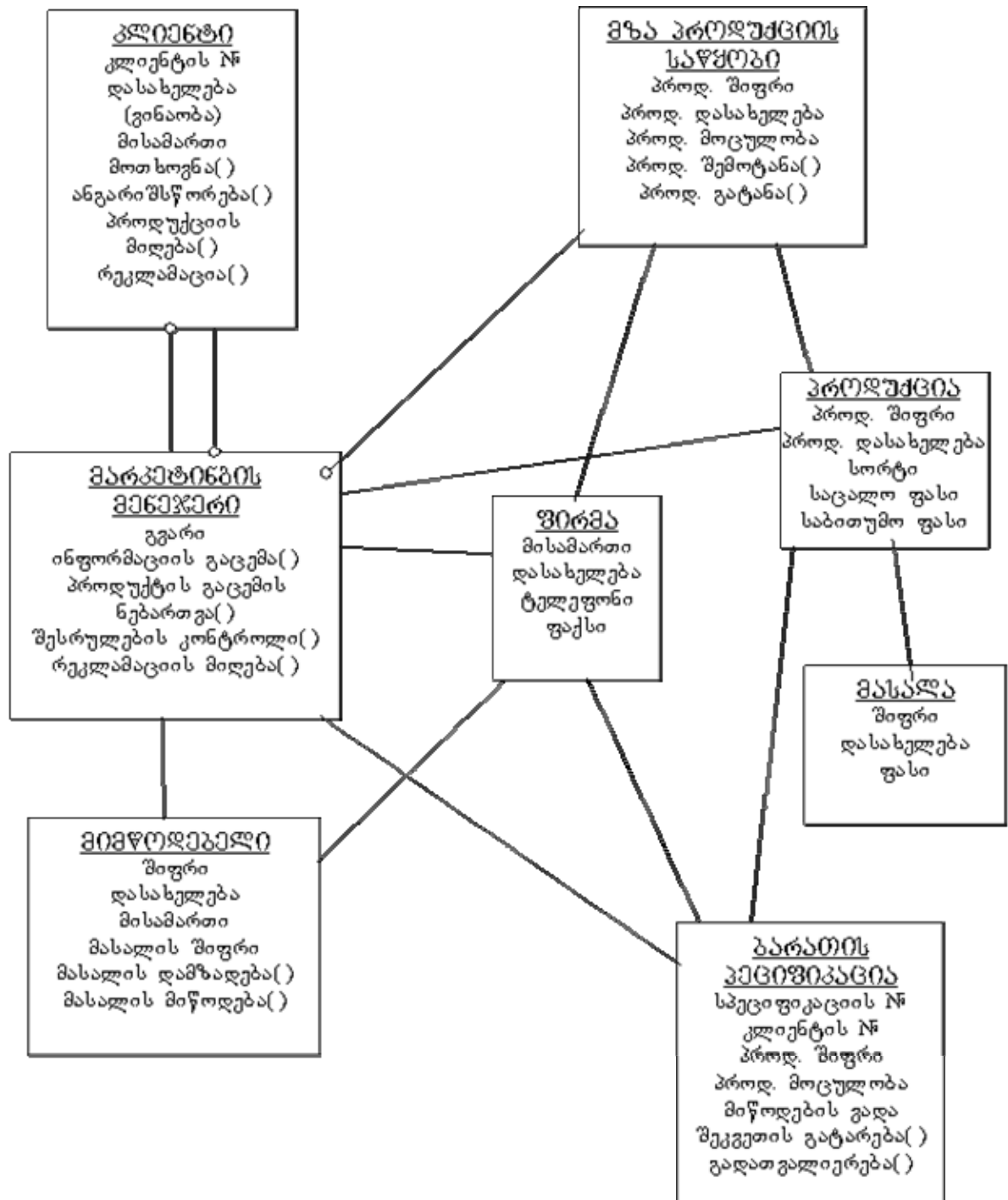
განაწილებული ოფის-ობიექტები მიეკუთვნება დიდი სისტემების კლასს. მათი პროცესების მართვისათვის პროგრამული პაკეტების შექმნა და თანხლება ფუნქციონირების ეტაპზე, საკმაოდ რთული საკითხია. იგი ბევრადაა დამოკიდებული იმაზე, თუ როგორ, რა მეთოდებითა და საშუალებებით იქნა ეს სისტემა შექმნილი. განსაკუთრებული ყურადღება არის გასამახვილებელი სისტემის დაპროექტების ეტაპზე.

ეს საკითხი უნდა განვიხილოთ ობიექტ-ორიენტირებული მიდგომის თვალსაზრისით, კერძოდ, ობიექტ-ორიენტირებული დაპროექტების ისეთი მნიშვნელოვანი საკითხი, როგორცაა ასაგები სისტემის ობიექტ-ორიენტირებული დიაგრამის (ოლ-დიაგრამის) გრაფიკული გამოსახვა.

შესაბამის ლიტერატურაში კარგადაა წარმოდგენილი ობიექტ-ორიენტირებული ანალიზისა და დაპროექტების ცნობილი კლასიკოსების ბუჩის, კოად-იორდონის, ჯაკობსონის, მარტინისა და შლეერ-მელორის გრაფიკული აღწერის საშუალებანი [2], ამიტომ ჩვენ მათ აქ დეტალურად არ შევეხებით. მხოლოდ გამოვიყენებთ ბოლო ვერსიების ინსტრუმენტებს ჩვენი კონკრეტული ამოცანების აღსაწერად.

3.1 ნახაზზე წარმოდგენილია ოლ-დიაგრამის ფრაგმენტი სპრობლემო სფეროსათვის „მარკეტინგის მართვა საწარმოო ფირმაში“. დიაგრამა აგებულია გ. ბუჩის აღწერის სისტემაში. ნახაზზე ნაჩვენებია სისტემის ძირითადი კლასები და მათი

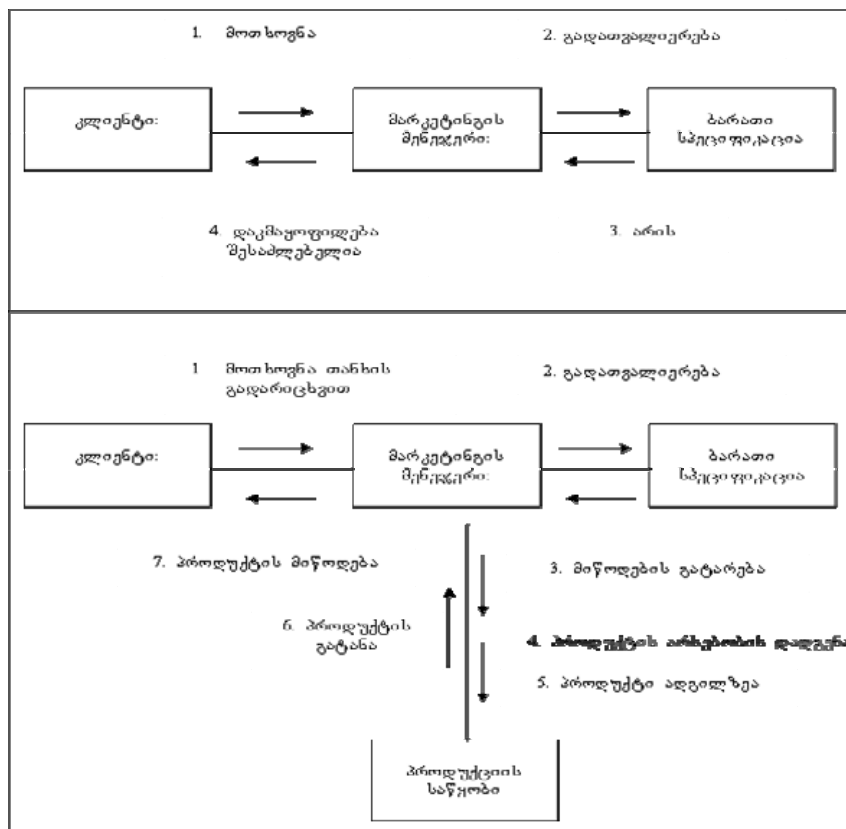
ურთიერთკავშირები, კლასების ცვლადები (მონაცემები) და მათი მეთოდები.



ნახ. 3.1. ობ-დიაგრამა საპრობლემო სფეროსათვის „მარკეტინგი“

ამ საბაზო ოო-დიაგრამის საფუძველზე შესაძლებელია მოთხოვნების შესრულება. 3.2 ნახაზზე მოცემულია ორი შეტყობინების მაგალითი. პირველი – კლიენტს აინტერესებს ინფორმაცია, არის თუ არა ფირმაში მისთვის საჭირო პროდუქცია (ა). მეორე – კლიენტმა უნდა გაიტანოს ფირმიდან მისი კუთვნილი პროდუქცია (ბ).

თუ შევაჯამებთ განხილულ საკითხებს, ადვილად დავრწმუნდებით, რომ გამოკვეთილია ორი სახის პრობლემა: პირველი – ობიექტთა მდგომარეობის მოდელირება (პროცესები შეტყობინების დამუშავებამდე) და მეორე, ობიექტთა ქცევის მოდელირება (პროცესები შეტყობინების დამუშავებით და შედეგებით).



ნახ. 3.2. შეტყობინების დამუშავების სცენარის ოო-დიაგრამა

არაფორმალური ინფორმაცია „საწარმოო ფირმის მარკეტინგის დეპარტამენტის“ შესახებ ასე შეიძლება წარმოვადგინოთ:

„თანამედროვე ფირმა ახორციელებს არა მხოლოდ პროდუქციის წარმოებას, არამედ მის რეალიზაციასაც. ამგვარად, ლაპარაკია პროდუქციის გასაღების პროცესების მართვაზე. ფირმას შეიძლება ჰქონდეს სავაჭრო ბაზები, მაღაზიები და პუნქტები გეოგრაფიულად სხვადასხვა რეგიონში, მათ შორის საზღვარგარეთაც. ამიტომაც, ფირმის ხელმძღვანელობას უხდება პროდუქციის წარმოებისა და რეალიზაციის პროცესების ორგანიზება, მისი მართვა საბაზრო მოთხოვნილებებისა და დიდი კონკურენციის პირობებში.

საწარმოო ფირმაში ახალი კომპიუტერული ტექნოლოგიები გამოყენებულ უნდა იქნეს როგორც წარმოების დაგეგმვის, აღრიცხვის, ანალიზისა და კონტროლის ამოცანების გადასაწყვეტად, ასევე საკონსტრუქტორო, სამხატვრო და ტექნოლოგიური პროცესების შესასრულებლად. იგი არ გამორიცხავს პროდუქციის წარმოებას პროგრამული ჩარხებით და რობოტოტექნიკის გამოყენებას სხვადასხვა რუტინული ოპერაციების შესასრულებლად.

მარკეტინგის დეპარტამენტის მენეჯერი ვალდებულია სწორად განსაზღვროს საბაზრო მოთხოვნილებანი, ერთის მხრივ და ფირმაში წარმოებული პროდუქციის ხარისხის კონტროლის საფუძველზე აწარმოოს ფასების პოლიტიკის რეგულირება, მეორეს მხრივ.

მარკეტინგის დეპარტამენტს კავშირი აქვს ფირმის თითქმის ყველა რგოლთან, როგორც მის შიგნით (საწარმოო და არასაწარმოო დეპარტამენტებთან), ასევე მის გარეთ - საბაზრო მაღაზიებისა და პუნქტების საინფორმაციო აგენტებთან და ა.შ. “.

კომპიუტერული ტექნოლოგიების გამოსაყენებლად მარკეტინგის მართვის პროცესებში საჭიროა მათი მოდელირება და შესაბამისი პროგრამული უზრუნველყოფის შექმნა.

UML-ტექნოლოგიის მიხედვით პირველ ეტაპზე განისაზღვრება ჩვენს მიერ ზემოთ აღწერილი ბიზნეს-პროცესების შესრულების ლოგიკურ მიმდევრობათა ე.წ. „გამოყენებითი შემთხვევების“ დიაგრამები (Use Case Diagrams). 3.3 ნახაზზე მოცემულია საკვლევი ობიექტის „მარკეტინგი“ ასეთი დიაგრამა, რომლის ოვალშიც მოთავსებულია კონკრეტული გამოყენებითი შემთხვევები (Actions), ხოლო „კაცუნები“ ამ შემთხვევებთან ფუნქციურად დაკავშირებული შემსრულებლებია (Actors).

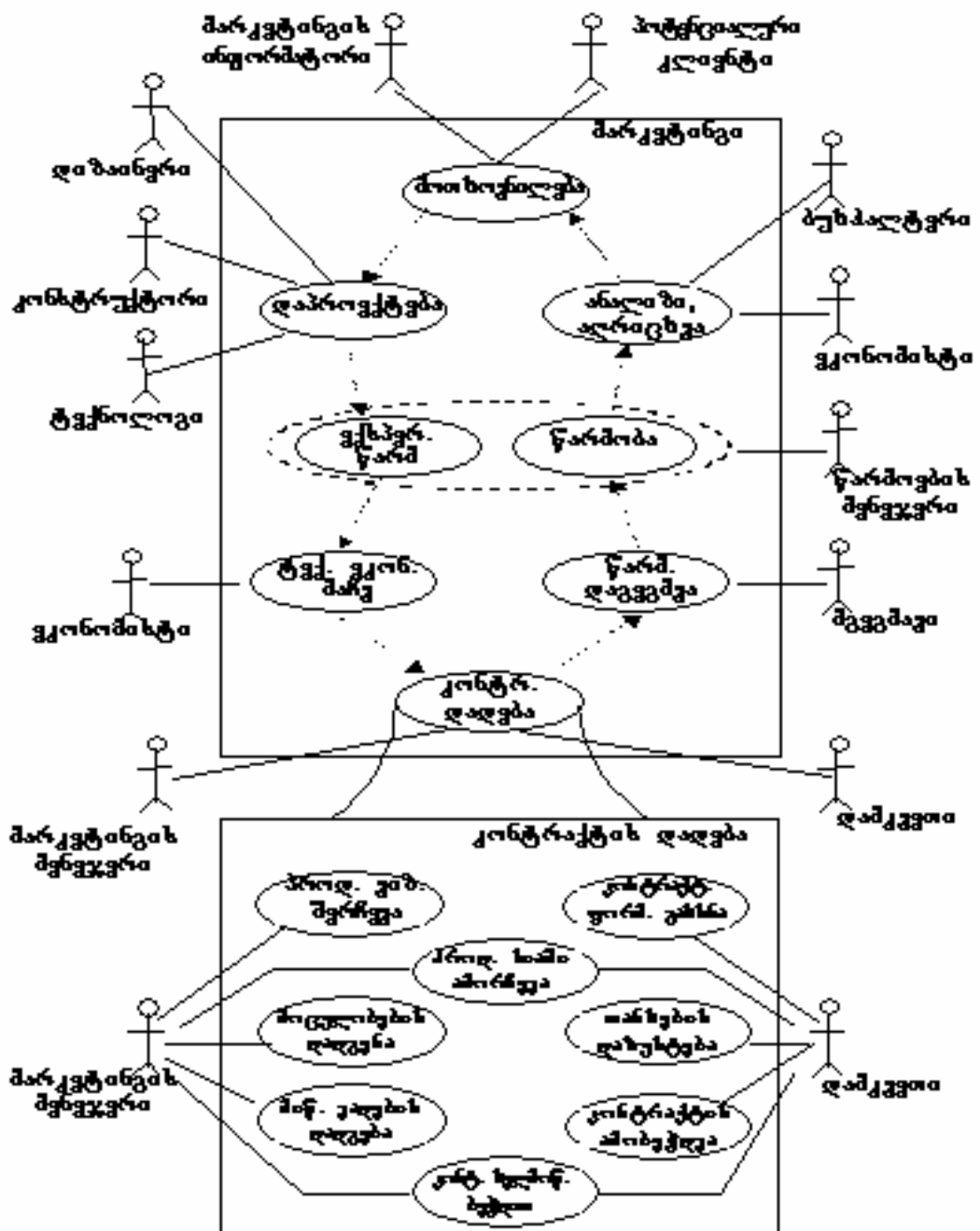
თითოეული ოვალი საპრობლემო გარემოს მართვის სფერო (Control sphere) და იგი შეიძლება თვითონ დაიშალოს უფრო დაბალი დონის კონკრეტულ გამოყენებით შემთხვევებად. ნახაზზე ამის მაგალითია ოვალი „კონტრაქტის-დადება“. სწორედ ამ შემთხვევისათვის განვიხილავთ ობიექტ-ორიენტირებულ ანალიზს და ობიექტ-ორიენტირებულ დაპროექტებას, რომლებიც UML-ტექნოლოგიის საფუძველია.

3.3 ნახაზზე მოცემულ დიაგრამასთან ერთად საჭიროა განისაზღვროს თითოეული გამოყენებითი შემთხვევის მოვლენები (Events) და სცენარები. „კონტრაქტის დადების“ გამოყენებითი შემთხვევის ძირითადი მოვლენათა ნაკადი შემდეგი სახისაა:

1. სისტემა (ან მარკეტინგის მენეჯერი) გახსნის პროდუქციის მონაცემთა ბაზას და საკონტრაქტო ფორმას (სპეციფიკაციას), დააფიქსირებს დაკვეთის ნომერს;
2. კლიენტი ეკრანიდან ვიზუალურად შეარჩევს პროდუქციას და „თავუს“ ორჯერ დაწკაპუნებით დააფიქსირებს მას;
3. დამკვეთი და მენეჯერი საკონტრაქტო ფორმაში ამოირჩევენ შესაბამის პროდუქციას;
4. დამკვეთი მიუთითებს შეკვეთის მოცულობებს (კვარტალური, წლიური);
5. დამკვეთი მიუთითებს შეკვეთის მიწოდების სასურველ ვადებს;
6. მენეჯერი აზუსტებს კვარტალურ და ჯამურ თანხებს;
7. მენეჯერი ბეჭდავს პრინტერზე კონტრაქტის ეგზემპლარებს;
8. დამკვეთი და მენეჯერი ხელს აწერენ კონტრაქტს და აფიქსირებენ ბეჭდებით;
9. მენეჯერი კონტრაქტს ინახავს მონაცემთა ბაზაში;

10. სისტემა განაახლებს პროდუქციის ნაშთების (ლიმიტების) ბაზას მომდევნო დაკვეთებისათვის.

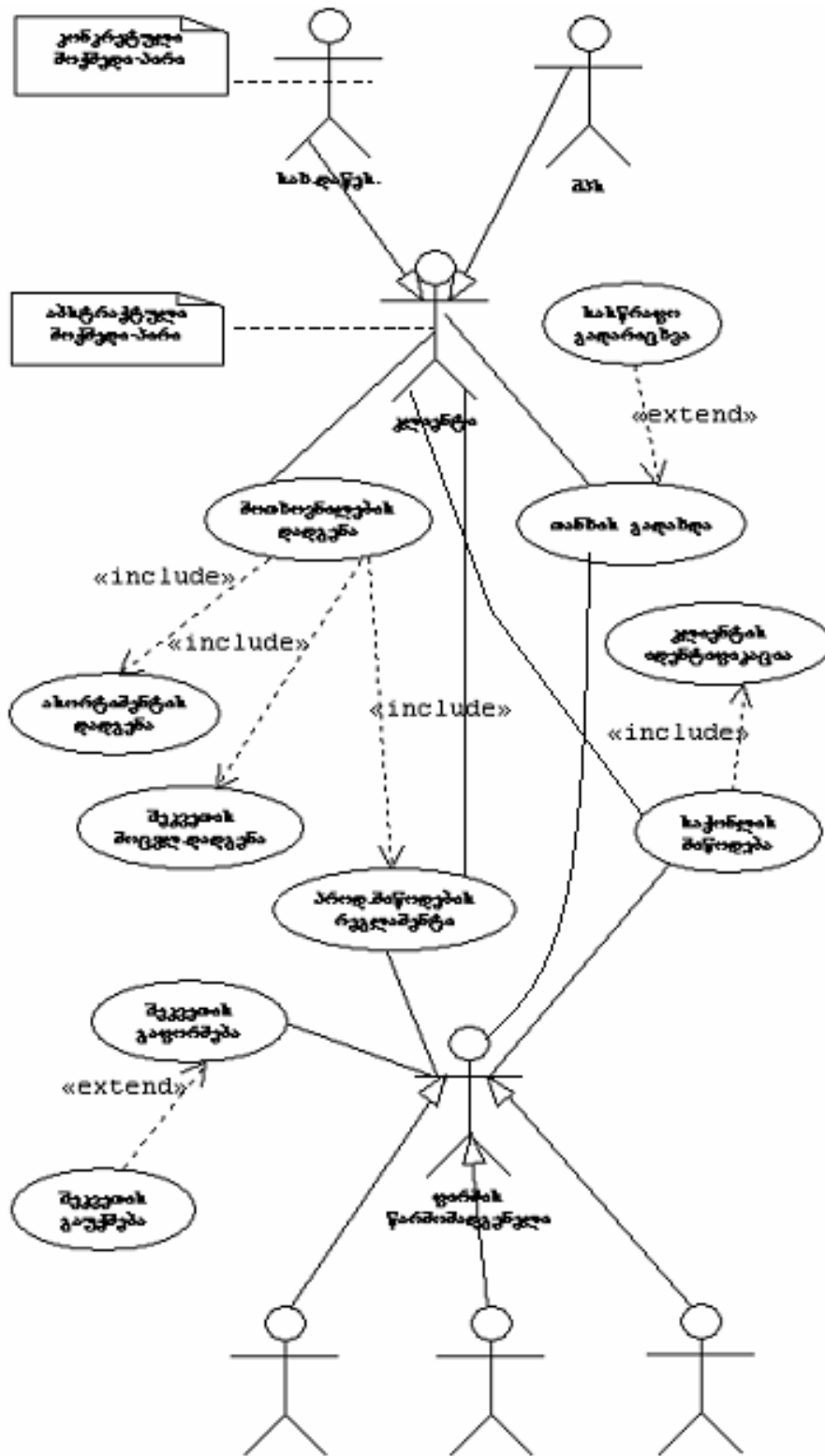
გამოყენებით შემთხვევათა დიაგრამების აგებისა და მათი შესაბამის მოვლენათა განსაზღვრის შემდეგ განიხილება ცალკეულ Action-ის დაპროექტების საკითხი, რომელიც კავშირშია Activity Diagram-ების აგებასთან.



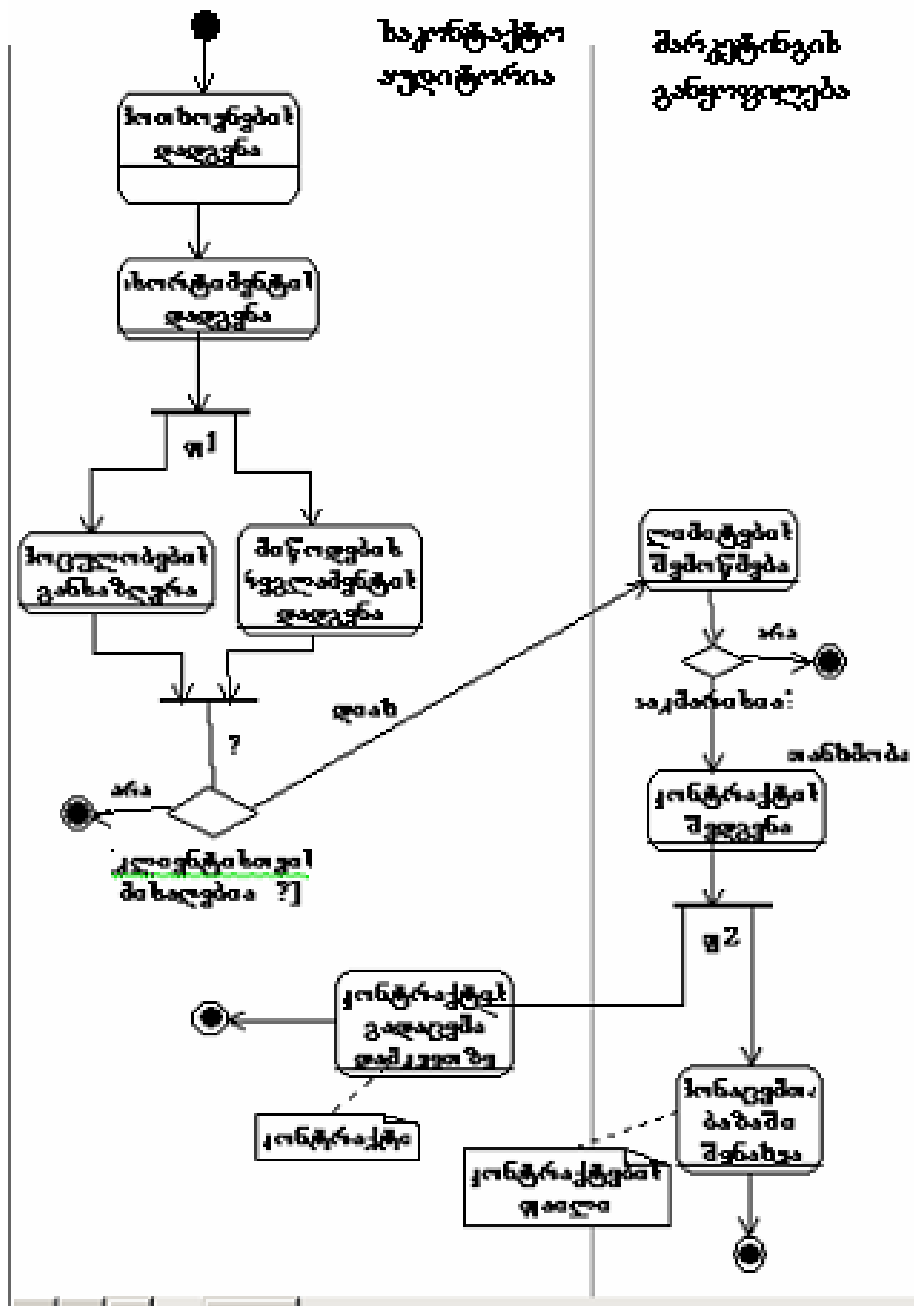
ნახ. 3.3. გამოყენებითი შემთხვევების (UseCase) ორდონიანი დიაგრამა



3.4 და 3.5 ნახაზებზე ნაჩვენებია ჩვენს მიერ აგებული გამოყენებით შემთხვევათა და მისი აქტიურობის დიაგრამის ფრაგმენტები ამოცანისათვის „კონტრაქტის დადება“.



ნახ.3.4. დიაგრამა „კონტრაქტი“



ნახ.3.5. აქტიურობის დიაგრამა „კონტრაქტი“

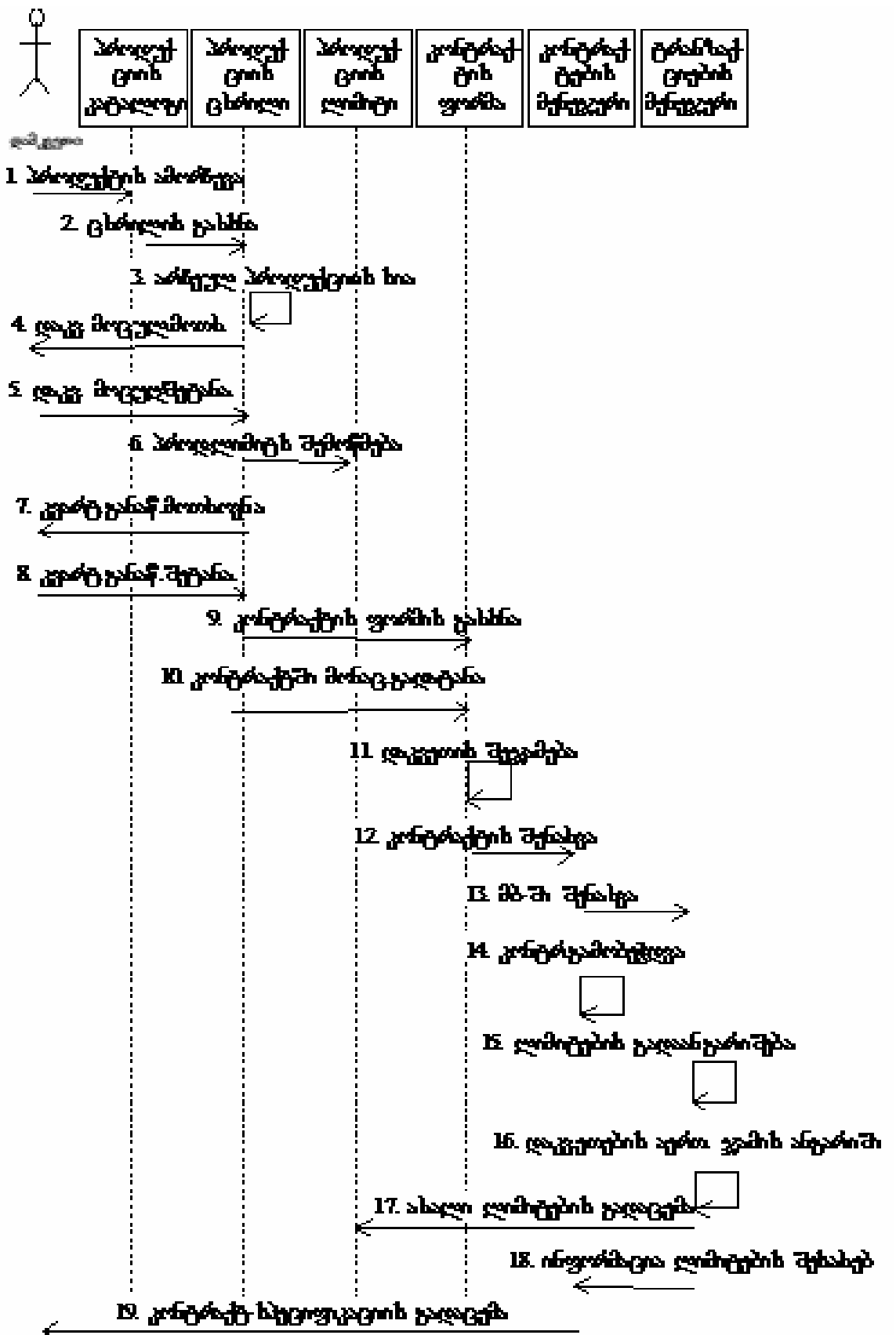
როგორც ნახაზიდან ჩანს, განიხილება ორი მართვის (საპასუხისმგებლო) სფერო (Swimlines): საკონტრაქტო აუდიტორია, სადაც ხდება კლიენტების მიერ საწარმოების პროდუქციის ვიზუალური დათვალიერება და მოთხოვნების განსაზღვრა (შესაძლებელია მათ კომპიუტერულ საინფორმაციო სისტემიდან

მიიღონ გასაყიდ პროდუქციაზე დამატებითი ტექნიკურ-ეკონომიკური მონაცემები).

კლიენტი აზუსტებს შესაკვეთ პროდუქციის ასორტიმენტს (სიას). ამის შემდეგ ორი პარალელური ოპერაციის შესრულებაა შესაძლებელი: შესაკვეთ პროდუქციის მოცულობების დადგენა (რაოდენობრივ და ღირებულებით ასპექტებში) და პროდუქციის მიწოდების კლიენტისათვის სასურველი რეგლამენტის დადგენა.

თუ კლიენტისათვის მისაღებია მის მიერ შერჩეული ვარიანტი, იგი აკეთებს განაცხადს და გადასცემს მარკეტინგის განყოფილების მენეჯერს. ე.ი. მართვა გადაეცემა საწარმოო ფირმის წარმომადგენელს. იგი შეამოწმებს პროდუქციის მონაცემთა ბაზაში სალიმიტო მონაცემებს (თითოეული პროდუქციისათვის ამ კლიენტის განაცხადიდან მოცულობით და დროით პარამეტრებში).

თუ მოთხოვნა მისაღებია (შესაძლებელია ყველა პროდუქციის შეკვეთა), მაშინ მარკეტინგის მენეჯერი შეადგენს კონტრაქტის ფორმას (სპეციფიკაციას), დაამტკიცებს მას ხელმოწერითა და ბეჭდით. ორ ეგზემპლარს გადასცემს შემკვეთ-კლიენტს (ერთი კომერციული ბანკისათვის მომავალი ანგარიშსწორების გასაფორმებლად), ერთს კომპიუტერულ სისტემას – მონაცემთა ბაზაში დასაფიქსირებლად და ცვლილებების შესატანად. ნახაზის ქვედა ნაწილში მოცემულია ორი კომენტარის „კონტრაქტი“ და „კონტრაქტების ფაილი“ მაგალითი. ასევე უნდა აიგოს აქტიურობათა დიაგრამები სხვა შემთხვევებისთვისაც. მაგალითად, როცა ლიმიტების ბაზა ვერ აკმაყოფილებს კლიენტის მოთხოვნებს და საჭიროა სხვა ალტერნატიული ვარიანტის შერჩევა. ან ხდება კონტრაქტის გაუქმება გარკვეული მიზეზებით და ა.შ.

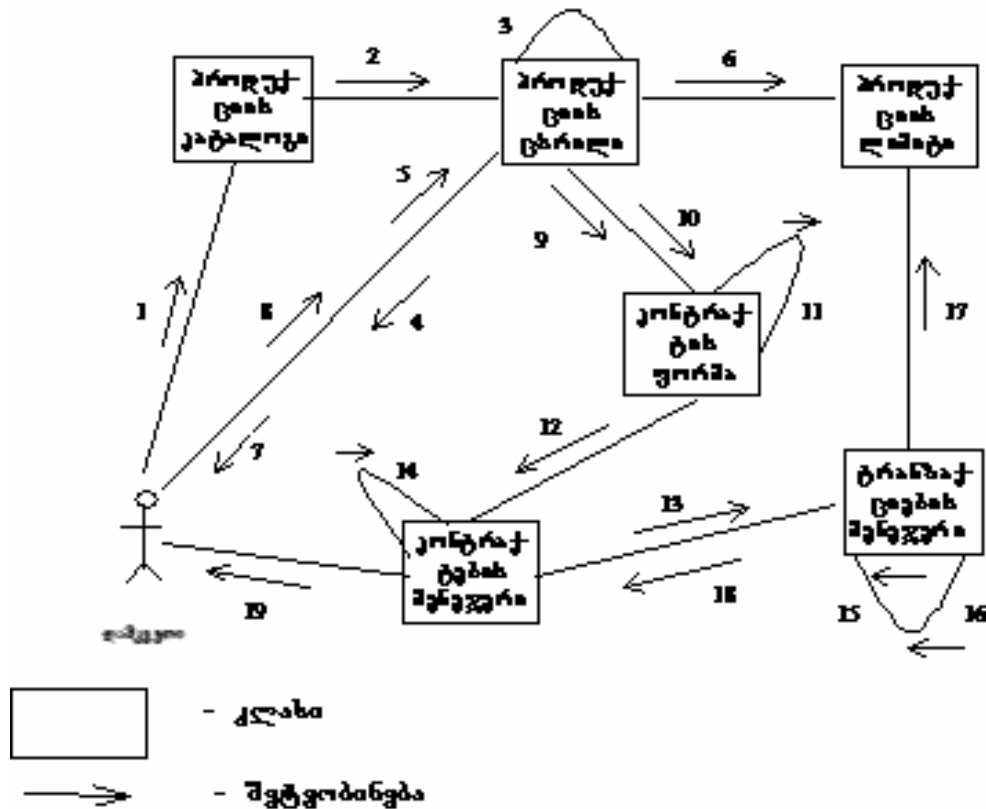


ნახ. 3.6. მიმღებრობის დიაგრამა

მომდევნო ეტაპზე საჭიროა გამოვიყენოთ სისტემის ობიექტებს შორის ურთიერთმოქმედების (Interaction) მოდელირების მეთოდები.

3.6 და 3.7 ნახაზებზე წარმოდგენილია „კონტრაქტის დადების“ გამოყენებითი შემთხვევის შესაბამისი მიმდევრობითობისა და თანამოქმედების დიაგრამები. ეს დიაგრამები გამოიყენებს ერთი და იმავე ელემენტებს, ოღონდაც მათი გამოსახვის სტრუქტურები განსხვავებულია.

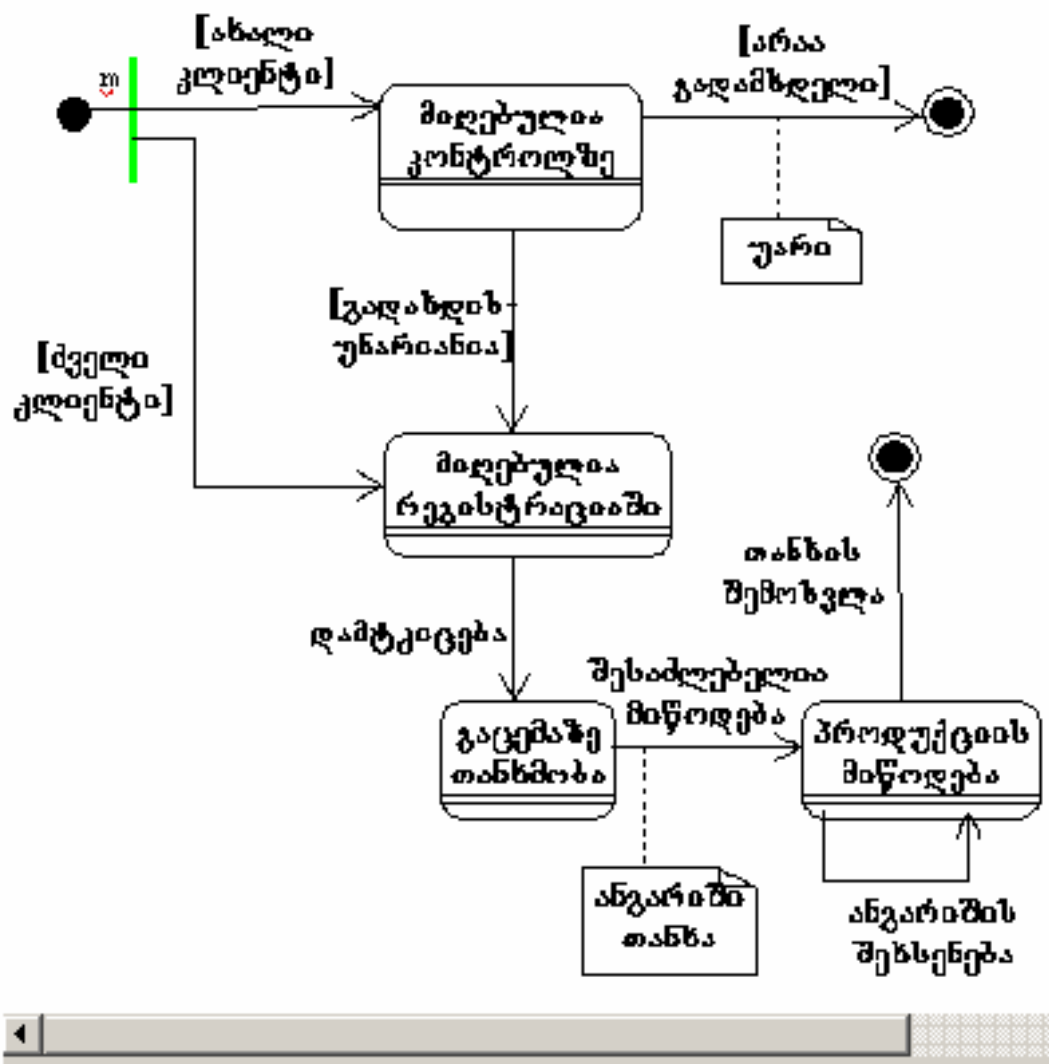
მომდევნო ეტაპზე აიგება კლასთა დიაგრამები, რომლებიც ითვლება სისტემათა სტრუქტურების დაპროექტების ინსტრუმენტად. კლასების გამოვლენა და მათი კონსტრუირება შესაძლებელია გამოყენებითი შემთხვევების სცენარების მოვლენათა ნაკადებისა და ურთიერთმოქმედების დიაგრამათა საფუძველზე. მოვლენათა ნაკადებში საჭიროა გაანალიზდეს არსებითი სახელით მოცემული ყველა ცნება.



ნახ. 3.7. თანამოქმედების დიაგრამა

მდგომარეობათა დიაგრამის (State Diagram) აგება მეტად მნიშვნელოვანია დინამიკური პროცესების გამოსაკვლევად. აქტიურობათა დიაგრამების მსგავსად, მაგრამ შედარებით იშვიათად, უფრო რთულ შემთხვევებში გამოიყენება აღნიშნული - მდგომარეობათა დიაგრამები (State Diagram). ეს მაშინ, როცა კლასის ობიექტს შეუძლია არსებობა რამდენიმე მდგომარეობაში და თითოეულში მისი ყოფაქცევა განსხვავებულია.

3.8 ნახაზზე მოცემულია მდგომარეობათა დიაგრამის აგების საილუსტრაციო მაგალითის ფრაგმენტი „კონტრაქტით პროდუქციის



ნახ.3.8. მდგომარეობათა დიაგრამა „პროდუქციის მიწოდება კონტრაქტით“

მიწოდებაზე“. კლიენტის მოთხოვნის შემოსვლისას მარკეტინგის განყოფილებაში ორი მდგომარეობა განიხილება (ერთი ძველი ანუ „ნაცნობი“ კლიენტის შემთხვევა, მეორე – ახალი ანუ „უცნობი“ კლიენტის შემთხვევა).

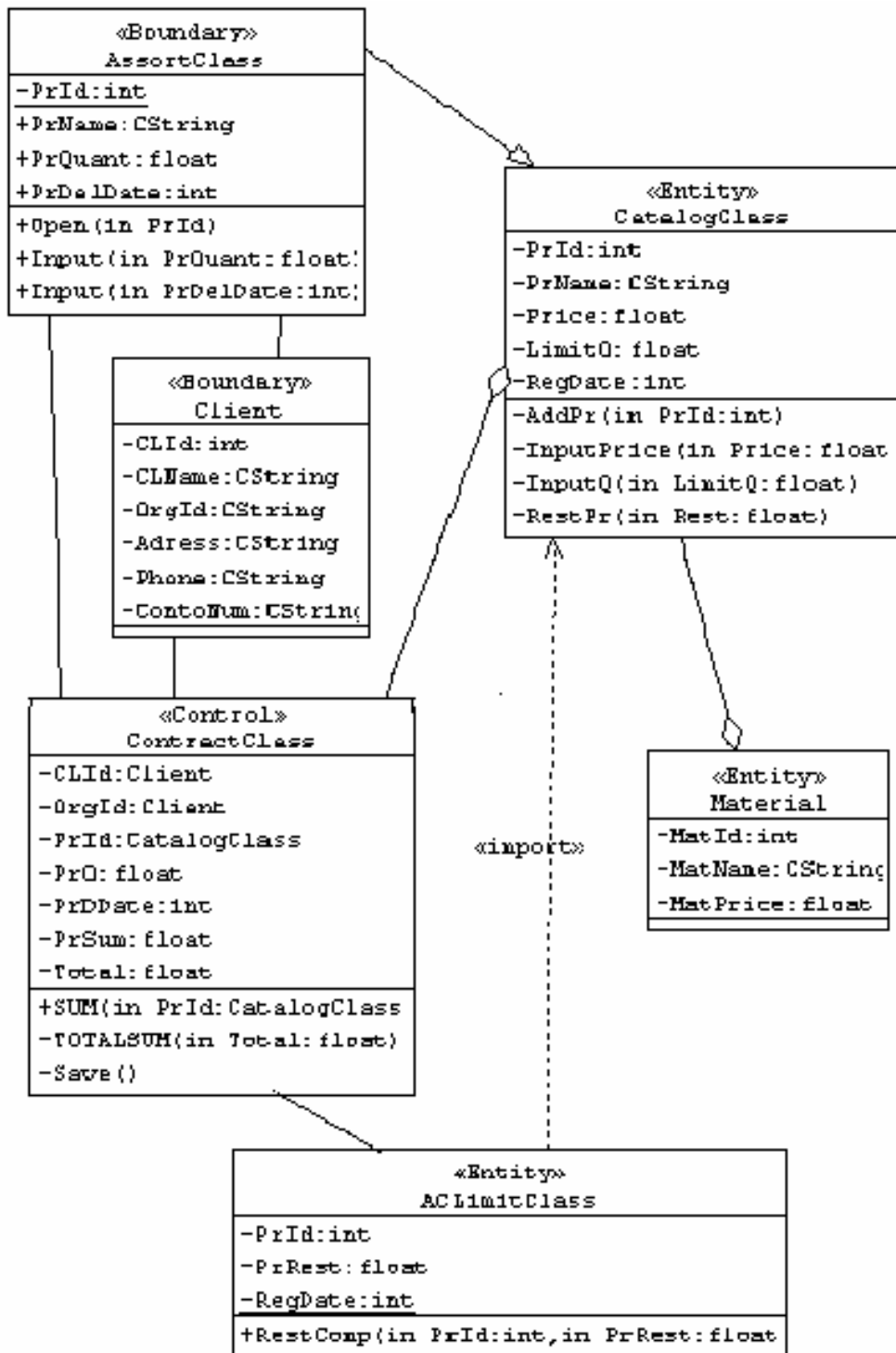
პირველ შემთხვევაში სისტემის ყოფაქცევა ასეთია: ფიქსირდება კლიენტის მოთხოვნები ასორტიმენტის, მოცულობებისა და მიწოდების ვადების მონაცემების საფუძველზე. მტკიცდება მისი გაცემის შესაძლებლობა, ხდება ჯამური თანხების გაანგარიშება და პროდუქციის მომზადება კლიენტზე გადასაცემად. ბოლოს პროდუქცია მიეწოდება კლიენტს, რომელიც იხდის (ან გადმორიცხავს) შესაბამის თანხებს. თუ გადახდა არ მოხდა, მაშინ კლიენტთან იგზავნება შეხსენება.

მეორე შემთხვევაში სახეზეა „ახალი“ კლიენტი. საწარმოო ფირმა ვალდებულია სწრაფად გამოიკვლიოს კლიენტის გადახდისუნარიანობა და მხოლოდ ამის შემდეგ გადაწყვიტოს – გასცეს თუ არა პროდუქცია ავანსად ამ კლიენტისათვის. როგორც ნახაზიდან ჩანს, დიაგრამას ერთი დასაწყისი და ორი დასასრული აქვს.

3.9 ნახაზზე საილუსტრაციო მაგალითის ფრაგმენტი ამოცანისათვის „კონტრაქტი“ საპრობლემო სფეროსათვის „მარკეტინგი“.

3.9 ნახაზზე ჩანს კლასები: კლიენტების (Client), მოთხოვნების ასორტიმენტების (AssortClass), პროდუქციის კატალოგის (CatalogClass), პროდუქციის ლიმიტების (ACLimitClass), მასალების (Material) და კონტრაქტების (ContractClass) მონაცემთა ფაილების შესანახად.

კლასთა შორის რეალიზებულია ურთიერთკავშირები, რომელთა ტიპები შეიარჩევა მათი ანალიზის საფუძველზე. ისინი



ნახ.3.9. კლასთა-ასოციაციის დიაგრამა

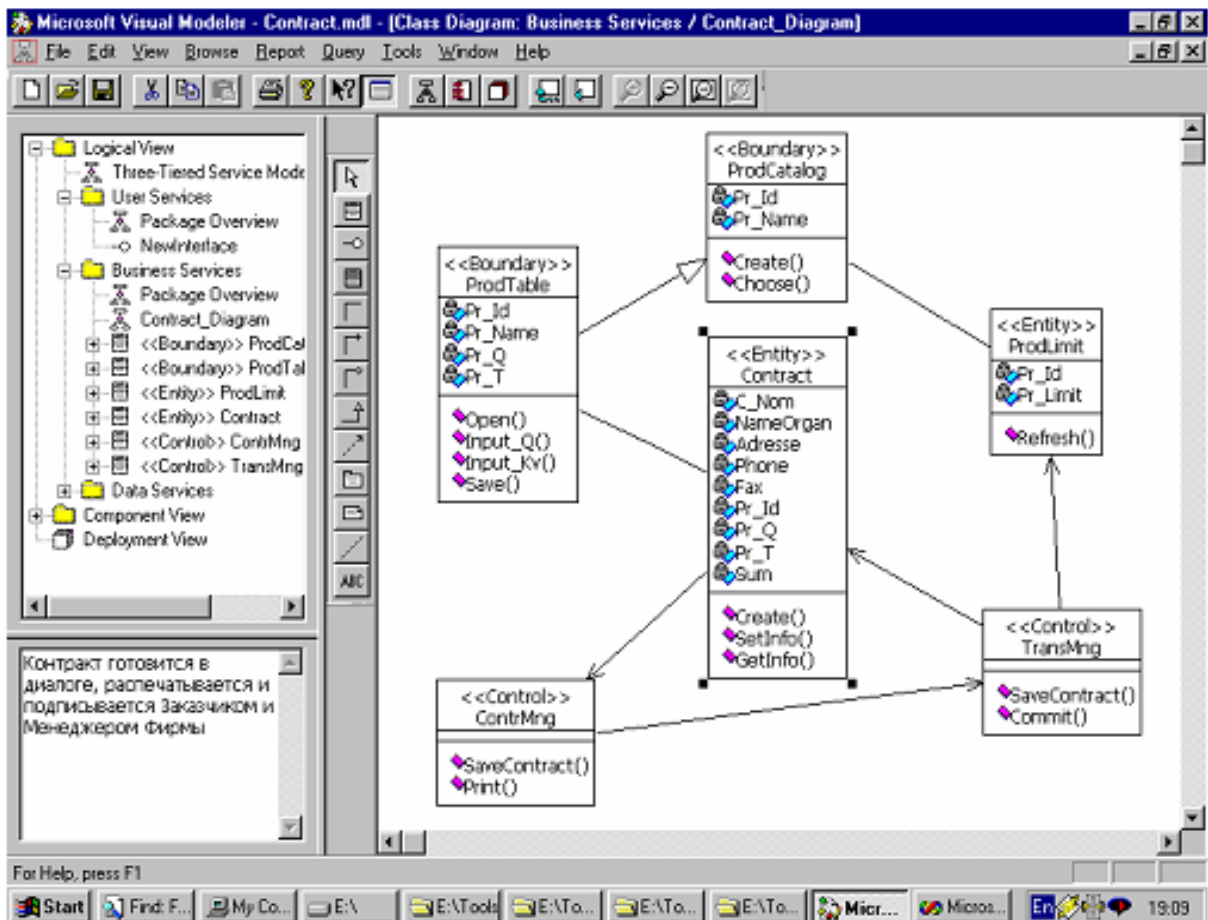
ასოციაციური, მემკვიდრეობითი, აგრეგაციული ან სხვა სახისაა. ეს თვით სისტემის დამპროექტებელმა უნდა გადაწყვიტოს.

კლასების დიაგრამის საფუძველზე გადავდივართ სისტემის რეალიზაციის (Implementation) ეტაპზე. ესაა პროგრამული კოდის



ავტომატიზებული აგების ეტაპი რომელიმე თანამედროვე დაპროგრამების ინსტრუმენტში (მაგ., C#, VB, C++, Java).

კლასების დაჯგუფება ხდება პაკეტებში. უპირატესად გამოიყენება სამი სტერეოტიპი: არსი (Entity), მოსაზღვრე (Boundary) და მმართველი (Control). 3.10 ნახაზზე მოცემულია კლასთა დიაგრამების ფრაგმენტები. იგი შეესაბამება „კონტრაქტის დადების“ გამოყენებითი შემთხვევის კლასთა დეტალიზებულ დიაგრამას ასოციაციებით.

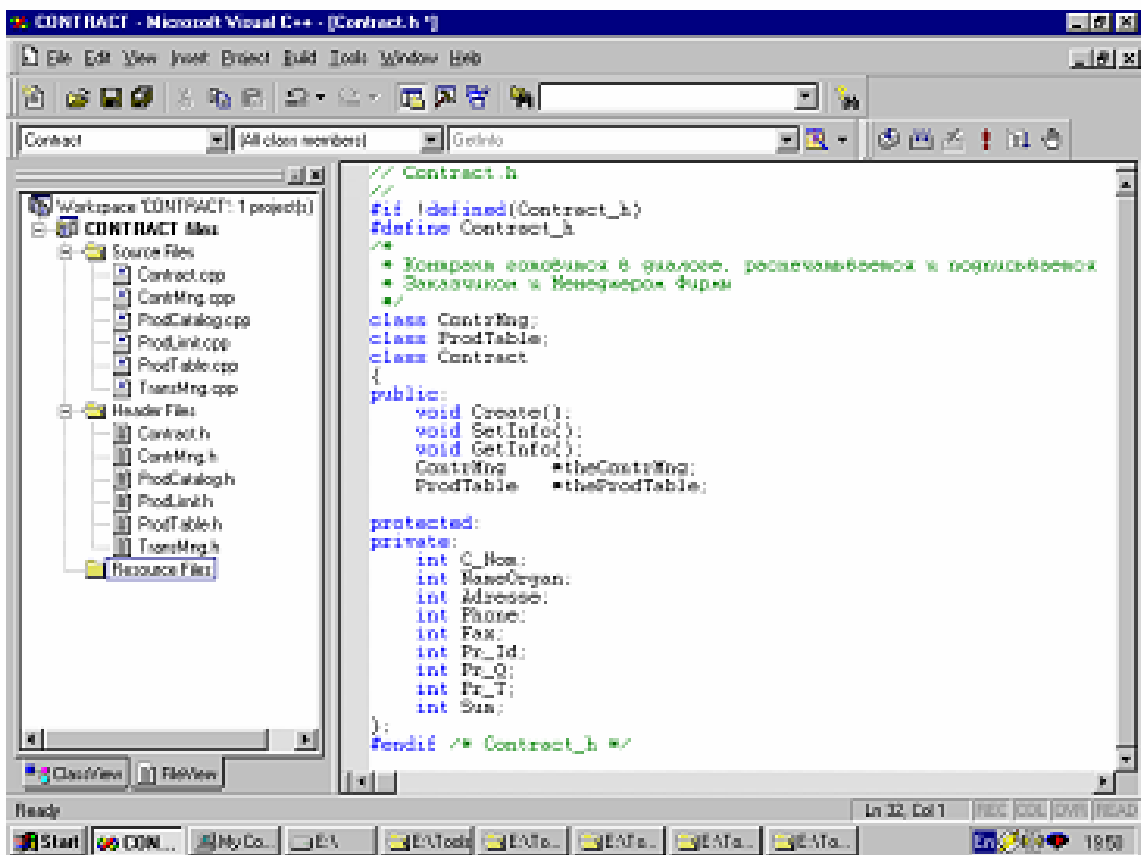


ნახ.3.10. Visual Basic-ის Modeler ინსტრუმენტი

ბოლო ეტაპი არის პროგრამული კოდის რეალიზაცია მიღებული კლასთა დიაგრამებიდან. ამ დროს საჭიროა მოდელირების პროგრამულ ინსტრუმენტს, მაგ., Rational\_Rose, Paradigm\_Plus, Ms VISIO, Visual Modeler მიეწოდოს ენის შესარჩევი პარამეტრი (C#, C++, Visual\_Basic, Java, Oracle, Power Builder ან სხვ.). ამჯერად ჩვენ

გამოვიყენეთ Visual Modeler-ში Visual\_C++ კოდის გენერირების მოთხოვნა [10].

3.11 და 3.12 ნახაზებზე მოცემულია ჩვენი მაგალითისთვის მიღებული პროგრამული კოდის ფრაგმენტები. აქ ნაჩვენებია თუ როგორ გარდაქმნის ვიზუალური დაპროგრამების ინსტრუმენტი კლასთა დიაგრამებს კლასების, ატრიბუტების, ოპერაციებისა და კავშირების ასოციაციების საფუძველზე VC++ პროგრამულ კომპონენტებად: h, cpp, exe და სხვა ტიპის ფაილებად.



ნახ. 3.11 Contract.h ფაილი Visual C++ გარემოში

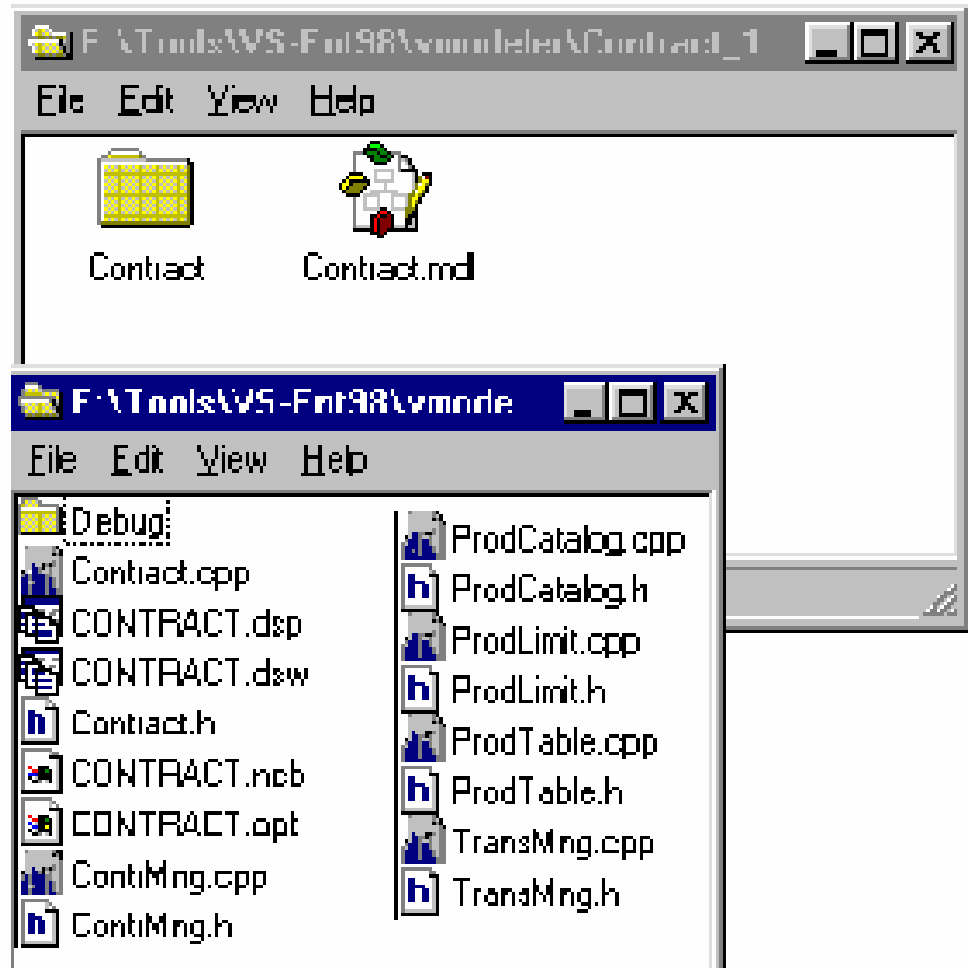
```
// Contract.cpp პროგრამული კოდი
#include "Contract.h"
#include "ContrMng.h"
#include "ProdTable.h"
void Contract::Create()
{
}
ა)
```

```

void Contract::SetInfo()
{
}

void Contract::GetInfo()

```



ბ)

ნახ. 3.12 ჩონტრაქტ.ცკპ პროგრამის ფრეიმი (ა) და პროგრამული პაკეტის კომპონენტების ღირექტორია (ბ)

ჩვენს მიერ გამოკვლეული იყო აგრეთვე საპრობლემო სფერო „უნივერსიტეტი“, რომლის დაპროექტებისა და რეალიაციის მათლები მოცემულია დანართში №2.

### 3.2 ოზის-სისტემების ინფორმაციული ბაზების აბეზა

#### Ms SQL Server პაკეტი

Ms SQL Server პაკეტი მონაცემებთან მიმართვისათვის იყენებს ოთხ ძირითად ინტერფეისს: OLE DB, ODBC, DB Library და Transact-SQL. მომხმარებლისათვის, რომელიც მუშაობს Windows-სისტემასთან, ეს ინტერფეისები რეალიზებულია დინამიკურად მიერთებადი ბიბლიოთეკის, .DLL-ფაილების სახით. Web-კლიენტებისთვის ქსელური ბიბლიოთეკის გამოძახება ხდება IPC (Interprocess Communication) კომპონენტებით.

SQL Server შედგება ოთხი ძირითადი კომპონენტისგან:

- Open Data Services SQL Server - უზრუნველყოფს ინტერფეისს ქსელურ ბიბლიოთეკებსა და თვით MSSQL Server-ის ბირთვის შორის;
- MSSQLServer - მართავს მონაცემთა ბაზის ყველა ფაილს, ამუშავებს მომხმარებელთა მოთხოვნებს, ანაწილებს სისტემურ რესურსებს, ამოწმებს მომხმარებელთა სააღრიცხვო ჩანაწერებს;
- SQLServer Agent - ახორციელებს დავალებათა დაგეგმვას და SQLServer მოვლენათა დამუშავების ავტომატიზაციას;
- MSDTC (Microsoft Distributed Transaction Coordinator) - როგორც განაწილებული ტრანზაქციების კოორდინატორი იგი მართავს მოთხოვნების შესრულებას მონაცემთა ბაზების რამდენიმე სერვერთან. MSDTC სერვისი შეიძლება ამუშავდეს როგორც SQL Server ბირთვიდან, ასევე კლიენტთა გამოყენებითი სისტემიდან.

ნახ. 3.13 და 3.15 ნაჩვენებია მონაცემთა ბაზების ფრაგმენტი ამოცანისათვის „კონტრაქტის დადება“.

ნახაზებზე 3.14 და 3.16 ნაჩვენებია მონაცემთა ბაზების ფრაგმენტი ამოცანისათვის „ტენდერში მონაწილეობის მიღება“, კერძოდ ტენდერის კომისიის წევრებისათვის.

Column Name	Data Type	Length	Allow Nulls
Tanam_num	int	4	
Firm_num	char	8	✓
GSM	char	30	✓
Tanamdeboba	char	30	✓
misamarTi	char	20	✓
Tel	char	10	✓
Mobiluri	char	10	✓
DabWeli	datetime	8	✓
Xelfasi	money	8	✓
Ganableba	char	15	✓
Sqesi	char	4	✓
SheniSvna	char	100	✓

Column Name	Data Type	Length	Allow Nulls
Nedl_num	char	10	
Nedl_dasax	char	10	✓
Masala	char	10	✓
Mimwod_num	int	4	✓
Erteulis_pasi	money	8	✓

Column Name	Data Type	Length	Allow Nulls
Contr_num	char	8	
Contr_name	image	16	
Contr_date	datetime	8	
Mimwod_num	int	4	
operac_code	int	4	
Firm_num	char	20	

Column Name	Data Type	Length	Allow Nulls
Firm_num	char	8	
Veli	char	10	
Qalaqi_num	char	10	
MisamarTi	char	10	
Tel	nchar	10	
Fax	nchar	10	✓

Column Name	Data Type	Length	Allow Nulls
Firm_num	char	8	
Contr_num	char	8	
Prod_code	int	4	✓
raodenoba	char	10	✓
TariRi	char	10	✓
Pasi	char	10	✓

6.3b. 3.13

Column Name	Data Type	Length	Allow Nulls
Firm_ID	int	4	
Weli	int	4	
Kap_Daband	money	8	✓
krediti	money	8	✓
Investicia	money	8	✓
Mater_danaxarj	money	8	✓
Mogeba	money	8	✓

Column Name	Data Type	Length	Allow Nulls
SamuC_ID	int	4	
Saxeli	char	100	✓
Firm_ID	int	4	✓
Dawyeb_Date	datetime	8	✓
Damtavr_Date	datetime	8	✓
Damkveti	char	25	✓

Column Name	Data Type	Length	Allow Nulls
Firm_ID	int	4	
CV_TanamSr	char	1	✓
Saj_Reestri	char	1	✓
Samew_Reestri	char	1	✓
SasamarTlo	char	1	✓
Sagadasaxado	char	1	✓
Tender_winadadeba	char	1	✓
Coprprac_Struct	char	1	✓
Company_Sacmian	char	1	✓

Column Name	Data Type	Length	Allow Nulls
Firm_ID	int	4	
Prod_ID	int	4	
Raad	int	4	✓

Column Name	Data Type	Length	Allow Nulls
Firm_ID	int	4	
Prod_Saxeli	char	50	
Modelis_ID	char	10	✓
ErTeuli	char	10	✓
Fasi	money	8	✓
SeniSvna	char	100	✓

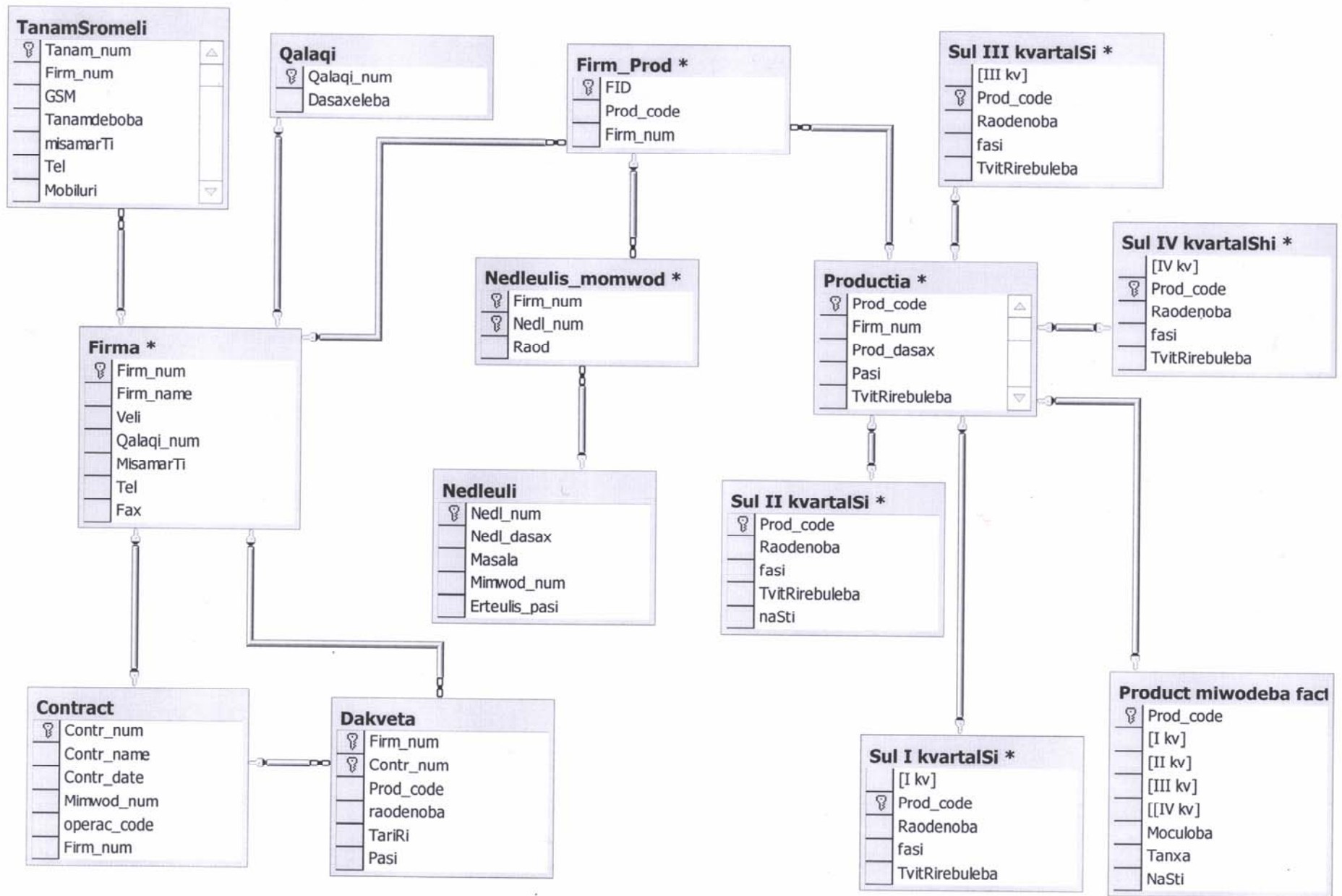
  

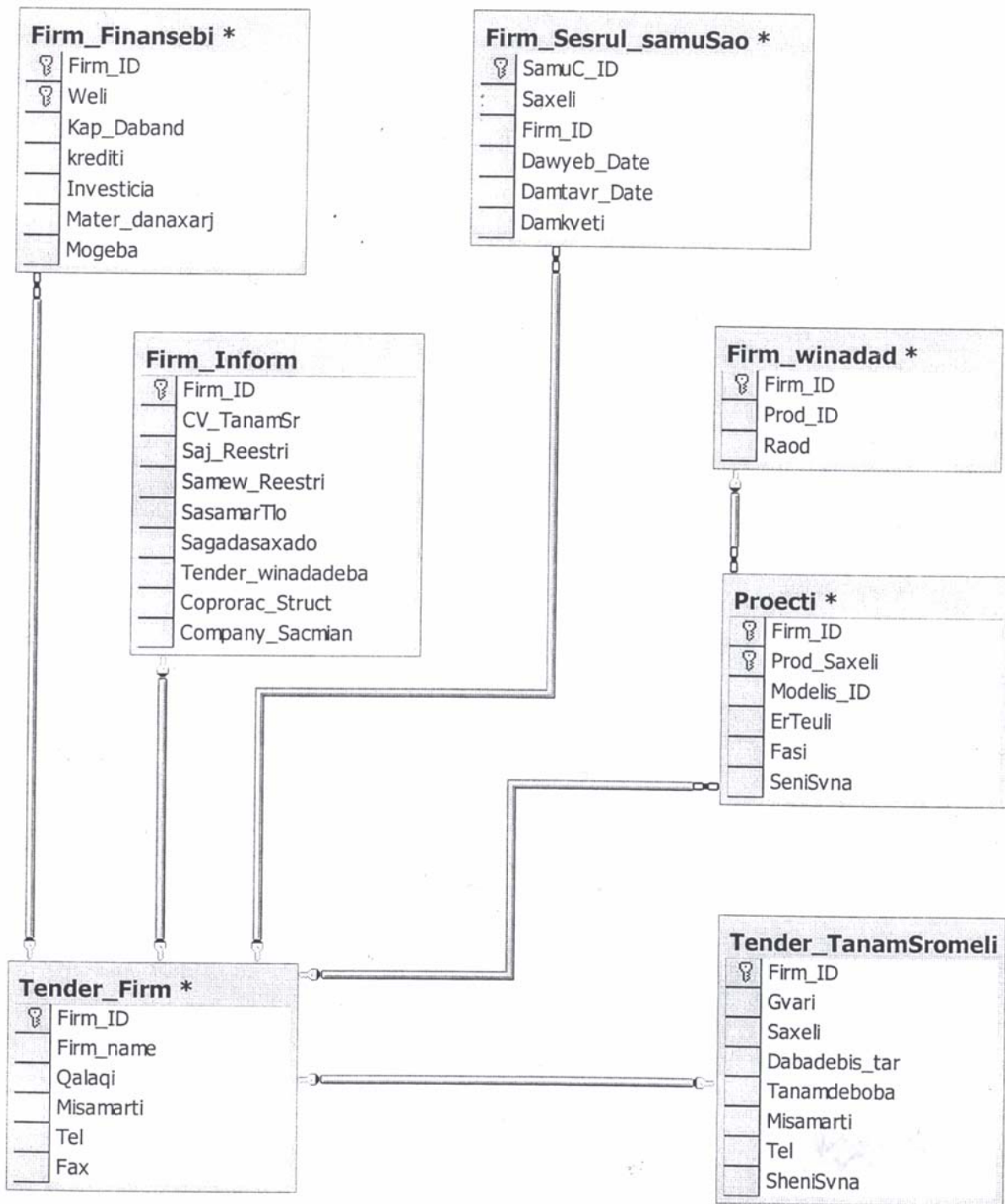
Column Name	Data Type	Length	Allow Nulls
Firm_ID	int	4	
Gvari	char	10	✓
Saxeli	char	10	✓
Dabadebis_tar	datetime	8	✓
Tanamdeboba	char	20	✓
Misamarti	char	40	✓
Tel	char	10	✓
SheniSvna	char	100	✓

Column Name	Data Type	Length	Allow Nulls
Firm_ID	int	4	
Firm_name	char	30	✓
Qalaqi	char	10	✓
Misamarti	char	30	✓
Tel	char	10	✓
Fax	char	10	✓

6.3b.3.14







### 3.3 SQL –ბაზებისათვის .DDL ფაილების მომზადება და ინსტრუქციები XML დოკუმენტის მისაღებად

მონაცემთა განაწილებული ბაზის დასაპროექტებლად, როგორც ეს UML-ტექნოლოგიით მოითხოვება, საჭიროა საპრობლემო სფეროს შესწავლა და მისი ტექსტური (არაფორმალური) აღწერა, აქ არსებული მიზნების, ამოცანების, ობიექტების, ფაქტორების და სხვა ელემენტების, აგრეთვე მათი ურთიერთკავშირების დეტალური ჩამოყალიბებით.

ესაა ცოდნა სამართავი ობიექტის შესახებ, მისი სტატიკური (მდგომარეობათა სიმრავლე) და დინამიკური (ქცევათა სიმრავლე) მოდელებით.

თუ ობიექტ-ორიენტირებული მოდელირების ტერმინებით ვისარგებლებთ, დასმული ამოცანის გადაწყვეტის „გასაღებს“ კლასების, ობიექტების, კლასთაშორისი კავშირების, ობიექტ-როლური და არსთა-დამოკიდებულების მოდელებისა და სხვა სახის დიაგრამების აგება წარმოადგენს. შემდეგ კი კლასთა-ასოციაციებისა და არსთა-დამოკიდებულების დიაგრამათა საფუძველზე განხორციელდება მიზნობრივი სისტემის პროგრამული კოდების რეალიზაციის ავტომატიზებული პროცესი.

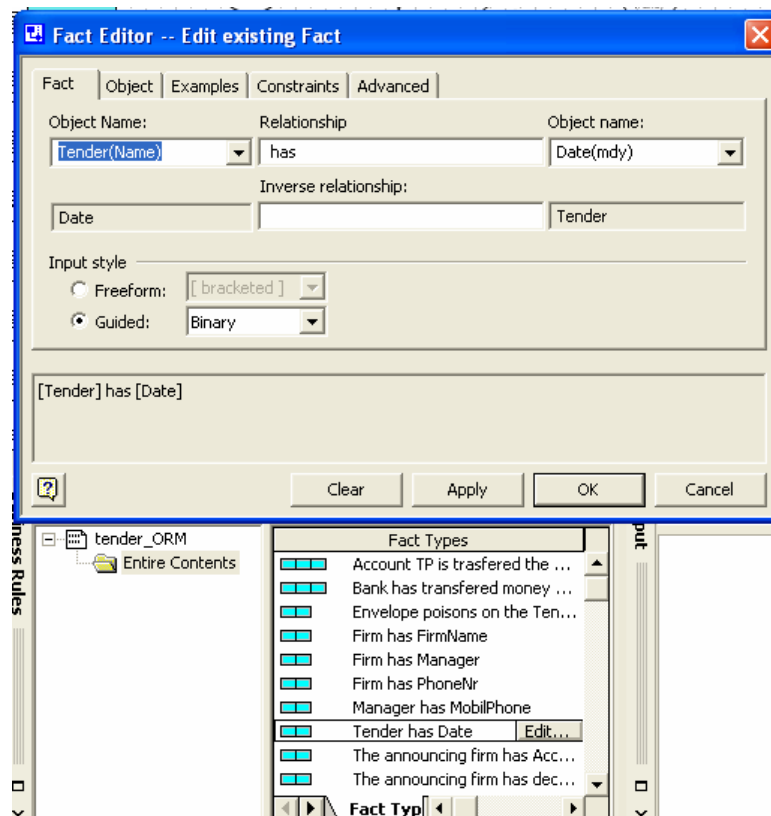
განვიხილოთ ეს საკითხები დეტალურად ტენდერის ჩატარების მაგალითზე.

კონცეპტუალური მოდელის დაპროექტებისათვის გამოყენებულია ობიექტ-როლური მოდელირება (ORM), რომლისთვისაც აუცილებელია საპრობლემო სფეროს აღწერა ფაქტების საშუალებით : „ზოგადად, გვაქვს ფირმა, რომელიც მონაწილეობას იღებს ტენდერში, ფირმა, რომელიც აცხადებს ტენდერს და ასევე გამარჯვებული ფირმა. ფირმას აქვს რეკვიზიტები და ჰყავს მენეჯერი, აქვს საკუთარი ანგარიშის ნომერი



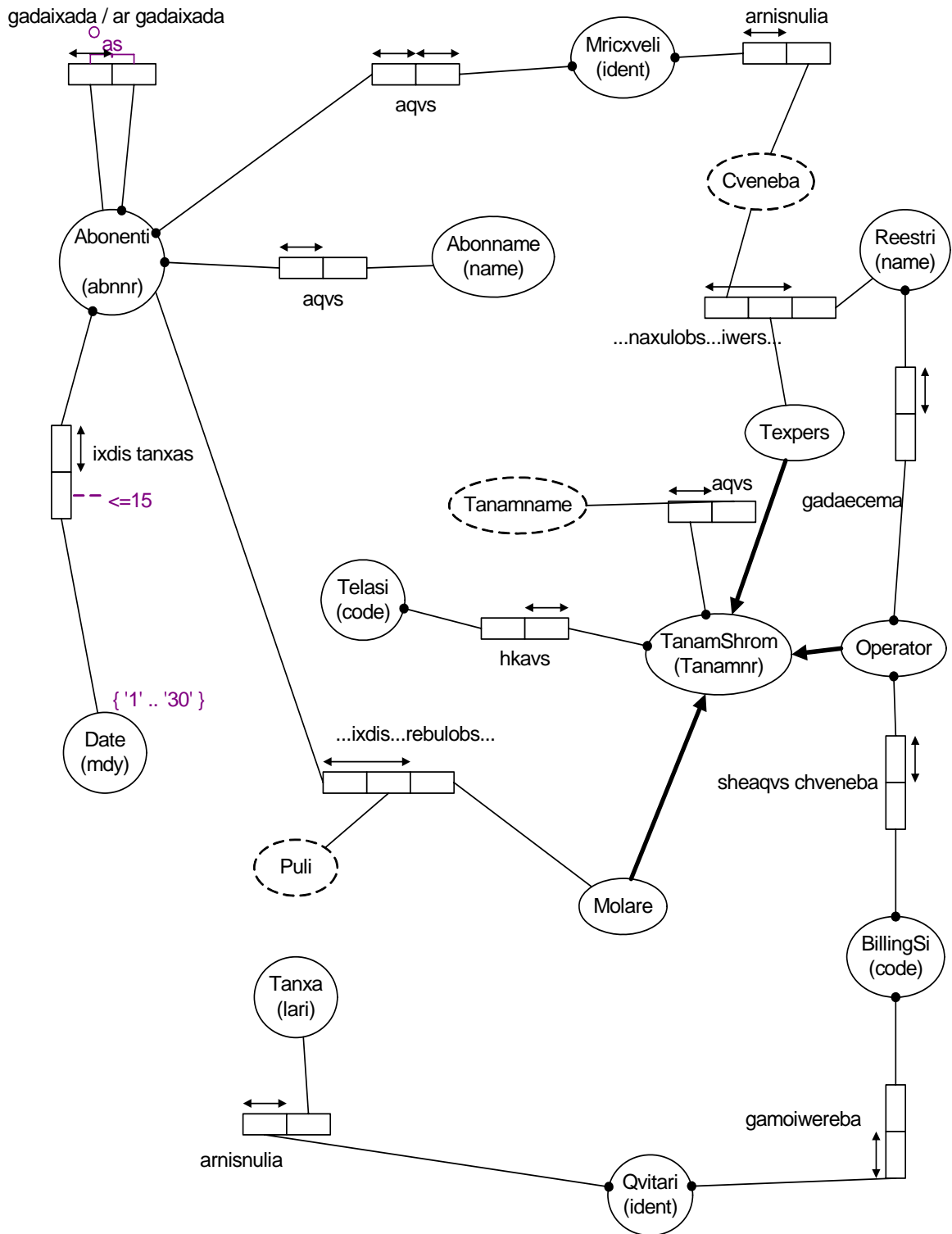
ბანში. ტენდერში მონაწილეობის მისაღებად საჭირო დოკუმენტები თავსდება კონვერტში, ილუქება და იგზავნება ტენდერის მოთხოვნებში მითითებულ მისამართზე. აქვე მითითებულია ტენდერის ჩატარების თარიღი და დრო. ამ დღეს, ტენდერში მონაწილეობის მიღების მსურველი ყველა ორგანიზაციის წარმომადგენელი, ვაღდებულება გამოცხადდეს მითითებულ მისამართზე. ყველას თვალწინ, სატენდერო კომისიის წევრები, გახსნიან დალუქულ კონვერტებს, გაეცნობიან შემოთავაზებულ წინადადებებს, გადაამოწმებენ საბუთებს, მსჯელობის შემდეგ ამოიხევენ საუკეთესო წინადადებას და შემდგომ დაასახელებენ ტენდერში გამარჯვებულ ორგანიზაციას. გამარჯვებული ფირმის ანგარიშზე გადაირიცხება სამუშაოს დაწყებისათვის აუცილებელი თანხა.“

პირველ ეტაპზე ხდება ზემოხამოთვლილი ფაქტების შეტანა Ms\_Visio ინსტრუმენტის საშუალებით. იხ. ნახაზი 3.17



ნახ.3.17

მეორე ეტაპზე საპრობლემო სფეროს შესახებ ცოდნა (კლასებისა და ობიექტების ძირითადი ტერმინები და ქცევის წესები) ჩვენს მიერ გადაიტანება Ms\_Visio ინსტრუმენტის საშუალებით ობიექტ-როლურ მოდელში (ნახ.3.18 ).



ნახ.3.18

ობიექტებისა და მათ შორის კავშირების (წესების) მომდევნო ეტაპზე განხორციელდება ობიექტ-როლური მოდელის ავტომატური გადაყვანა არსთა-დამოკიდებულების მოდელში.

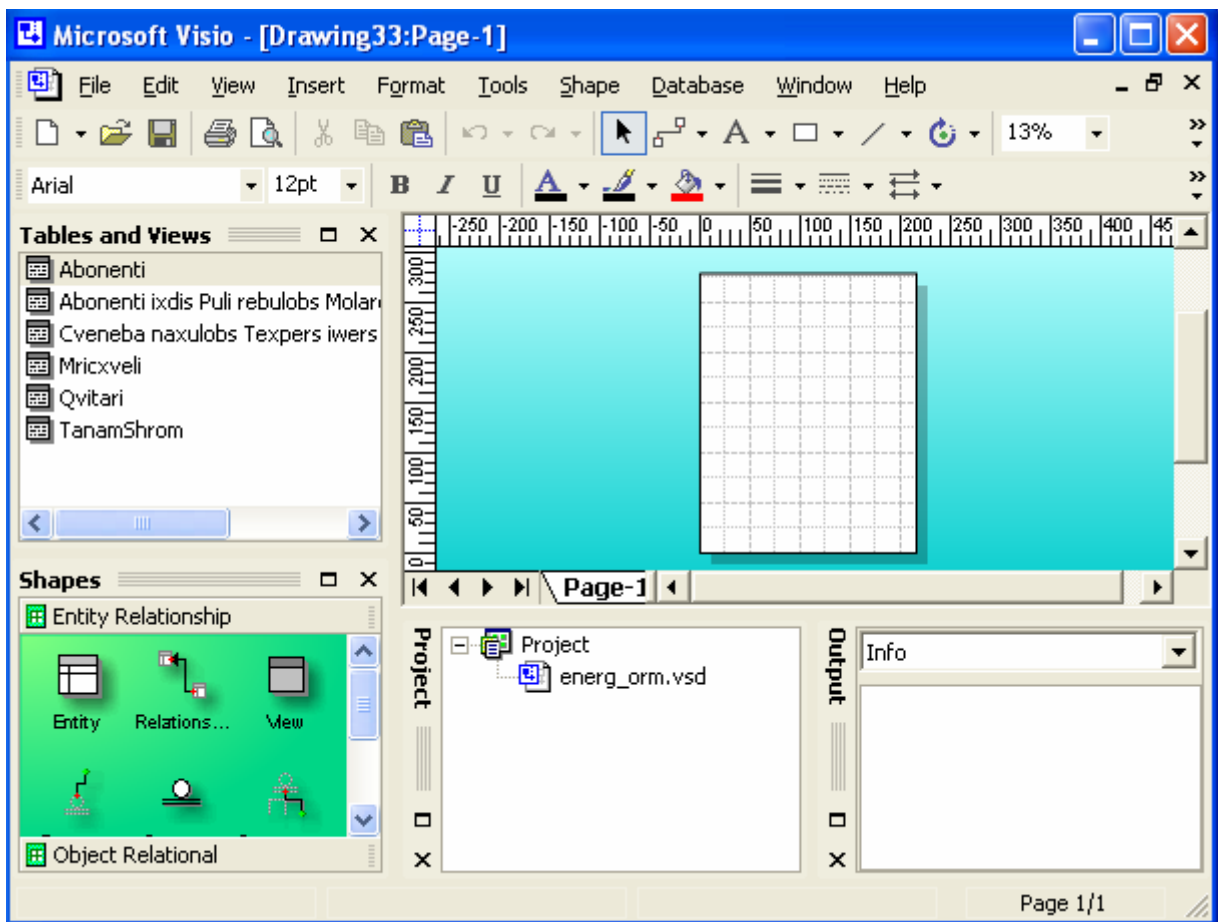
ამისათვის Ms\_Visio ინსტრუმენტის (ORM-source) მენიუდან გამოვიძახებთ სტრიქონს :

File | New | Database | Database Model Diagram

და ახალი ფანჯრის სუფთა ფურცელზე გადმოვიტანთ წინა ეტაპზე აგებულ ORM-დიაგრამის შესაბამის ER-დიაგრამას. ამისათვის აქ მენიუდან მივუთითებთ:

Database | Project | Add Existing Document

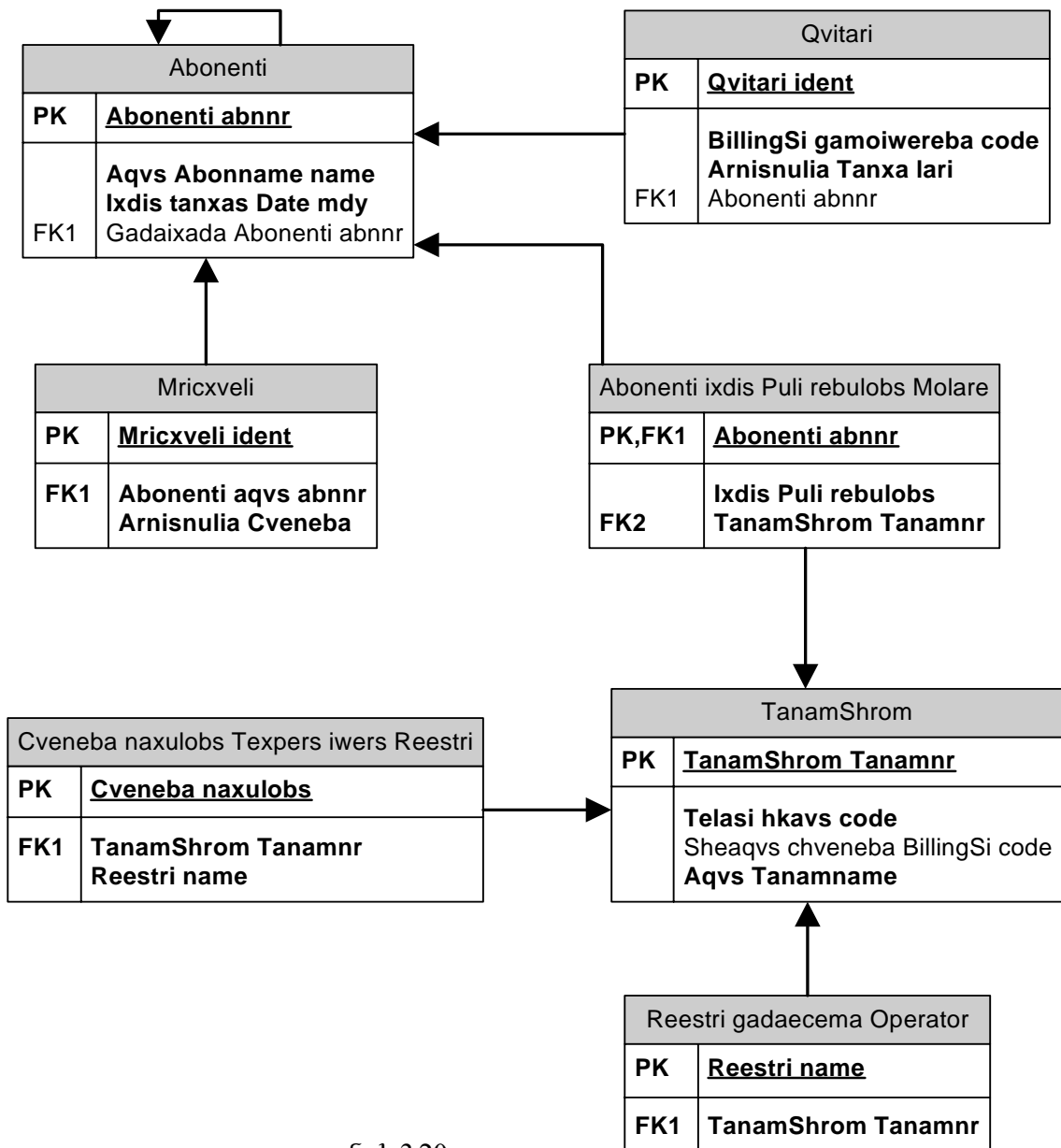
მივიღეთ ნახ. 3.19-ზე ნაჩვენები სურათი.



ნახ 3.19

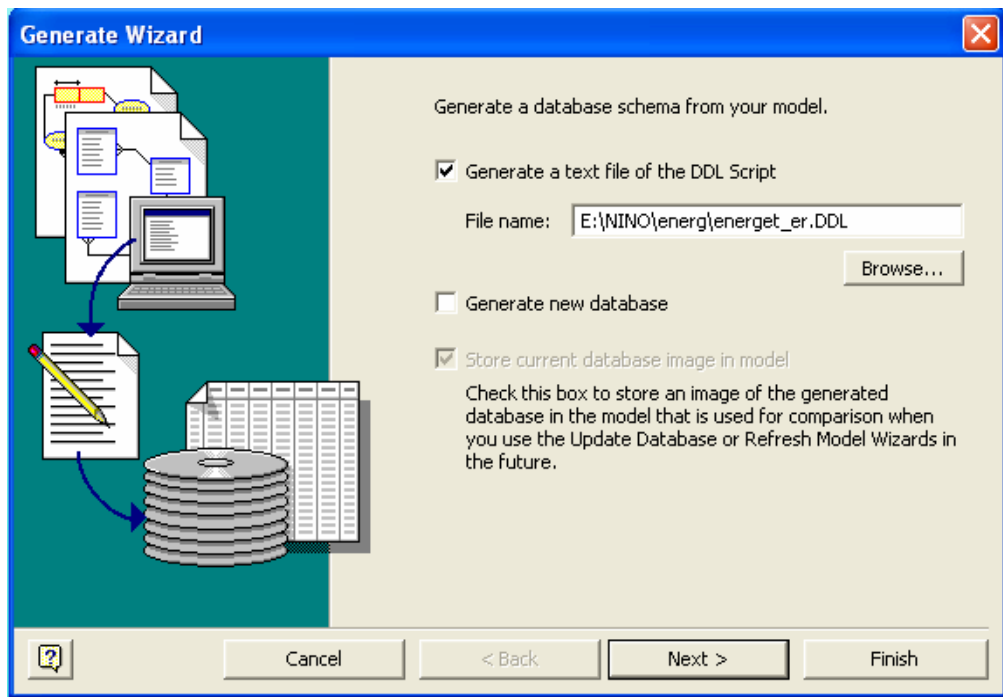
Microsoft Studio.Net საშუალებას გვაძლევს ORM-დიაგრამიდან ავტომატიზებურად ავაგოთ ER-დიაგრამა., რომლის საფუძველზეც

შემდგომში აიგება რელაციურ მონაცემთა ბაზების ლოგიკური სტრუქტურა. იხ ნახ.3.20



ნახ.3.20

ER-დიაგრამის საფუძველზე ასევე ავტომატურად მიიღება .DDL ფაილი, რომელიც ავტომატურად ჯდება SQL შერვერ მონაცემთა ბაზაში. იხილეთ ნახაზი 3.21



ნახ.3.21

ქვემოთ მოცემულია . DDL ფაილის კოდის ფრაგმენტი:

```

/* Create energ database.                                     */
use master
go
create database "energ"
ON PRIMARY
    (NAME = ener, FILENAME = 'c:\data\datafile', SIZE = 3 MB, MAXSIZE = UNLIMITED,
FILEGROWTH = 256 KB )
go
use "energ"
go
/* Create new table "Reestri gadaecema Operator".           */
/* "Reestri gadaecema Operator" : Table of Reestri gadaecema Operator */
/* "Reestri name" : Reestri identifies Reestri gadaecema Operator */
/* Role two (Reestri) of fact: Reestri gadaecema Operator is identified by */
/* Role two (Reestriname) of fact: Reestri is identified by */
/* "TanamShrom Tanamn" : TanamShrom Operator has Reestri gadaecema Operator */
create table "Reestri gadaecema Operator" (
    "Reestri name" char(10) not null,
    "TanamShrom Tanamn" char(10) not null)
go
alter table "Reestri gadaecema Operator"
    add constraint "Reestri gadaecema Operator_PK" primary key ("Reestri name")
go
/* Create new table "Abonenti ixdis Puli rebulobs Molare". */
/* "Abonenti ixdis Puli rebulobs Molare" : Table of Abonenti ixdis Puli rebulobs Molare */
/* "Abonenti abnnr" : Abonenti has Abonenti ixdis Puli rebulobs Molare */
/* Role two (Abonenti) of fact: Abonenti ixdis Puli rebulobs Molare is of */
/* Role two (abnnr) of fact: Abonenti is identified by */
/* "Ixdis Puli rebulobs" : Puli is of Abonenti ixdis Puli rebulobs Molare */
/* "TanamShrom Tanamn" : Molare has Abonenti ixdis Puli rebulobs Molare */
/* TanamShrom is a Molare */
/* Role two (Tanamn) of fact: TanamShrom is identified by */
create table "Abonenti ixdis Puli rebulobs Molare" (
    "Abonenti abnnr" char(10) not null,
    "Ixdis Puli rebulobs" char(10) not null,

```

```

        "TanamShrom Tanamn" char(10) not null)
go
alter table "Abonenti ixdis Puli rebulobs Molare"
    add constraint "Abonenti ixdis Puli rebulobs Molare_PK" primary key ("Abonenti abnr")
go
/* Create new table "Cveneba naxulobs Texpers iwers Reestri". */
/* "Cveneba naxulobs Texpers iwers Reestri" : Table of Cveneba naxulobs Texpers iwers Reestri */
/* "Cveneba naxulobs" : Cveneba identifies Cveneba naxulobs Texpers iwers Reestri */
/* "TanamShrom Tanamn" : TanamShrom has Cveneba naxulobs Texpers iwers Reestri */
/* "Reestri name" : Reestri is of Cveneba naxulobs Texpers iwers Reestri */
create table "Cveneba naxulobs Texpers iwers Reestri" (

```

ახალი საინფორმაციო ტექნოლოგიების გამოყენებით, რომლებიც მოდელირების უნიფიცირებული ენის (UML) ინსტრუმენტებზეა ორიენტირებული, შესაძლებელია მოდელირებისა და დაპროგრამების პროცესების ავტომატიზაცია, რაც საბოლოო ჯამში შესაძლებელს ხდის მონაცემთა ბაზების მართვის სისტემების (მაგალითად, Ms SQL Server, SyBase, Oracle, ADO.NET) და დაპროგრამების საინტერფეისო ენების (მაგალითად, C#.NET, VB.NET) ინტეგრირებული გამოყენებით ვიზუალური მოდელირების პაკეტებთან ერთად (მაგალითად, Ms Visio, Rational Rose, Paradigm Plus) დაპროექტდეს და რეალიზებულ იქნას ხარისხიანად და სწრაფად მართვის ავტომატიზებული სისტემების Windows- და Web-დანართები.

SQL Server-იდან შეგვიძლია მივიღოთ XML-დოკუმენტი SELECT-ინსტრუქციის საშუალებით. განვიხილოთ რამდენიმე შემთხვევა Query Analyzer უტილიტის გამოყენებით. ყურადღება უნდა მივაქციოთ FOR წინადადებას (FOR XML RAW, FOR XML AUTO, FOR XML AUTO, ELEMENTS, FOR XML EXPLICIT). განვიხილოთ მაგალითები ჩვენი შემთხვევისათვის.

3.22 ნახაზზე ნაჩვენებია მოთხოვნა, რომლის შედეგადაც მიიღება პირველი სამი სტრიქონი ცხრილიდან "პროდუქცია", ჩაწერილი XML AUTO ფორმატით.

```

SELECT TOP 3 dasaxeleba, fasi
FROM producti
FOR XML AUTO

```

XML\_F52E2B61-18A1-11d1-B105-00805F4991...

```

<producti dasaxeleba="producti-1 "
  fasi="155.00" />
<producti dasaxeleba="producti-2 "
  fasi="75.00" />
<producti dasaxeleba="producti-2 "
  fasi="95.00" />

```

Grids Messages

ნახ. 3.22

3.23 ნახაზზე მოცემულია იგივე მოთხოვნა, რომელიც შედეგებს ჩაწერს FOR XML AUTO, ELEMENTS ფორმატით.

```

SELECT TOP 3 dasaxeleba, fasi
FROM producti
FOR XML AUTO, ELEMENTS

```

XML\_F52E2B61-18A1-11d1-B105-00805F49916B

```

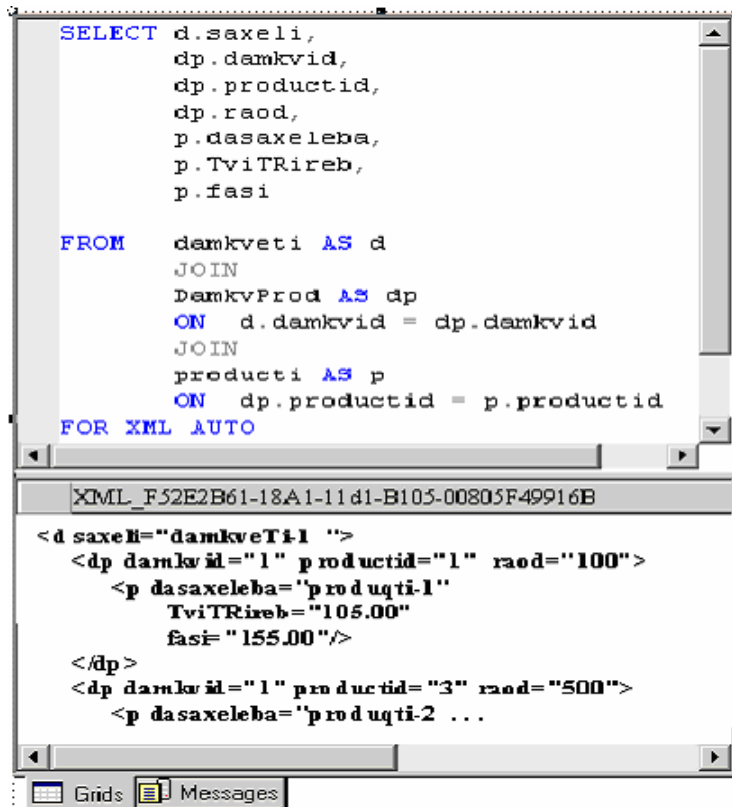
<producti><dasaxeleba>producti-1 </dasaxeleba>
  <fasi>155.00</fasi>
</producti>
<producti><dasaxeleba>producti-2</dasaxeleba>
  <fasi>75.00</fasi>
</producti>
<producti><dasaxeleba>producti-2 </dasaxeleba>
  <fasi>95</fasi>
</producti>
...

```

Grids Messages

ნახ. 3.23

3.24 ნახაზზე ილუსტრირებულია სამი დაკავშირებული ცხრილის მონაცემების საფუძველზე JOIN გაერთიანების ოპერატორის გამოყენების შემთხვევა. გაერთიანების პირობა ჩაწერილია ON წინადადებაში. საბოლოო შედეგები გამოიტანება FOR XML AUTO.



ნახ.3.24

### 3.4. Ms .ADO.NET პროგრამული პაკეტი მონაცემთა ბაზებთან სამუშაოდ

MS ADO.NET (Microsoft ActiveX Data Objects) არის Visual Studio.NET ინტეგრირებული პროგრამული ტექნოლოგია, რომელიც გამოიყენება ინტერნეტში სტრუქტურული მონაცემების გასაცვლელად კლიენტ-სერვერ არქიტექტურისათვის, მონაცემთა ბაზებთან (მაგალითად, Oracle, SQL Server და სხვ.) სამუშაოდ და მომხმარებელთა ინტერფეისების ასაგებად.

SQL Server და ADO.NET პროგრამული პაკეტები, როგორც მონაცემთა ბაზებისა და ბაზებთან მუშაობის მომხმარებელთა ინტერფეისების დამუშავების კლასიკური ინსტრუმენტული საშუალებანი, ფლობს აგრეთვე ჯგუფურ CASE- ტექნოლოგიებს და UML- ენას. მისი საშუალებით ხორციელდება ეფექტური მრავალდონიანი გამოყენებითი სისტემების დამუშავება მონაცემთა ბაზებთან სამუშაოდ ინტრაქსელსა და ინტერნეტში.



სისტემის ობიექტური მოდელი ორი ნაწილისგან შედგება: მარცხენა -მიერთებადი ობიექტები (connected objects) და მარჯვენა- განცალკევებადი ობიექტები (disconnected objects).

მიერთებადი ობიექტები უშუალოდ უკავშირდება მონაცემთა ბაზას, ახორციელებს მასში ტრანზაქციებს, მონაცემთა ძებნას, განახლებას და გადაცემას. განცალკევებადი ობიექტების დანიშნულებაა მონაცემთაბაზებიდან მიღებული ობიექტების (ცხრილები, სტრიქონები, სვეტების და ა.შ.) ავტონომიურად დამუშავება.

ერთ-ერთი მთავარი ტერმინი „მონაცემთა მიმწოდებელი“ (DataProvider), რომლიდანაც სისტემა ღებულობს მონაცემებს. ამგვარად, .NET-ის მონაცემთა მიმწოდებელი არის კლასების ერთობლიობა, რომელიც უზრუნველყოფს მონაცემთა საცავებთან (Repository) ურთიერთქმედებას. ამ სისტემაში მონაცემთა ორი მიმწოდებელია გათვალისწინებული: SQL Client .NET Data Provider, რომელიც SQL Server მონაცემთა ბაზასთან მუშაობს, და OLE DB .NET Data Provider, რომელიც ურთიერთქმედებს მონაცემთა სხვადასხვა საცავებთან OLE DB მიმწოდებლის საშუალებით.

მონაცემთა ყველა მიმწოდებელი ერთიდაიმავე საბაზო კლასებს იყენებს: Connection, Command, DataProvider, Parameter, Transaction, ოღონდაც თავიანთ კონტექსტში. მაგალითად, თუ ის SQL Server-ია, მაშინ გვექნება SqlConnection, თუ OLE DB, მაშინ OleDbConnection.

- ობიექტი Connection გამოიყენება მონაცემთა ბაზასთან ფიზიკური კავშირის დასამყარებლად (ან ამ კავშირის გასაწყვეტად). ობიექტის თვისებებში მიეთითება მონაცემთა წყაროს ტიპი, მისი ადგილმდებარეობა და სხვა პარამეტრები.

- Command ობიექტები გამოიყენება მონაცემთა ბაზასთან SQL-მოთხოვნების მისაწოდებლად, შენახვადი პროცედურების (Stored

Procedures) გამოსაძახებლად, ცხრილებისა (Tables) და წარმოდგენების (Views) მნიშვნელობების მისაღებად, შესაცვლელად და დასაბრუნებლად. მაგალითად, SqlCommand ობიექტს აქვს ExecuteXmlReader მეთოდი, რომელიც მოთხოვნების შედეგებს აბრუნებს XML-ფორმატში.- ობიექტი DataReader გამოიყენება მოთხოვნების საფუძველზე მონაცემთა ბაზიდან ჩანაწერების სწრაფად ამოსარჩევად და დასათვალიერებლად. მისი საშუალებით ჩანაწერებში მონაცემთა ცვლილება არ ხდება.

- ობიექტი Transaction გამოიყენება მონაცემთა ბაზაში ერთდროულად რამდენიმე მიმდევრობითი ცვლილების განსახორციელებლად. ეს რამდენიმე ცვლილება ჯგუფდება როგორც ერთი მთლიანი პროცედურა, რომელსაც ტრანზაქციას უწოდებენ. ობიექტს Connection აქვს მეთოდი BeginTransaction, რომელიც ქმნის Transaction ობიექტს. მისი დანიშნულებაა ჯგუფში შემავალი ცვლილებების განხორციელების დამოწმება ან უარყოფა. თუ, მაგალითად, ჯგუფში არ შესრულდა ყველა დაგეგმილი ცვლილება, მაშინ ტრანზაქცია მონაცემთა ბაზაში არ მოახდენს ნაწილობრივ შეცვლილი ჩანაწერების დაფიქსირებას.

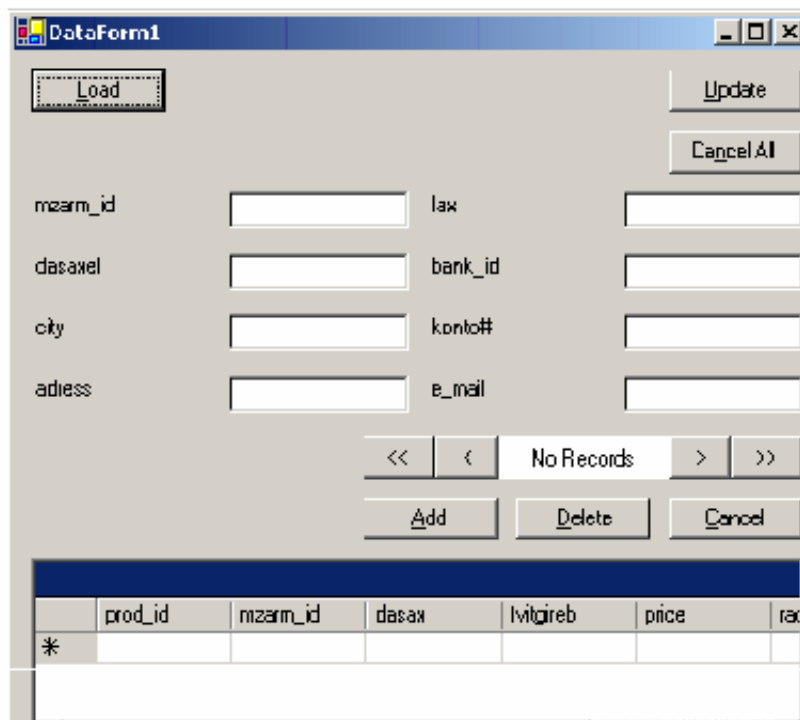
- ობიექტი Parameter გამოიყენება SQL-მოთხოვნის ფორმირების პროცესში Where-კონსტრუქციაში „ ? “ - მარკერის ჩასმით (მაგალითად, Where Client\_Id = ?). ე.ი. შესაძლებელი იქნება ყველა კლიენტისთვის ამ პარამეტრის გამოყენება.

- ობიექტი DataAdapter არის „ხიდი“ მონაცემთა ბაზასა და ADO.NET მოდელის განცალკევებულ ობიექტებს შორის. მაგალითად, მეთოდი DataAdapterFill ახორციელებს მონაცემთა ეფექტურ ამორჩევას ბაზაში შესაბამისი მოთხოვნის საფუძველზე. შემდეგ კი ეს მონაცემები მუშავდება ავტონომიურად DataSet ან DataTable ობიექტებით. ბოლოს, შეცვლილი და გადამუშავებული

მონაცემები DataSet-ის ობიექტიდან DataAdapter-ის საშუალებით ჩაიწერება მონაცემთა ბაზაში. ძირითადი ამოცანა, რომელსაც ჩვენ აქ განვიხილავთ, არის ADO.NET პროგრამული პაკეტის გამოყენებით მომხმარებელთა სამუშაო ინტერფეისის დამუშავების სადემონსტრაციო მაგალითის აგება.

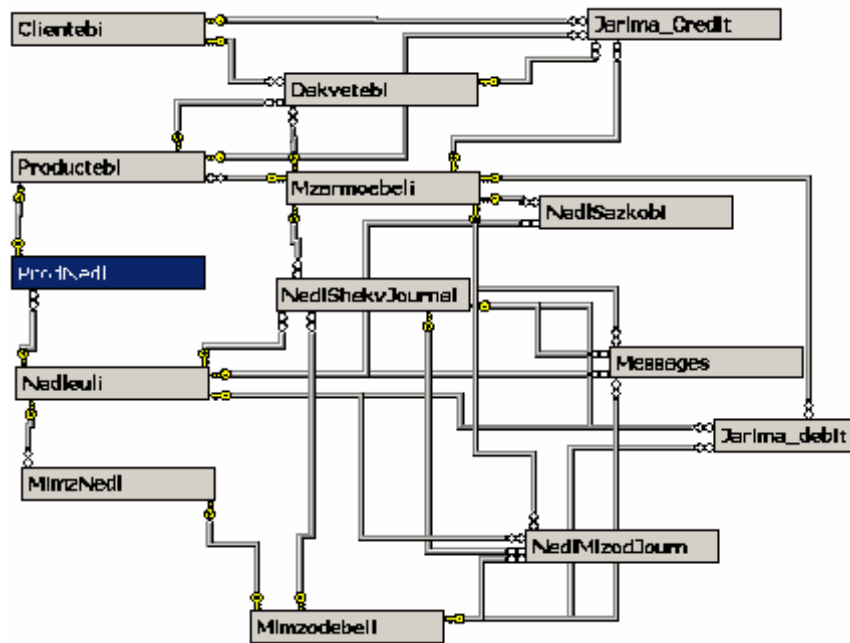
ამისათვის მენიუდან View | Server Explorer-ით ავირჩიოთ HOMESERVER. BusinProc.dbo. გამოჩნდება SQL Server-ის ბაზა, ცხრილები და ველები. DIAGRAM1-ის არჩევით მივიღებთ ცხრილებს კავშირებით, რომელიც იდენტურია SQL Server-ში ჩვენს იერ შექმნილი ბაზისა. ამგვარად თავსებადობა კარგადაა რეალიზებული.

3.25 ნახაზზე ნაჩვენებია ცხრილთა კავშირების ფრაგმენტი, მხოლოდ მათი სათაურების ჩვენებით. ამასთანავე ფიქსირდება კავშირების დამოკიდებუ-ლების სახელი.



ნახ.3.25 ინტერფეისი ფორმის შედეგი

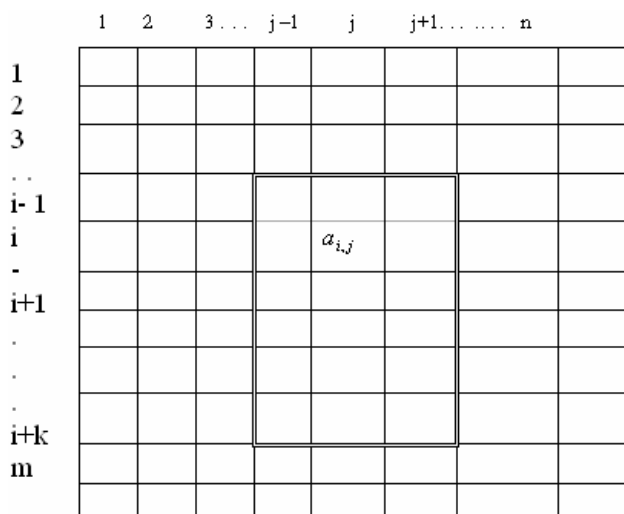
3.26 ნახაზზე გამოტანილია საინტერფეისო ფორმის შემუშავებული ვარიანტის შედეგი.



ნახ 3.26

### 3.5 VBA ენის ინტეგრირებული გამოყენება Microsoft Office-ის აპლიკაციებში

როგორც ვიცით, Word პაკეტი ტექსტების დამუშავების რედაქტორია და მასში მათემატიკური ანგარიშების ან რთული დოკუმენტების (ფორმების) ინფორმაციის დამუშავების მექანიზმები არაა მძლავრად განვითარებული. ამ შემთხვევაში, VBA – Word მაკროსები სწორედ რომ საქმის მნიშვნელოვან გადაწყვეტას უწყობს ხელს. Word – მაკროსის ასაგებად ცხრილური (Table) მონაცემის დასამუშავებლად განვიხილოთ მისი შესაბამისი დოკუმენტის მოდელი, რომელიც 3.27 ნახაზზეა მოცემული.



ნახ.3.27

(m x n) განზომილების ცხრილში, სადაც m სტრიქონების რაოდენობაა (Rows) და n — სვეტებისა (Columns). ცხრილში შეირჩევა ქვეცხრილი, რომელიც ერთგვარ პროექციას წარმოადგენს კოორდინატებით [(j-1), (j-1), (i+k), (j+1)]. პირობითად ავიღეთ სამი სვეტი (j-1, j, j+1) და ათამდე სტრიქონი (i-1, i, i+1, ..., i+k). ამოცანა მდგომარეობს შემდეგში: ამ ცხრილის [(j-1), (j+1)] უჯრაში მოთავსებულია რიცხვი, მაგალითად, int ტიპისა (მთელი რიცხვი),  $a(i-1, j+1) = 4$ . უნდა დაიწეროს მაკროსი Word – VBA ენაზე, ისეთი, რომელიც (j+1) სვეტებში ქვემოთ ჩაწერს  $a(i-1, j+1)$  უჯრაში არსებული რიცხვის მნიშვნელობას, 1-ით ნაკლებს, მანამ, სანამ ის გახდება უარყოფითი რიცხვი.

დავუშვათ, რომ  $a(i-1, j-1)$  უჯრაში ჩაწეროს 2005,  $a(i-1, j)$  უჯრაში კი ჩაიწეროს 2006. ამ ორი უჯრის ქვეშ ჩაიწერება 1-ით ნაკლები რიცხვები  $a(i-1, (j-1))-1$ ,  $a(i-1, j)-1$  და ა.შ. მანამ, სანამ (j+1) –ის შესაბამის უჯრებში არ შეგხვდება უარყოფითი რიცხვი “-1”. საბოლოო შედეგს უნდა ჰქონდეს ასეთი სახე (ნახ.3.28):

2005	2006	4
2004	2005	3
2003	2004	2
2002	2003	1
2001	2002	<b>უარყოფითია</b>

ნახ.3.28

მაკროსი, რომელიც ასეთ პროცედურას შეასრულებს გამოყენებულ იქნება, მაგალითად თანამშრომელთა მუშაობის რეგლამენტის კონტროლისათვის. ამ რთულ დოკუმენტში მონაცემები აიღება Tools – Merge Mail ინსტრუმენტით Excel ან Ms Access ფაილებიდან, ერთ-ერთი ველი “მუშაობის სტაჟი” ეთანადება რიცხვით ფაილს, რომელშიც ჩაწერილია 0,1,2,...,4 და ა. შ. . ამ საწყისი, ერთი რიცხვის საფუძველზე ჩვენი მაკროსი შეავსებს

გრაფებს “წელი” და “სტაჟი”. ცხრილის უჯრა „უარყოფითი“ შეესაბამება იმ წლებს, როცა პიროვნება არ ჰქონდა სტაჟი.

ქვემოთ მოცემულია VBA-მაკროსის შესაბამისი მოდული (პროგრამის კოდის) ფრაგმენტი.

```
‘ VBA – kodi cxrilSi gaangariSebuli
‘ ricxvebis Casawerad
Sub kaplan ()
j = 5
Set mycell = ActiveDocument.Tables(1).Cell(Row:=11, Column:=j)
ActiveDocument.Tables(1).Cell(10, j).Select
Selection.SelectColumn
i = Val(mycell)
smif = i
a = 11
c = 2006
d = 2005
For Each Ocell In Selection.Cells
    i = i - 1
    k = 4
    Set mycell1 = ActiveDocument.Tables(1).Cell(Row:=a, Column:=k)
    mycell1.Range.Text = c
    k = 3
    Set mycell1 = ActiveDocument.Tables(1).Cell(Row:=a, Column:=k)
    mycell1.Range.Text = d
    ActiveDocument.Tables(1).Cell(4, j).Select
    Selection.SelectColumn
    a = a + 1
    c = c - 1
    d = d - 1
    If i < 0 Then
        i_h = Str(i + 1)
        Set mycell1 = ActiveDocument.Tables(1).Cell(Row:=a - 1, Column:=j)
        mycell1.Range.Text = "uaryofiTia"
        Exit For
    End If
    Set mycell1 = ActiveDocument.Tables(1).Cell(Row:=a, Column:=j)
    mycell1.Range.Text = i
Next Ocell
' -----
Dim x1, x2, y1, y2
i = smif
x1 = 194.4
x2 = 324
y2 = 633.6
If i = 0 Then
    y1 = 410.4
Else:
    If i = 1 Then
        y1 = 424
    Else:
        If i = 2 Then
            y1 = 446
        Else:
            If i = 3 Then
                y1 = 475
            Else:
                If i = 4 Then
```



(მაგალითად, ბუჰხალტერი) შეიტანს დოკუმენტის 1,5,6 სვეტების და „კურსის“ მნიშვნელობას (პროდუქციის შემოსვლის შემთხვევაში) ან მე-8 სვეტს (პროდუქციის გაყიდვის შემდეგ).

პროდუქტის დასახელება	რაოდენობა	თვით ღირებულება	ჯამი	შემოსული რაოდენობა	კურსი ფასი Euro	ფასი ლარი	გაყიდული რაოდენობა
1	2	3	4	5	6	7	8

ნახ.3.29 დასაპროგრამებელი დოკუმენტის ფორმა

პროგრამულმა სისტემამ ავტომატურად უნდა დაიანგარიშოს პროდუქციის შემოსვლისა და გაყიდვის შემთხვევებში „რაოდენობა“ (2), „თვითღირებულება“ (3) და „ჯამი“ (4).

სავალუტო კურსის ხშირი ცვალებადობის გამო, მისი რიცხვითი მნიშვნელობა ოპერატიულად შეიტანება „კურსის“ უჯრედში, ხოლო პროგრამა შემოსული პროდუქციის ფასებს ლარებით, ავტომატურად გადაიანგარიშებს ყველა სტრიქონისთვის.

Excel-ზე დამუშავებული სისტემა მუშაობს ორი ღილაკის (button) საშუალებით. პირველი – „დამატება“ უზრუნველყოფს ბუჰხალტრის მიერ მე-5,6 სვეტებში შეტანილი მნიშვნელობების შეჯამებას და მე-2,3 და 4 სვეტების გადაანგარიშებას.

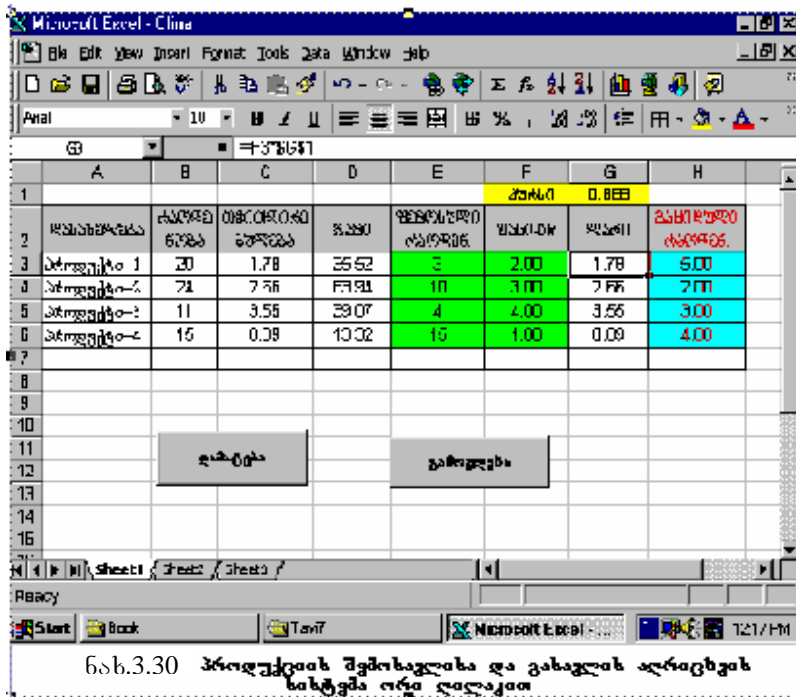
მეორე ღილაკი – „გამოკლება“ კი მე-8 სვეტში ჩაწერილი რაოდენობრივ მნიშვნელობებს დააკლებს მე-2 სვეტის მნიშვნელობებს.

დოკუმენტზე ისრებით მითითებულია ოპერატიული ინფორმაციის შეტანის უჯრედები. ესაა „შემოსული რაოდენობა“, „გაყიდული რაოდენობა“ და „სავალუტო კურსი“. მომხმარებელი (მაგალითად, ბუჰხალტერი) შეიტანს დოკუმენტის 1,5,6 სვეტების და „კურსის“ მნიშვნელობას (პროდუქციის შემოსვლის შემთხვევაში) ან მე-8 სვეტს (პროდუქციის გაყიდვის შემდეგ).

პროგრამულმა სისტემამ ავტომატურად უნდა დაიანგარიშოს პროდუქციის შემოსვლისა და გაყიდვის შემთხვევებში „რაოდენობა“ (2), „თვითღირებულება“ (3) და „ჯამი“ (4). სავალუტო კურსის



ხშირი ცვალებადობის გამო, მისი რიცხვითი მნიშვნელობა ოპერატიულად შეიტანება „კურსის“ უჯრედში, ხოლო პროგრამა შემოსული პროდუქციის ფასებს ღარებით, ავტომატურად გადაიანგარიშებს ყველა სტრიქონისთვის.

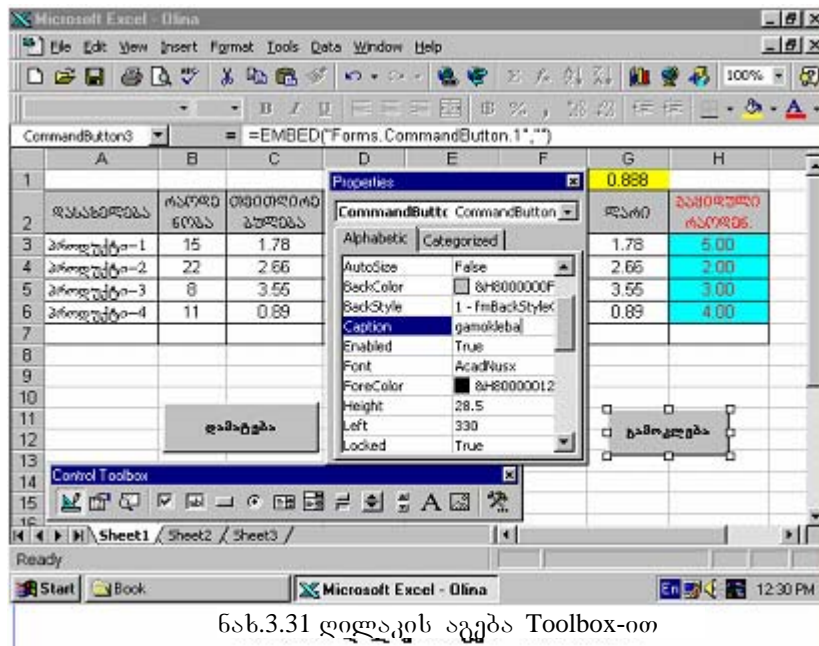


3.30 ნახაზზე ნაჩვენებია MsExcel-სისტემაში აგებული დოკუმენტის ფრაგმენტი აღნიშნული ამოცანისათვის.

Excel-ზე დამუშავებული სისტემა მუშაობს ორი დილაკის (ბუტტონ) საშუალებით. პირველი – „დამატება“ უზრუნველყოფს ბუჰხალტრის მიერ მე-5,6 სვეტებში შეტანილი მნიშვნელობების შეჯამებას და მე-2,3 და 4 სვეტების გადაანგარიშებას.

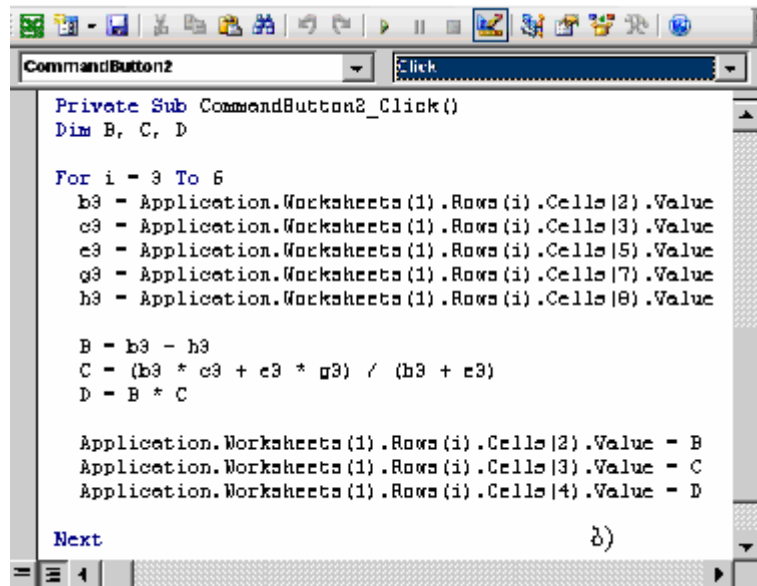
მეორე დილაკი – „გამოკლება“ კი მე-8 სვეტში ჩაწერილი რაოდენობრივ მნიშვნელობებს დააკლებს მე-2 სვეტის მნიშვნელობებს.

ამგვარად, დილაკები „დამატება“ და „დაკლება“ მუშაობს, ობიექტ-ორიენტირებული დაპროგრამების ტერმინებით რომ ვთქვათ, „მოვლენის“ (event) პრინციპზე. ე.ი. დილაკი არის ობიექტი, რომელიც მემკვიდრეობითობის პრინციპით აიგება სპეციალური ინსტრუმენტების მეშვეობით (ნახ.3.31).



ნახ.3.31 ღილაკის აგება Toolbox-ით

მომდევნო ეტაპზე საჭიროა ღილაკებს მივამაგრეთ ფუნქციები (ან მეთოდები). ღილაკ – „დამატებას“ დაჭერით უნდა მოხდეს ახალი პროდუქციის რაოდენობისა და ფასების ჩამატება მონაცემთა ბაზაში, ხოლო ღილაკ „გამოკლების“ დაჭერით უნდა მოხდეს სტრიქონებიდან პროდუქციის რაოდენობის შემცირება.

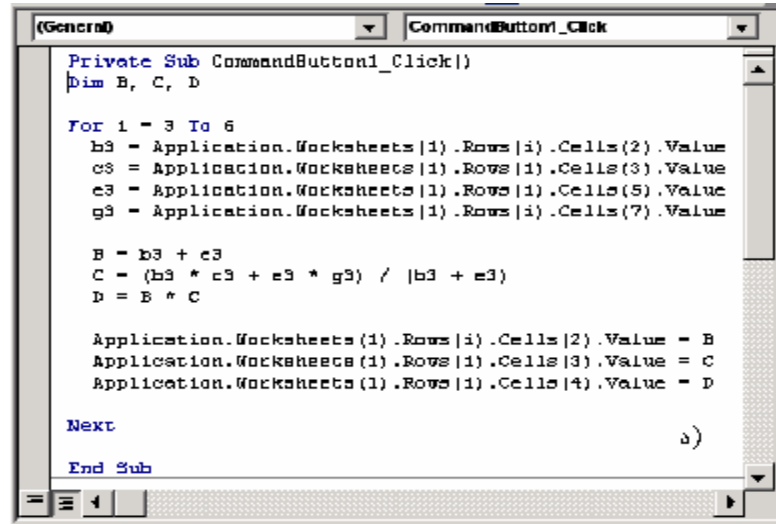


ნახ. 3.32 ღილაკის „დამატება“ პროგრამა ვიზუალ ბეისიკზე

ეს პროცედურები უნდა დაიწეროს VBA-ენის გამოყენებით. ამისათვის ღილაკის მართკუთხედს (მაგალითად, „დამატება“) თავის მარცხენა კლავიშს ორჯერ დაუწკაპუნებთ, რის შემდეგაც

გამოჩნდება მეთოდის (პროცედურის) ჩასაწერი ფანჯარა ვიზუალ ბეისიკზე.

3.32 და 3.33 ნახაზებზე ნახვენებია „დამატება“ და „გამოკლება“ ლილაკების შესაბამისი მეთოდების აღმწერი პროგრამული კოდები.



```
Private Sub CommandButton1_Click()  
    Dim B, C, D  
  
    For i = 3 To 6  
        b3 = Application.Worksheets(1).Rows(i).Cells(2).Value  
        c3 = Application.Worksheets(1).Rows(i).Cells(3).Value  
        g3 = Application.Worksheets(1).Rows(i).Cells(5).Value  
        g3 = Application.Worksheets(1).Rows(i).Cells(7).Value  
  
        B = b3 + c3  
        C = (b3 * c3 + e3 * g3) / (b3 + e3)  
        D = B * C  
  
        Application.Worksheets(1).Rows(i).Cells(2).Value = B  
        Application.Worksheets(1).Rows(i).Cells(3).Value = C  
        Application.Worksheets(1).Rows(i).Cells(4).Value = D  
  
    Next  
End Sub
```

ნახ. 3.33 ლილაკის „გამოკლება“ პროგრამა ვიზუალ ბეისიკზე

### 3.6. საჭმიანი პროცესების ობიექტ-ორიენტირებული მოდელირება და ანალიზი კეტრის ქსელებით.

აქ განუხულება განაწილებული ბიზნეს-ობიექტების დოკუმენტ-უზრუნველყოფისა და დოკუმენტბრუნვის პროცესების ავტომატიზაცია. ასეთი დიდი და რთული სისტემების ობიექტ-ორიენტირებული ანალიზი და ობიექტ-ორიენტირებული მოდელირება განხორციელებულია უნიფიცირებული მოდელირების ენის (UML) სტანდარტებით და პეტრის ქსელების გრაფო-ანალიზური ინსტრუმენტებით. შედეგად მიღებულია ალტერნა-ტიული სქემები და მახასიათებლები საერთო რესურსების ეფექ-ტურად გამოყენებისა და მოთხოვნების დამუშავების დროის შესამცირებლად.

განაწილებულ ოფის-სისტემებში ფუნქციური ამოცანების ეფექტური გადაწყვეტის მიზნით, სწრაფი დოკუმენტუზრუ-ნველყოფის თვალსაზრისით საერთო გამოყენების რესურსების

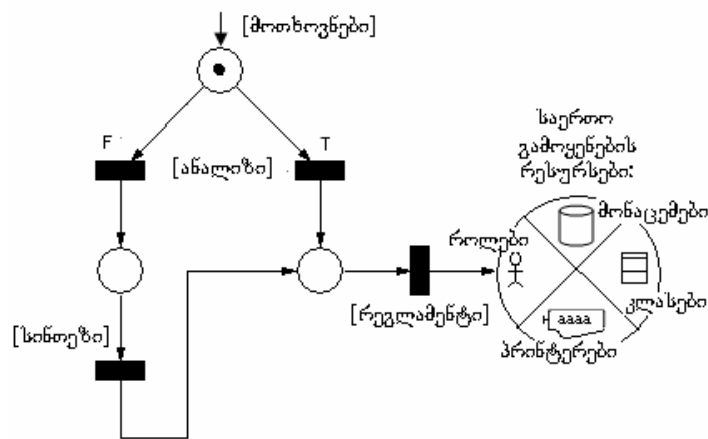
პირობებში, შესაძლებელია გარკვეული ალგორითმული სქემების შემუშავება და მათი ანალიზი პეტრის ქსელის ბაზაზე [13,70].

3.34 ნახაზზე მოცემულია ოფისის საერთო ქსელის მომხმარებელთა მოთხოვნების დამუშავების (მაგალითად დოკუმენტების მომზადების) პროცესის ზოგადი პეტრის ქსელის მოდელი.

მარკერი საწყის პოზიციაში მიუთითებს მოთხოვნის არსებობაზე, რომელიც ანალიზის ეტაპზე განშტოვდება ორ კონფლიქტურ (.T. ან .F. ) გადასასვლელს შორის. თუ მოთხოვნა ეხება დეტერმინირებულ ამოცანათა კლასის ფორმებს, მაშინ იგი აირჩევს .T. – გზას, ხოლო შემთხვევითი (ოპერატიული) მოთხოვნებისათვის აუცილებლად გაივლის .F. – გზას. შესაბამისი დოკუმენტის ფორმის სინთეზის პროცედურით.

გადასასვლელი რეგლამენტი ასახავს კონკრეტული დოკუმენტის ფორმირების წესებს, სისტემის საერთო რესურსების მიმდევრობით-პარალელური ოპერაციების შესრულების ბაზაზე,

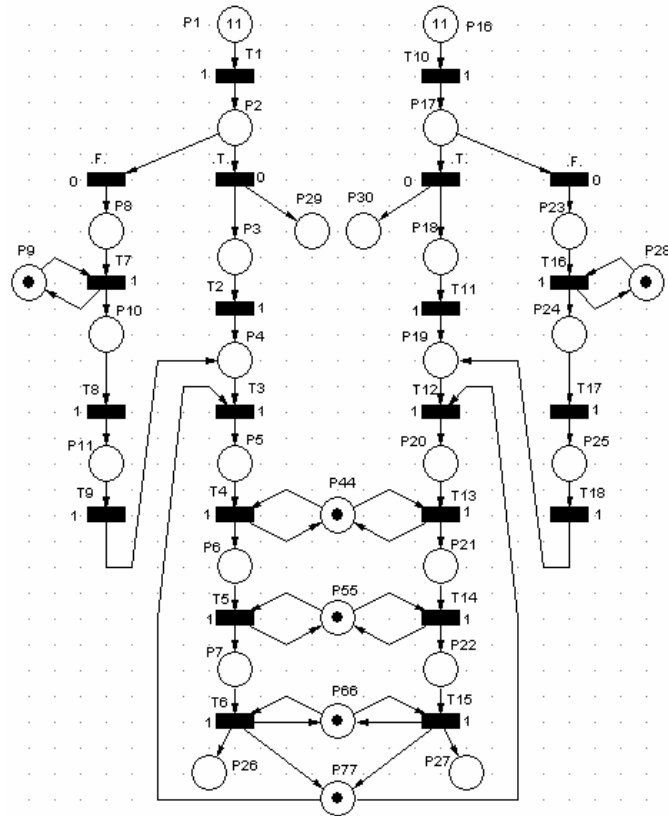
მათი როლების, მონაცემების, პროგრამული მოდულებისა და სხვა ორგტექნიკის კოლექტიური გამოყენების თვალსაზრისით



**ნახ.3.34.მოთხოვნების დამუშავების ზოგადი მოდელი**

.აღსანიშნავია პროცესის მართვის დეტალიზება და მისი ეფექტურობის ანალიზი გარკვეული რაოდენობრივი კრიტერიუმებით, შესაძლებელია პეტრის ქსელის შემდგომი

დაზუსტებით. 3.35 ნახაზზე ნაჩვენებია ასეთი სქემის მაგალითი, ხოლო 1-2 ცხრილებში მოცემულია მისი შესაბამისი პოზიციებისა და გადასასვლელების აღწერა. მოდელი შეიძლება განზოგადებულ იქნას  $n$ -მომხმარებლისათვის.



ნახ.3.35. პეტრის ქსელის ფრაგმენტი 2-მომხმარებლისთვის

3.36 ნახაზზე ილუსტრირებულია აღნიშნული პეტრის ქსელის იმიტაციური პროცესის ეტაპები. აქ მარკერების (მოთხოვნების) გადაადგილება ხორციელდება  $P_1$  და  $P_{16}$  პოზიციებიდან  $P_{26}$  და  $P_{27}$  პოზიციებამდე. მაგალითად, 1-ელ საწყის ეტაპზე ჩანს 11-11 მოთხოვნა ქსელის ორ კვანძში და საერთო რესურსების  $P_{44}$  – მონაცემთა ბაზები,  $P_{55}$ -კლასები,  $P_{66}$ -ქსელური პრინტერები და  $P_{77}$ -ადამიანური რესურსი) მზადყოფნა (მარკერების არსებობით).

3.37 ნახაზზე ნაჩვენებია იმიტაციური პროცესის მართვის აპარატი (Run, Stop და სხვა დილაკებით) და თვით პეტრის ქსელის ერთ-ერთი შუალედური მდგომარეობა, რომელზეც კარგად ჩანს მარკერის განაწილება და პარალელურად მომუშავე გადასასვლელები (მონიშნულია). ამ მომენტში შესრულებულია მხოლოდ ერთი დავალება (P<sub>26</sub> – ში ერთი მარკერია).

3.38 ნახაზის ბოლო ეტაპზე P<sub>26</sub> და P<sub>27</sub> ჩანს, რომ ყველა მოთხოვნა შესრულებულია. P<sub>29</sub> და P<sub>30</sub> პოზიციები მიუთითებს, რომ 1-ელ კვანძში 11 შემოსული მოთხოვნიდან 6 იყო დეტერმინირებულ ამოცანათა კლასიდან, ხოლო მეორეში – 7.

ცხრ.3.1

**პოზიციების ცხრილი**

P <sub>j</sub>	პოზიცია (მარკერებით)
P <sub>1, 16</sub>	მოთხოვნა არის (თუ მარკერია)
P <sub>2, 17</sub>	დოკუმენტის ფორმა (კოდი)
P <sub>3, 18</sub>	ცნობილი დოკუმენტის ფორმა
P <sub>4, 19</sub>	დასამუშავებელი დოკუმენტები
P <sub>5, 20</sub>	დოკუმენტი როლთან
P <sub>6, 21</sub>	დოკუმენტი მონაცემთა რესურსით
P <sub>7, 22</sub>	დოკუმენტი კლასთა რესურსით
P <sub>26, 27</sub>	დაბეჭდილი დოკუმენტი
P <sub>8, 23</sub>	დასამუშავებელი მოთხოვნა
P <sub>10, 24</sub>	ატრიბუტთა სიმრავლე
P <sub>11, 25</sub>	რელაციის სქემა დოკუმენტის ფორმისათვის
P <sub>9, 28</sub>	ატრიბუტების ანალიზატორი თავისუფალია (მარკერით)
P <sub>44</sub>	მონაცემთა რესურსი თავისუფალია (მარკერით)
P <sub>55</sub>	კლასების რესურსი თავისუფალია (მარკერით)
P <sub>66</sub>	საბეჭდი რესურსი თავისუფალია (მარკერით)
P <sub>77</sub>	ადამიანური რესურსი თავისუფალია (მარკერით)
P <sub>26, 27</sub>	დამუშავებული ფორმების ჯამური რაოდენობა
P <sub>29, 30</sub>	დამუშავებული სტანდარტული ფორმების რაოდენობა

ცხრ.3.2

**გადასასვლელუბის ცხრილი**

<b>T<sub>i</sub></b>	<b>გადასასვლელის ფუნქცია</b>
T <sub>1, 10</sub>	მოთხოვნის ანალიზი
.T. / .F.	ფორმა ცნობილია/არაა ცნობილი
T <sub>2, 11</sub>	დოკუმენტის ფორმის არჩევა
T <sub>3, 12</sub>	ადამიანური რესურსით ფორმის შევსების წესის განსაზღვრა
T <sub>4, 13</sub>	მონაცემთა რესურსის მიღება
T <sub>5, 14</sub>	კლასთა რესურსის მიღება
T <sub>6, 15</sub>	საბეჭდი მოწყობილობის რესურსის მიღება
T <sub>7, 16</sub>	ატრიბუტთა სიმრავლის დადგენა
T <sub>8, 17</sub>	რელაციათა სქემის სინთეზი
T <sub>9, 18</sub>	რელაციათა სქემით დოკუმენტის ფორმის განსაზღვრა

P<sub>4</sub> და P<sub>19</sub> - ის პოზიციებია, სადაც კონკრეტულ როლთან (ადამიანის რესურსი) თავს იყრის სხვადასხვა მოთხოვნები და საიდანაც უნდა მოხდეს მათი უზრუნველყოფა მონაცემთა ბაზებით, პროგრამებით და საბეჭდი მოწყობილობით. სქემის აღნიშნულ ფრაგმენტს განსაკუთრებული მნიშვნელობა აქვს მრავალმომხმარებლურ რეჟიმში მუშაობისას, როდესაც სისტემის ადამიანური და სერვერული რესურსები ეფექტურად უნდა იმართოს.

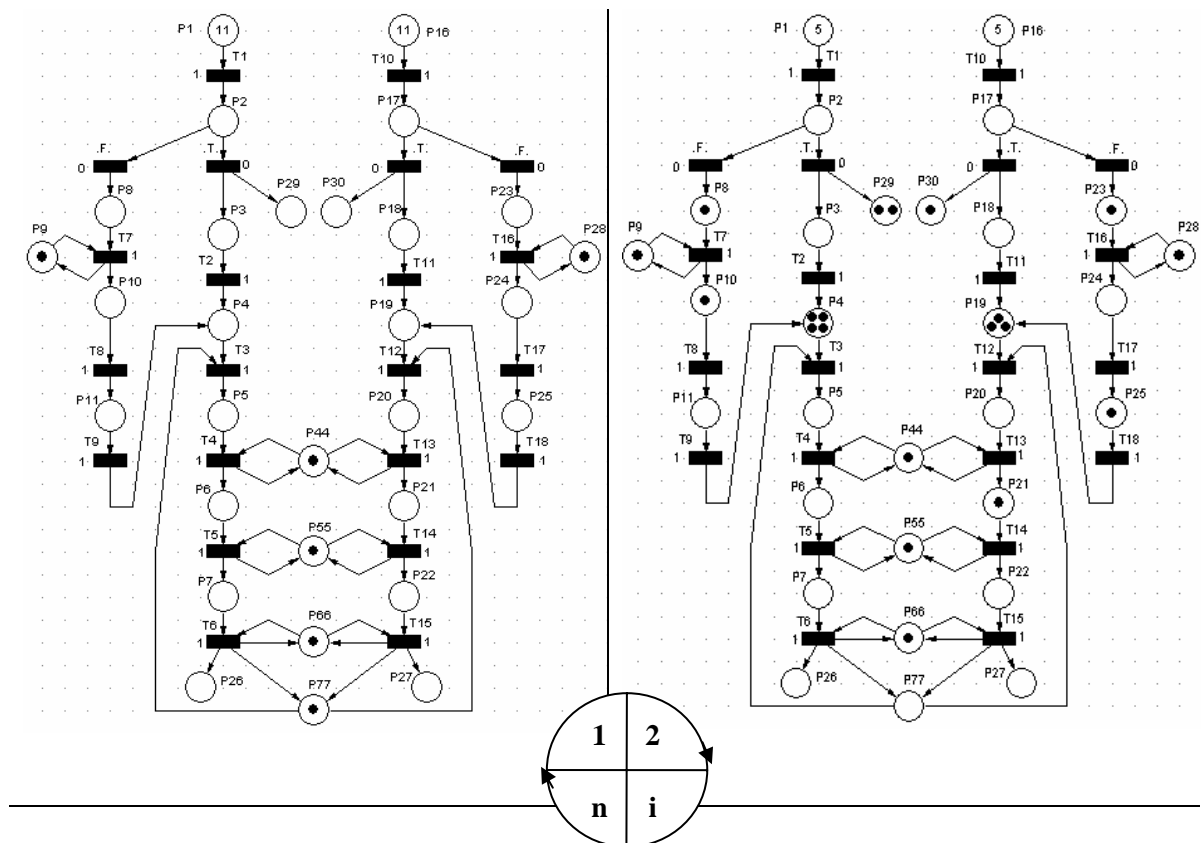
**3.37** ნახაზზე მოცემულია P<sub>4</sub> და P<sub>19</sub> პოზიციების მდგომარეობები დროის მიხედვით.

ტრიგერული მართვისას რესურსები მიმდევრობით გადაეცემა მომხმარებლებს ყოველგვარი პრიორიტეტის გარეშე. ამიტომაც აქ მოთხოვნების დამუშავება სინქრონულად მიმდინარეობს (ნახ.3.38 და ნახ.39)

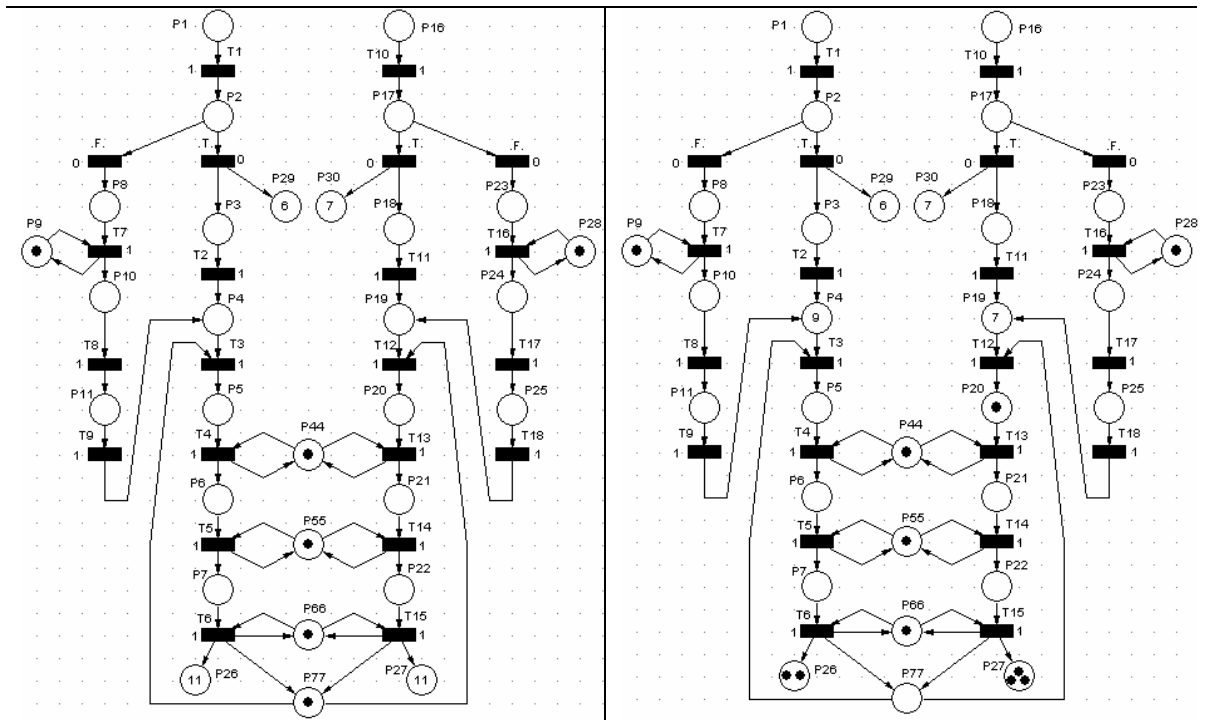
წინა შემთხვევაში ადგილი ჰქონდა რესურსების ასინქრონულ რეჟიმს. ამ დროს მართვის სისტემა შემთხვევით, თვითონ ირჩევს კონფლიქტური და პარალელური პროცესების შესრულების მიმდევრობას.

მოდევნო მაგალითებში ნაჩვენებია გვაქვს რესურსების ექსტენსიონალური (ნახ.3.40) და ინტენსიონალური (ნახ.3.41) გაფართოებით მოთხოვნების დამუშავების დროითი დიაგრამები. ნახაზებზე კარგად ჩანს მოთხოვნების დროის შემცირების ტენდენცია.

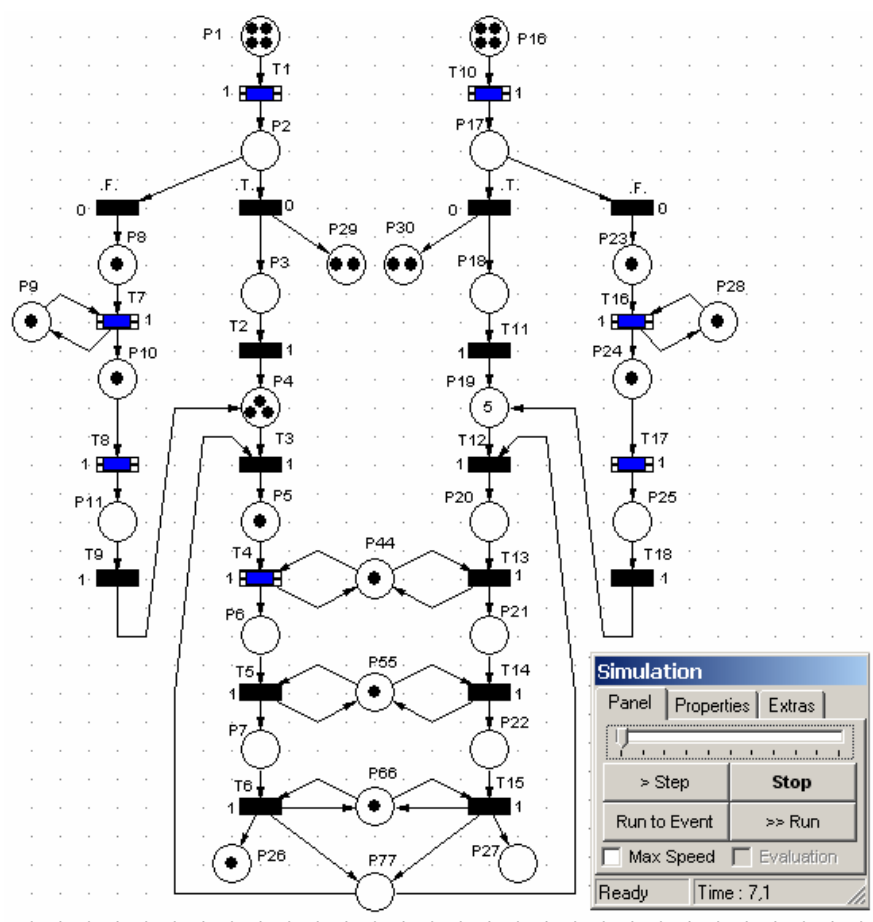
ამრიგად, განაწილებული ოფის-სისტემების ბიზნეს-პროცესების დოკუმენტური უზრუნველყოფის და რაციონალური დოკუმენტბრუნვის დასაპროექტებლად მნიშვნელოვანი ინსტრუმენტია კლასებზე და პროცესებზე ორიენტირებული საპრობლემო სფეროს მოდელირებისა და ანალიზის მეთოდები, რომელთა რეალიზაცია და კვლევა ხორციელდება პეტრის ქსელებით საერთო რესურსების პირობებში.



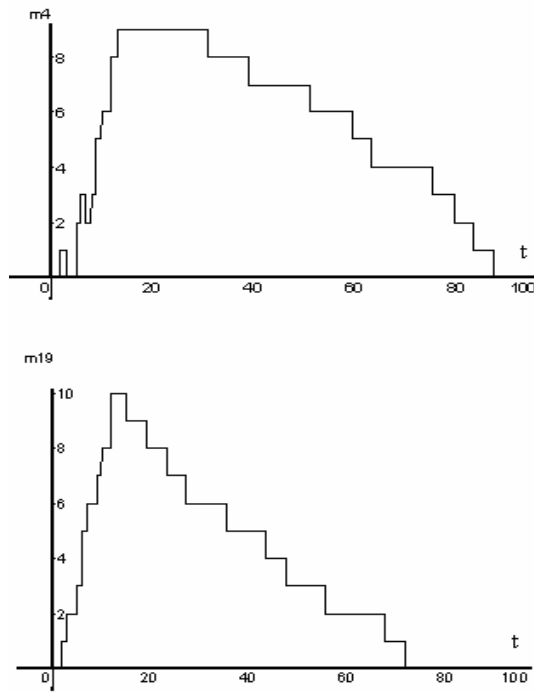




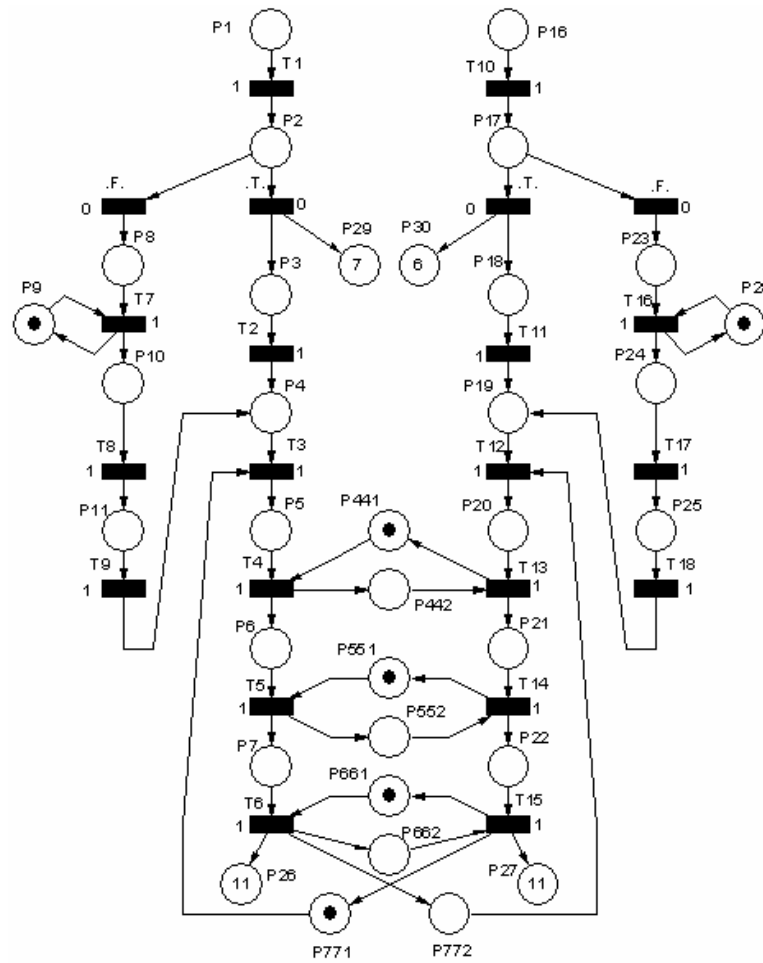
ნახ.3.36 პეტრის ქსელის სიმულატორით მიღებული ეტაპობრივი შედეგები



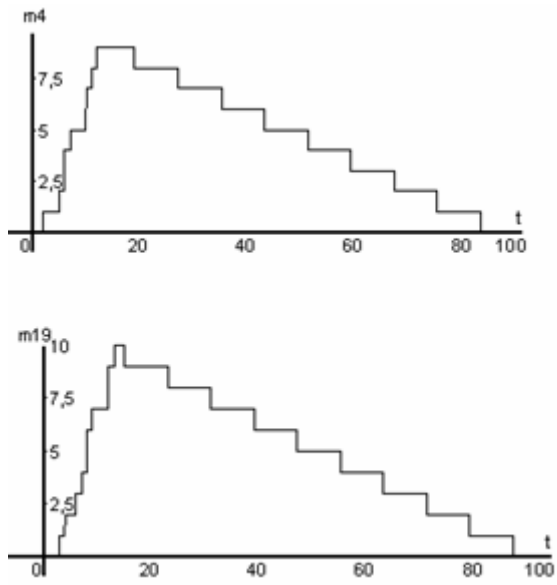
ნახ.3.37. იმიტაციის პროცესის ფრაგმენტი პეტრის ქსელით



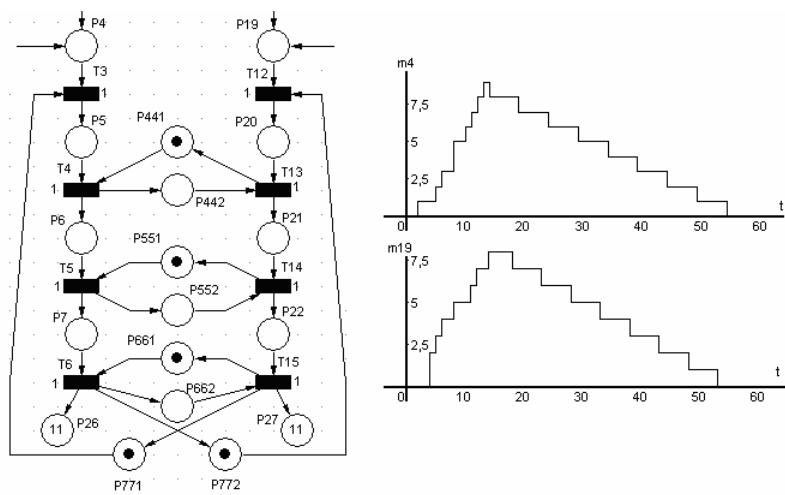
ნახ.3.38. P4 და P19 პოზიციების მდგომარეობათა ცვლილებები დროში



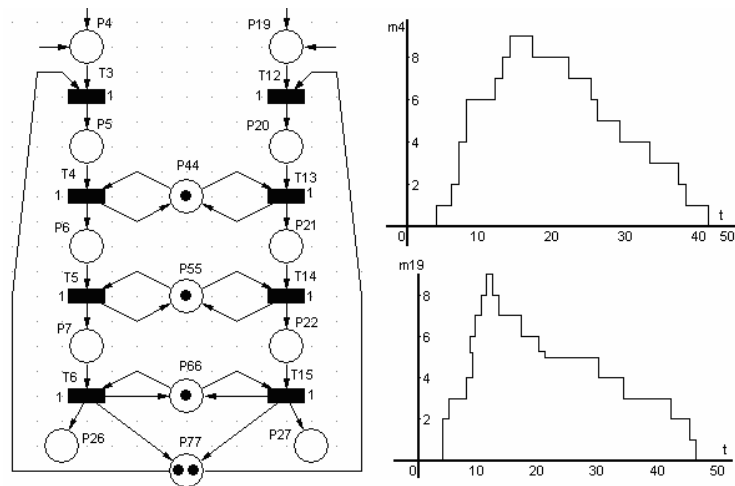
ნახ.3.39 რესურსების ტრიგერული მართვის ფრაგმენტი



ნახ.3.40 P4 და P19 პოზიციების მდგომარეობათა ცვლილებები დროში ტრიგერული მართვისას



ნახ.3.41 P4 და P19 პოზიციების მდგომარეობათა ცვლილებები დროში ტრიგერული მართვისას გაორკეცებული ადამიანური რესურსით



ნახ.3.42. P<sub>4</sub> და P<sub>19</sub> პოზიციების მდგომარეობათა ცვლილებები დროში ადამიანური რესურსის გაორკეცებული სწრაფქმედებით

### მესამე თავის დასკვნები

1. მესამე თავში შემოთავაზებულია ოფის-სისტემების მარკეტინგული პროცესების მოდელირების ობიექტ-ორიენტირებული მეთოდი, რომლის საფუძველზეც აგებულია უნიფიცირებული მოდელირების ენის ძირითადი დიაგრამები ამ პროცესებისათვის;
2. დამუშავებულია ალგორითმული სქემები და მათი რეალიზაციის პროგრამები ობიექტ-როლური მოდელის და მათი შესაბამისი ფიზიკური მოდელის (რელაციური მოდელის) ასაგებად Ms SQL Server მონაცემთა ბაზების სისტემისათვის;
3. მონაცემთა ბაზების მიღებული სტრუქტურებისათვის შესაძლებელია ADO.NET პროგრამული დრაივერების გამოყენება მომხმარებელთა საინტერფეისო პროგრამებთან სამუშაოდ. შესაძლებელია აგრეთვე XML ფორმატის გამოყენება ინფორმაციის ინტერნეტში გადასაცემად.
4. შემოთავაზებულია განაწილებული, საერთო გამოყენების რესურსების (მონაცემთა ბაზები, პროგრამები, საბეჭდო მოწყობილობანი, გადაცემის არხები და სხვ.) მართვის პროცესების მოდელირებისა და ანალიზის ეფექტური ალგორითმები პეტრის ქსელების ბაზაზე.

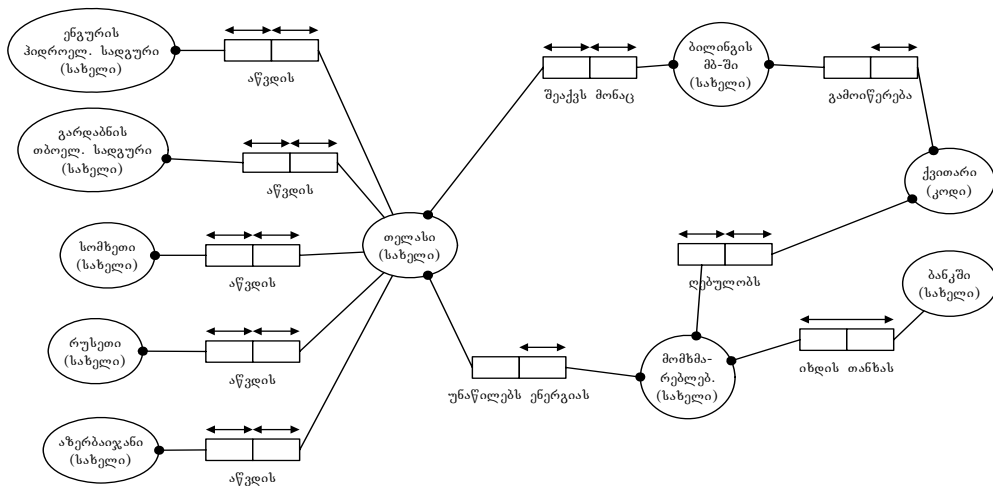
## IV ტაპი

### განაწილებული ენერგეტიკული ოფის-სისტემების ამოცანების კომპიუტერული რეალიზაცია

#### 4.1. ენერგოსისტემის ობიექტ-როლური მოდელი

თანამედროვე ენერგეტიკული ობიექტების ქსელი უდავოდ მიეკუთვნება რთული და დიდი სისტემების კლასს, რომლისთვისაც დამახასიათებელია ძნელადფორმალიზებადი მართვის ამოცანების, დიდი მოცულობის საინფორმაციო ბაზების და გადაწყვეტილების მიღების რთული ალგორითმების არსებობა [20,35,48]. ყოველივე ეს კი აუცილებლად მოითხოვს ენერგეტიკაში მართვის ახალი სისტემების გამოყენებას.

ჩვენი მიზანია ორგანიზაციული და ბიზნესის მართვის განაწილებული ენერგოსისტემის ოფის-ობიექტებისათვის საქმიანი პროცესების წარმოების უნიფიცირებული დოკუმენტბრუნვის ტექნოლოგიის შემუშავება და პროგრამული რეალიზაცია, თანამედროვე ობიექტ-ორიენტირებული დაპროგრამების მეთოდებისა და კლიენტ-სერვერული არქიტექტურის საფუძველზე.



ნახ.4.1 ORM მოდელის ფრაგმენტი

4.1 ნახაზზე მოცემულია განაწილებული ენერგობიექტების ზოგადი ORM-დიაგრამის ფრაგმენტი, რომელიც აერთიანებს

ელენერგის მწარმოებლებს, მის დამკვეთებს (მაგ., თელასი) და საბოლოო აბონენტებს (იურიდიულ და ფიზიკურ მომხმარებლებს).

ენერგეტიკის ოფის-ობიექტებზე საქმიანი პროცესების თანხმლები უნიფიცირებული და ინდივიდუალური დოკუმენტაციისა და დოკუმენტბრუნვის სისტემის დიაგნოსტიკური ანალიზი, ფუნქციურ განყოფილებათა ურთიერთკავშირის სტრუქტურებისა და მართვის პროცესში მონაწილე პირთა „როლების“ გათვალისწინებით.

მოცემულ პარაგრაფში დამუშავებულია განაწილებული ენერგეტიკული ოფის-სისტემის მონაცემთა ბაზის ფიზიკური სტრუქტურები, მათი მოდიფიკაციის ალგორითმები და ადმინისტრირების ფუნქციები, კლიენტ-სერვერული არქიტექტურის გამოყენებით. შემოთავაზებულია ენერგეტიკული ოფის-სისტემის მომხმარებელთა ინტერფეისები, მათი როლების შესაბამისად.

მართვის ავტომატიზებული სისტემის ამოცანები განიხილება განაწილებული ენერგოსისტემის მთლიან სპექტრში, დაწყებული, მაგალითად, ენგურჰესის ჰიდროელექტროსადგურზე ელენერგის გამომშავების აღრიცხვიდან, დამთავრებული თელასის სააბონენტო გადასახადების აღრიცხვისა და კონტროლის ამოცანებით, ინტერნეტული მოხმარებისათვის.

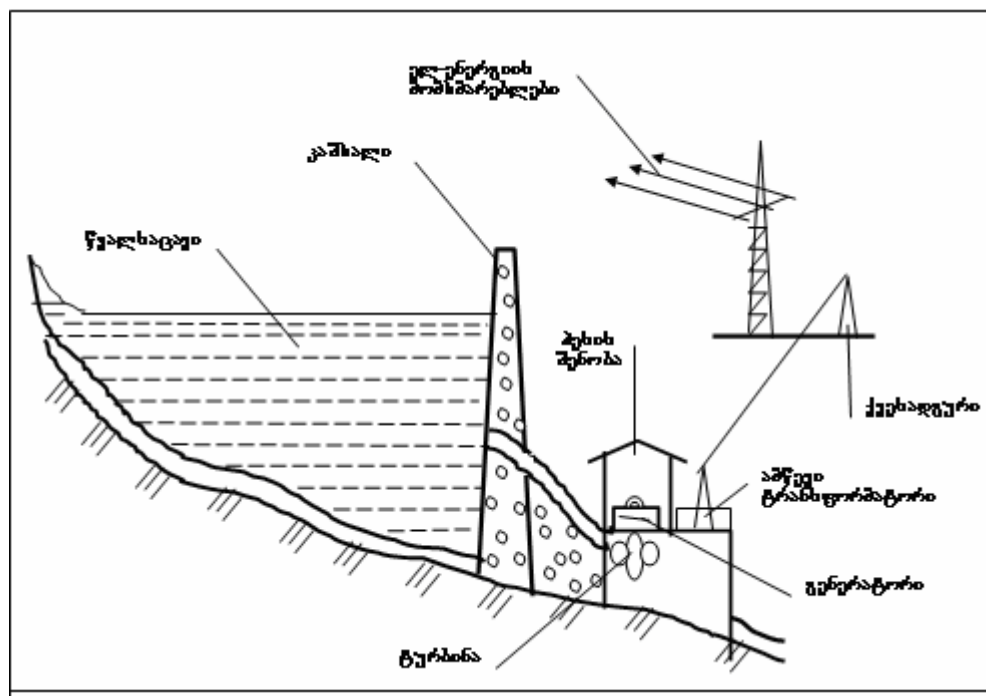
#### **4.2. „ენგურჰესის“ მართვის ავტომატიზებული სისტემის სტრუქტურა**

კვლევის ობიექტია ჰიდროელექტრო სადგური, კერძოდ „ენგურჰესი“. იგი შეიძლება განვიხილოთ როგორც საწარმოო ფირმა, რომელიც გამოიმუშავებს ელექტრო ენერგიას და მიაწვდის ენერგო ბაზარს. ამგვარად მას ჰყავს დამკვეთები.

ელექტრო ენერგის საწარმოებლად, ანუ ტექნოლოგიური პროცესის სტაბილურად განსახორციელებლად ფირმას სჭირდება გარკვეული ნედლეულის შემოტანის, დამხმარე წარმოების

გადლოის, ადმინისტრაციული აპარატისა და სხვა აუცილებელი სამუშაოების ჩატარების ხარჯების გათვალისწინება. ჰიდროელექტროსადგურს აქვს თავისი შემადგენელი ელემენტები ნახ.4.2: კაშხალი, წყალსაცავი, ჰესის შენობა – გენერატორებითა და ტურბინებით, გამაძლიერებელი ტრანსფორმატორებით, ქვესადგურებითა და გადამცემი ხაზებით და ა.შ. [59].

ჰიდროელექტროსადგური არის კომპლექსური ტიპის ობიექტი, თავისი ტექნიკური, ტექნოლოგიური, ეკონომიკური, სოციალური და ეკოლოგიური პირობებით. ამიტომაც მისი მართვის სისტემა მიეკუთვნება რთული სისტემების კლასს, რომელთა ოპტიმალური მართვისათვის აუცილებელია კომპიუტერული ტექნიკისა და თანამედროვე საინფორმაციო ტექნოლოგიების გამოყენება.



ნახ. 4. 2 „ენგურჰესის“ სადგურის პირობითი სქემა

4.3 ნახაზზე ნაჩვენებია „ენგურჰესის“ საწარმო-ორგანიზაციული სტრუქტურის ფრაგმენტი. მაგ: ნედლეულისა და მზა პროდუქციის საწყობები, არასაწარმოო რგოლები, ანუ ფირმის მართვის აპარატი დეპარტამენტებით: ბუღალტერია, მარკეტინგის, დაგეგმვისა და ეკონომიკური ანალიზის, ტექნოლოგიური, კადრების,

ინფორმაციის კომპიუტერული დამუშავების, საზოგადოებასთან ურთიერთობის და ა.შ. მოდიფიცირებულ იქნება დაზუსტების თვალსაზრისით, შესაბამისად.

ახალი ინფორმაციული ტექნოლოგიები საშუალებას იძლევა ენგურჰესზე გამოყენებულ იქნეს როგორც წარმოების დაგეგმვის, აღრიცხვის, ანალიზისა და კონტროლის ამოცანების ავტომატიზაცია, ასევე საპროექტო-საკონსტრუქტორო და ტექნოლოგიური პროცესების ავტომატიზაცია.

ჰესების ეფექტური მართვისათვის დიდი მნიშვნელობა აქვს მის სამეურნეო და საწარმოო მახასიათებლების საფუძველზე სწორი გადაწყვეტილებების მიღებას. მისმა ხელმძღვანელობამ კარგად უნდა იცოდეს ფირმის ორგანიზაციულ-ტექნიკური დონე. ეს უკანასკნელი განმსაზღვრელია როგორც გამოსაშვები პროდუქციის (ელექტრო ენერჯის) ხარისხისა, ასევე ფირმაში სოციალურ-ეკონომიკური მდგომარეობისა.

როგორც აღვნიშნეთ „ენგურჰესის“ ავტომატიზაცია ინტეგრირებული-მას საფუძველზე უნდა განხორციელდეს. აქ ვგულისხმობთ როგორც ელ-ენერჯის წარმოების პროცესის თანმხლები პროცესების ინფორმატიზაციას, ასევე ტექნოლოგიური პროცესების მართვას.

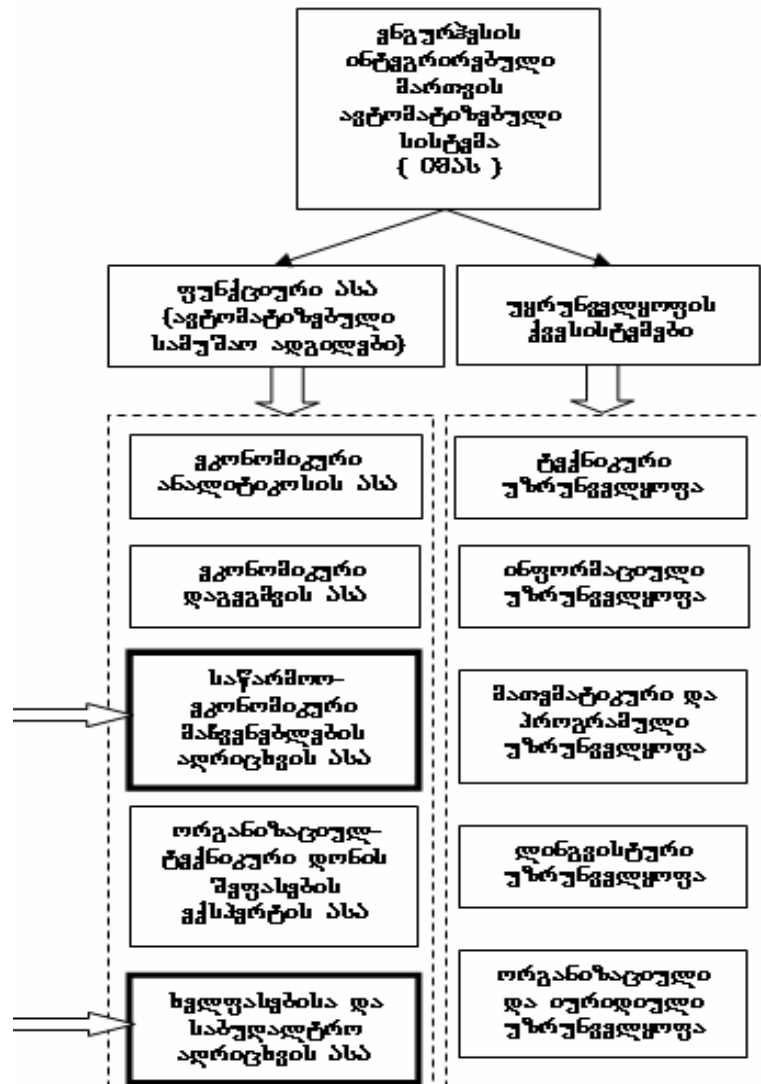
4.3 ნახაზზე წარმოდგენილი გვაქვს ასეთი სისტემის ფუნქციური და უზრუნველყოფის ქვესისტემების სტრუქტურა, ავტომატიზებული სამუშაო ადგილების გამოყენებით.

საჭიროა ინფორმაციული ბაზების შექმნა:

- ელექტროენერჯის ყოველდღიური გამომუშავების შესახებ;
- ფირმის სამეურნეო-ეკონომიკური მაჩვენებლებისათვის;
- ენერჯო ბაზარზე არსებული მდგომარეობის შესახებ, გარკვეული პროგნოზის ჩატარების შესაძლებლობით.



- ფირმაში მიმდინარე საწარმოო პროცესების სააღრიცხვო ინფორმაციისთვის;
- ფირმის წინა წლების სამეურნეო-ეკონომიკური ანგარიშებისთვის დინამიკის შესასწავლად და ა.შ.



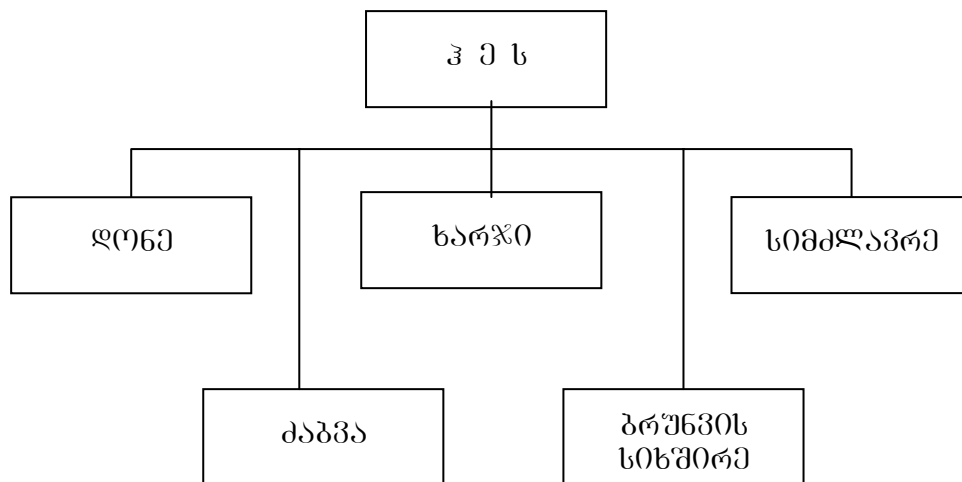
ნახ.4.3. ენგურჰესის ინტეგრირებული მართვის ავტომატიზებული სისტემის სტრუქტურა და შედგენილობა ასა – (ავტომატიზებული სამუშაო ადგილი)

შედგენების საფუძველზე შესაძლებელი იქნება „ენგურჰესის“ ორგანიზაციისათვის პერსპექტიული და ოპერატიული გეგმების აგება და კორექტირება მიზნობრივი ორგანიზაციულ-ტექნიკური დონის მისაღწევად.

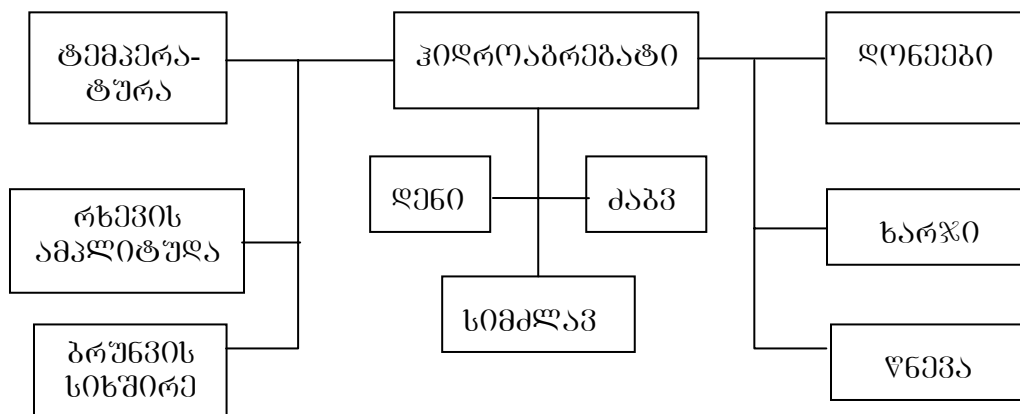
სადემონსტრაციო მაგალითისათვის აქ კონკრეტულად განვიხილავთ „ენგურჰესის“ ობიექტს და მის ერთ-ერთ ძირითად ამოცანას, როგორცაა ელექტროენერჯის წარმოების აღრიცხვის ავტომატიზაცია საწარმო-ტექნიკურ განყოფილებაში.

საწარმო-ტექნიკური მაჩვენებლები და მდინარის ჰიდროენერგეტიკული რესურსების ანგარიში

„ენგურჰესის“ მდგომარეობის კონტროლის მახასიათებლებიდან შეიძლება გამოვყოთ თვით ჰესის მართვის პარამეტრები (ნახ.4.4) და ჰიდროაგრეგატის მართვის პარამეტრები (ნახ.4.5).

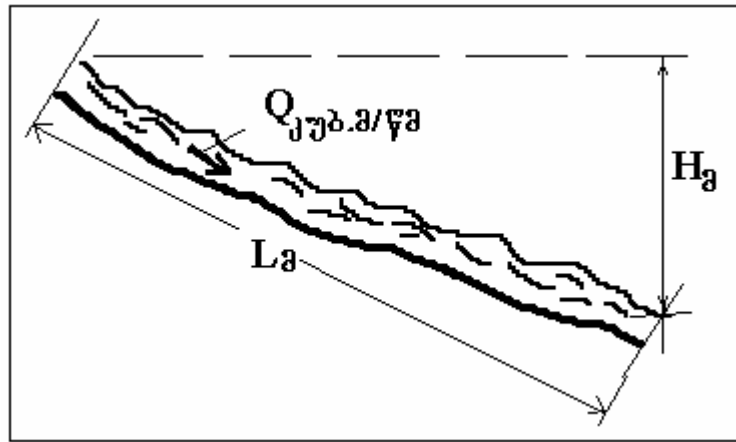


ნახ.4.4 ჰესის მართვის პარამეტრები



ნახ.4.5. ჰიდროაგრეგატის მართვის პარამეტრები

ჰესებზე განსაკუთრებით საყურადღებოა მდინარის ჰიდროენერგეტიკული რესურსების გაანგარიშება. ეს ამოცანა შეიძლება ამგვარად წარმოვადგინოთ (ნახ.4.6):



ნახ.4.6.

ნახაზზე მოცემულია მდინარის შემდეგი ჰიდროენერგეტიკული პარამეტრები:

$Q$  - წყლის ხარჯი;

$H$  - წყლის ვარდნის სიმაღლე;

$L$  - წყლის ჩადინების სიგრძე.

წყლის ჩადინების სიმძლავრე წარმოადგენს გამდინარე წყლის მუშაობას 1 წამში. მისი პოტენციალური მნიშვნელობა გაითვლება ფორმულით:

$$N = \rho g Q H = 9810 Q H \quad (\text{ვატი ან ჯოული/წმ}), \text{ სადაც}$$

$\rho$  - წყლის სიმკვრივეა ( $= 1000$  კგ/კვ.მ);

$g$  - თავისუფალი ვარდნის აჩქარებაა ( $9,81$  მ/წმ.კვ).

შეიძლება ასევე ჩავწეროთ:

$$N = 9,81 Q H \quad (\text{კვტ}).$$

წყალჩადინების ენერჯია გაითვლება ასე:

$$E = \frac{9,81 * Q * H * t}{3600} = \frac{W * H}{367} \quad (\text{კვტ/სთ}), \text{ სადაც}$$

$W = Q * t$  არის გამოყენებული ჩამდინარე წყლის მოცულობა.

დანაკარგების არსებობის გამო (ჰიდროტექნიკურ ნაგებობებსა და ტურბინებში) ელექტროენერჯის გამომუშავების რეალური ან ტექნიკური სიმძლავრე შემცირდება. ეს დანაკარგი გაითვალისწინება მარგი ქმედების კოეფიციენტით -  $\eta$ .

ამგვარად, სასარგებლო სიმძლავრე იქნება:

$$N = 9,81QH\eta \text{ (კვტ)},$$

ხოლო ელექტროენერჯია:

$$E = \frac{W * H * \eta}{367} \text{ (კვტ/სთ)}.$$

ლიტერატურიდან ცნობილია მდინარეების ჰიდროენერგეტიკული რესურსების ეკონომიკური პოტენციალის მაჩვენებლები (მილიარდ/კვტ.სთ), მაგალითად:

ენისეი – 158,3;	მტკვარი – 16,6;
ლენა – 144;	დნეპრი – 14,6;
ვოლგა – 51,4;	კამა – 12,7 და
ამუდარია – 36;	ენგური – 12,3 მლრდ/კვტ.სთ.

#### 4.3 გამომუშავებული ელექტროენერჯის აღრიცხვის მათემატიკური მოდელი და მომხმარებლის ინტერფეისი

ერთ-ერთ მთავარ ამოცანას ენგურჰესზე წარმოებული ელექტროენერჯის ყოველდღიური (ყოველთვიური) აღრიცხვის ავტომატიზებული პროცედურის დამუშავება წარმოადგენს. მისი მათემატიკური მოდელის ასაგებად გამოიყენება შემდეგი მაჩვენებლები:

1	$i \in \overline{1,31}$	კალენდარული დღეების ნომრები თვეში
2	$j \in \overline{1,12}$	თვის ნომრები
3	N	ექსპლუატაციის დაწყებიდან გამომუშავებული ელექტროენერჯის რაოდენობა (ათასი კვტ.სთ)

4	$B_j$	მიმდინარე წლის დასაწყისიდან თვიური ელენერჯის გარდამავალი მნიშვნელობა (ნაშთი)
5	$E_{ji}$	$j$ -თვის $i$ -რიცხვში გამომუშავებული ელ-ენერჯია, რომელიც მრიცხველიდან შეიტანება სააღრიცხვო ჟურნალში ათასი კვტ.სთ).
6	$E_j$ (i-მდე)	$j$ -თვის წლიური გამომუშავებული ჯამური ელ-ენერჯია $i$ -რიცხვამდე, რომელიც გაითვლება და შეიტანება სააღრიცხვო ჟურნალში (ათასი კვტ.სთ).
7	$E_y$ (j,i-მდე)	წლიური გამომუშავებული ჯამური ელ-ენერჯია $j$ თვის $i$ -რიცხვამდე, წინა თვეების გათვალისწინებით, რომელიც გაითვლება და შეიტანება სააღრიცხვო ჟურნალში (ათასი კვტ.სთ).
8	$E_{expl}$ (j,i-მდე)	ექსპლუატაციის დაწყებიდან გამომუშავებული ჯამური ელ-ენერჯია $j$ თვის $i$ -რიცხვამდე, წინა წლებისა და თვეების გათვალისწინებით, რომელიც გაითვლება და შეიტანება სააღრიცხვო ჟურნალში (ათასი კვტ.სთ).
9	$S_{j,i}$	$j$ -ურითვის გამომუშავებული ელ-ენერჯიათვის რიცხვების ჯამით
10	$S_j$	თვიური ჯამური გამომუშავებული ელ-ენერჯიათვის დასაწყისიდან
11	$S_{y,j}$	$y$ -წლიური ჯამური გამომუშავებული ელ-ენერჯია $j$ -ურითვის ჩათვლით წლის დასაწყისიდან
12	$S_{expl,j}$	ექსპლუატაციის დაწყებიდან გამომუშავებული ჯამური ელ-ენერჯია $j$ -ურითვის ჩათვლით
13	$K_j$	$j$ -თვეში დღეების რაოდენობა
14	$C_e$	ელექტროენერჯის თვითღირებულება
15	$C_x$	წლიური ხარჯი ელ-ენერჯის გამომუშავებაზე


ნახ. 4.7

ენგურჰესზე ყოველდღიური გამომუშავებული ელექტრო-ენერჯის ანგარიში და აღრიცხვა წარმოებს შემდეგი მათემატიკური ფორმულებით:

1.  $j$ -ური თვის დასაწყისიდან  $i$ -ური რიცხვის ჩათვლით გამომჟღავნებული ელ-ენერგია:

$$E_{j,i} = E_{j,i-1} + E_{ji} ;$$

მაგალითად, თვის დასაწყისში, 1-ელ რიცხვში  $E_{j,i-1}=0$ , ამიტომ დღის ბოლოს გამომჟღავნებული ელ-ენერგია (10000 კვტ.სთ) ჩაჯდება თვის დასაწყისის პირველ სტრიქონში.



№	დღიური ელ-ენერგია	თვის დასაწყისიდან	წლის დასაწყისიდან
1	10000	$E_{j,i-1} \rightarrow 10000$	10000
2	$E_{ji} \rightarrow 9000$	$E_{j,i} \rightarrow 19000$	19000
3	8000	 27000	27000
...		...	
	ჯამი :	311000	$B_1=311000$

ნახ. 4.8

2. წლის დასაწყისიდან  $j$ -ური თვის წინა თვეების ჩათვლით გამომჟღავნებული ელ-ენერგია  $i$ -ური რიცხვისთვის:

$$E_{y,j,i} = B_j + E_{ji} ;$$

მაგალითად, 1-ელი თვის ბოლოს  $B_1=311000$  კვტ.სთ (იხ. წინა ცხრილი), ამიტომაც ცხრილიში წლის დასაწყისის 1-ელ სტრიქონში ჩაჯდება  $311000+10000=321000$  კვტ.სთ:



№	დღიური ელ-ენერგია	თვის დასაწყისიდან	წლის დასაწყისიდან
1	10000	10000	321000
2	$E_{ji} \rightarrow 7000$	17000	$E_{y,j,i} \rightarrow 328000$
3	10000	27000	 338000
4	2000	29000	 340000

ნახ. 4.9

3. ექსპლოატაციის დაწყებიდან  $j$ -ური თვის წინა თვეების ჩათვლით გამომუშავებული ელ-ენერგია  $i$ -ური რიცხვისთვის:

$$E_{y,j,i} = N + B_j + E_{ji} ;$$

მაგალითად, ახალი წლის დასაწყისში  $N=5000000$  კვტ.სთ., 1-ელი თვის ბოლოს  $B_1=311000$  კვტ.სთ (იხ. წინა ცხრილი), ამიტომაც თებერვლის ცხრილიში ექსპლოატაციის დაწყებიდან 1-ელ სტრიქონში ჩაჯდება  $311.000+5.000.000=5.321.000$  კვტ.სთ:

№	დღიური ელ-ენერგია	თვის დასაწყისიდან	წლის დასაწყისიდან	ექსპლოატაც. დაწყებიდან
1	10000	10000	321000	5321000
2	$E_{ji} \rightarrow 7000$	17000	328000	$E_{expl} \rightarrow 5328000$
3	10000	27000	338000	 5338000
4	2000	29000	340000	 5340000

ნახ.4.10

4.  $j$ -ური თვის გამომუშავებული ელ-ენერგია თვის რიცხვების ჯამით:

$$S_{j,i} = \sum_{i=1}^{K_j} E_{j,i} , \text{ სადაც } j \in \overline{1,12} .$$

5. თვიური ჯამური გამომუშავებული ელ-ენერგია თვის დასაწყისიდან:

$$S_j = S_{j,i} , \text{ სადაც } j \in \overline{1,12}, \text{ და } i \in \overline{1, K_j} .$$

6. წლიური ჯამური გამომუშავებული ელ-ენერგია  $j$ -ური თვის ჩათვლით წლის დასაწყისიდან:

$$S_{y,j} = B_j + S_j , \text{ სადაც}$$

$$B_j = \begin{cases} 0, & \text{if } j = 1; \\ S_{j-1}, & \text{if } j > 1. \end{cases}$$

7. ექსპლუატაციის დაწყებიდან გამომდინარე გამომუშავებული ჯამური ელ-ენერგია  $j$ -ური თვის ჩათვლით:

$$S_{\text{expl},j} = N_{j-1} + S_j, \text{ სადაც}$$

$$N_j = \begin{cases} N_0, & \text{if } j = 1; \\ N_0 + S_{j-1}, & \text{if } j > 1. \end{cases}$$

მაგალითად, ჩვენი შემთხვევისათვის წლის დასაწყისში  $N_0 = 5.000.000$  კვტ.სთ., შემდეგ მას ყოველთვიურად ემატება წინა თვის ჯამური  $S_{j-1}$  ელ-ენერგიის მოცულობა.

8. გამომუშავებული ელექტროენერგიის თვითღირებულება გაანგარიშდება წლიური ჯამური ხარჯების გაყოფით ელ-ენერგიის წლიურ მოცულობაზე:

$$C_e = \frac{C_x}{S_y}, \text{ სადაც } S_y = \sum_{j=1}^{12} B_j \text{ და } j \in \overline{1,12}.$$

აქ  $B_j$  არის  $j$ -ური თვის გამომუშავებული ელექტროენერგია.

აღნიშნული ანგარიშების გარდა შეიძლება კიდევ სხვა ეკონომიკური, საწარმოო თუ ტექნიკური მაჩვენებლების ანგარიშიც, მაგრამ ჩვენ მათ აქ არ შევეხებით.

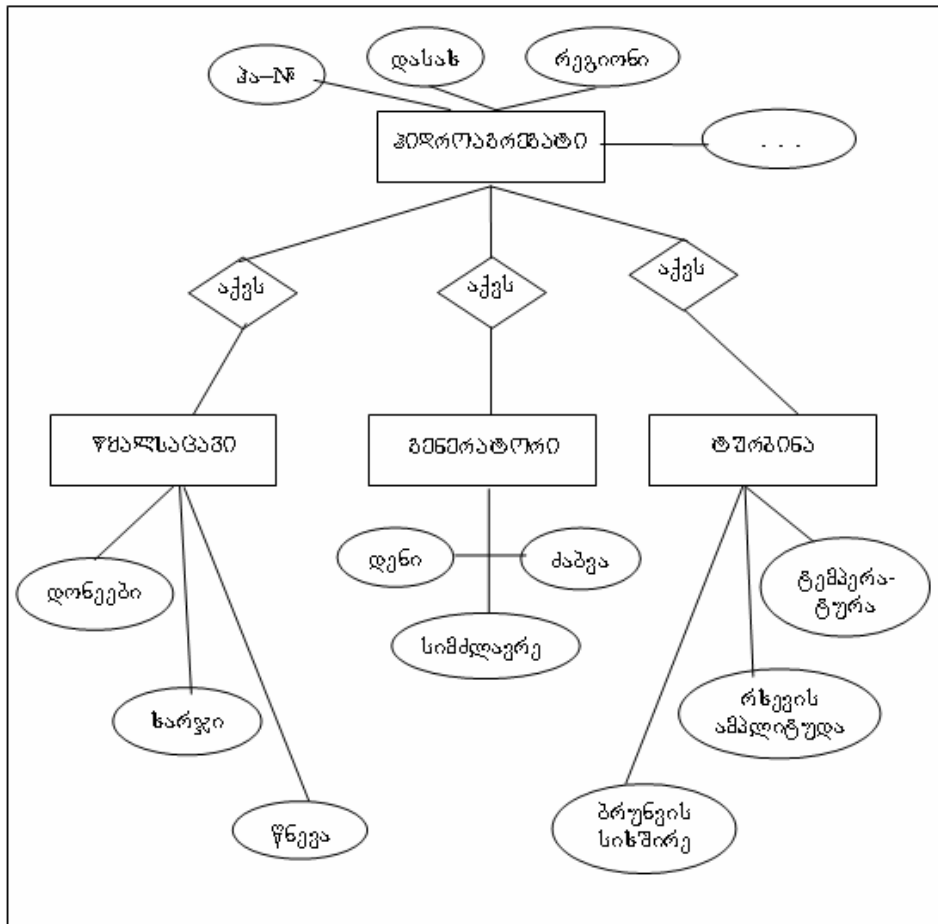
#### საპრობლემო სფეროს კონცეპტუალური მოდელი

ენგურჰესზე ელექტროენერგიის გამომუშავების აღრიცხვისა და თანამშრომელთა ხელფასების დარიცხვის ქვესისტემებისათვის საჭიროა მონაცემთა ბაზების აგება. ეს პროცესი სამი ძირითადი ეტაპისაგან შედგება: - საპრობლემო სფეროს კონცეპტუალური მოდელის დამუშავება; - მონაცემთა ბაზის ლოგიკური სტრუქტურის დაპროექტება; - მონაზემთა ბაზის ფიზიკური ჩანაწერების შეტანა.

საპრობლემო სფეროს კონცეპტუალური მოდელი მუშავდება არსთა-დამოკიდებულების მეთოდის საშუალებით და მას ER\_Model-ს უწოდებენ (Entity-Relations Model). ესაა გრაფიკული ასახვის ინსტრუმენტი მონაცემთა ბაზებისათვის საპრობლემო სფეროს კონცეპტუალური მოდელის ასაგებად.



4.11 ნახაზზე მოცემულია ჩვენი ამოცანის შესაბამისი სისტემის კონცეპტუალური მოდელის სქემა.



ნახ.4.11

„ენგურჰესი“, როგორც მართვის ავტომატიზებული სისტემის კონკრეტული მაგალითი, ადამიან-მანქანურ სისტემათა კლასს მიეკუთვნება. მისი პროექტი მოიცავს დაგეგმვის, აღრიცხვის, ტექნოლოგიური და ეკონომიკური პროცესების ოპერატიული მართვის, სამეურნეო ანალიზისა და სხვა მრავალი შრომატევადი ამოცანის ერთობლიობას.

კომპიუტერულ სისტემასთან უნდა იმუშაონ ენერგეტიკოს-ტექნოლოგებმა და ჰიდრო-ტექნიკოსებმა (ინჟინრებმა), დისპეტჩერებმა, ეკონომისტებმა, ფინანსებისა და საბუღალტრო დარგის სპეციალისტებმა. შეიძლება ვთქვათ, რომ უმრავლესობას მათ შორის აქვთ კომპიუტერული ცოდნა საოფისე სისტემების ექსპლუატაციის დონეზე. ცოდნა, გამოცდილება და მართვის უნარი

ასეთი ობიექტებისა. მაგრამ მათ არ გააჩნიათ კომპიუტერული სისტემების აგებისა და დაპროგრამების გამოცდილება.

ინჟინრებმა და ბუღალტრებმა რომ შეძლონ ჩვენს სისტემაში მუშაობა, მცირე კომპიუტერული კონსულტაციის შემდეგ, საჭიროა თვით ეს ავტომატიზებული სისტემა იყოს დაპროექტებული ასეთ სპეციალისტებზე ორიენტირებულად.

ამის განხორციელება კი შეიძლება კომპიუტერზე შესაბამისი დიალოგური პროცედურების შერჩევით და რეალიზებით. ჩვენი შემთხვევისათვის ეს პროცედურები უნდა იყოს მსგავსი ტექნიკურ განყოფილებაში ელექტროენერჯის აღრიცხვის საქმის წარმოებისა და ბუღალტრიაში ხელფასების აღრიცხვის ტრადიციული პროცესის განხორციელებისა.

დიალოგური პროცედურები არის „სცენარი“ იმისა, თუ როგორ უნდა იმუშაოს მომხმარებელმა კომპიუტერულ სისტემასთან. ყველა დეტალი, დაწყებული სისტემის სტარტიდან მის ფინიშამდე უნდა იყოს წინასწარ გათვლილი.

კომპიუტერულ სისტემას უნდა ჰქონდეს „დახმარების“ (Help) უნარი, ანუ ბუღალტერმა, მაგალითად, გარეშე კონსულტანტის დაუხმარებლად უნდა მიიღოს რჩევა ან ცნობა თვით სისტემისაგან.

საჭიროა აგრეთვე განხორციელდეს სისტემის საწყისი მონაცემების შეტანისა და საბოლოო შედეგების ავტომატიზებული კონტროლი. ეს უკანასკნელიც დიალოგური პროცედურებითაა რეალიზებული.

დღეისათვის საკმაოდ გავრცელებულია მსგავსი ამოცანების გადასაწყვეტად MicroSoft Office (მაიკროზოფტ ოფისის) Excel-პროგრამული პაკეტის გამოყენება. ამ ელექტრონული ცხრილების „მძლავრ კალკულატორს“ მრავალი სასარგებლო ფუნქცია გააჩნია, მათ შორის მათემატიკური სტატისტიკის, მათ-ლოგიკის, მონაცემთა ბაზების და თვით საფინანსო-ეკონომიკურიც კი.

ამგვარად, დისერტაციაში გამოვიყენებთ Excel პაკეტს და დიალოგური პროცედურებიც განვიხილავთ გვექნება ამ სისტემასთან მუშაობის სტილში.

MS-Excel პროგრამული პაკეტი თანამედროვე მძლავრ კომპიუტერულ ინსტრუმენტებს მიეკუთვნება, რომელშიც რეალიზებულია როგორც ვიზუალური დაპროგრამების საშუალებანი (არატრადიციული დაპროგრამება), ასევე ობიექტ-ორიენტირებული მოდელირების მეთოდი.

შემდგომში ჩვენ განვიხილავთ დიალოგური პროცედურების დამუშავებისა და მისი გამოყენების რეჟიმებს.

ნახაზებზე 4.12- 4.15 მოცემულია კომპიუტერზე რეალიზებული ამოცანების საინტერფეისო ცხრილების ფრაგმენტები: ელენერგის აღრიცხვისა და ხელფასისათვის.

ელექტროენერგიის გამომწვავების  
აღრიცხვა

თვე: **თებერვალი - 2**

ნაწილი

თვის რიცხვი	ელ-ენერგიის დღიური გამომწვავება (ათასი კვტ.სთ)	თვის დასაწყისიდან (ათასი კვტ.სთ)	წლის დასაწყისიდან (ათასი კვტ.სთ)	მსკლუბატაციის დაწყებიდან (ათასი კვტ.სთ)	თვის № j	წლის დასაწ. B	ექსპლ.დ აწყ. N
1	10000	10000	321000	5321000	1	0	5000000
2	7000	17000	328000	5328000	2	311000	5311000
3	10000	27000	338000	5338000	3	350000	5350000
4	2000	29000	340000	5340000	4	0	0
5	10000	39000	350000	5350000	5	0	0
6	0	39000	350000	5350000	6	0	0
7	0	39000	350000	5350000	7	0	0
8	0	39000	350000	5350000	8	0	0
9	0	39000	350000	5350000	9	0	0
10	0	39000	350000	5350000	10	0	0
11	0	39000	350000	5350000	11	0	0
12	0	39000	350000	5350000	12	0	0
13	0	39000	350000	5350000			
14	0	39000	350000	5350000			
15	0	39000	350000	5350000			
16	0	39000	350000	5350000			
17	0	39000	350000	5350000			
18	0	39000	350000	5350000			
19	0	39000	350000	5350000			
20	0	39000	350000	5350000			
21	0	39000	350000	5350000			
22	0	39000	350000	5350000			
23	0	39000	350000	5350000			
24	0	39000	350000	5350000			
25	0	39000	350000	5350000			
26	0	39000	350000	5350000			
27	0	39000	350000	5350000			
28	0	39000	350000	5350000			
29	0	0	0	0			
30	0	0	0	0			
31	0	0	0	0			
<b>ჯამი:</b>	<b>39000</b>	<b>39000</b>	<b>350000</b>	<b>5350000</b>			

კალენდარი

თვე	დღეების რაოდ.
1	31
2	28
3	31
4	30
5	31
6	30
7	31
8	31
9	30
10	31
11	30
12	31

ნახ.4.12

ელექტროენერგიის  
ბამომშუშავების აღრიცხვა

თვე: მარტი-

თვის რიცხვი	ელ-ენერგიის დღიური ბამომშუშავებ ა (ათასი კვტ.სთ)	თვის დასაწყისიდან ნ (ათასი კვტ.სთ)	წლის დასაწყისიდან ან (ათასი კვტ.სთ)	ექსპლუატაცი ის დაწყებიდან (ათასი კვტ.სთ)
1	10000	10000	360000	5360000
2	5000	15000	365000	5365000
3	0	15000	365000	5365000
4	0	15000	365000	5365000
5	0	15000	365000	5365000
6	0	15000	365000	5365000
7	0	15000	365000	5365000

ნახ. 4.13

ნაშთები

თვის№ j	წლის დასაწ. B	ექსპლ.დ აწყ. N
1	0	5000000
2	311000	5311000
3	350000	5350000
4	365000	5365000
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0

კალენდარი

თვე	დღეების რაოდ.
1	31
2	28
3	31
4	30
5	31
6	30
7	31
8	31
9	30
10	31
11	30
12	31

ნახ.4.14

თანამშრომლები

№	გვარი ს.	თანამდებობა	ხელფასი	ტელეფონი
9401	აბულაძე ა.	ასისტენტი	100.00	23-23-23
9402	ბალაღაძე ბ.	დოცენტი	240.00	22-22-22
9407	გიორგაძე გ.	პროფესორი	500.00	29-29-29
9415	დოლიძე დ.	ლაბორანტი	11.00	95-95-95
9422	დოლიძე თ.	დოცენტი	360.00	22-11-11
9423	თოფურია ნ.	ასისტენტი	250.00	32-32-32
9430	შონია შ.	მენეჯერი	700.00	98-98-98

ხელფასის შკალა		დასაბმელი %
<<	9	0
10	200	12
201	350	15
351	600	17
601	>>	20

24  
46.5  
89

ხელფასის უწყისი

December

№	ხელფასი	დარიცხული თანხა	დასაბმელი თანხა	საშემდგომი აბონენტი	საშემდგომი მითითების თვის	ხელზე ბანკში
9401	100.00	1200.00	1092.00	169.2	187.4	81.80
9402	240.00	2880.00	2772.00	477.2	523.4	193.80
9407	500.00	6000.00	5892.00	1049.2	1147.4	401.80
9415	11.00	132.00	24.00	2.64	2.88	10.76
9422	360.00	4320.00	4212.00	741.2	811.4	289.80
9423	250.00	3000.00	2892.00	499.2	547.4	201.80
9430	700.00	8400.00	8292.00	1489.2	1627.4	561.80
	2161.00				4847.28	1741.56

მიმდინარე თვის №
12

ნახ.4.15

4.4 მონაცემთა ბაზების ავტომატური დაპროექტება ენერგეტიკის სფეროში მომუშავე პერსონალისათვის.

აქ განხილულია ენერგეტიკის სფეროში მომუშავე პერსონალისათვის მონაცემთა ბაზების ავტომატიზებული დაპროექტების ამოცანა. მისი გადაწყვეტა საშუალებას აძლევს ენერგეტიკოს-სპეციალისტს მონაცემთა ბაზა ააგოს კომპიუტერის დახმარებით საკუთარი ცოდნისა და გამოცდილების საფუძველზე ენერგო-სფეროში. მოდელირების პროცესის წარმართვისათვის მას ესაჭიროება კონკრეტული ამოცანის ირგვლივ გარკვეული არესებული კანონზომიერებების (ფაქტების) ტექსტური ფორმით ჩაწერა [2,3]. მაგალითად:

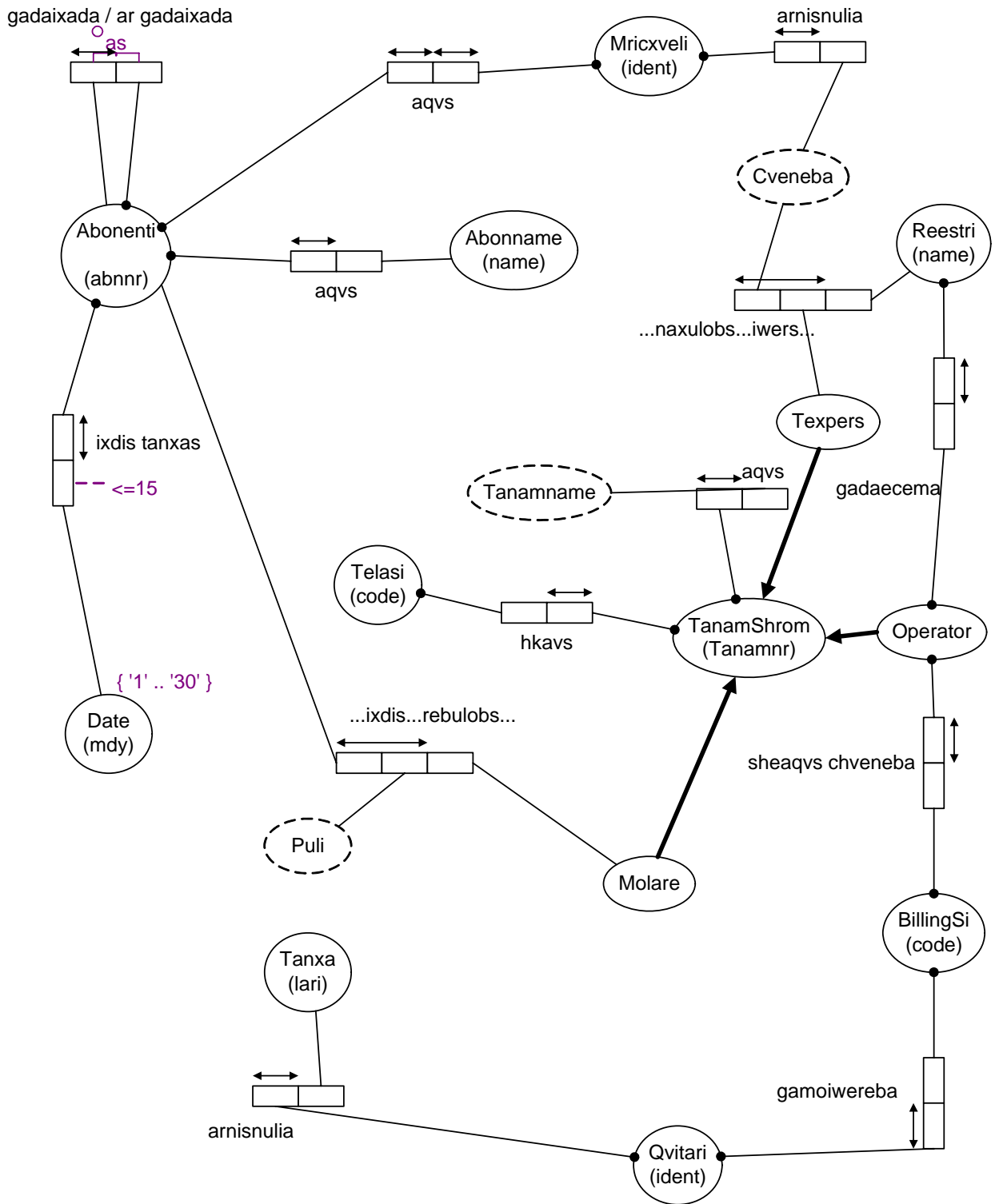
- f1 აბონენტს აქვს გვარი ;
- f2 აბონენტს აქვს მრიცხველი
- f3 აბონენტს იხდის თანხას ყოველი თვის 15 რიცხვამდე ;

- f4 მრიცხველზე აღნიშნულია მოხმარებული ელ.ენერგია კვტ/სთ ;
- f5 არსებობენ აბონენტები, რომლებმაც გადაიხადეს ან არ გადაიხადეს მოხმარებული ელ.ენერგიის საფასური ;
- f6 თელასში მუშაობენ თანამშრომლები;
- f7 თანამშრომელს აქვს გვარი
- f8 ოპერტორი არის თანამშრომელი ;
- f9 ტექნიკური პერსონალი არის თანამშრომელი ;
- f10 მოლარე არის თანამშრომელი ;
- f11 ტექნიკური პერსონალი აღნიშნავს აბონენტთა მრიცხველების ჩვენებებს რეესტრში ;
- f12 რეესტრში არსებული მონაცემები ოპერატორს შეაქვს ბილინგის მონაცემთა ბაზაში ;
- f13 ბილინგის მონაცემთა ბაზიდან გამოიწერება ქვითარი;
- f14 ქვითარზე აღნიშნულია თანხა ;
- f15 ქვითარს ღებულობს აბონენტი ;
- f16 აბონენტი იხდის ფულს მოლარესთან ;  
და ა.შ.

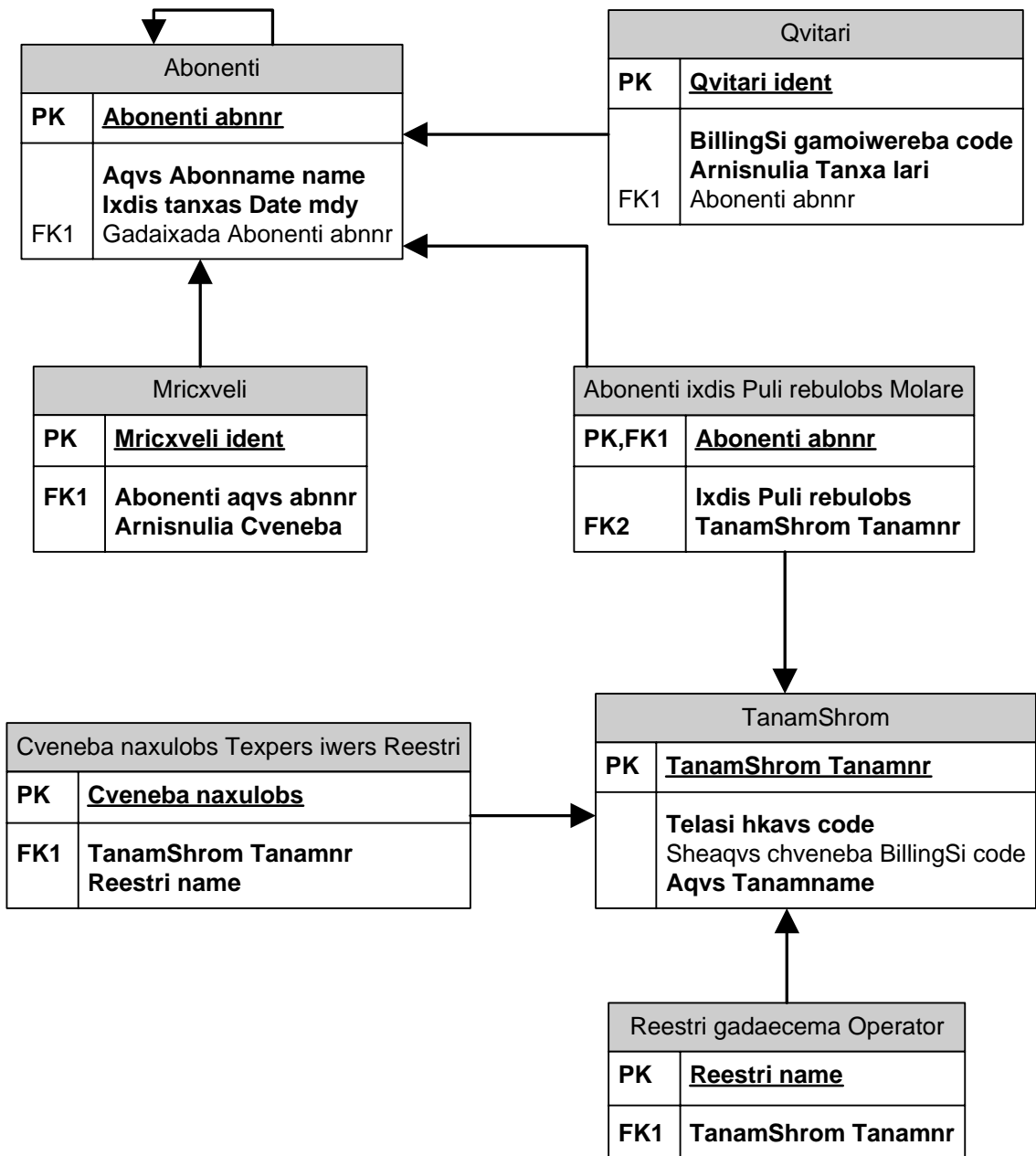
ზემოჩამოთვლილი ფაქტების შესაბამისი ORM (Object-Role Model)–დიაგრამა, რომელიც აიგება MsVisio მოდელირების ინსტრუმენტით, მოცემულია 4.16 ნახაზზე [4]. რაც მეტი ფაქტია აღწერილი, მით უფრო ზუსტია ობიექტის საშედეგო სემანტიკური (კონცეპტუალური) მოდელი.

შემდგომ ეტაპზე ORM-დიაგრამიდან ავტომატურად ვაპროექტებთ ERM (Entity Relationship Model) დიაგრამას (ნახ.4.17), რომლის საფუძველზეც აიგება რელაციურ მონაცემთა ბაზების ლოგიკური სტრუქტურა.

მესამე ეტაპზე ER-მოდელის საფუძველზე ვახდენთ პროგრამული სახით მონაცემთა აღწერის .DDL ფაილის ავტომატურ გენერირებას.



ნახ.4.16 ORM-დიაგრამის ფრაგმენტი



ნახ.4.17 ERM-დიაგრამის ფრაგმენტი

ქვემოთ მოცემულია .DDL ფაილის კოდის ფრაგმენტი:

```

Create new table "Reestri gadaecema Operator". */
/* "Reestri gadaecema Operator" : Table of Reestri gadaecema Operator */
/* "Reestri name" : Reestri identifies Reestri gadaecema Operator */
/* Role two (Reestri) of fact: Reestri gadaecema Operator is identified by
*/
/* Role two (Reestri name) of fact: Reestri is identified by */
/* "TanamShrom Tanamnrr" : TanamShrom Operator has Reestri gadaecema Operator
*/
create table "Reestri gadaecema Operator" (

```



```

"Reestri name" char(10) not null,
"TanamShrom Tanamn" char(10) not null)
go
alter table "Reestri gadaecema Operator"
add constraint "Reestri gadaecema Operator_PK" primary key ("Reestri name")
go
create table "Abonenti ixdis Puli rebulobs Molare" (
"Abonenti abnnr" char(10) not null,
"Ixdis Puli rebulobs" char(10) not null,
"TanamShrom Tanamn" char(10) not null)
go
alter table "Abonenti ixdis Puli rebulobs Molare"
add constraint "Abonenti ixdis Puli rebulobs Molare_PK" primary key ("Abonenti
abnnr")
go

```

.....

ამგვარად, კომპიუტერულ სისტემაში მონაცემთა ბაზების სტრუქტურები და ცხრილთაშორისი კავშირები მზადაა კონკრეტული მონაცემების შესატანად. აუცილებლობის შემთხვევაში შესაძლებელია მიღებული მონაცემთა ბაზის სტრუქტურის მანუალური მოდიფიკაცია.

#### 4.5. ენერგეტიკული ობიექტების კონტროლის სისტემა

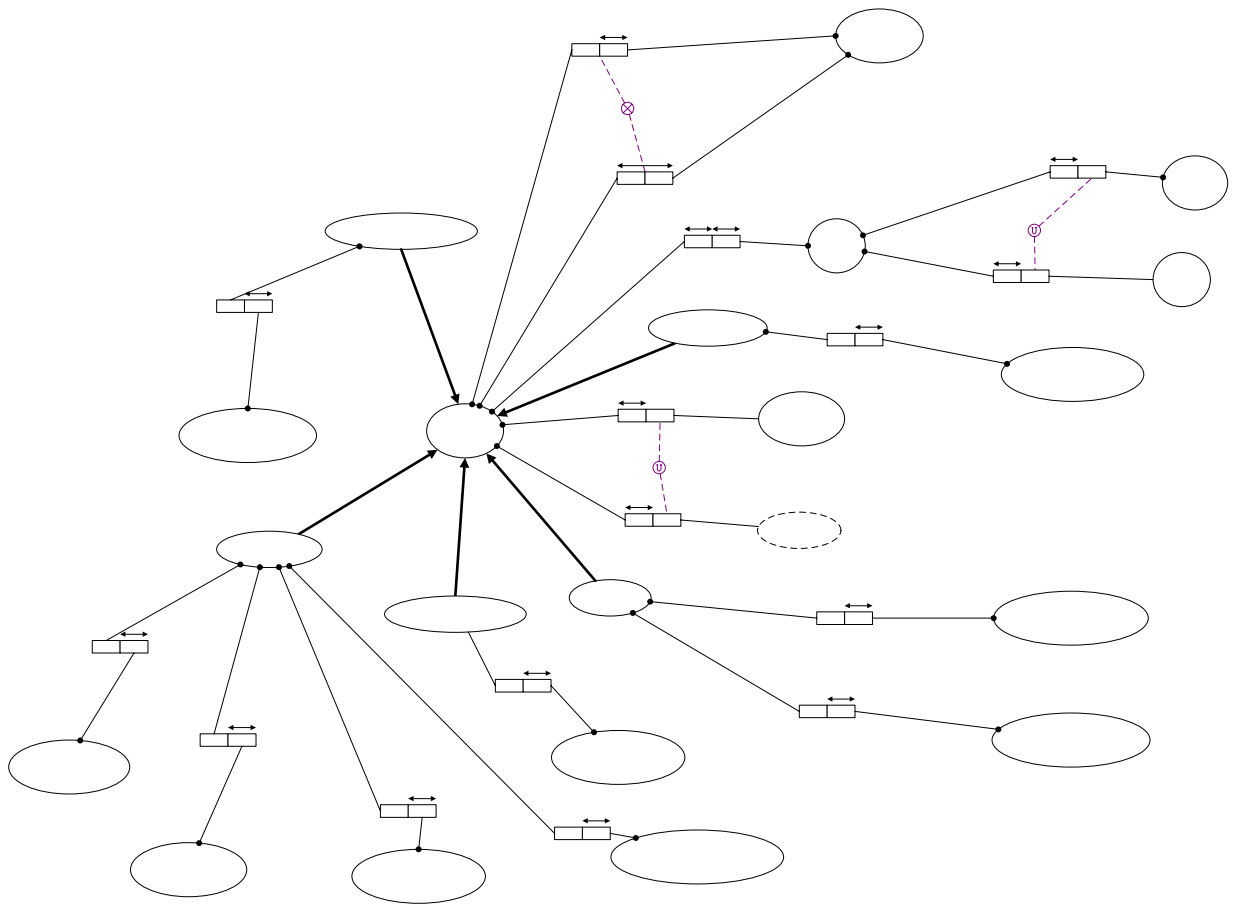
##### კლიენტ-სერვერული არქიტექტურის ბაზაზე

ახლა განვიხილოთ ამოცანა, რომელიც უშუალოდ უკავშირდება მონაცემთა ბაზისათვის ინფორმაციის შეკრებისა და მომზადების საკითხს. ჩვენს შემთხვევაში ეს მომხმარებელზე მიწოდებული ელ-ენერჯის აღრიცხვის და კონტროლის ამოცანაა.

„კომერციული ხარჯების“ შემცირების ერთ-ერთი ეფექტური საშუალებაა სადარბოვებში მრიცხველების დადგმა [21]. დატაცებული ელექტროენერჯის დაფიქსირების ავტომატიზებული სისტემა პროგრამულად რეალიზებულია კლიენტ-სერვერ არქიტექტურით MsSQL\_Server პაკეტის გამოყენებით. კლიენტ-მომხმარებელთათვის დამუშავებულია ინტერფეისები MsAccess, Excel,

VBA და Visual Studio.NET ინტეგრირებული პროგრამული პაკეტებით, ინტერნე-ტული პროგრამის ASP.NET გამოყენებით [5].

დატაცებული ელექტროენერჯის დაფიქსირების მიზნით აუცილებელია ჩატარდეს შემდეგი სამუშაო. სადარბაზოს მრიცხველზე აღირიცხება შემავალი ელ-ენერჯის მოცულობა. როგორც ცნობილია, თელასის ტექნიკური ჯგუფის თანამშრომელი, თვეში ერთხელ აფიქსირებს აბონენტთა მრიცხველის ჩვენებებს რეესტრში. ამავე დროს, იგი ჩაიწერს სადარბაზოს მრიცხველის ჩვენებებსაც, რომლებსაც ოპერატორი შეიტანს კომპიუტერში, კლიენტ-მომხმარებელთათვის განკუთვნილი ინტერფეისის საშუალებით. აღსანიშნავია, რომ კლიენტ-მომხმარებელს შეუძლია ქსელში ჩართული ნებისმიერი კომპიუტერიდან (ასევე მობილური ტელეფონიდან) შეიტანოს მონაცემები, ბრაუზერში საჭირო მისამართისა და პაროლის მითითების შემდეგ.

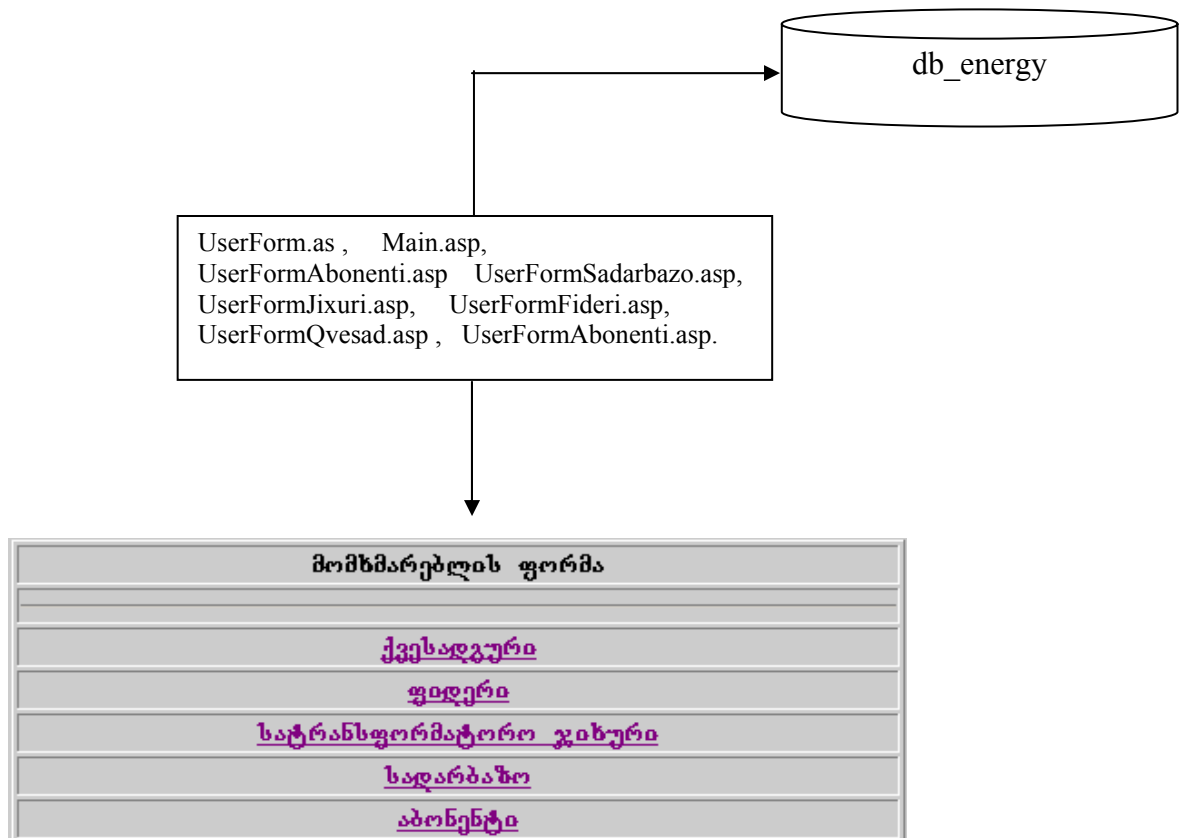


ნახ.4.18

4.18 ნახაზზე წარმოდგენილი ORM-დიაგრამა აღწერს ენერგეტიკის სფეროში მომუშავე პერსონალის მიერ ტექნიკური დანაკარგების შეტანის პროცესს.

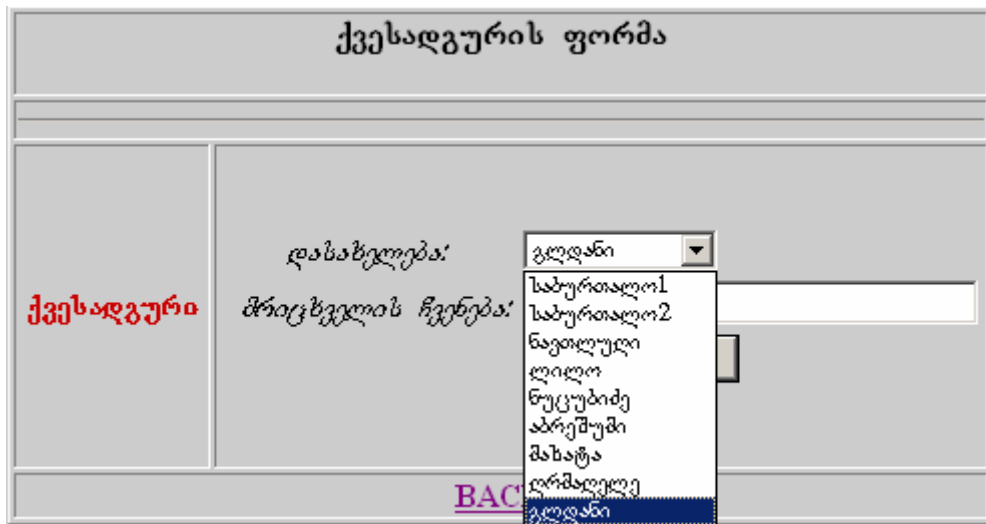
მაგალითად, Internet Explorer-ის მისამართის ველში კომპიუტერის ლოკალური ქსელის მისამართის

192.168.8.55/db\_energy/main.asp



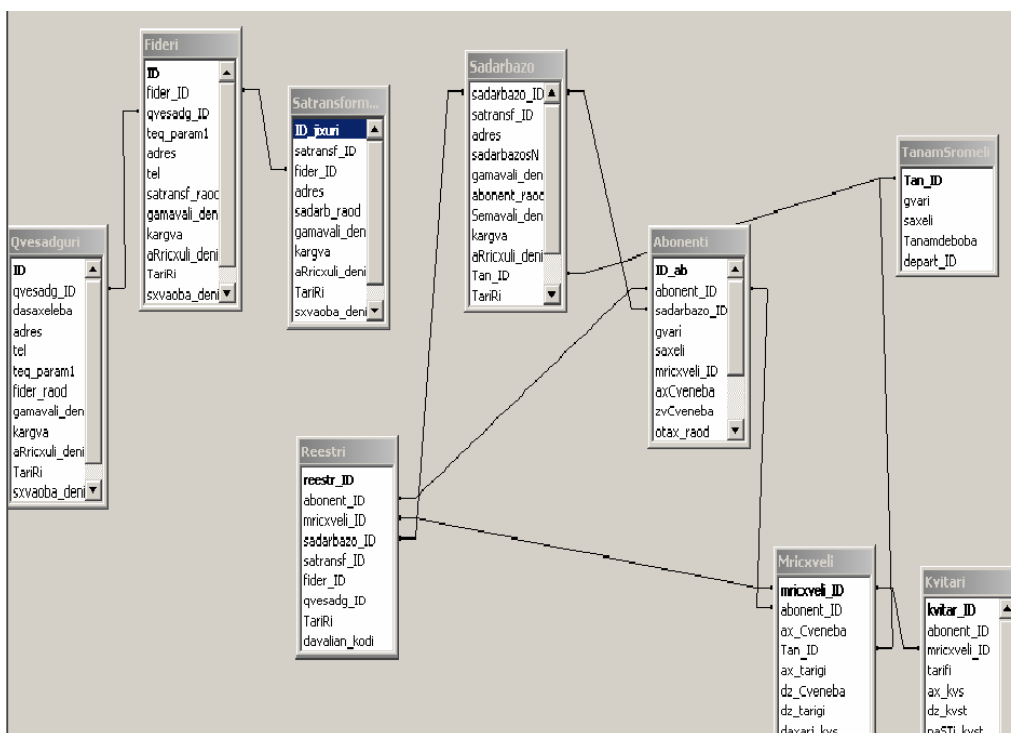
ნახ. 4.19

მითითებისა და პაროლის აკრების შემდეგ ეკრანზე გამოჩნდება მე-4.19 ნახაზზე ნაჩვენები კლიენტ-მომხმარებლის ინტერფეისი. ამ ინტერფეისის საშუალებით ქვესადგურების (იხ. ნახ.4.20), სატრანსფორმატოროებისა და ელ-ჯიხურების უფროსებს, პაროლის მითითების შემდეგ შეუძლიათ შეიტანონ შესაბამისი ჩვენებები. კლიენტ-მომხმარებლიდან შეტანილი ინფორმაცია .ASP მოდულების საშუალებით ჩაიწერება მონაცემთა ბაზაში (db\_energy).



ნახ.4.20

ნახ.4.21 - ნახ.4.23 მოცემულია MsAccess-ში შესრულებული db\_energy მონაცემთა ბაზები და ნაჩვენებია თუ როგორ ჩაიწერა აბონენტის შესახებ ინფორმაცია რეესტრიდან db\_energy მონაცემთა ბაზაში თავის მხრივ, ადმინისტრატორს მისთვის განკუთვნილი ინტერფეისის საშუალებით, შეუძლია დაათვალიეროს ნებისმიერი ქვესადგურის, ფიდერის, სატრანსფორმატორო ჯიხურის ან სადარბაზოს ჩვენება. ასევე შეუძლია აღმოაჩინოს კონკრეტულად სად მოხდა ელექტრო ენერჯიის დატაცება.



ნახ.4.21

4.24 ნახაზზე ნაჩვენებია ენერგომომხმარებლის ავტომატიზებული კონტროლის სისტემის კლიენტ-სერვერული არქიტექტურით რეალიზებული პაკეტის პროგრამების კატალოგი.

SadarMricxvelis_ID	satransf_ID	adres	sadarbazosN	gamavali_deni	abonent_raod	Semavali_deni	
1122	0157	qavTaraZis 10	3	123123	0	8000	
abonent_ID	gvari	saxeli	mricxveli_ID	axCveneba	zvCveneba	daxarjuli_kvt	otax
1234	MumlaZe	Mindia	037522	3000	1800		
5678	SalariZe	Dato	567893	1770	1200		
9870	ladze	la	234510	5600	1245		
1256	AbashiZe	Lali	098765	2345	1235		
0				0	0	0	0
1123	0157	42342sdasdfs	7	1000000		0	0
abonent_ID	gvari	saxeli	mricxveli_ID	axCveneba	zvCveneba	daxarjuli_kvt	otax
2345	BorcvaZe	Liza	045673				
1345	Mikava	Gocha	234156				
2340	AsaTiani	Vaxtang	234510	123123	0		
0				0	0	0	0
1124	0157	cxsgsfdfs	4	234234		0	0
abonent_ID	gvari	saxeli	mricxveli_ID	axCveneba	zvCveneba	daxarjuli_kvt	otax
3456	Kantaria	Dato	098765	3000	2000		
3450	Gilauri	Mariam	234510	12312124	0		
4560	KobiZe	Zaza	234156	12312312	0		
0				0	0	0	0
1125	0368	kazbegi 12	1			10	0
abonent_ID	gvari	saxeli	mricxveli_ID	axCveneba	zvCveneba	daxarjuli_kvt	otax
4567	Margiani	Lela	945678				
0				0	0	0	0
1126	0368	asatiani 9	2			15	0
1127	0368	tabidze 24	3			14	0
1128	0134	kipSidze 12	4			13	0
1129	0134	kavtaradze 10	1			12	0
1130	0134	djikia 12	2			15	0
1131	0134	kostava 35	3			0	0
0						0	0

ნახ. 4.22

სადარბაზო მონიტორინგის-№ 1122 თარიღი

სატრანსფ-№ 0157 მდგომარეობა-კოდი

მისამართი ქავთარაძის 10

სადარბაზოს № 3

შემავალი-დენი 8000

აბონენტების ჯამური 7235 სხვაობა-დენი 765

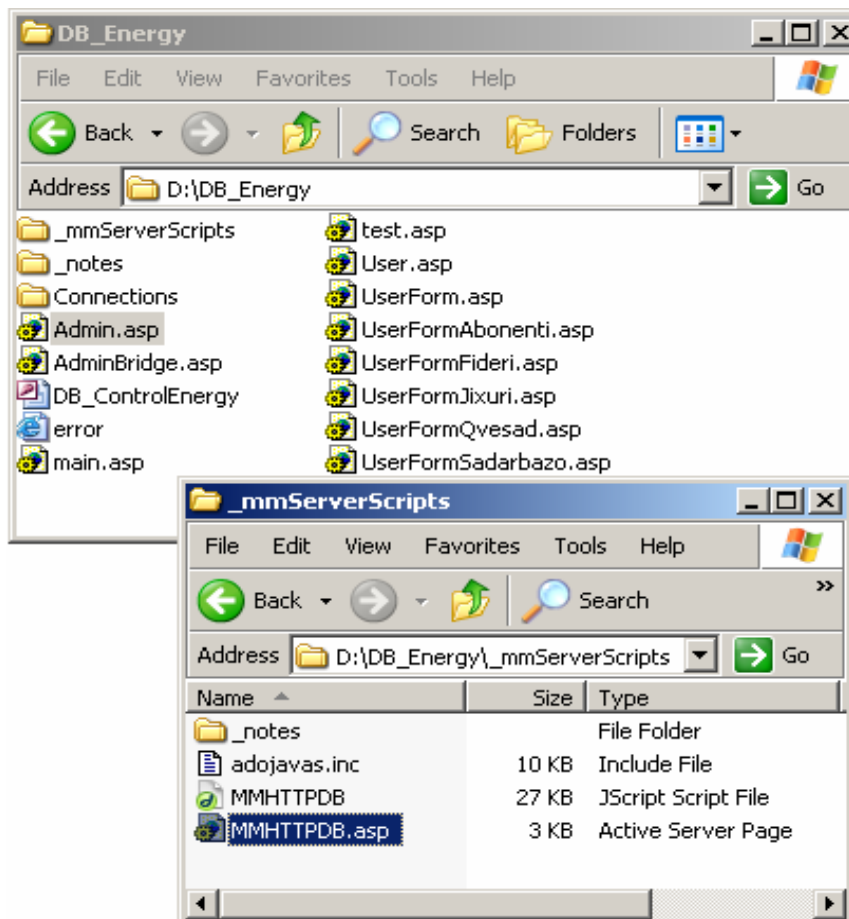
დანახარჯი

აბონენტის ფორმა

აბონენტ_№	ანჩენება	ქველი ჩენება	სხვაობა:
1234	3000	1800	1200
5678	1770	1200	570
9870	5600	1245	4355
1256	2345	1235	1110
0	0	0	0

Record: 1 of 4

ნახ.4.23



ნახ.4.24

### მეოთხე თავის დასკვნები

1. შემოთავაზებულია განაწილებული ენერგეტიკული ოფის-ობიექტების ბიზნეს-პროცესების ობიექტ-ორიენტირებული მოდელირების, ანალიზისა და დაპროექტების ამოცანების კომპიუტერული რეალიზაცია.
2. დამუშავებულია განაწილებული ენერგეტიკული ოფის-სისტემის მონაცემთა ბაზის ფიზიკური სტრუქტურები, მათი მოდიფიკაციის ალგორითმები და ადმინისტრირების ფუნქციები, კლიენტ-სერვერული არქიტექტურის გამოყენებით.
3. შემოთავაზებულია ენერგეტიკული ოფის-სისტემის მომხმარებელთა ინტერფეისები, მათი როლების შესაბამისად. კლიენტ-მომხმარებელთათვის დამუშავებულია ინტერფეისები

MsAccess, Excel, VBASCRIPТ და Visual Studio.NET ინტეგრირებული პროგრამული პაკეტებით, ASP.NET და ADO.NET გამოყენებით.

4. ასევე განხილულია ORM–დიაგრამიდან ER-დიაგრამის და შემდგომ .DDL კოდის ავტომატური ფორმირებისა და შემდგომ მისი MsSQL Server-თან მიერთების წესები კონკრეტული მაგალითისათვის.

5. განაწილებული ენერგოსისტემის ამოცანების კომპიუტერული რალიზაცია განხილება მთლიან სპექტრში, დაწყებული, მაგალითად, ენგურჰესის ჰიდროელექტროსადგურზე ელენერჯის გამომშავების აღრიცხვიდან, დამთავრებული თელასის სააბონენტო გადასახადების აღრიცხვისა და კონტროლის ამოცანებით, ინტერნეტული მოხმარებისათვის (მონიტორინგის სისტემა).

## დასკვნები

ჩატარებული თეორიული და ექსპერიმენტული გამოკვლევების საფუძველზე შეიძლება გავაკეთოთ შემდეგი დასკვნები:

1. გამოკვლეულია ენერგეტიკული დარგის განაწილებულ ოფის-ობიექტებზე საქმიანი პროცესების ეფექტური მართვის პრობლემები და ამოცანები უნიფიცირებული დოკუმენტაციისა და დოკუმენტბრუნვის სისტემის კლასიფიკაციისა და დიაგნოსტიკური ანალიზის საფუძველზე;

2. განსაზღვრულია ენერგეტიკული ოფის-ობიექტების სტრუქტურებისა და მათი ფუნქციონირების წესების აღწერის სისტემა, არაფორმალური მეთოდებით;

3. დამუშავებულია ოფის-სისტემის ობიექტ-როლური მოდელების ერთობლიობა და მათი ქცევის წესები, კატეგორიული მიდგომით,

ფორმალური ენის გრამატიკისა და ლოგიკურ-ალგებრული მეთოდების საფუძველზე;

4. დამუშავებულია განაწილებული ენერგეტიკული ოფის-სისტემის კონცეპტუალურ სქემათა აგების ალგორითმები ობიექტ-ორიენტირებული, კლასთა სემანტიკური კავშირების საფუძველზე;

5. განსაზღვრულია ოფის-სისტემის მონაცემთა განაწილებული ბაზის ოპტიმალური ლოგიკური სტრუქტურები და დამუშავებულია მათი რეალიზაციის ალგორითმები მონაცემთა რელაციური მოდელის საფუძველზე;

6. დამუშავებულია განაწილებული ენერგეტიკული ოფის-სისტემის მონაცემთა ბაზის ფიზიკური სტრუქტურები, მათი მოდიფიკაციის ალგორითმები და ადმინისტრირების ფუნქციები, კლიენტ-სერვერული არქიტექტურის გამოყენებით;

7. დამუშავებულია ენერგეტიკული ოფის-სისტემის მომხმარებელთა ინტერფეისები, მათი როლების შესაბამისად, აგრეთვე განაწილებული ავტომატიზებული სისტემის საერთო გამოყენების რესურსების ეფექტური მართვის მოდელი და მისი კვლევის ალგორითმები პეტრის ქსელებით;

8. თეორიული შედეგები რეალიზებულია პროგრამულად მონაცემთა თანამედროვე ობიექტ-ორიენტირებული, განაწილებული რელაციური ბაზების ( sSQL Server, MsAccess, ADO.Net), უნიფიცირებული მოდელირების ტექნოლოგიის (UML, MsVisio) და ქსელური პროგრამული სისტემების (Internet/Intranet) გამოყენებით.



### ბამოყენებული ლიტერატურა:

1. ბახია გ., თოფურია ნ., სურგულაძე გ. ბიზნეს-პროცესების ავტომატიზაციის სისტემების ევოლუცია // სტუ-ს შრ., №1(451), თბ., 2004.
2. ბოტჭე კ., სურგულაძე გ., დოლიძე თ., შონია ო., სურგულაძე გ. თანამედროვე პროგრამული პლატფორმები და ენები // სტუ, თბილისი, 2003.
3. გოგიჩაიშვილი გ., სურგულაძე გ., შონია ო. დაპროგრამების მეთოდები // სტუ, თბილისი, 1997.
4. გოგიჩაიშვილი გ., სურგულაძე გ., გულუა დ. და სხვ. ეკონომიკური რეფორმების მართვის სრულყოფა უახლესი ინფორმაციული ტექნოლოგიებით. //სტუ-შრ.კრებ. №7(446). თბილისი, 1996.
5. გუგუშვილი ა., ხუროძე რ., იმედაძე თ., გარგი დ., კოტრიკაძე ო. და სხვ., XXI საუკუნის წიგნი // სტუ, 1999.
6. დოლიძე თ., გულუა დ., სურგულაძე მ. ჯანდაცვის ობიექტების ბიზნეს-პროცესების მოდელირებისა და მართვის ავტომატიზებული სისტემის დაპროექტების ტექნოლოგია. საქმეცნ.აკად. მართვის სისტ. ინსტ. საერთაშ. სამეცნ.კონფ.შრ. „მართვისა და ენერჯეტიკის პრობლემები“. თბილისი, 20047.
7. კამკამიძე კ., სამხარაძე რ., შენგელია ნ. რეჟიმების ტექნოლოგის სამუშაო ადგილის სრულყოფის შესახებ. // თბილისი, 1989.
8. კიკნაძე მ., კაპანაძე დ., თოფურია ნ. ტესტურ დავალებათა ბაზით შესასწავლი თემების დაფარვის ამოცანა. //სტუ-ს შრ.კრ. №3(437), თბილისი, 2006.
9. სამხარაძე რ.. მონაცემთა ბაზები Visual Studio.NET .სტუ, თბილისი, 2005.
10. სურგულაძე გ., შონია ო., ყვავაძე ღ. მონაცემთა ბაზების მართვის სისტემები: MsAccess, MsSQLServer, InterBase, Oracle, JDBC, CORBA. //სტუ, თბილისი, 2004.
11. სურგულაძე გ., დოლიძე თ., თურქია ე., ოხანაშვილი მ. საწარმოო ფორმებში მარკეტინგული პროცესების მართვის ინფორმაციული სისტემის დაპროექტება და რეალიზაცია UML- ტექნოლოგიით //სტუ-ს შრომები, № 4(437), თბილისი, 2001.
12. სურგულაძე გ., თურქია ე. ბიზნეს-გეგმის ავტომატიზებული დამუშავების პროცესის სამუშაო ნაკადების მართვის სისტემა // სტუ-ს შრომები, № 8(424), თბილისი, 1998.

13. სურგულაძე გ., ჩაჩანიძე გ., გულუა დ.. ობიექტ-ორიენტირებული მოდელირება **AMIS** ტექნოლოგიისა და პეტრის ქსელების გამოყენებით განათლების სისიტემაში. // თბილისი, „ინტელექტი“, №1(7), თბილისი, 2000.
14. სურგულაძე გ., ვედეკინდი ჰ., თოფურია ნ. განაწილებული ოფის-სისტემების მონაცემთა ბაზების დაპროექტება და რეალიზაცია -ტექნოლოგიით. // მონოგრაფია, სტუ. 2006.
15. სურგულაძე გ., თოფურია ნ., ქაჩიბაია ვ., ილდიზი ი., გელიტაშვილი ნ. კონცეპტუალური მოდელის დაპროექტება UML-ტექნოლოგიით უნივერსიტეტის მონაცემთა ბაზების აგებისას.// სტუ-ს შრ.კრებ. 1 მას-კათ.35-ე წლ.საიუბილეო გამოცემა. თბილისი, 2006.
16. სურგულაძე გ. დაპროგრამების ვიზუალური მეთოდები და ინსტრუმენტები: UML, Ms VISIO, C++Builder. // სტუ, თბილისი, 2005.
17. რეისიგი ვ., სურგულაძე გ., გულუა დ., ვიზუალური ობიექტ-ორიენტირებული დაპროგრამების მეთოდები // სტუ, თბილისი, 2002.
18. ჩაჩანიძე გ., სურგულაძე გ., ქელბაქიანი ვ.. სწავლების პროფილური დიფერენცირების მათემატიკური მოდელი.// თბილისი, „ინტელექტი“, №1. 1996.
19. ჩოგოვაძე გ., სურგულაძე გ., შონია ო.: მონაცემთა და ცოდნის ბაზების აგების საფუძვლები. // გამომცემლობა „განათლება“. თბილისი, 1996 წ.
20. ფრანგიშვილი ა., სამხარაძე რ. გადაწყვეტილების მიღების ექსპერტული სისტემების დაპროექტების თეორია. // სტუ, თბილისი, 2004.
21. ყიფშიძე მ, ჯამარაშვილი ვ., არაბიძე გ. საქართველოს სათბობ-ენერგეტიკული კომპლექსის პრობლემები და განვითარების პერსპექტივები. სტუ, თბ., 2004.
22. Александровский А., Кнеллер М., Коробова Д. и др. // Гидроэнергетика. Москва, Энергоатомиздат. 1988.
23. Биргкоф Т.Барти . Современная прикладная Алгебра , // Мир, Москва, 1976.
24. Буч Г., Рамбо Дж., Джакобсон А. Язык UML. Руководство пользователя. //Пер.с англ., Питер, 2004.
- 25.Голдблат Р. Топосы Категорный анализ логики // Мир, Москва, 1983.
26. Гулуа Д., Петриашвили Л., Топурия Н., М. Оханашвили, Л. Квавадзе. Использование системных сетей Петри для задач объектно-ориентированного моделирования в АСУ // Сб. Тр. ГТУ, Тб., 2006.

27. Гулуа Д., Петриашвили Л., Оханашвили М. Кикнадзе М. Использование системных сетей петри для задач объектно-ориентированного моделирования в АСУ. // Сб.Тр.ГТУ, посв.35-летию образ.каф.АСУ. Тб., 2006.
28. Заложнев А. Модели и методы внутрифирменного управления – [urss.ru/cgi-bin/db.pl?lang=sp&blang=ru&page=Book&list=189&](http://urss.ru/cgi-bin/db.pl?lang=sp&blang=ru&page=Book&list=189&)
29. Исмаилова Л., Косиков С. Соотнесение семантик против семантики соотнесений – <http://conf/infosoc.ru/03-r1f08.html>.
30. Кипшидзе М., Арабидзе Г., Верулава Н. Анализ основных проблем, стоящих перед энергетикой Грузии, и пути преодоления кризиса . // Энергия №1-2, Тбилиси, 2001.
31. Кульба В., Волков А., Климов А., Швецов А. Анализ и синтез систем контроля и защиты данных с использованием сетей Петри. // Междунар. акад. бизнеса и банк. дела. Тольятти, 1998.
32. Леонтиев Б. MS Office Visio 2003: Не для Дилетантов. Построение проектов, диаграмм и бизнес-схем в ОС MS Windows XP. //Москва, ЗАО “Новый издат.дом”, 2005.
33. Мамиконов А., Кульба В. и др. Оптимизация структур распределенных баз данных в АСУ. // Москва, Наука, 1990.
34. Мамиконов А., Деметрович Я., Кульба В. и др. Использование сетей Петри при проектировании систем обработки данных. Оптимизация структур распределенных баз данных в АСУ. // Москва, Наука, 1988.
35. Мачарашвили Г. “Математическая модель производимой электроэнергии”. // Georgian Engineering News. №2, 2006
36. Модин А., Яковенко Е., Погребной Е., Справочник Разработчика АСУ. Москва «ЭКОНОМИКА, 1978
37. Организация и автоматизация документооборота – <http://www.termica.ru/dou/resh/avtomatiz.htmlid>
38. Пауэрс Ш. Чего мы ждем от XML? // Мир ПК , № 3, Москва, 1998.
39. [Печерский](http://www.citforum.ru/internet/XML) А.: Язык XML - практическое введение. // Центр Информационных Технологии – [www.citforum.ru/internet/XML](http://www.citforum.ru/internet/XML), 1999
40. Сургуладзе Г., Топурия Н., Петриашвили., Кикнадзе М. Автоматизированное проектирование классов и межклассовых отношений для распределенных, реляционных баз данных. // Georgian Engineering News. N3, 2006.
41. Г.Сургуладзе, Н. Топурия, И. Илдиз. Проектирование и реализация АСУ учебным процессом кафедры на базе UML-технологии // Сб. Тр. ГТУ, Тб., № 4(437), 2001.

42. Топурия Н. Автоматизированное проектирование концептуальной модели распределенных офисных систем на основе объектно-ролевого моделирования. // Georgian Engineering News. N3, 2006.
43. Цаленко М. Итоги науки и техники Информатика Том 9 // Мир, Москва, 1985.
44. Цаленко М. Моделирование Семантики в базах данных // Наука, Москва, 1989.
45. Чоговадзе Г., Качибая В., Сургуладзе Г. Теория реляционных зависимостей и проектирование логической схемы базы данных. // «Мецниереба», Тб., 1988.
46. Чоговадзе Г., Гогичаишвили Г., Сургуладзе Г., Шерозия Т., Шония О. Проектирование и построение АСУ. // ГТУ, Тбилиси., 2001.
47. Чоговадзе Г. Основы проектирования Автоматизированных Систем Управления Предприятиями // Тбилиси , Изд. Тб. Унив.,1972
48. Элиава Ш., Картвелишвили., Катамашвили., Гогишвили. О состоянии приема-передачи электроэнергии между энергосистемами России и Грузии. Заключение Контрольной палаты Грузии с участием специалистов отрасли. // Тбилиси, 2002.
49. Язык XML — практическое введение [www.citforum.ru](http://www.citforum.ru)
50. Aarts E. Investigations in Logic, Language and Computation // ILLC Dissertation Series 1995 - [www.illc.uva.nl/Publications/Dissertations](http://www.illc.uva.nl/Publications/Dissertations)
51. Barker R. CASE\*Method. Entity-Relationship Modelling. Copyright Oracle Corporation UK Limited // Addison-Wesley Publishing Co., 1990.
52. Barr M. and Wells C.. Category Theory for Computing Science.// Prentice-Hall, Englewood Cliffs, New Jersey, 1990.
53. Booch G., Jacobson I., Rumbaugh J. Unified Modeling Language for Object-Oriented Develop. Rational Software Corporation, Santa Clara, 1996.
54. Brackett J., C. McGowan. Applying SADT to Large System Problems // SofTech Technical Paper TP059, 1977.
55. Case Method- <http://www.interface.ru/fset.asp?Url=/case/sadt4.htm>
56. Document Object Model (DOM) Level 1 Specification <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/>.
- 57.. Elmasri R, Weeldreyer J., and Hevner A.. The category concept: An extension to the entityrelationship model. // Data & Knowledge Engineering, 1985.
58. Halpin T., Object Role Modeling: an overview, white paper, [www.orm.net](http://www.orm.net), 2001
59. Halpin T. and Orłowska M. Fact-oriented modelling for data analysis. // Journal of Information Systems, 2(2):97–119, April 1992.

60. Halpin, T.A., Information Modeling and relational Databases // Morgan Kaufmann Publishers, San Francisco, 2001  
[www.mkp.com/books\\_catalog/catalog.asp?ISBN=1-55860-672-6](http://www.mkp.com/books_catalog/catalog.asp?ISBN=1-55860-672-6).
61. Halpin T. and Proper H.,. Subtyping and Polymorphism in Object-RoleModelling // Data & Knowledge Engineering, 15(3), 1995.
62. Halpin A., Information Modeling and Relational Databases From Conceptual Analysis to Logical Design // Elsevier, 2006.
63. Haas E. LOGICS FOR OO INFORMATION SYSTEMS a semantic study of object orientation from a categorial-substructural perspective  
<http://www.illc.uva.nl/Publications/Dissertations/DS-2001-03.text.pdf> 1
64. Levis A., “System Architectures,” in Sage A. and Rouse W. (eds.) Handbook of Systems Engineering and management, Wiley, to appear.
65. Lippe E. and Hofstede A. A Category Theory Approach to Conceptual Data Modeling.// RAIRO Theoretical Informatics and Applications, 30(1):31–79, 1996.
66. Ovchinnikov V., Vahromeev Y., A Declarative Concept-Based Query Language as a mean for Relational Database Querying Journal of Conceptual Modeling – [www.inconcept.com/jcm](http://www.inconcept.com/jcm), 2005.
67. Ovchinnikov V., A Semantically Complete Conceptual Modeling Technique // Journal of Conceptual Modeling [www.inconcept.com/jcm](http://www.inconcept.com/jcm) , 2004 .
68. Paradigm Plus. Methods Manual. PLATINUM technology. 1997.
69. Rumbaugh. J., Blaha M., Premerlani W., Eddy F., and Lorenson W. Object-Oriented Modeling and Design.// Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
70. Surguladze G., Petriashvili L., Shonia O., Surguladze Gr.. Visual modeling and dynamic analysis of distributed business-processes on the basis of net-technologies and petri networks. // Geo.Scienc.Acad.-“Moambe”, No172-2, 2005.
71. Wedekind H. Objektorientierte Schemaentwicklung. Ein kategorialer Ansatz fuer Datenbanken und Programmierung. Wissenschaftsverlag, Mannheim/Wien/Zuerich. 1991
72. H. Wedekind. What is the right Object Model for a Database ? // **Transactions** Automated control systems. №1, 2006

## კათედრის მას-ის დაპროექტება და აგება UML ტექნოლოგიის გამოყენებით

როგორც ცნობილია, ობიექტ-ორიენტირებული ანალიზის ეტაპზე ხდება საპრობლემო გარემოს დეტალური ანალიზი, რომლის შედეგადაც ვლელობთ მასში მიმდინარე პროცესების შინაარსობრივ აღწერას, ობიექტებისა და მათ შორის კავშირების პირველად სქემას.

ჩვენი ამოცანისათვის ასეთ არაფორმალურ აღწერას ექნება შემდეგი სახე: „კათედრაზე არსებობენ სტუდენტები და ლექტორები, როგორც თანამშრომლები, ისე მოწვეულები, საშტატო და საათობრივი დატვირთვით. არსებობს საგნები, რომლებიც იკითხება კათედრაზე. თითოეული საგნისათვის გამოყოფილია ლექციების, პრაქტიკული და ლაბორატორიული სამუშაოების გარკვეული რაოდენობა. კათედრაზე ხდება დატვირთვის განაწილება. შტატში მყოფმა თანამშრომლებმა აუცილებლად უნდა ჩაატარონ საათების გარკვეული რაოდენობა. დარჩენილი საათები ნაწილდება საათობრივ დატვირთვაზე. კათედრაზე ხდება საათობრივ დატვირთვაზე ჩატარებული ლექციებისათვის თანხის გამოწერა. ამასთან გასათვალისწინებელია ლექტორის წოდება, მოწვეულია თუ შტატში მყოფია ლექტორი. კათედრის თითოეულ ლექტორსა და სტუდენტს აქვს მეცადინეობის ცხრილი, რომლის მიხედვითაც ტარდება მეცადინეობები. თითოეული ლექტორი კათედრაზე წარადგენს სასწავლო პროგრამას და ა.შ.“

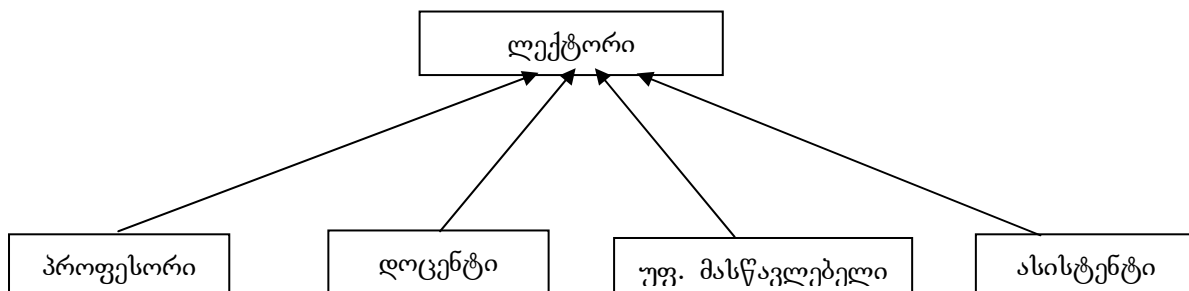
განსაკუთრებული ყურადღება ექცევა უმაღლეს სასწავლებლებში სტუდენტთა შეფასების ტესტირების დანერგვას. კათედრის დისციპლინებში შემუშავებულია საგამოცდო ტესტები და სტუდენტთა ტესტირების ჩატარების სისტემა. მიღებული შეფასება ფორმირდება სტუდენტთა ჯგუფების უწყისში. სესიების ჩაბარების შემდეგ სტუდენტი გადადის კურსიდან კურსზე. თეორიული კურსის დახურვის შემდეგ სტუდენტი იცავს დიპლომს. კათედრას ჰყავს მაგისტრანტები და მაძიებლები, ასევე სწავლობენ მოსამზადებელი ჯგუფები და ა.შ.“

სტუდენტთა სწავლებისა და მათი საგამოცდო ტესტირების პროცესები შეიძლება ფორმალურად წარმოვადგინოთ მიმდევრობით და პარალელურად შესასრულებელი პროცედურების სახით, რომლის მოდელირებისა და შემდგომი ანალიზის საუკეთესო ინსტრუმენტად ითვლება პეტრის ქსელები.

ზემოთ აღწერილი პროცესების და UML-ტექნოლოგიის საფუძველზე შეიძლება აიგოს მომხმარებელთა შემთხვევების Use Case დიაგრამა. ეს დიაგრამა გვიჩვენებს მართვის სფეროების პროცესებისა და შემსრულებლების ერთობლიობას, მათ შორის კავშირებს.

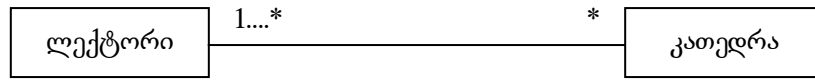
ჩვენი შემთხვევისათვის Use Case დიაგრამას აქვს 1.1 ნახაზზე ნაჩვენები სახე. გვაქვს სამი სახის კავშირის ასოციაცია, რომლებიც აუცილებელია ობიექტ-ორიენტირებული მოდელირებისათვის.

მემკვიდრეობითობა (Generalization). ერთი კლასი წარმოადგენს მეორე კლასის კერძო შემთხვევას. გვიჩვენებს, რომ კლასი „შვილი“ ობიექტები შეიძლება გამოყენებული იყოს ყველგან, სადაც გვხვდება კლასი „მშობლის“ ობიექტები, და არა პირიქით.



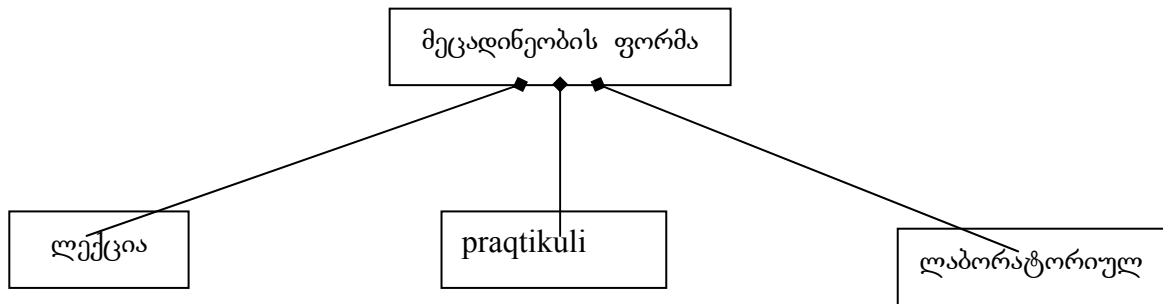
ნახ.1.1

ასოციაცია (Association). გვიჩვენებს, რომ ერთი ტიპის ობიექტები გარკვეული სახით დაკავშირებული არიან მეორე ტიპის ობიექტებთან.



ნახ.1.2

აგრეგაცია, ასოციაციის კერძო შემთხვევაა. გვიჩვენებს ასოციაციას „მთელი/ნაწილი“.



ნახ.1.3

საგარაუდო კლასების ჩამონათვალი ასეთია:

1. კლასი STUDENT ინახავს ინფორმაციას სტუდენტის შესახებ.  
 აქვს შემდეგი ატრიბუტები: სტუდენტის რიგითი ნომერი ST\_N, სტუდენტის სახელი, გვარი, მამის სახელი ST\_NAME, ასაკი AGE, სქესი SEX, ჯგუფის ნომერი GR\_N, საცხოვრებელი ადგილის მისამართი ADDRESS.
  2. კლასი LECTOR ინახავს ინფორმაციას ლექტორის შესახებ. აქვეა მითითებული კათედრის თანამშრომელია ლექტორი თუ მოწვეული.  
 აქვს შემდეგი ატრიბუტები: ლექტორის რიგითი ნომერი LEC\_N, ლექტორის სახელი, გვარი, მამის სახელი LEC\_NAME, სამეცნიერო ხარისხი STEP, ასაკი AGE, წოდება TITULI, იმ საგნის რიგითი ნომერი, რომელსაც ის კითხულობს SUB\_N, საცხოვრებელი ადგილის მისამართი ADDRESS.
  3. კლასი GROUP ინახავს ინფორმაციას ჯგუფის შესახებ.  
 აქვს შემდეგი ატრიბუტები: ჯგუფის ნომერი GR\_N, სპეციალობა SPEC, ჯგუფში მყოფი სტუდენტების რაოდენობა RAOD, მერამდენე კურსია COURS, ჯგუფხელის გვარი JGUPXELI.
  4. კლასი SUBJECT ინახავს ინფორმაციას საგნის შესახებ.  
 აქვს შემდეგი ატრიბუტები: საგნის რიგითი ნომერი SUB\_N, სახელწოდება SUB\_NAME, ლექტორის გვარი LEC\_NAME, ლექციების რაოდენობა LEC\_RAOD, პრაქტიკულების რაოდენობა PRAC\_RAOD, ლაბორატორიული სამუშაოების რაოდენობა LAB\_RAOD.
- ზემოთჩამოთვლილ კლასებში გვექნება შემდეგი მეთოდები:
- მონაცემთა დამატება ADD(), მონაცემთა ამოგდება REMOVE(), მონაცემთა ძებნა FIND(), მონაცემთა მოდიფიკაცია EDIT(), მონაცემთა ფილტრაცია FILTER(), მონაცემთა სორტირება SORT().
5. კლასი უწყისი UZKISI ინახავს ინფორმაციას უწყისის შესახებ.

გვაქვს შემდეგი ატრიბუტები: უწყისის ნომერი USK\_N, საგნის დასახელება SUB\_NAME, თარიღი DATE, ჯგუფის ნომერი GR\_N, ლექტორის ნომერი LEC\_N.

ამ კლასში გვაქვს შემდეგი მეთოდები: შეფასებების შეტანა GET\_USK(), ძებნა FIND().

6. კლასი საგამოცდო SAGAMOCDO ინახავს ინფორმაციას საგამოცდო ფურცლის შესახებ.

გვაქვს შემდეგი ატრიბუტები: საგამოცდო ფურცლის ნომერი SAG\_N, საგნის დასახელება SUB\_NAME, თარიღი DATE, ჯგუფის ნომერი GR\_N, ლექტორის გვარი LEC\_N, სტუდენტის გვარი STUD\_N.

გვაქვს შემდეგი ატრიბუტები: საგამოცდოს ფურცლის შევსება GET\_SUG(), ძებნა FIND().

7. კლასი ტესტი TEST ინახავს ინფორმაციას ტესტების შესახებ.

გვაქვს შემდეგი ატრიბუტები: ტესტის ნომერი TEST\_N, საგნის ნაწილი SUBNAC, შეკითხვების რაოდენობა TEST\_RAOD, შეკითხვა QUEST, სწორი პასუხი TRQUES, ი პასუხების ვარიანტები VAR1, VAR2, VAR3, VAR4.

ამ კლასში გვაქვს შემდეგი მეთოდები:

პასუხისა და ვარიანტის შედარება COMPAR(), სწორი პასუხების ჯამის დათვლა JAMI(), პროცენტის გამოთვლა PERS(), ქულა QULA().

8. კლასი დიალოგი-1 LECDIAL, რომელიც აღწერს ლექტორის დიალოგს ტესტების მენეჯერთან. ლექტორს შეჰყავს თავისი რიგითი ნომერი, გვარი და ჯგუფის ნომერი, რათა მიიღოს ტესტირების ჩაბარების უფლება.

გვაქვს შემდეგი ატრიბუტები: ლექტორის რიგითი ნომერი LEC\_N(), ლექტორის გვარი LEC\_NAME, ჯგუფის ნომერი GR\_N(), რომელებიც შეჰყავს ლექტორს. შემდეგ ხდება მათი კონტროლი, CONTROL().

9. კლასი დიალოგი-2 STDIAL, რომელიც აღწერს სტუდენტის მუშაობას ტესტებზე.

გვაქვს შემდეგი ატრიბუტები: საგნის სახელწოდება SUB\_NAME, ჯგუფის ნომერი GR\_N, სტუდენტის გვარი ST\_NAME, ტესტის ნომერი TEST\_N, სწორი პასუხი PASUXI.

გვაქვს შემდეგი მეთოდები: პასუხების შეტანა PASSHET(), კორექტირება EDIT().

კლასების ურთიერთქმედებათა დიაგრამის საშუალებით შეიძლება კვლევის ობიექტის სხადასხვა მხრიდან დანახვა. მაგალითისათვის მე-5 ნახაზზე მოყვანილია თანმიმდევრობათა დიაგრამა (Sequence diagram), რომელშიც აქცენტი გამახვილებულია შეტყობინებების დროის მიხედვით დალაგებაზე. აქ განიხილულია დიაგრამა პროცესისათვის „ტესტირება“.

1.4 ნახაზზე ნაჩვენებია კლასების დიაგრამა, რომელიც აგებულია Visual Modeler პროგრამულ პაკეტში.

დასასრულ, კლასების დიაგრამის საფუძველზე იქმნება სისტემის პროგრამული კოდი Visual C++ ენაზე.



ენბურჰისის პრექტის პროგრამული უზრუნველყოფა

პროგრამული უზრუნველყოფა შეიძლება განვიხილოთ როგორც პერსონალური კომპიუტერის სისტემური პროგრამული პაკეტებისა და უშუალოდ ენგურჰისის ავტომატიზებული ამოცანების სამომხმარებლო სპეციალური პროგრამული მოდულების ერთობლიობა.

**მანქანის სისტემური პროგრამული პაკეტი**

სისტემური პროგრამული პაკეტიდან უნდა გამოვეყოთ პირველ რიგში :

- Windows- XP/2000 ოპერაციული სისტემა;
- MS Office – მაიკროსოფტის საოფისო სისტემები,
- Excel - ელექტრონული ცხრილების პროგრამა;
- Equation – ფორმულების ასაგები პროგრამა;
- Internet Explorer – ინტერნეტის სამუშაო პროგრამა;
- ქართული შრიფტების პროგრამები;

კომპიუტერში აუცილებლად უნდა გვქონდეს სხვა სისტემური სერვისული პროგრამული პაკეტებიც, რომლებიც უზრუნველყოფს სისტემის საიმედო ფუნქციონირებას და დაცვას არასანქცირებული მიმართვისგან.

**MS EXCEL - პროგრამული პაკეტი**

Ms\_Excel პაკეტს აქვს მომხმარებლისთვის მოხერხებული ინტერფეისი დაპროგრამების ვიზუალური კომპონენტებით.

კომპიუტერული სისტემის აგება ხდება სპეციალური Wizard–პროგრამებით და მდიდარი ინსტრუმენტების პანელის დახმარებით, რომელზეც მრავლადაა წარმოდგენილი ტექსტური, გრაფიკული და მათემატიკური ფუნქციების პიქტოგრამები (სტანდარტული მაკროსები დოკუმენტების დასახაზად, ფორმულების დასაწერად, დიაგრამების ასაგებად, მონაცემთა ბაზაში ძებნის პროცედურის განსახორციელებლად და ა. შ.).

ტრადიციული პროგრამული კოდების გარეშე შეიძლება აქ დოკუმენტის, მონაცემთა ფაილების, ბაზების, გრაფიკების და თვით კომპლექსური სისტემების აგებაც. მაგ., ”ხელფასების დარიცხვის მართვის ავტომატიზებული სისტემა” შეიძლება მთლიანად იქნას აგებული Excel -პაკეტით.

როგორც ცნობილია, Excel-ის სამუშაო გარემოს ცხრილი (Table) წარმოადგენს, რომლის ძირითადი ელემენტია „უჯრა“ (Cell). უჯრა იდენტიფიცირდება სვეტისა და სტრიქონის კოორდინატებით, მაგ., C5 უჯრა შეესაბამება ელ-ენერჯის 1-ელი იანვრის გამომუშავებულ მნიშვნელობას.

Excel – ის ცხრილი მანქანაში ინახება xls ტიპის ფაილების სახით. ერთ ფაილში შეიძლება რამდენიმე ცხრილის შექმნა, რომლებიც განლაგებულ იქნება მის სამუშაო ფურცლებზე (Worksheets).

მაგალითად, მე-2 ფურცელზე (Janvari) განთავსებულია სამი ცხრილი: ძირითადი ჟურნალი, გარდამავალი ნაშთებისა და კალენდარული დღეების.

დანარჩენი ფურცლები შეესაბამება სხვა თვეების მონაცემებს. ეს ფურცლები დანომრილია 1,2,...16 ან შეიძლება მიენიჭოთ სემანტიკური სახელები, მაგალითად, 1Janvari, Tebervali და ა.შ.

ასევე ხელფასის ამოცანისათვის გვაქვს სამუშაო ფურცლები: Xelpasi (ხელფასები), Dynamika (დინამიკა) და ა.შ.

თითოეულ გვერდში შეიძლება საკმაოდ დიდი მოცულობის ცხრილებისა და შესაბამისი დიაგრამების აგება. ცხრილებსა და დიაგრამებს შორის არსებობს კავშირი. თუ შეიცვალა ცხრილში რაიმე მონაცემები, მაშინ იგი ავტომატურად აისახება დიაგრამის ცვლილებით და პირიქით.

სამუშაო გვერდებს შორის არსებობს კავშირები. მე-3 Sheet-ში (Tebervali) ნაშთების ცხრილში გვაქვს ორი მნიშვნელობა „წლის დასაწყისში“ ელ-ენერჯის ნაშთი (ანუ I6-უჯრაში B-ს მნიშვნელობა). იგი აიღება წინა თვის Sheet-იდან ფორმულით „ = Janvari ! D36 “.

ასევე ხდება კავშირის განხორციელება ჟრ-უჯრაში თებერვლისათვის N-ის ნაშთის მნიშვნელობის დასადგენად იანვრის თვიდან: „ = Janvari ! F36 “.

Excel პროგრამის გამოყენებისას, როგორც ცნობილია, ფორმულის ჩაწერა “უჯრაში” შესაძლებელია ხელით “=” სიმბოლოს გამოყენებით. მაგალითად, =SUM(C5 : C35).

აქ SUM( ) ჯამის ფუნქცია ახორციელებს (C5 : C35) უჯრებში ჩაწერილი ყოველდღიური ელ-ენერჯის შეკრებას და მოთავსებს C36 უჯრაში.

ამგვარი სტანდარტული მათემატიკური ფუნქციების გამოყენება დიდ დამსახურებად ითვლება Excel-ის სისტემაში. იგი აადვილებს არაპროფესიონალი-პროგრამისტი მომხმარებლებისათვის ფორმულების შეტანის პროცედურას.

Worksheet-1: Janvari

თვის მონაცემები	თვიანი მონაცემები	თვის მონაცემები	თვიანი მონაცემები	თვის მონაცემები
თვიანი მონაცემები	311000	311000	311000	5311000

Worksheet-2: Tebervali

თვის მონაცემები	თვიანი მონაცემები	თვიანი მონაცემები	თვიანი მონაცემები
10000	10000	321000	5321000

Formula bar: =Janvari!D36