

## ირინა ვაჭარაძე

ბიზნეს პროგრამების ექსპერტულ შეფასებებში მხარდამჭერი  
გადაწყვეტილებების მიღების მეთოდებისა და მოდელების  
შემუშავება

წარმოდგენილია დოქტორის აკადემიური ხარისხის  
მოსაპოვებლად

საქართველოს ტექნიკური უნივერსიტეტი  
თბილისი, 0175, საქართველო  
ივნისი, 2008

საავტორო უფლება © 2008 ”ვაჭარაძე ირინა”

## საქართველოს ტექნიკური უნივერსიტეტი

”ინფორმატიკისა და მართვის სისტემების ფაკულტეტი”

ჩვენ, ქვემოთ ხელისმომწერნი ვადასტურებთ, რომ გავეცანით ”ვაჭარაძე ირინას” მიერ შესრულებულ სადისერტაციო ნაშრომს დასახელებით: ”ბიზნეს პროგრამების ექსპერტულ შეფასებებში მხარდამჭერი გადაწყვეტილებების მიღების მეთოდებისა და მოდელების შემუშავება” და ვაძლევთ რეკომენდაციას საქართველოს ტექნიკური უნივერსიტეტის ”ინფორმატიკისა და მართვის სისტემების ფაკულტეტის” სადისერტაციო საბჭოში მის განხილვას დოქტორის აკადემიური ხარისხის მოსაპოვებლად.

17 ივნისი 2008 წ.

ხელმძღვანელი: პროფ., ტ.მ.დ. გია სურგულაძე

---

რეცენზენტი: პროფ., ტ.მ.დ. ავთანდილ კვიტაშვილი

---

რეცენზენტი: ტ.მ.კ. ვალერი ბურჯანაძე

---

რეცენზენტი:

## საქართველოს ტექნიკური უნივერსიტეტი

2008 წ.

ავტორი: "ვაჭარაძე ირინა"

დასახელება: ბიზნეს-პროგრამების ექსპერტულ შეფასებებში  
მხარდამჭერი გადაწყვეტილებების მიღების მეთოდებისა  
და მოდელების შემუშავება

ფაკულტეტი: "ინფორმატიკისა და მართვის სისტემების ფაკულტეტი"

ხარისხი: დოქტორი

სხდომა ჩატარდა:

ინდივიდუალური პიროვნებების ან ინსტიტუტების მიერ  
ზემომოყვანილი დასახელების დისერტაციის გაცნობის მიზნით მოთხოვნის  
შემთხვევაში მისი არაკომერციული მიზნებით კოპირებისა და გავრცელების  
უფლება მინიჭებული აქვს საქართველოს ტექნიკურ უნივერსიტეტს.

---

ავტორის ხელმოწერა

ავტორი ინარჩუნებს დანარჩენ საგამომცემლო უფლებებს და არც  
მთლიანი ნაშრომის და არც მისი ცალკეული კომპონენტების გადაბეჭდვა ან  
სხვა რაიმე მეთოდით რეპროდუქცია დაუშვებელია ავტორის წერილობითი  
ნებართვის გარეშე.

ავტორი ირწმუნება, რომ ნაშრომში გამოყენებული საავტორო  
უფლებებით დაცული მასალებზე მიღებულია შესაბამისი ნებართვა (გარდა  
ის მცირე ზომის ციტატებისა, რომლებიც მოითხოვს მხოლოდ სპეციფიურ  
მიმართებას ლიტერატურის ციტირებაში, როგორც ეს მიღებულია  
სამეცნიერო ნაშრომების შესრულებისას) და ყველა მათგანზე იღებს  
პასუხისმგებლობას.

”ვუძღვნი საქართველოს ტექნიკური უნივერსიტეტის  
ქსელებისა და სისტემების მიმართულების ხელმძღვანელის,  
ბატონი კონსტანტინე კამკამიძის 80 წლის იუბილეს.

კათედრას, რომლის ბაკალავრიატი და  
მაგისტრატურაც მე დავამთავრე და მის პედაგოგებს,  
რომლებმაც მნიშვნელოვანი და დაუვიწყარი როლი  
შეასრულეს ჩემს განათლებასა და ცხოვრებაში”.

## რეზიუმე

სადისერტაციო ნაშრომში, ”ბიზნეს-პროგრამების ექსპერტულ შეფასებებში მხარდამჭერი გადაწყვეტილებების მიღების მეთოდებისა და მოდელების შემუშავება”, განიხილება ორგანიზაციული მართვის სისტემებში კორპორაციული დაგეგმვის პროცესების სრულყოფის საკითხები მმართველობითი კონსულტირების ბიზნეს-პროცესების ავტომატიზაციის საფუძველზე. შემოთავაზებულია ამ პროცესებში ექსპერტულ შეფასებათა მხარდამჭერი გადაწყვეტილებების მიღების ობიექტ-ორიენტირებული მეთოდებისა და მოდელების შემუშავება, მათი დინამიკური პროცესების დაპროექტება უნიფიცირებული მოდელირების ენის საფუძველზე და პროგრამული რეალიზაცია ჯგუფური CASE-მეთოდებით, თანამედროვე ინფორმაციული ტექნოლოგიების, კერძოდ .NET-პლატფორმის ბაზაზე.

გადმოცემულია კომერციულ-საფინანსო ბანკებისათვის მონაცემთა საცავის აგებისა და ოპერატიული ინფორმაციის ანალიზის თანამედროვე სისტემების გამოყენების საკითხები. შემუშავებულია მონაცემთა მრავალგანზომილებიანი, რელაციული არსთა-დამოკიდებულების ვარსკვლავური სქემის დაპროექტების ვიზუალურ-ანალიზური მოდელი ობიექტ-როლური დიაგრამების ინსტრუმენტის საფუძველზე. საილუსტრაციო მაგალითისათვის განიხილება ბანკში კლიენტთა შემოსავლების მონიტორინგის ამოცანა.

განხილულია ბიზნეს-პროცესების ოპერატიული ანალიზის OLAP-ისტრუმენტის გამოყენების საკითხი. შემოთავაზებულია განაწილებული მონაცემთა ბაზებიდან შერჩეული მონაცემების ერთიან გადაწყვეტილების მიღების მხარდამჭერ სისტემაში სტრუქტურიზებული ორგანიზება და შესაბამისი პროგრამული პაკეტის რეალიზაცია ობიექტ-ორიენტირებული დაპროგრამების C++ ენის Decision Cube კომპონენტის გამოყენებით მრავალფაქტორული ანალიზის ამოცანებისთვის.

შემოთავაზებულია ბიზნეს-პროგრამების ექსპერტულ შეფასებათა პროცესების ობიექტ-ორიენტირებული მოდელირების, დაპროექტებისა და რეალიზაციის საკითხები, კატეგორიალური და ლოგიკურ-ალგებრული მეთოდების საფუძველზე ბიზნეს-პროცესების ობიექტ-როლური მოდელირების პროცედურების რეალიზაცია UML/ORM გარემოში და კლიენტ-სერვერ არქიტექტურის განაწილებული სისტემის მონაცემთა ბაზის სტრუქტურის (ERM) ავტომატიზებული დაპროექტების შედეგები Ms SQL Server პაკეტისათვის.

განიხილება ორგანიზაციული მართვის განაწილებული ბიზნეს-ობიექტების დოკუმენტბრუნვის პროცესების ავტომატიზაცია. ასეთი დიდი და რთული სისტემების ობიექტ-ორიენტირებული ანალიზი და ობიექტ-ორიენტირებული მოდელირების ენის (UML) სტანდარტებით და პეტრის ქსელების გრაფო-ანალიზური ინსტრუმენტით. შედეგად მიღებულია ალტერნატიული სქემები და მახასიათებლები საერთო რესურსების ეფექტურად გამოყენებისა და მოთხოვნების დამუშავების დროის შესამცირებლად.

კლიენტ-სერვერული არქიტექტურის ბიზნეს-პროცესების მოდელირებისა და იმიტაციური ანალიზისათვის აგებულია პეტრის ქსელის ეკვივალენტური გრაფები, გამოკვლეულია მათში მიმდინარე პროცესების დროითი მახასიათებლები და დადგენილია შედარებით ეფექტური, მისაღები მოდელების ერთობლიობა.

სისტემის ინფორმაციული უზრუნველყოფის ასაგებად ნაშრომში გადმოცემულია განაწილებული, მონაცემთა რელაციური ბაზების დაპროექტებისა და რეალიზაციის ამოცანის გადაწყვეტის საკითხები ახალი საინფორმაციო ტექნოლოგიების ბაზაზე, კერძოდ განიხილება კონცეფცია რევერსიული, ობიექტ-ორიენტირებული დაპროგრამების, ობიექტ-როლური მოდელის (ORM), არსთა-დამოვიდებულების მოდელის (ERM) და მონაცემთა აღწერის ენის (DDL) ურთიერთთავსებადობის შესახებ. შემოთავაზებულია რეალიზაციის შესაბამისი ალგორითმული სქემები MySQL Server და ADO.NET გარემოში სამუშაოდ.

ორგანიზაციული მართვის სისტემებში ექსპერტულ შეფასებათა გადაწყვეტილების მიღების მხარდამჭერი ვინდოუს-პროგრამების გარდა შემოთავაზებულია ვებ-აპლიკაციების დაპროექტების და რეალიზაციის საკითხები. კერძოდ საფინანსო ბანკის კლიენტთა მომსახურების მაგალითზე ილუსტრირებულია მათი ინტერფეისული კომპონენტების აწყობისა და მონაცემთა სერვერული ბაზების ორგანიზების ამოცანები. სისტემა დამუშავებულია .NET-პლატფორმაზე, C#, ASP.NET, ADO.NET და MySQL Server ობიექტ-ორიენტირებული ინსტრუმენტების გამოყენებით.

## **Abstract**

In presented thesis “Development of Methods and Models of Support of Decision-Making in Expert Estimations for Business Programs” the problems of perfection of corporate planning processes in organizational control systems, based on the automation of the management consulting business-processes, are considered.

Development of object-oriented methods and models of support of decision-making in expert estimations for these processes is offered as well, as designing of their dynamic processes, based on unified modeling language, and program realization by the group CASE-methods, based on the modern information technologies, particular, on NET-platform.

The problems of use of the modern systems of database construction and analysis of the operative information for commercial-financial banks are displayed. Visual-analysis model of designing of multi-dimensional, relational, meaning-dependent star scheme, based on the tools of objective-role diagrams, is developed. As the illustration example, the problem of monitoring of clients' income in the bank is considered.

The problem of use of OLAP-tools for the operative analysis of business-processes is considered. The structured organization of data, selected from the distributed databases, into the uniform system of support of decision-making is offered as well, as the corresponding program realization of an applied package of system on the basis of components of Decision Cube of object-oriented programming C++ language for a problem of multi-factorial analysis.

The problems of object-oriented modeling, designing and realization of processes of expert estimations of business-programs, realization of procedures of objective-role modeling of business-processes on the basis of categorial approach and logical-algebraic methods in UML/ORM environment are offered as well, as the results of automated designing of the distributed database system of the client-server architecture (ERM) for Ms SQL Server packet.

Automation of processes of documents circulation in distributed business-objects of the organizational control is considered. The object-oriented analysis and the object-oriented modeling of such large and complicated systems are carried out by the standards of unified modeling language (UML) and by the

graph-analytical tools of Petri networks. As a result, the alternative networks and characteristics are received for effective use of the common resources and for decrement in the time of requests elaboration.

For modeling, simulation and analysis of the client-server architecture of business-processes the equivalent graphs of Petri network are constructed, the time-dependent characteristics of current processes are investigated and the community of relatively effective, acceptable models is stated.

For creating of information supply of the system the questions of solution of the problems of designing and realization of distributed, relational bases, based on the modern information technologies, are displayed in the thesis. In particular, the conception concerning the inter compatibility of reversible, object-oriented programming, objective-role model (ORM), meaning-dependent model (ERM) and data description language (DDL) is considered. The corresponding realization, algorithmic schemes for work in MsSQL Server and ADO NET environments are offered.

In organizational control systems except of Windows programs of support of decision-making in expert estimations, the problems of designing and realization of web-applications are offered. In particular, the example of service of clients of the financial bank illustrates the problems of construction of their interface components and of organization of server databases as well. The system is elaborated on NET-platform, using the object-oriented tools of C#, ASP.NET, ADO.NET and MsSQL Server.

## შინაარსი

შესავალი . . . . .	15
1 თავი. ლიტერატურის მიმოხილვა: კორპორაციათა ბიზნეს-პროგრამების დაგენერირების პრობლემები და მათი სრულყოფის ამოცანები საინფორმაციო ტექნოლოგიების ბაზაზე . . . . .	18
1.1. კორპორაციული ობიექტების მართვის მექანიზმების სრულყოფის მიზნები და ამოცანები . . . . .	18
1.2. საკვლევი ობიექტის დიაგნოსტიკური ანალიზის ოთხეტაპიანი მოდელი . . . . .	22
1.3. მიზნის მისაღწევად შესასრულებელი ამოცანები . . . . .	26
1.4. გადაწყვეტილების მიღების ხელშემწყობი სისტემები . . . . .	29
1.5. მონაცემთა საცავი და ოპერატიული ანალიზის OLAP-ინსტრუმენტი . . . . .	30
1.6. საბანკო სისტემების კომუნიკაციები და საინფორმაციო ტექნოლოგიები . . . . .	38
1.7. მდგომარეობათა და პროცესების მოდელირება პეტრის ქსელებით . . . . .	44
1.8. დაპროგრამების UML ტექნოლოგია NET პლატფორმის ბაზაზე . . . . .	46
1.9. პირველი თავის დასკვნები . . . . .	49
2 თავი. შედეგები და მათი განსჯა: ბიზნეს-პროგრამების მართვის ხელშემწყობი კომპიუტერული სისტემის აგების მეთოდებისა და მოდელების შემუშავება . . . . .	50
2.1. ბიზნეს-პროგრამების ექსპერტულ შეფასებათა გადაწყვეტილების მიღების პროცესების მოდელირება UML-ის სტატიკური და დინამიკური დიაგრამებით . . . . .	50
2.2. მონაცემთა ბაზაში ექსპერტთა ცოდნის ასახვის ობიექტ-როლური მოდელების შემუშავება კატეგორიალური მიდგომით . . . . .	61
2.3. ობიექტ-როლური მოდელების დაპროექტება და აგება ავტომატიზებულ რეჟიმში . . . . .	70
2.4 კონცეპტუალური მოდელებისა და კლასთა ურთიერთკავშირების სქემის დაპროექტება ობიექტ-როლური მოდელების ბაზაზე . . . . .	74
2.5. განაწილებული ბიზნეს-პროცესების ობიექტ-ორიენტირებული მოდელირება და ანალიზი პეტრის ქსელებით . . . . .	77
2.6. მეორე თავის დასკვნები . . . . .	86
3 თავი. შედეგების პრაქტიკული რეალიზაცია: ინფორმაციულ-პროგრამული უზრუნველყოფის დამუშავება . . . . .	87
3.1. განაწილებული მონაცემთა ბაზების აგების ავტომატიზაცია NET გარემოში . . . . .	87
3.2. მონაცემთა საცავის დაპროექტება გადაწყვეტილებათა მიღების მხარდამჭერი სისტემისათვის . . . . .	93
3.3. გადაწყვეტილების მიღების მხარდამჭერ სისტემაში მრავალფაქტორული ამოცანის გადაწყვეტის პროგრამული რეალიზაცია . . . . .	97
3.4. ინფორმაციული ნაკადების დიაგრამები ავტომატიზებული საბანკო სისტემისათვის . . . . .	103
3.5. კორპორაციული მართვის სისტემის Web-აპლიკაციის დამუშავება Internet-Intranet გარემოში .NET-პლატფორმაზე . . . . .	108
3.6. ავტომატიზებული ანალიზის სისტემის დაპროექტება სატენდერო კომისიის ექსპერტებისათვის. . . . .	114
3.7. მესამე თავის დასკვნები . . . . .	120
4. დასკვნა.....	122
გამოყენებული ლიტერატურა . . . . .	124
დანართი . . . . .	128

## ცხრილების ნუსხა

რ/ნ	ცხრილის ნომერი. დასახელება	გვ.
1	ცხრ.2.1. ფაქტების ჩაწერის ფორმა კატეგორიალური მიღღომით	72
2	ცხრ.2.2. DDL კოდის ფრაგმენტი	77
3	ცხრ.2.3. პეტრის ქსელის პოზიციების ცხრილი	81
4	ცხრ.2.4. პეტრის ქსელის გადასასვლელების ცხრილი	81
5	ცხრ.3.1. რეალიზაცია ქალაქების და თვეების მიხედვით	100
6	ცხრ.3.2. პროდუქციის რეალიზაცია ქალაქებისა და თვეების მიხედვით	26

## ნახაზების ნუსხა

რ/ნ	ნახაზის ნომერი. დასახელება	გვ.
1	ნახ.1.1. UML -ტექნოლოგიის ოთხი ეტაპი	20
2	ნახ.1.2. სისტემის არქიტექტურის მოდელირება	20
3	ნახ.1.3. სტრატეგიული დაგეგმვის ოთხეტაპიანი სქემა	22
4	ნახ.1.4. პირველი სტადიის ეტაპები	23
5	ნახ.1.5. მეორე სტადიის ეტაპები	24
6	ნახ. 1.6. მესამე სტადიის ეტაპები	25
7	ნახ.1.7. მეოთხე სტადიის ეტაპები	26
8	ნახ.1.8. ლოკალური (1) და გლობალური (2) მიზნები	28
9	ნახ.1.9. მონაცემთა საცავის ზოგადი სქემა	31
10	ნახ.1.10. საბანკო ქსელის მართვის სიტემის არქიტექტურა	43
11	ნახ.1.11. S/T - პეტრის ქსელის ფრაგმენტი	45
12	ნახ.1.12. UML-ტექნოლოგიის ეტაპები	47
13	ნახ.1.13. Windows და .NET-პლატფორმათა ურთიერთოქმედება	48
14	ნახ.2.1. ექსპერტულ შეფასებათა დერძი	55
15	ნახ.2.2. ექსპერტულ შეფასებათა პროცესის Use Case დიაგრამა	56
16	ნახ.2.3. „პრობლემის ანალიზის“ დეტალიზაციის დონე	57
17	ნახ.2.4. ექსპერტულ შეფასებათა მხარდამჭერი სისტემის აგების ტექნიკური პროცესის აქტიურობის დიაგრამა	59
18	ნახ.2.5. მიმდევრობითობის დიაგრამის ფრაგმენტი	61
19	ნახ.2.6. თანამოქმედების დიაგრამის ფრაგმენტი	62
20	ნახ.2.7. ობიექტ-როლური მოდელის (ORM) ფრაგმენტი	70
21	ნახ.2.8. ფაქტების ჩაწერის ინსტრუმენტი	73
22	ნახ.2.9. ORM მოდელის ფრაგმენტი ექსპერტულ შეფასებათა სისტემისთვის	74
23	ნახ.2.10. ERM მოდელის ფრაგმენტი, აგებული ავტომატიზებულ რეჟიმი ORM მოდელის საფუძველზე	76
24	ნახ.2.11. კოდის გენერირების ვიზარდ-პროგრამა	77
25	ნახ.2.12. საქმიანი პროცესების ზოგადი ანალიზური მოდელი	79
26	ნახ.2.13. განაწილებულ სისტემაში მოთხოვნების დამუშავების ზოგადი მოდელი	80
27	ნახ.2.14. დეტალიზებული პეტრის ქსელის ფრაგმენტი	81
28	ნახ.2.15. პეტრის ქსელის სიმულატორით მიღებული ეტაპობრივი შედეგები	83
29	ნახ.2.16. იმიტაციის პროცესის ფრაგმენტი პეტრის ქსელით	84
30	ნახ.2.17. P <sub>4</sub> და P <sub>19</sub> პოზიციების მდგომარეობათა ცვლილებები დროში	85

რ/ნ	ნახაზის ნომერი. დასახელება	გვ.
31	ნახ.2.18. რესურსების ტრიგერული მართვის ფრაგმენტი	85
32	ნახ.2.19. P <sub>4</sub> და P <sub>19</sub> პოზიციების მდგომარეობათა ცვლილებები დროში ტრიგერული მართვისას	86
33	ნახ.2.20. P <sub>4</sub> და P <sub>19</sub> პოზიციების მდგომარეობათა ცვლილებები დროში ტრიგერული მართვისას გაორკეცებული ადამიანური რესურსით	86
34	ნახ.2.21. P <sub>4</sub> და P <sub>19</sub> პოზიციების მდგომარეობათა ცვლილებები დროში ადამიანური რესურსის გაორკეცებული სწრაფქმედებით	87
35	ნახ.3.1. კლასთა-ასოციაციის დიაგრამა	92
36	ნახ.3.2. ტოპოლოგია: ფიფქი და ვარსკვლავი	92
37	ნახ.3.3. ER-დიაგრამის ფრაგმენტი	93
38	ნახ.3.4. მბ-ის დაპროექტების პროცესის რევერსიული მექანიზმის კომპონენტები	96
39	ნახ.3.5. მგბ-ის დაპროექტების პროცესის ავტომატიზაციის ზოგადი სქემა	97
40	ნახ.3.6. DDL ფაილების ავტომატიზებული კორექცია	98
41	ნახ.3.7. მონაცემები სამგანზომილებიან კუბში	100
42	ნახ.3.8. ჯამური რეალიზაციის თანხები	102
43	ნახ.3.9. თანხები ქალაქების მიხედვით	102
44	ნახ.3.10. პროდუქციის რეალიზაცია თვეების მიხედვით	102
45	ნახ.3.11. პროდუქციის რეალიზაცია ქალაქებისა და თვეების მიხედვით	103
46	ნახ.3.12. ცხრილი სამი ფაქტორის გათვალისწინებით	104
47	ნახ.3.13. პროცესთა ნაკადების დიაგრამის ფრაგმენტი	105
48	ნახ.3.14. მონაცემთა ნაკადების დიაგრამის ფრაგმენტი	106
49	ნახ.3.15. აქტიური ობიექტების დიაგრამის ფრაგმენტი	106
50	ნახ.3.16. მონაცემთა საცავის დიაგრამის ფრაგმენტები	107
51	ნახ.3.17. მართვის ნაკადის დიაგრამის ფრაგმენტი	108
52	ნახ.3.18. ოპერაციის აღწერის მაგალითი	103
53	ნახ.3.19. ინტერფეისის მთავარი გვერდი	113
54	ნახ.3.20. გვერდი: „ინფორმაცია ანგარიშზე“	113
55	ნახ.3.21. გვერდი: „თანხის შეტანა“	114
56	ნახ.3.22. თანხის შეტანის საბოლოო შედეგი	115
57	ნახ.3.23. სატენდერო პროცესის მომზადების აქტიურობის დიაგრამა	117
58	ნახ.3.24. სატენდერო პროცესის ORM-დიაგრამა	119
59	ნახ.3.25. „ტენდერის“ საპრობლემო სფეროს ER-მოდელი	120
60	ნახ.3.26. ტენდერის მონაცემთა ბაზის ფრაგმენტის ცხრილების სტრუქტურა	121

## დისერტაციაში გამოყენებული აბრევიატურები

ADO	- Active Data Objects
AD4E	- Activity Diagram for Expert
API	- Application Programming Interface
ASP	- Active Server Pages
ATM	- Asynchronous Transfer Mode
CLR	- Common Language Runtime
CLS	- Common Language Specification
CTS	- Common Type System
DBS	- Data Base System
DDBS	- Distributed Data Base System
DDL	- Data Definition Language
DML	- Data Manipulation Language
DSS	- Decision Support Systems
DWH	- Data Warehouse
ERM	- Entity-Relationship Model
IL	- Intermediate Language
JIT	- Just-In-Time
OLAP	- Online Analytical Processing
OMG	- Object Management Group
ORM	- Object Role Model
RDBS	- Relational Data Base System
S/T PN	- State/Transition Petri Nets
SPN	- System Petri Nets
SQL	- Structured Query Language
UML	- Unified Modeling Language
XML	- eXtended Markup Language
გმბს	- გადაწყვეტილების მიღების ხელშემწყობი სისტემები

## მადლიერება

მინდა მადლიერება გამოვხატო ყველა იმ პირის მიმართ, ვინც მეხმარებოდა სადისერტაციო ნაშრომის შესრულებისას. პირველ რიგში, მინდა მადლიერება გამოვხატო ჩემი ხელმძღვანელის მიმართ, რომლის ამაგი წინამდებარე სადისერტაციო ნაშრომში და ჩემი სპეციალისტად ჩამოყალიბებაში ფასდაუდებელია. ბატონი გია სურგულაძის რჩევებით და დახმარებით, მე დავეუფლე თანამედროვე ინფორმაციულ ტექნოლოგიებს და შევისწავლე მათი გამოყენება კორპორაციული მართვის ამოცანების გადასაწყვეტად. მისი ხელშეწყობითა და მოთხოვნით ყოველთვის ვიღებდი აქტიურ მონაწილეობას ჩვენი ფაკულტეტის სამეცნიერო სემინარებსა და კონფერენციებში.

ასევე მადლიერების დიდი გრძნობა მინდა გამოვხატო პროფ. არჩილ ფრანგიშვილის მიმართ, რომელიც იყო ჩემი ხელმძღვანელი მაგისტრატურაში სწავლის დროს. სწორედ მისი უშუალო დახმარებით და ბატონი ზურაბ გასიტაშვილის მხარდაჭერით ჩამოყალიბდა ჩემი სადისერტაციო კვლევის მიმართულება და თემის დასახელება.

დიდი მადლობა მინდა გადავუხადო ქალბატონ ნინო თოფურიას, რომლის უშუალო პრაქტიკული კონსულტაციებით მოვახერხე სადისერტაციო ნაშრომის ექსპერიმენტული ნაწილის ამოცანების გადაწყვეტა.

და ბოლოს, მინდა უდიდესი მადლობა გადავუხადო ჩემს მამას, ვალერი ვაჭარაძეს, იმ სულიერი, ორგანიზაციული და მატერიალური მხარდაჭერისათვის, რაც მან გამიწია სადისერტაციო სამუშაოს შესრულების პროცესში.

## შესავალი

თანამედროვე კორპორაციულ სისტემებს, რომელთა რიცხვს  
მიეკუთვნება დარგობრივი დაწესებულებები, საწარმოები, სავაჭრო  
ფირმები, კომერციული ბანკები და ა.შ., ზოგადად – ბიზნეს-ობიექტები,  
აქვს რთული ორგანიზაციული სტრუქტურა, რომელიც ტოპოლოგიურად  
განაწილებული ფილიალების სახით შეიძლება წარმოვიდგინოთ. მათი  
ეფექტური მართვა შესაძლებელია ყველა დეპარტამენტის, ფილიალის  
და განყოფილების მდგომარეობისა და მოქმედების შესახებ სრული  
ოპერატორიული ინფორმაციის არსებობისას. ამ ფაქტორმა გამოიწვია  
ორგანიზაციული სისტემების, და კერძოდ კომერციული ბანკების  
ავტომატიზაციის აუცილებლობა და თანამედროვე ინფორმაციული  
ტექნოლოგიების, ტელეკომუნიკაციების საშუალებათა, მონაცემთა  
ბაზების მართვის სისტემების ფართო გამოყენება.

ბიზნეს-ობიექტებზე განსაკუთრებული ყურადღება ექცევა  
მარკეტინგისა და მენეჯმენტის პროცესების სწორად და დროულად  
წარმართვას. ახალი ბიზნეს-პროგრამები, რომელთა განხორციელება,  
ერთის მხრივ სასიცოცხლოა თანამედროვე კონკურენციის პირობებში,  
ხოლო მეორეს მხრივ, ძნელადრეალიზებადია საკმაო მოცულობის  
ინვესტიციების მოზიდვის თვალსაზრისით, მოითხოვს ფირმის  
ხელმძღვანელებისა და ექსპერტ-სპეციალისტებისგან სწორი  
ეკონომიკური პოლიტიკის გატარებას გააზრებული ჯგუფური  
გადაწყვეტილების მიღების საფუძველზე [1].

წინამდებარე ნაშრომის მიზანია ორგანიზაციული მართვის  
სისტემის სრულყოფა, მისი ბიზნეს-პროგრამების შეფასების  
მხარდაჭერი გადაწყვეტილებათა მიღების ახალი ტექნოლოგიების  
შემუშავებისა და მათი უნიფიცირებული პროგრამული რეალიზაციის  
საფუძველზე, რაც უდავოდ აქტუალური და მნიშვნელოვანი  
მიმართულებაა [2,3].

აღნიშნული მიზნის მისაღწევად დისერტაციაში განიხილება  
შემდეგი საკითხები:

- გაანალიზებულია ორგანიზაციული მართვის სისტემებში კორპორაციული დაგეგმვის პრობლემები და გამოვლენილია მათი გადაწყვეტის ამოცანები და გზები;
- შემუშავებულია ბიზნეს-პროგრამების ექსპერტულ შეფასებათა გადაწყვეტილების მიღების მხარდამჭერი კომპიუტერული სისტემის სტრუქტურა და შედგენილობა, ობიექტ-ორიენტირებული მეთოდები და მოდელები;
- განსაზღვრულია განაწილებული კორპორაციული სისტემის მონაცემთა რელაციური საცავის აგებისა და მისი ოპერატიული ანალიზის ეფექტური ალგორითმული სქემები მრავალფაქტორული ამოცანებისათვის, ვიზუალურ-კომპონენტური დაპროგრამების მეთოდის საფუძველზე;
- დამუშავებულია ბიზნეს-პროგრამების ექსპერტულ შეფასებათა პროცესების ობიექტ-ორიენტირებული მოდელირების, დაპროექტებისა და რეალიზაციის საკითხები კატეგორიალური და ლოგიკურ-ალგებრული მეთოდების გამოყენებით;
- შემოთავაზებულია კლიენტ-სერვერული არქიტექტურის განაწილებული ბაზების დაპროექტებისა და აგების პროცესების ავტომატიზაცია ობიექტ-როლური და კონცეპტუალური მოდელების საფუძველზე;
- აგებულია ბიზნეს-პროცესების მართვის იმიტაციური მოდელირებისა და ანალიზის ალტერნატიული სქემები სისტემური პეტრის ქსელების გრაფების გამოყენებით;
- დამუშავებულია Windows- და Web- აპლიკაციების დანართები, მომხმარებელთა სერვისული ინტერფეისები კომურციული ბანკების მაგალითზე, .NET-პლატფორმის, C#, ASP.NET, ADO.NET და Ms SQL Server პაკეტების გამოყენებით.

ნაშრომის პირველ თავში გადმოცემულია ორგანიზაციული მართვის ობიექტების კორპორაციული დაგეგმვის არსებული მექანიზმების ანალიზი, გამოვლენილია პრობლემები და ამოცანები თანამედროვე საბაზოო კონკურენციისა და მმართველობითი კონსულტირების ეფექტური მეთოდების გამოყენების თვალსაზრისით. ყურადღება გამახვილებულია პროდუქციის საწარმოო ფირმებისა და

კომერციული ბანკების ბიზნეს პროგრამების მართვის საკითხებზე. გადმოცემულია დღეისათვის არსებული მეთოდებისა და ინსტრუმენტების, აგრეთვე ინფორმაციული ტექნოლოგიების მოკლე, კრიტიკული ანალიზი. დასმულია ამოცანა ბიზნეს-პროგრამების ექსპერტულ შეფასებათა ჯგუფური გადაწყვეტილების მხარდამჭერი კომპიუტერული სისტემის ასაგებად.

მეორე თავში აღწერილია ბიზნეს-პროგრამების მართვის და ჯგუფური შეფასების ტექნოლოგიური პროცესების უნიფიცირებული Case-მეთოდები და ობიექტ-ორიენტირებული მოდელები; სისტემის მონაცემთა ბაზაში ექსპერტთა ცოდნის ასახვის ობიექტ-როლური მოდელები კატეგორიალური მიდგომით, აგრეთვე ამ პროცესების ავტომატიზებულ რეჟიმში გადაწყვეტის პროცედურები ვიზუალურ-კომპონენტური დაპროგრამების საფუძველზე; წარმოდგენილია განაწილებული ბიზნეს-პროცესების ობიექტ-ორიენტირებული მოდელირებისა და იმიტაციური ანალიზის ალტერნატიული სქემები სისტემური პეტრის ქსელების გამოყენებით.

მესამე თავი ეხება ბიზნეს-პროგრამების ექსპერტულ შეფასებათა მხარდამჭერი კომპიუტერული სისტემის ინფორმაციული და პროგრამული უზრუნველყოფების დამუშავებას და რეალიზაციას. შემოთავაზებულია კორპორაციების მონაცემთა საცავების დაპროექტების, აგებისა და მათი ოპერატიული მრავალფაქტორული ანალიზის ამოცანების გადაწყვეტა. გამოყენებულია კლიენტ-სერვერული კონფიგურაცია ექსპერტთა შეფასებების ავტომატიზებული დამუშავებისათვის, Borland C++Builder, MsSQL Server, Ms Office პაკეტებისა და .NET პლატფორმის (C#.NET, ADO.NET, ASP.NET) ბაზაზე.

წარმოდგენილია ექსპერიმენტული ნაწილის შედეგები, რომლებიც განხორციელდა პროდუქციის საწარმოო ფირმის, კომერციულ-საფინანსო ბანკის, განათლების და სხვ. სფეროს ობიექტებზე. შემოთავაზებულია მომხმარებელთა ინტერფეისების სადგმონსტრაციო პაკეტი Web-აპლიკაციის სახით, კომერციული ბანკის სერვისის ამოცანების გადასაწყვეტად.

სადისერტაციო ნაშრომის თემატიკაზე გამოქვეყნებული მაქვს 9 სამეცნიერო სტატია.

## 1 თავი. ლიტერატურის მიმოხილვა:

პორაორაციათა ბიზნეს-კომპრამების დაბეგმვის პროცესები და  
მათი სრულყოფის ამოცანები საინიციაციო ტექნოლოგიების  
გაზაზე

### 1.1. კორპორაციული ობიექტების მართვის მექანიზმების სრულყოფის მიზნები და ამოცანები

კორპორაციულ ობიექტებს მიეკუთვნება სახელმწიფო ან კურძო  
სტრუქტურების საწარმოო, ორგანიზაციული, საბანკო-საფინანსო და  
სხვა ტიპის იურიდიული სუბიექტები. მათი ფუნქციონირების  
კეთილდღეობა, კონკურენციის მძაფრ პირობებში, ბევრადაა  
დამოკიდებული მართვის აპარატის მოქნილობასა და საიმედოობაზე.  
რთული ბიზნეს-პროცესების კორპორაციული დაგემვისა და  
ოპერატიული მართვის მექანიზმების ეფექტური ორგანიზებითა და  
მუდმივი სრულყოფით შესაძლებელი ხდება ამ ობიექტების  
სასიცოცხლო ციკლის გახანგრძლივება.

კორპორაციული ობიექტების მართვის ავტომატიზებული სისტემა  
მოიცავს დიაგნოსტიკური ანალიზის, ექსპერტული შეფასებების, ბიზნეს-  
კომპრამების დაგეგმვის, მათი განხორციელების ორგანიზების, ფაქტ-  
შედეგების აღრიცხვის, ეკონომიკური ანალიზისა და შეფასების,  
ობიექტზე ეფექტური ზემოქმედების მმართველი გადაწყვეტილების  
მიღების პროცესების ხელშემწყობი მექანიზმების შემუშავებას და მათ  
კომპიუტერულ რეალიზაციას.

ასეთი ობიექტების მართვის მექანიზმების მოდელი საკმაოდ  
რთულია და მიეკუთვნება ძნელად ფორმალიზებად დიდი სისტემების  
კლასს. მათი აგებისა და ეფექტური გამოყენებისათვის მიზანშეწონილია  
არაერთგვაროვანი, რაოდენობრივი და ხარისხობრივი მეთოდების  
კომპლექსური გამოყენება. კერძოდ, ერთის მხრივ, კოგნიტური  
მოდელების ასაგებად ექსპერტულ შეფასებათა სხვადასხვა მეთოდების  
ინტეგრირებული გამოყენება, და მეორეს მხრივ, ბიზნეს-პროგრამების  
დაგეგმვის პროცესების ოპტიმიზაცია და კვლევა მასობრივი  
მომსახურების სისტემების თეორიის საფუძველზე, მარკოვის  
პროცესებით.

ექსპერტულ შეფასებათა დამუშავების ავტომატიზებული სისტემის ძირითადი მიზანია საწარმოო ფირმებისა ან ორგანიზაციების სტრატეგიული განვითარების გეგმების (კორპორაციული დაგეგმვა) შედგენის პროცესის ავტომატიზაცია.

თვით ასეთი კორპორაციული გეგმების ოპტიმალური ან „უკეთესი“ ვარიანტების შერჩევას აქვს გლობალური მიზანი – ესაა ფირმის მუშაობის ეფექტურობის ამაღლება (სწორი დაგეგმვის გზით). ამ საკითხებს ყოველთვის განსაკუთრებული ყურადღება ექცეოდა და მქონება საზღვარგარეთ [1].

ჩვენშიც, საბაზრო ეკონომიკაზე გადასვლის შემდეგ, როცა მრავალი საწარმოო ფირმა და ორგანიზაცია პრივატიზებულია და კერძო საკუთრების ფორმით მოგვევლინა, აქტუალური გახდა აღნიშნული მიმართულება.

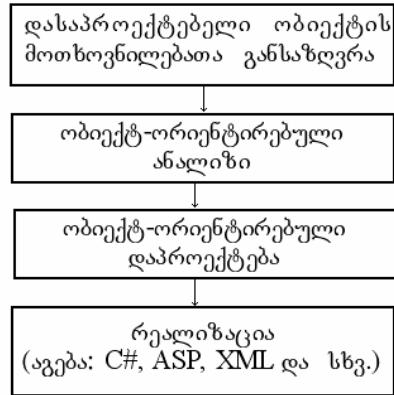
ამგვარად, დისერტაციაში დასმული ამოცანების გადაწყვეტის მიზანი ფირმის სასიცოცხლო ციკლთან – არსებობის საკითხთანაა დაკავშირებული და ამგვარად მისი აქტუალურობა და მნიშვნელობა ეჭვს არ იწვევს.

კორპორაციული ობიექტების მართვის პროცესისთვის ჩვენ უნდა გამოიყენოთ ექსპერტულ შეფასებათა მეთოდებით მოპოვებული ინფორმაცია და გადავამუშავოთ ისინი კომპიუტერის გამოყენებით. ეს გადამუშავება კი გულისხმობს სპეციალური პროგრამული პაკეტის ("Expert\_UML") დამუშავებას, ანუ უნდა შეიქმნას კომპიუტერული დიალოგური სისტემა - ახალი, ობიექტ-ორიენტირებული მოდელირებისა და UML-ტექნოლოგიის საფუძველზე [3].

დღეისათვის ერთ-ერთი აქტუალური და პერსპექტიული მიმართულებაა სამეცნიერო-ეკონომიკური ობიექტების მართვის კომპიუტერული სისტემების დაპროექტება და რეალიზაცია უნიფიცირებული მოდელირების ენის (UML/2 – Unified Modeling Language) საშუალებით [4].

ეს UML-ენა არის პროგრამული უზრუნველყოფის დამუშავების თანამედროვე ტექნოლოგია, რომელიც განაპირობებს პროგრამული პაკეტების მაღალ დონეზე სტანდარტიზაციას.

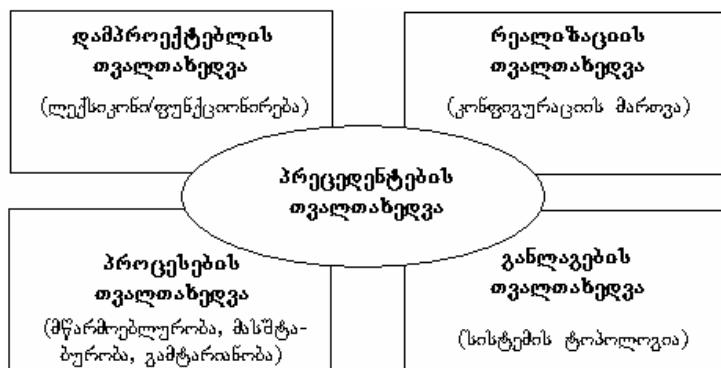
UML ტექნოლოგია ეყრდნობა ობიექტ-ორიენტირებული მოდელირებას და რამდენიმე ეტაპისაგან შედგება:



### ნახ. 1.1 UML -ტექნოლოგიის ოთხი ეტაპი

პირველ ეტაპზე ხორციელდება საკვლევი ობიექტის შესწავლა და მისი ფუნქციონირების მოთხოვნილების დადგენა.

სისტემის არქიტექტურა მნიშვნელოვანი კომპონენტია, რომელიც გამოიყენება მომხმარებელთა განსხვავებული თვალსაზრისების მიზანმიმართული მართვისათვის, მაგალითად, პროგრამული სისტემის ორგანიზება; სტრუქტურული ელემენტების არჩევა, რომელთაგანაც შედგება სისტემა და ინტერფეისები; ამ ელემენტების ყოფაქცვა სხვებთან მიმართებაში და ა.შ. პროგრამული სისტემის არქიტექტურა შეიძლება კარგად აისახოს ხუთი ურთიერთდაკავშირებული ელემენტით (ნახ.1.2). თითოეული თვალთახედვა დამოუკიდებულია, მაგრამ სისტემის აგების დროს UML ტექნოლოგიით ხდება მათი ურთიერთმოქმედება.



### ნახ. 1.2. სისტემის არქიტექტურის მოდელირება

ჩვენ საკვლევ ობიექტად განვიხილავთ პროდუქციის მწარმოებელ ფირმას (ან კომერციულ ბანკს), მის ორგანიზაციულ-საწარმოო სტრუქტურას. საწარმოო ფირმა ამზადებს განსხვავებული

ასორტიმენტის პროდუქციას (კომერციული ბანკი ემსახურება იურიდიულ და ფიზიკურ პირებს სხვადასხვა სერვისული ამოცანებით). მას აქვს საფირმო მაღაზიები. ჰყავს როგორც პარტნიორები: სავაჭრო ბაზები, მუდმივი დამკვეთები, ასევე კონკურენტები ბაზარზე. ფირმას უხდება „ბრძოლა“ თავისი არსებობის შესანარჩუნებლად. მისი მდგომარეობა შეიძლება იყოს კარგი, დამაკმაყოფილებელი ან ცუდი, ანუ გაკოტრების ფირმის მდგომარეობის გასაანალიზებლად საჭიროა ექსპერტების შეფასებების შეგროვება და მათი კომპლექსური გადამუშავება. ამ პროცესის საკმაო სირთულისა და ინფორმაციის დიდი მოცულობის გამო, სასურველია გადამუშავდეს იგი კომპიუტერის დახმარებით.

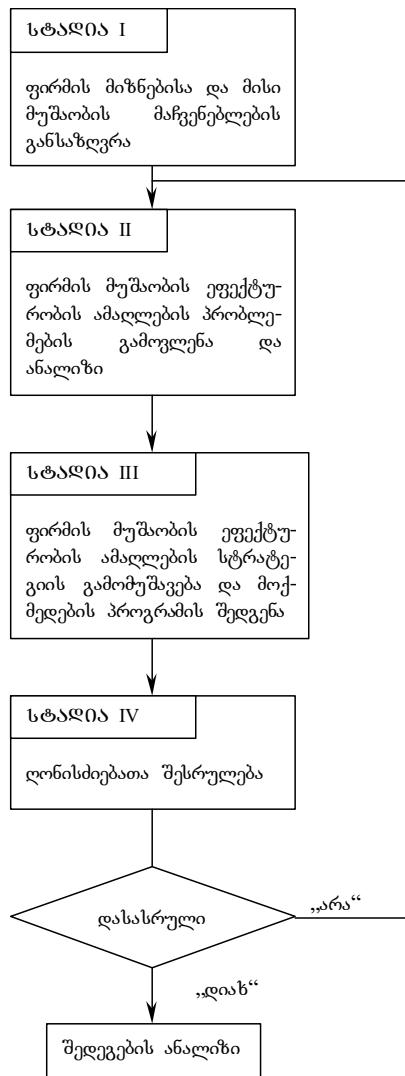
ჩვენი სისტემის აქტიური მომხმარებლებია ექსპერტები (საწარმოო ფირმის ან ბანკის ხელმძღვანელები და მთავარი სპეციალისტები) და კონსულტანტი, რომელიც მეთოდურად ხელმძღვანელობს ექსპერტებს და წარმართავს როგორც მათი განსწავლის პროცესს, ასევე ჯგუფური ინფორმაციის შეგროვებას, შეფასებას და გადამუშავებას.

აქედან გამომდინარე, სისტემის ძირითადი მოთხოვნები შეიძლება ასე ჩამოვაყალიბოთ: - შეიქმნას კონსულტანტის ავტომატიზებული სამუშაო ადგილი (სერვერზე, ან ცალკე კლიენტის სახით); - აიგოს მონაცემთა განაწილებული ბაზა ფირმის ხელმძღვანელებისა და სპეციალისტების ცოდნის ამსახველი ინფორმაციით (კლიენტების სახით); - შეიქმნას ალგორითმული და პროგრამული პაკეტები, რომლებიც შეაჯერებს ექსპერტულ შეფასებათა შედეგებს და გამოიმუშავებს კოლექტიურ გადაწყვეტილებებს; - მოხდება ამ გადაწყვეტილებათა ცხოვრებაში გატარების ღონისძიებათა შემუშავება და მათი დანახარჯების ანალიზი და შეფასება; - აიგება კონსულტანტისა და ექსპერტების სამუშაო ადგილების ინტერფეისული პროგრამები.

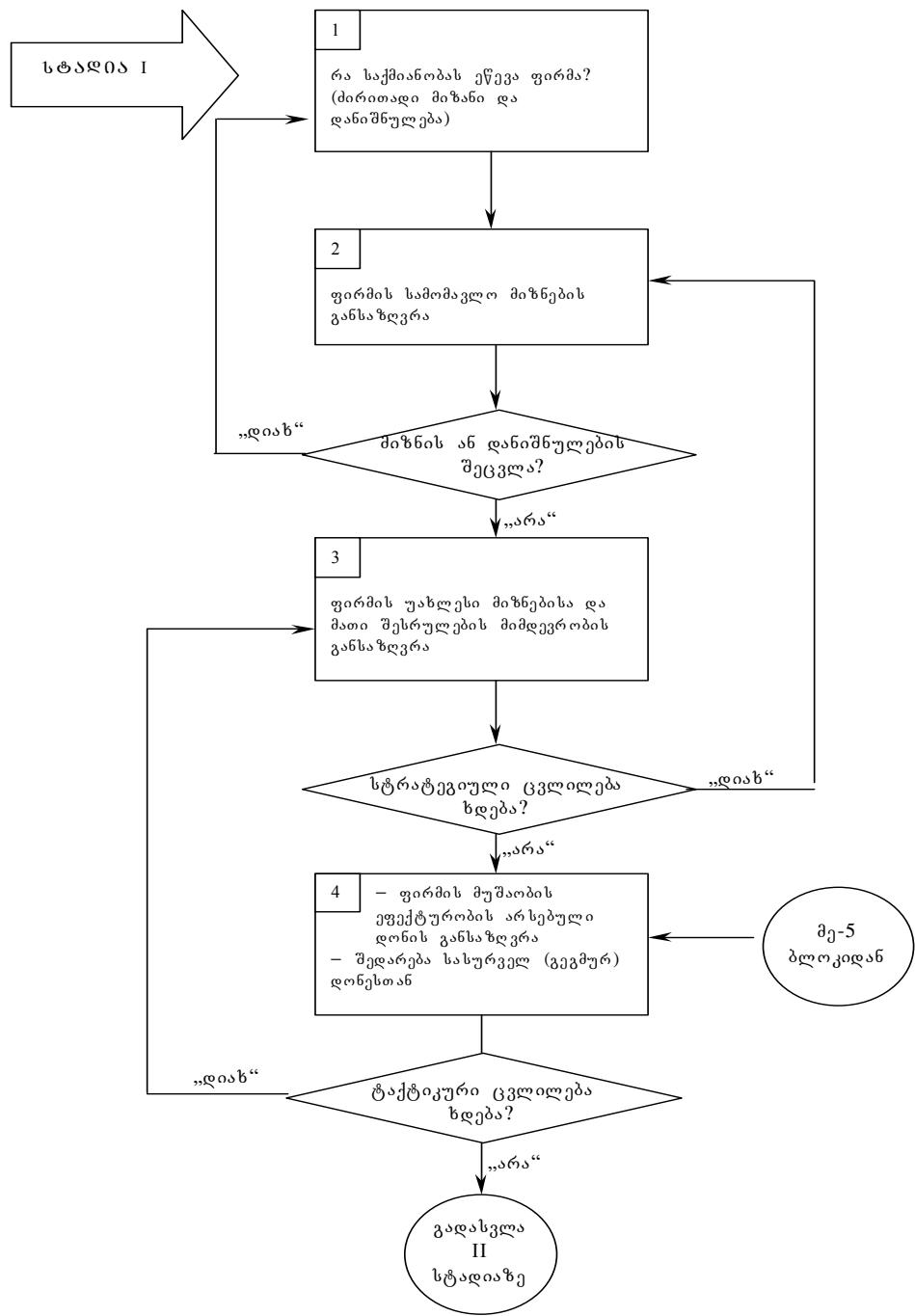
ამგვარად, ფირმის ხელმძღვანელობა და მთავარი სპეციალისტები თვითონ არიან ექსპერტები, ეხმარებიან კონსულტანტს საბოლოო გადაწყვეტილებათა მოძებნაში. კონსულტანტს აქვს მეთოდიკა, თუ როგორ წარმართოს მუშაობა ექსპერტებთან. ეს მეთოდიკა გულისხმობს ობიექტის კვლევისათვის სისტემური მიდგომისა და ექსპერტულ შეფასებათა მეთოდების გამოყენებას. ჩვენ მათ შემდგომში განვიხილავთ.

## 1.2. საკვლევი ობიექტის დიაგნოსტიკური ანალიზის ოთხეტაპიანი მოდელი

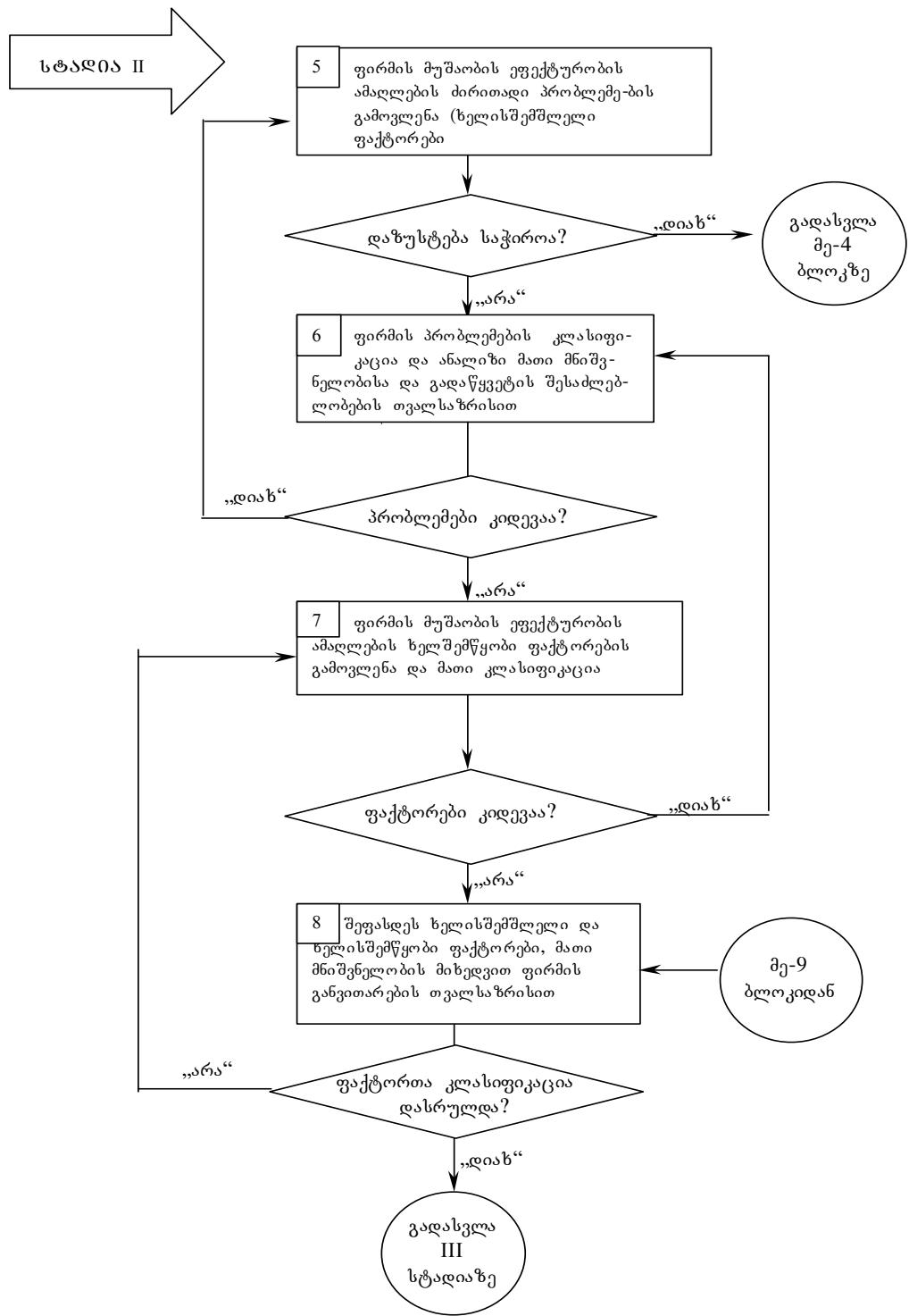
ამერიკელი მეცნიერები ეკონომიკური დაგეგმვისა და ფირმების  
მუშაობის კონსულტირების სფეროში რობერტ აბრამსონი და უოლტი  
ჰალსები გვთავაზობენ ოთხეტაპიან მოდელს ორგანიზაციის  
გამოსაკვლევად და სტრატეგიული ღონისძიებების შესადგენად (ნახ.  
1.3÷1.7) [1].



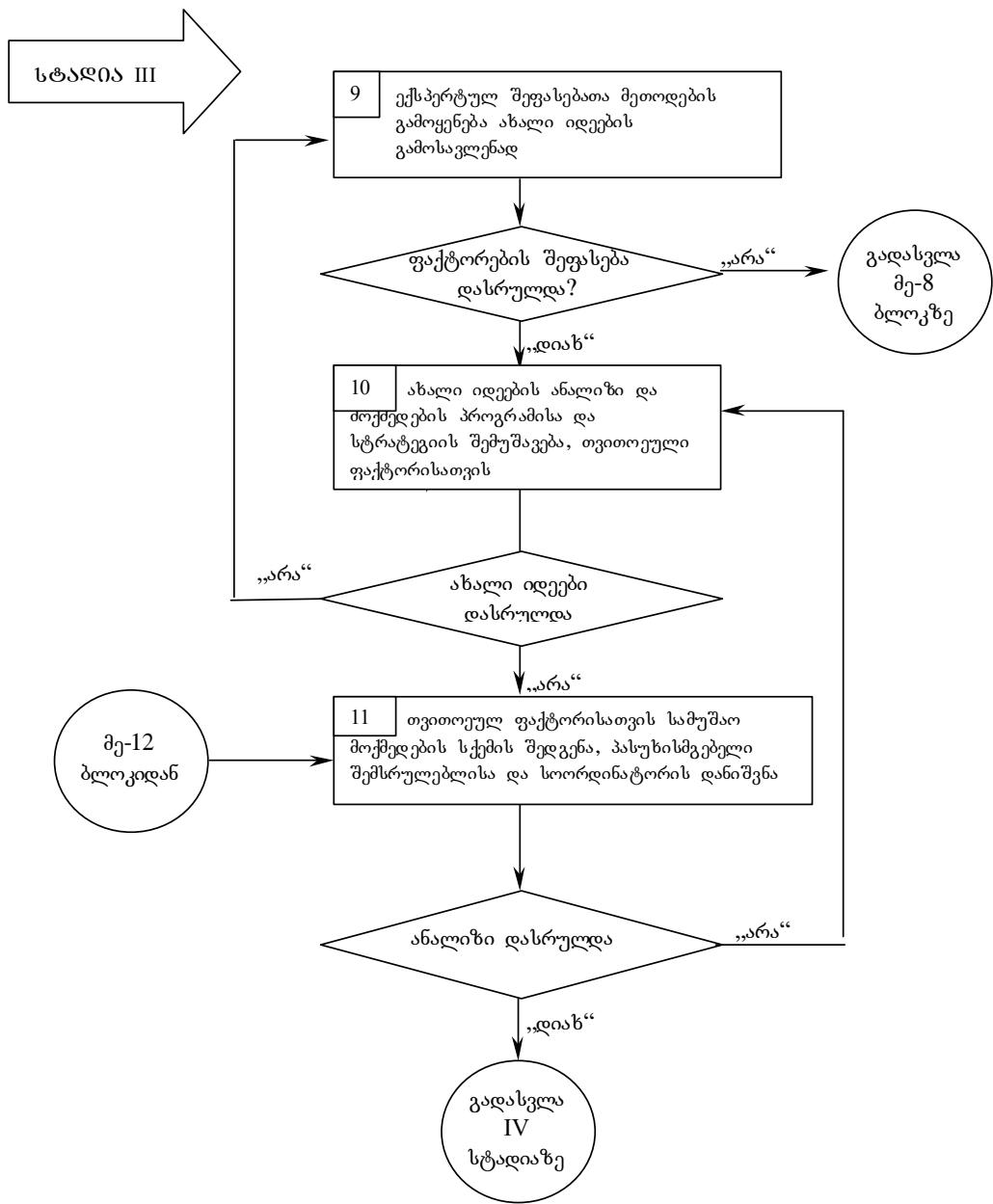
ნახ. 1.3 სტრატეგიული დაგეგმვის ოთხეტაპიანი სქემა



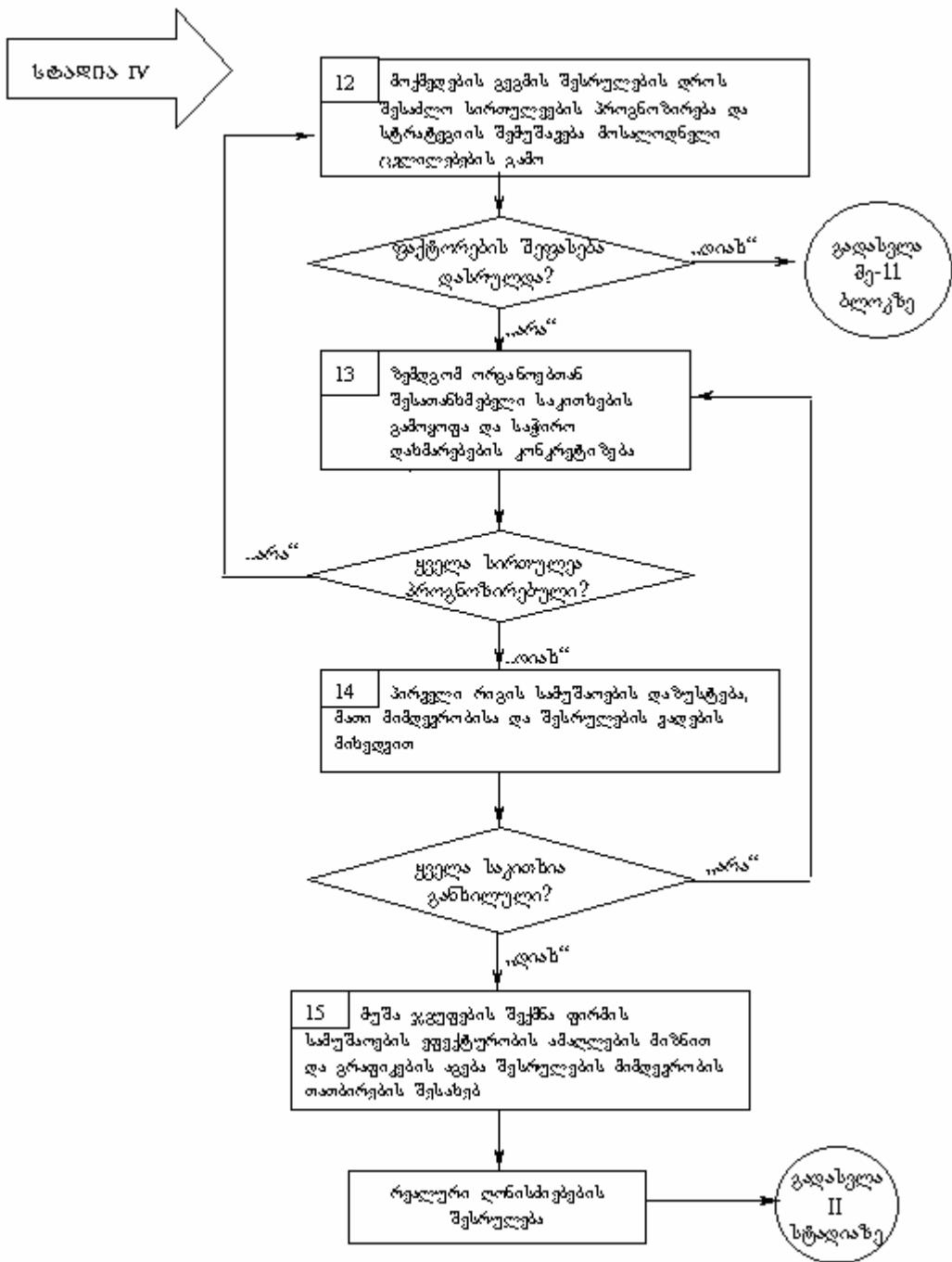
ნახ. 1.4. პირველი სტადიის ეტაპები



ნახ. 15. მეორე სტადიის ეტაპები



ნახ. 1.6. მესამე სტადიის ეტაპები



### **ნახ. 1.7. მეოთხე სტადიის ეტაპები**

1.3. მიზნის მისაღწევად შესასრულებელი ამოცანები

ექსპერტულ შეფასებათა დამუშავების ავტომატიზებული სისტემა  
არის ავტომატიზებული სამუშაო ადგილი კონსულტაციების და  
ექსპერტებისათვის.

როგორც აღვნიშნეთ, კონსულტანტი მოწვეული პიროვნებაა და  
იგი წარმართავს ექსპერტებთან (ფირმის ხელმძღვანელობა, მთავარი

სპეციალისტები) დისკუსიებს. მოსაპოვებელი ინფორმაცია ექსპერტებს ცოდნის, გამოცდილების სახით აქვთ დამახსოვრებული. არაა გამორიცხული მათი სუბიექტური შეხედულებების გავლენა ამ ცოდნის გამოყენებისას.

გარკვეული ინფორმაციის ნაწილი კონსულტანტმა შეიძლება მიიღოს ფირმის მართვის ავტომატიზებული სისტემის სხვა ფუნქციონალური ქვესისტემებიდან. მაგალითად, დაგეგმვის მართვის ავტომატიზებული ქვესისტემა – არსებული ხუთწლიანი და 1 წილიანი – ოპერატორული გეგმების მისაღებად; აღრიცხვისა და ეკონომიკური ანალიზის ქვესისტემები – ფირმაში პროდუქციის ფაქტობრივი გამოშვებისა და მისი რეალიზაციის მდგომარეობის შესახებ; სასაწყობო მეურნეობის ქვესისტემები – ფირმაში ნედლეულისა და მზა პროდუქციის არსებობის მდგომარეობა და ა.შ. (კომერციულ ბანკს ექნება მისთვის დამახიასთებელი ბიზნეს-პროცესები).

თუ ფირმის მართვის ავტომატიზებული სისტემა არ ფუნქციონირებს, მაშინ ყველა ზემოაღნიშნული ინფორმაცია კონსულტანტმა უნდა მიიღოს ფირმის ტერიტორიაზე განლაგებული საწარმოო ქვედანაყოფების ხელმძღვანელებიდან უშუალოდ. მთავარია სრული და სწორი ინფორმაციის შეკრება, რათა კონსულტანტმა ზუსტად დასვას დიაგნოზი, თუ რა უჭირს ფირმას.

ბიზნეს-პროგრამების ექსპერტულ შეფასებათა გადაწყვეტილების მიღების მხარდამჭერი ავტომატიზებული სისტემა ემსახურება ფირმის განვითარების (სტრატეგიული) გეგმის სწორად ჩამოყალიბებას. ალტერნატიული ბიზნეს-პროგრამებიდან უკეთესის შერჩევას. ამგვარად, სისტემის საბოლოო პროდუქტი თრგანიზაციის განვითარების სტრატეგიული გეგმა, რომელიც კომპიუტერშია ჩადებული თავისი პროგრამული პაკეტითა და ინფორმაციული ბაზებით, და რომელსაც სისტემა ინფორმაციის სახით მიაწვდის გადაწყვეტილების მიმღებ პირს, ანუ შესთავაზებს საუკეთესო ალტერნატიულ ვარიანტებს. საბოლოო გადაწყვეტილებას იღებს ადამიანი – პასუხისმგებელი პირი.

მომავალში ასეთი ინფორმაციული ბაზები შეიძლება გამოყენებულ იქნეს დაგეგმვის განყოფილებაში ე.წ. ოპერატიული (1 წლიანი) გეგმების შესადგენად. ექსპერტთა შეფასებების დამუშავების

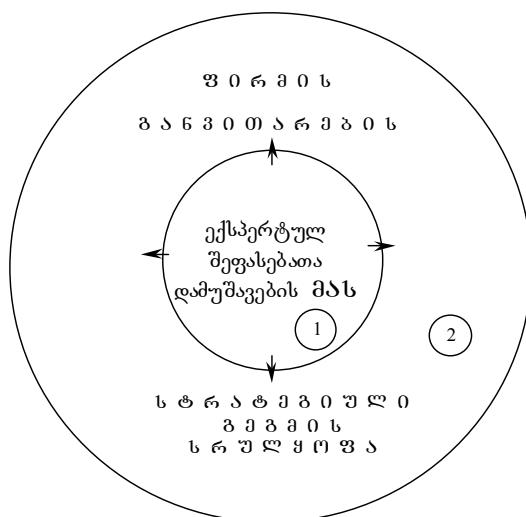
ავტომატიზებული სისტემა შეიძლება გამოყენებულ იქნეს არა მხოლოდ ფირმის (ზედა) დონეზე, არამედ ფირმის ქვეგანყოფილებებსა და საამქროებში, თუ ამ უკანასკნელთა განვითარების კონცეფცია იქნება შესაქმნელი.

მთავარია ის, რომ ამ სისტემის საშუალებით შესაძლებელი უნდა იქნეს კოლექტიური გადაწყვეტილების მიღება ექსპერტთა შეფასებების შეჯერების ხარჯზე, რომელიც ჩადებულია ამ მეთოდებში (მაგალითად, დელფისა და პატერნის მეთოდებში). მათ ჩვენ მე-2 თავში შევეხებით.

დისერტაციის მიზანია კორპორაციული ობიექტების ბიზნეს-პროგრამების აგების ავტომატიზაცია და მისთვის გადაწყვეტილების მიღების მხარდაჭერი საინფორმაციო სისტემის შექმნა.

როგორც აღვნიშნეთ, ამისათვის საჭიროა მონაცემთა მრავალდონიანი საცავების აგება, მათი ფუნქციურ-ანალიზური პაკეტების დამუშავება, ექსპერტულ შეფასებათა მეთოდების გამოყენება ფირმის განვითარების სტრატეგიული გეგმის შესამუშავებლად (მესამე სტადია, ნახ. 1.6). ამაგვარად, არსებობს ორი მიზანი: ლოკალური და გლობალური. ლოკალური მიზანია მონაცემთა საცავისა და ექსპერტულ შეფასებათა დამუშავების კომპიუტერული სისტემის შექმნა, ანუ ჩვენი დისერტაციის თემის პრაქტიკული რეალიზაცია.

გლობალური მიზანია ამ კომპიუტერული სისტემის გამოყენებით კორპორაციული ორგანიზაციის განვითარების სტრატეგიული გეგმის ოპტიმალური ვარიანტის შედგენა და ექსპერტული შეფასება. ეს აზრი ფორმალიზებულია 1.8 ნახაზზე.



ნახ.1.8. ლოკალური (1) და გლობალური (2) მიზნები

ჩვენ შემოვიფარგლებით ლოკალური მიზნით. ამიტომ აქ ჩამოვთვლით იმ ძირითად ფუნქციებს, რომლებიც ექსპერტულ შეფასებათა დამუშავების ავტომატიზებული სისტემის აგებისათვისაა საჭირო:

- 1) ობიექტის დიაგნოსტიკური ანალიზი სისტემური მიღებით;
- 2) ექსპერტულ შეფასებათა მეთოდების ანალიზი და მათი შერჩევა დასმული მიზნის მისაღწევად;
- 3) კონსულტაციის მეთოდიკის ფორმალიზება და ფუნქციონალური ბლოკ-სქემის აგება;
- 4) ინფორმაციული უზრუნველყოფის საკითხების დამუშავება;
- 5) დიალოგური პროცედურების აგება კონსულტაციისა და ექსპერტებისათვის;
- 6) პროგრამული პაკეტის დამუშავება და ტესტირება;
- 7) მომხმარებლის ინსტრუქციების ჩამოყალიბება.

#### 14. გადაწყვეტილების მიღების ხელშემწყობი სისტემები

როგორც აღვნიშნეთ, დიდი კორპორაციების უფასტურად სამართავად გადაწყვეტილების მიღების მხარდაჭერის მიზნით, აუცილებელია სრული და უტყუარი ინფორმაციის არსებობა ბიზნესის სხვადასხვა ასპექტებზე. ამაზეა დამოკიდებული ორგანიზაციის მართვის ხარისხი, დაგეგმვის ეფექტურობა და ა.შ. ამასთანავე მნიშვნელოვანია ინფორმაციის ასახვის ფორმების თვალსაჩინოება, ახალი სახის რეპორტების მიღების სისტრაფე, მიმდინარე და ისტორიული მონაცემების ანალიზის შესაძლებლობა. ასეთ შესაძლებლობათა მქონე სისტემებს უწოდებენ გადაწყვეტილების მიღების მხარდამჭერ სისტემებს (გმმს) [5]. ისინი ფართოდ გამოიყენება სხვადასხვა დარგებში, როგორიცაა, მაგალითად ტელეკომუნიკაციები, ბიზნესი და ფინანსები, ვაჭრობა, მრეწველობა და სხვ., თანამედროვე ინფორმაციული ტექნოლოგიის ინსტრუმენტის სახით.

გადაწყვეტილებათა მიღების მხარდამჭერი სისტემები (DSS - Decision Support Systems), ზოგადად, აგებულია მონაცემთა ბაზების, ხელოვნური ინტელექტის, იმიტაციური მოდელებისა და მენეჯმენტის საინფორმაციო სისტემების საფუძველზე. გმმს-ების საშუალებით შესაძლებელია არასტრუქტურირებადი და სუსტად სტრუქტურირებადი მრავალკრიტერიუმიანი ამოცანების გადაწყვეტა. ამ მიზნით გამოიყენება აგრეთვე კატეგორიალური ანალიზისა და კოგნიტური მოდელირების შემცნებითი მეთოდები [25,26,84,85].

გმმს შედგება ორი ძირითადი, აუცილებელი კომპონენტისგან: მონაცემთა საცავისა (data warehouse) და ანალიზური საშუალებებისგან (ინსტრუმენტებისგან) [62,68,71,72]. მონაცემთა საცავი წარმოადგენს კორპორაციული მონაცემების ერთიან სივრცეს, რომელშიც მონაცემები ისეთი სტრუქტურებითაა მოწესრიგებული, რომელიც იძლევა ანალიზური ოპერაციების ჩატარების ოპტიმიზაციის საშუალებას.

ანალიზური ინსტრუმენტები საშუალებას აძლევს ფუნქციურ მომხმარებელს, რომელსაც არ გააჩნია სპეციალური განათლება ინფორმაციულ ტექნოლოგიებში, მოახდინოს მონაცემების მარტივად გამოყენება და მათი ასახვა საგნობრივი სფეროს ტერმინებში. სხვადასხვა კვალიფიკაციის მომხმარებელთათვის გმს-ს აქვს მონაცემებთან მიმართვის სერვისის სხვადსხვა ტიპის ინტერფეისები.

### **1.5. მონაცემთა საცავი და ოპერატიული ანალიზის OLAP-ინსტრუმენტი**

მონაცემთა საცავი, როგორც ახალი ინფორმაციული ტექნოლოგია, აქტუალური გახდა ამერიკელი მეცნიერის, ვ. ინმონის შრომებიდან [6,7].

მონაცემთა საცავი აღიწერებოდა, როგორც „მონაცემთა სუპერმარკეტი“, „სუპერ მონაცემთა ბაზა“. ამ მიმართულებით პირველი პროექტი „ევროპის ბიზნეს ინფორმაციული სისტემა“ 1988 წელს IBM ფირმის მიერ განხორციელდა. გლობალურ მონაცემთა შენახვა და მათი ანალიზი სავაჭრო კომპანიების მუშაობისას არა მარტო ამაღლებს მუშაობის ეფექტურობას, არამედ უძლებს მკაცრ კონკურენციას, რომელიც საერთაშორისო ბაზარზე ყოველწლიურად ძლიერდება [8].

მონაცემთა საცავის ფართო გამოყენება, მსოფლიო ბიზნეს გაერთიანებაში ადასტურებს იმ ფაქტს, რომ ამ ტექნოლოგიაზე დაყრდნობით მოგება ყოველწლიურად იზრდება ათობით მილიარდი დოლარით.

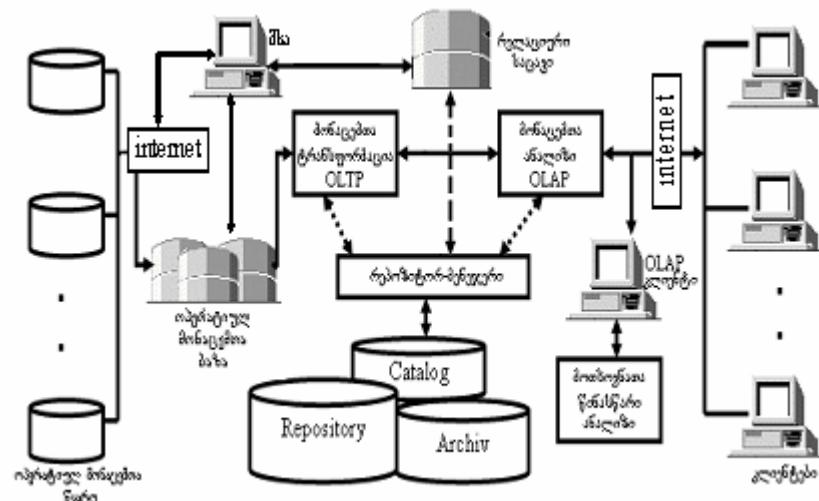
კომერციულ ბაზებში ჩატარებულმა ანალიზმა აჩვენა, რომ საბაზო ტრანზაქციების და სასურველ მონაცემებზე არსებული ინფორმაციის მოცულობა ძალიან დიდია. პირველადი ანალიზისათვის უნდა მომზადდეს გაფართოებული მონაცემები და მიეცეს ინდექსაცია.

ამისათვის საჭიროა ფართო კომპიუტერული რესურსები, რომელიც საშუალებას იძლევა მცირე დროის განმავლობაში შესრულდეს ძებნა რამდენიმე ცხრილიში, რომლიც ათობით მიღიონ ჩანაწერს შეიცავს და განხორციელდეს მონაცემთა შერჩევა.

მონაცემთა საცავი არის კომპლექსური სისტემა, რომელიც შედგება შემდეგი ძირითადი ფუნქციური ბლოკებისაგან:

- მონაცემთა განაწილებული, რელაციური ბაზების მართვის სისტემები;
  - ინფორმაციის წყაროებიდან ოპერატიულ მონაცემთა ჩატვირთვის და გარდაქმნის საშუალება;
  - საცავის დაპროექტების მეთოდური და ინსტრუმენტული საშუალებანი;
  - საცავის აგებისა და მოდიფიკაციის საშუალებანი;
  - საცავის მეტამონაცემთა იერარქიული ორგანიზების პიპერლინკური საშუალებანი;
  - საცავის ფუნქციურ მომხმარებელთა მოთხოვნების წინასწარი ანალიზისა და ტრანზაქციების ეფექტურად შესრულების დაგეგმვის საშუალებანი;
  - საცავის საინფორმაციო ბლოკებისა და არქივის ოპერატიული ანალიზის ინსტრუმენტული საშუალებანი.

1.9 ნახაზზე მოცემულია განაწილებული ავტომატიზებული მართვის სისტემის მონაცემთა საცავის ზოგადი სქემა.



#### **ნახ.19. მონაცემთა საკავის ზოგადი სქემა**

მონაცემთა საცავის მუშაობის პრინციპი ასეთია: პირველ ეტაპზე DWH-ს ტექნოლოგიის გამოყენების საშუალებით, რელაციურ ბაზებში ერთად თავმოყრილი მონაცემები ლაგდება გარკვეული სტრუქტურული თანამიმდევრობით, ხდება მათი „დაწმენდა“. მეორე ეტაპზე წარმოებს მათი ტექნოლოგიური დამუშავება მონაცემთა ოპერატიული ანალიზის OLAP – ტექნოლოგიის გამოყენებით [68,69]. ხოლო მესამე ეტაპზე ეს მონაცემები მომხმარებლებს მიეწოდებათ ინტერნეტის საშუალებით.

ინფორმაციული ბლოკები, რომლებიც მონაცემთა საცავებშია განაწილებული, მიზანმიმართულად თავსდება ინტერნეტ – გვერდებზე და ხელმისაწვდომია ფართო მომხმარებლისთვის.

მონაცემთა რელაციურ მოდელებზე აგებულ ბაზებისა და საცავებისათვის SQL (Structured Query Language) მონაცემთა სტრუქტურირებადი ენა არის ერთ–ერთი ეფექტური, საუკეთესო საშუალებაა მონაცემთა მანიპულირებისათვის. იგი აღიარებულია საერთაშორისო სტანდარტით, ამიტომაც ინფორმაციული საცავებისათვის მიზანშეწონილია რელაციური პლატფორმა.

მონაცემთა საცავი ორიენტირებულია განსაზღვრულ საგნობრივ სფეროზე და ორგანიზებულია მონაცემთა ოპერატიული ბაზიდან შემოსულ სტრუქტურულად გადამუშავებულ მონაცემთა ქვესიმრავლების საფუძველზე. ინფორმაციის წყაროს წარმოადგენს სხვადასხვა ორგანიზაციათა დანართები (აპლიკაციები), რომლებიც გამოიყენებს განსხვავებულ პროგრამულ პლატფორმებს და უკავშირდება ოპერატიულ მონაცემთა ბაზას ინტერნეტის საშუალებით (on-line რეჟიმი). შესაძლებელია აგრეთვე სხვა სახის კავშირების (off-line რეჟიმი) გამოყენებაც.

მონაცემთა საცავში ინახება მონაცემთა სტრიქონების არა მთელი სიმრავლე, არამედ ამა თუ იმ ხარისხით გაერთიანებული (აგრეგატული) ინფორმაცია, რაც ხელს უწყობს მეხსიერების ეფექტურად გამოყენებას.

მონაცემთა საცავი მოიცავს ტრანსფორმაციისა და კონვერტაციის პროგრამებს, საბაზო მეტამონაცემთა სისტემას, არქივირებული შენახვის სისტემას და ინტეგრირებულ მონაცემთა საცავს [9].

**მონაცემთა ბაზების სიტემა:** მონაცემთა საცავს ემსახურება ისე, როგორც სხვა მონაცემთა ბაზების სისტემები, რომელთა

საშუალებითაც ხდება, მონაცემთა დამოუკიდებელ პროგრამათა ინტეგრაცია, შენახვა და მართვა. მონაცემთა ბანკების მართვის სისტემა მომხმარებელთათვის უზრუნველყოფს ადამიანსა და კომპიუტერს შორის ინფორმაციის გაცვლას. იგი შეიცავს მონაცემთა დიდ რაოდენობას, რომელთა შორისაც დამყარებულია კანონზომიერი კავშირები. მონაცემთა ბანკების მართვის სისტემა შეადგენს მონაცემთა საცავის ცენტრალურ სისტემას.

**ტრანსფორმაციის პროგრამა:** ახორციელებს ინტერფეისის ფუნქციას მონაცემთა საცავსა და მონაცემთა წყაროებს შორის. მონაცემები (ინფორმაცია) განსხვავებულ მონაცემთა ბაზებიდან ან განსხვავებული ფორმატის ფაილებიდან (ASCII, ANSI, EBCDIC და ა.შ.) ექსტრაგირდება. ტრანსფორმაციის წესის თანახმად ისინი ერთიანდება (Bridging), როგორც ინტეგრირებული, სუბიექტ-ორიენტირებული, მუდმივი და დროში ცვალებადი სტრუქტურები. ტრანსფორმაციის პროგრამამ უნდა უზრუნველყოს ტრანსპორტირებისათვის ფუნქციათა წარმოდგენა და აგრეთვე მონაცემთა მომზადება საცავში გადასაგზავნად. მონაცემთა საცავში შიგა მონაცემების (ინფორმაციის) უდიდესი ნაწილი შესაძლოა მიღებული იქნას განაწილებული ოპერატიული სისტემიდან. მონაცემთა საცავში ისტორიული და მიმდინარე მონაცემების შევსება, ხდება ბაზებიდან, სადაც პერიოდულად მიმდინარეობს მონაცემთა აქტუალიზაცია. თუ ოპერატიულ მონაცემთა ჩანაწერების რეგისტრაციაში მონაცემის შეტანის თარიღი უფრო ახალია, ვიდრე ბოლო ტრანსფორმაციის შეტანის დრო, მაშინ მოხდება ამ უკანასკნელის ლიკვიდაცია. ე.ი. აუცილებლად გათვალისწინებული უნდა იქნეს ცალკეულ ტრანსფორმაციის პროცესებს შორის ვადები.

მონაცემთა გარე მიმწოდებლებია შეტყობინებათა სამსახურები, ბირჟები, პოლიტიკური საინფორმაციო სამსახურები, სამეცნიერო კვლევითი ინსტიტუტები და ბაზარი. ინფორმაციის მოწოდებისათვის გამოიყენება ისეთი საშუალებები, როგორიცაა მაგალითად: Internet, CD-ROM, Flash-Memory, FD და ა. შ. ბევრი ამ მონაცემთაგან ტრანსფორმაციის პროგრამის საშუალებით, სანამ გადავა მონაცემთა საცავში უნდა წარმოდგეს გარკვეული Internet-სტანდარტული-ფორმატის სახით.

**მეტამონაცემთა საბაზო სისტემა:** მეტამონაცემები არის მონაცემები - მონაცემების შესახებ. მეტამონაცემთა საბაზო სისტემა მონაცემთა საცავის შემადგენელი ნაწილია. მონაცემთა საცავში იგი ერთ-ერთ მთავარ როლს ასრულებს, მეტამონაცემების დანიშნულებაა მონაცემთა საცავში არსებული მონაცემების აღწერა და მათზე დამატებითი ცნობების მოგროვება. მეტამონაცემთა ინსტრუმენტის გამოყენება მომხმარებელს საშუალებას აძლევს აწარმოოს მონაცემთა მასივებში მანევრირება და ეხმარება მას მონაცემთა საცავში ორიენტირებაისათვის. მეტამონაცემთა საბაზო სისტემა მომხმარებელს ეხმარება მოთხოვნების შესაბამისად მონაცემთა შერჩევაში. ეს ხორციელდება მონაცემთა საცავში მეტამონაცემთა გამჭვირვალე ასახვის შედეგად.

**არქივირებული შენახვის სისტემა:** უზრუნველყოფს მონაცემთა დაცვას და მათ არქივირებულ შენახვას მონაცემთა საცავში. მონაცემთა არქივირებული შენახვა, როგორც ცალკე სისტემა, მონაცემთა საცავში ამცირებს მეხსიერების უჯრედებს და ზრდის მუშაობის ეფექტურობის ხარისხს. არქივირების ეფექტური სისტემა მნიშვნელოვანია, რადგან მოკლე ვადაში შესაძლებელია მონაცემთა გადმოტვირთვა მომხმარებელთა მოთხოვნების შესაბამისად.

ხშირად სისტემაში თავს იყრის უსარგებლო ინფორმაციათა ნაკადი და იკავებს დიდ ადგილს, რაც აფერხებს სისტემის მუშაობის ეფექტურობას, არქივირებული სისტემის დახმარებით ხდება ასეთი ინფორმაციის განადგურება. შესაძლოა ასევე დაფიქტური ტრანზაქციის შედეგად მოხდეს მონაცემთა „დაზიანება“. ამ შემთხვევაში ამოქმედდება მონაცემთა დაცვის სისტემა, რაც უზრუნველყოფს დაფიქტების აღმოფხვრას და არასასურველი ინფორმაციის განადგურებას.

არქივირებული შენახვისას ყურადღება უნდა გავამახვიდოთ ინფორმაციის შენახვის კანონებზე, რომელიც ითვალისწინებს არქივში ინფორმაციის შენახვის ვადებს, რადგან შენახვის ინფორმაცია გარკვეული პერიოდის შემდეგ კარგავს აქტუალობას.

**ანალიზური დამუშავების OLAP-ტექნოლოგია:** მას მონაცემთა მრავალგანზომილებიან, კომპლექსური ანალიზის ტექნოლოგიას

უწოდებენ. **OLAP** (Online Analytical Processing) ნიშნავს „მონაცემთა ოპერატორიული ანალიზს“. იგი პირველად მონაცემებს წარმოადგენს ინფორმაციის სახით, რომლის დახმარებითაც შესაძლებელი ხდება საწარმოს მოცულობის შესახებ ვიქონიოთ რეალური წარმოდგენა. ამავე დროს იგი უნიკალური ინსტრუმენტია, რომლიც საშუალებას გვაძლევს სხვადასხვა ანალიზური ჭრილით ჩავატაროთ ინფორმაციის მრავალგანზომილებიანი ანალიზი.

მონაცემთა საცავების დაპროექტებისა და მისი ფუნქციონირებისათვის, მეთოდური თვალსაზრისით ჩვენ ვიყენებთ კვ-კოდის მიერ ჩამოყალიბებულ პრინციპებს [10]. ესაა ის 12 წესი, რომელსაც უნდა აკმაყოფილებდეს ნებისმიერი განაწილებული სისტემა მონაცემთა საცავით, რათა ჩატარდეს საინფორმაციო ბლოკების სრულფასოვანი ოპერატორიული ანალიზური სამუშაოები.

**1. მონაცემთა მრავალგანზომილებიანი კონცეპტუალური წარმოდგენა.** მომხმარებელ-ანალიტიკოსი საპრობლემო სფეროს, თავისი ბუნების მიხედვით ხედავს როგორც მრავალგანზომილებიანს. შესაბამისად OLAP- მოდელიც უნდა იყოს მრავალგანზომილებიანი. ასეთი ტიპის კონცეპტუალური სქემა (მომხმარებელთა წარმოდგენები) აითლებს მოდელირებას, ანალიზს და გამოთვლებს;

**2. გამჭვირვალეობა.** OLAP წარმოდგენილი უნდა იყოს ღია არქიტექტურის კონტექსტში, სადაც მომხმარებელს საშუალება ექნება დროის ნებისმიერ მონაკვეთში, ანალიზური ინსტრუმენტის საშუალებით დაუკავშირდეს სერვერს და მიიღოს მისთვის სასურველი ინფორმაცია;

**3. მიღწევადობა.** OLAP – ის მომხმარებელ ანალიზატორს უნდა ჰქონდეს ანალიზის ჩატარების საშუალება, რომელიც ემყარვება საერთო კონცეპტუალურ სქემას, რომელშიც განთავსებულია რელაციური მონაცემთა ბაზა სწარმოთა შესახებ არსებული ყველა ახალი და ძველი მონაცემებით. ეს ნიშნავს, რომ OLAP – მა უნდა წარმოადგინოს თვისი საკუთარი ლოგიკური სქემა, რათა შეასრულოს შესაბამისი გარდაქმნა და მომხმარებელს წარუდგინოს მონაცემები. გარდა ამისა აუცილებელია წინასწარ იმაზე ზრუნვა, თუ სად, როგორ და როგორი სახის ფიზიკური ორგანიზაციის მონაცემები იქნას გამოყენებული.

OLAP სისტემაში უნდა შეასრულოს ისეთი მონაცემების დამუშავება, რომელთა მოთხოვნაც რეალურად არსებობს;

**4. ანგარიშთა დამუშავებისას მუდმივი წარმადობა.** თუ ანალიტიკოსის მიერ ჩატარებული გაზომვათა რაოდენობა ან მონაცემთა ბაზების რიცხვი მნიშვნელოვნად იზრდება, მომხმარებელ ანალიზატორისთვის ეს პროცესი უნდა დარჩეს შეუმჩნეველი და არ უნდა აისახებოდეს საწარმოო პროცესების წარმადობის შემცირებაზე;

**5. კლიენტის სერვერის – არქიტექტურა.** მონაცემთა დიდი ნაწილი, რომელიც მოითხოვს ოპერატორულ ანალიზურ გადამუშავებას, უნდა მუშაობდეს კლიენტ-სერვერულ რეჟიმში. ამ თვალსაზრისით აუცილებელია, რომ ანალიზური ინსტრუმენტის სერვერული კომპონენტები იყოს „ინტელექტუალური“, რადგან განსხვავებულ კლიენტებს შეეძლოთ დაუკავშირდნენ სერვერს და გამოიყენონ პროგრამიული პაკეტები. „ინტელექტუალურ“ სერვერს უნდა შეეძლოს მონაცემთა ბაზის შეუთავსებადი ლოგიკური და ფიზიკური სქემის ასახვა და გაერთიანება. ეს უზრუნველყოფს გამჭვირვალეობას და საშუალებას იძლევა აიგოს საერთო კონცეპტუალური, ლოგიკური და ფიზიკური სქემები;

**6. მრავალგანზომილება.** გაზომვის ყოველი მცდელობის დროს გამოყენებული უნდა იყოს მიუკერძოებებითი სტრუქტურა და ოპერაციული შესაძლებლობა. დამატებითი ოპერაციული შესაძლებლობა უნდა მიეცეს ერთ-ერთ რომელიმე ცდას და თუ ეს გაზომვა სიმეტრიული იქნება სხვა გაზომვის შედეგების, მაშინ ცალკე აღებული ფუნქცია შეიძლება წარმოვადგინოთ ნებისმიერი გაზომვის სახით;

**7. დინამიკური მართვა გამონთავისუფლებული რეჟიმით.** OLAP – ინსტრუმენტის ფიზიკური სქემა უნდა ადაპტირდებოდეს სპეციფიკურ ანალიტიკურ მოდელთან, რათა ოპტიმალურად მართოს გამონთავისუფლებული მატრიცა. ერთი დაცლილი მატრიცისთვის არსებობს ერთადერთი ოპტიმალური ფიზიკური სქემა. OLAP – ინსტრუმენტის ბაზური ფიზიკური მონაცემები პრაქტიკული ოპერაციებისთვის, რომელთაც აქვთ დიდი ანალიზური მოდელი უნდა კონფიგურირდებოდეს ნებისმიერ ქვესიმრავლესთან. თუ OLAP –

ინსტრუმენტს არ შეუძლია გააკონტროლოს და დაარეგულიროს საანალიზებელი მონაცემების მნიშვნელობები, ის ჩაითვლება უსარგებლოდ და არასაიმედოდ;

**8. მრავალმომხმარებლობა.** ხშირ შემთხვევაში მომხმარებელ-ანალიტიკოსი დასმულ მოთხოვნებს აყენებს ერთ ანალიზურ მოდელთან ან ქმნის განსხვავებულ მოდელს ერთი სახის მონაცემებიდან. **OLAP** – ინსტრუმენტი კი საშუალებას გვაძლევს მივიღოთ უსაფრთხო, სრულყოფილი და ზუსტი ანალიზური შედეგები;

**9. შეუზღუდავი გადამკვეთი ოპერაციები.** მონაცემთა შემოწმების სხვადასხვა დონე და გაერთიანების გზა, მათი იერარქიული ბუნების გათვალისწინებით მჭიდრო კავშირშია **OLAP** - მოდელთან ან დანართთან. თვითონ ინსტრუმენტი უნდა მოიაზრებოდეს შესაბამის გამოთვლებთან და არ უნდა მოსთხოვოს მომხმარებელს თავიდან განსაზღვროს გამოთვლები და ოპერაციები. გამოთვლები მოითხოვს რომელიმე გამოყენებულ ენაში განსხვავებული ფორმულების განსაზღვრას. ასეთი ენა შეიძლება გამოვიყენოთ ნებისმიერი სიდიდის მონაცემთა მანიპულირებისთვის და არ შეზღუდოს მონაცემები არსებული კუბის უჯრედებს შორის და კონკრეტული უჯრედების საერთო ატრიბუტებზე;

**10. მონაცემთა ინტუიციური მანიპულაცია.** მონაცემთა დეტალიზაციის, გაერთიანების და სხვა მანიპულაციები უნდა იყენებდეს ცალკეულ უჯრედებზე ანალიზური მოდელის შედეგებს და არ უნდა იყენებდეს მომხმარებლის ინტერესებს. მომხმარებელ ანალიტიკოსს უნდა ჰქონდეს ყველა აუცილებელი პირობა იმისა, რომ მიიღოს სრულყოფილი ინფორმაცია;

**11. ანგარიშების მიღების მოქნილი საშუალება.** შეტყობინებათა დამუშავება და პასუხის გაცემა უნდა იყოს მოქნილი და ელასტიკური. მომხმარებელს უნდა შეეძლოს მონაცემთა კომბინირება და გაანალიზება. მოქნილობა მნიშვნელოვანია, რათა ყურადღება გამახვილდეს მონაცემთა განმასხვავებელ ნიშნებზე. თუ რაიმე სირთულე წარმოიქმნება, უნდა შევჩერდეთ ინდივიდუალურ ინფორმაციულ მოთხოვნაზე და შეირჩეს მხოლოდ მომლოდინე მოთხოვნა;

**12. შეუზღუდავი ზომები და აგრეგაციათა რაოდენობა.**  
გამოკვლევებმა აჩვენეს, რომ აუცილებელი გაზომვა ერთდროულად შეიძლება ჩატარდეს 19-ჯერ. აქედან გამომდინარე შეიძლება ვთქვათ, რომ ანალიზური ინსტრუმენტი საშუალებას გვაძლევს ერთდროულად ვაწარმოოთ 15-დან 20-მდე გაზომვა, ამასთან თითოეული გაზომვის მცდელობა არ არის შემოსაზღვრული დადგენილი რიცხვით.

ეს პირობები შეიძლება ჩავთვალოთ, ოპერატორული ანალიზური დამუშავების თეორიულ ბაზისად. როგორც უკვე ავღნიშნეთ **OLAP**-ზი ძევს მონაცემთა დამუშავების მრავალგანზომილებიანი სტრუქტურის იდეა. როდესაც ვსაუბრობთ **OLAP**-ზე, უნდა ვიგულისხმოთ, რომ ეს არის მონაცემთა ლოგიკური სტრუქტურის მრავალგანზომილებიანი ანალიზური ინსტრუმენტი.

### **1.6. საბანკო სისტემების კომუნიკაციები და საინფორმაციო ტექნოლოგიები**

კომურციულ ბანკში მომუშავე პერსონალისა და მისი კლიენტების თვალსაზრისით ბანკი არის ფინანსური დაწესებულება. ტელეკომუნიკაციების სპეციალისტების თვალსაზრისით კი ბანკი ინფორმაციის შენახვის, დამუშავების და გადაცემის დაწესებულებაა. ბანკში მიმდინარე ფინანსური და ფულადი პროცესები შეიძლება და უნდა იქნას ინტერპრეტირებული, როგორც ინფორმაციის დამუშავების, შენახვის და გადატანის პროცესები (ელექტრონული ფულის კონცეფცია). ეს თანაბრად ეხება როგორც კლიენტის ანგარიშის მდგომარეობის შესახებ მანიპულირებადი ინფორმაციის გამოთვლით პროცესებს, ასევე ბანკის მართვის პროცესებს და მის სხვადასხვა სფეროში გადაწყვეტილების მიღებას. განსაკუთრებით ეს ინტერპრეტაცია თავს იჩენს ბანკების, საქმიანი მსოფლიოს და მთლიანად საზოგადოების გადასვლაში ფულის ბრუნვის ახალ მეთოდებზე, როდესაც საკრედიტო და სადებეტო ბარათებს, ბანკომატებს, კლიენტის ელექტრონულ მომსახურებას და სხვა მსგავს პროცესებს მივყავართ იმისკენ, რომ ყველა საგადასახადო, გამოთვლით და სხვა ფინანსურ პროცედურას არ დასჭირდება ქაღალდის ფული და დოკუმენტები და შემოიფარგლება ინფორმაციის კომპიუტერული

დამუშავებით და გადაცემით. ასეთი პერსპექტივის არსებობის დროს შეუძლებელია კომპიუტერული ინფორმაციული სისტემების და კომპიუტერული ტელეკომუნიკაციების როლის შემცირება საბანკო სისტემაში.

თანამედროვე კომერციულ ბანკებს გააჩნია ფილიალები მთელ მსოფლიოში. ასეთი ბანკის ეფექტური მართვა შესაძლებელია მხოლოდ ყველა განყოფილების მოქმედების შესახებ ოპერატიული ინფორმაციის არსებობისას. ყველა ამ ფაქტორმა გამოიწვია საბანკო სისტემის ავტომატიზაციის აუცილებლობა და საბანკო სისტემებში ინფორმაციული ტექნოლოგიების, ტელეკომუნიკაციების საშუალებების, მონაცემთა ბაზების გამოყენება.

საბანკო ტექნოლოგიების ავტომატიზაციის პროცესების აქტიური განვითარება დაიწყო 80-იანი წლების ბოლოს და განსაკუთრებით გამოაჩინა თავი 90-იანი წლების ბოლოს, როდესაც გაჩნდა ბევრი კომერციული ბანკი. ამ დროს გამოთვლითი ცენტრები, რომლებზეც ხორციელდებოდა საბანკო ინფორმაციის დამუშავება, უკვე ვერ ხთავაზობდა ბანკს ყველა სფეროს მომსახურებას, რომელიც საჭირო იყო რუტინული სამუშაოს შესამცირებლად და ბანკის ფინანსური მდგრმარეობის ანალიზისთვის და პროგნოზირებისთვის.

კომპიუტერული ტექნიკის და ინფორმაციული ტექნოლოგიების განვითარებამ საშუალება მისცა ბანკების უმეტესობას შეექმნა საკუთარი გამოთვლითი კომპლექსები, რომელთა ბაზაზე ავტომატიზებული იქნა ბანკის შემოქმედების ძირითადი მიმართულებები.

ბანკის საინფორმაციო და სხვა ტექნოლოგიების ავტომატიზაცია ხელს უწყობს მომსახურების ხარისხის ამაღლებას ავტომატიზებული სამუშაო ადგილების შექმნით ყველა დონის სპეციალისტებისთვის. საბანკო ტექნოლოგიების ავტომატიზაციაში ადგილს პოულობს როგორც მარტივი პროგრამული პროდუქტები, რომლებიც საშუალებას გვაძლევს შევავსოთ რამდენიმე ფორმა, და ასევე საკმაოდ ინტელექტუალური კომპლექსები, რომლებიც წყვეტს ბანკის მართვის ამოცანებს.

მართვის ინფორმაციული სისტემა უნდა უზრუნველყოდეს შემდეგ პირობებს:

- სისტემამ უნდა უზრუნველყოს ინფორმაციის უსაფრთხოება და დაცვა;
- სისტემასთან ურთიერთობა არ უნდა იყოს რთული, მისი მომსახურება უნდა იყოს ადვილი;
- სისტემას უნდა ჰქონდეს მომხმარებელთან ურთიერთობის ე.წ. მეგობრული ინტერფეისი.

**ტექნიკური აღჭურვილობა:** მნიშვნელოვანი ფაქტორი, რომელიც მოქმედებს საბანკო სისტემის ეფექტურ მუშაობაზე და ფუნქციონალურ შესაძლებლობებზე, არის ტექნიკური საშუალებების შემადგენლობა და არქიტექტურა.

თანამედროვე საბანკო სისტემები აღჭურვილია აპარატურული საშუალებების რიგით, რომელშიც შედის:

- გამოთვლითი ტექნიკის საშუალებები;
- ლოკალური გამოთვლითი ქსელების აპარატურა;
- ტელემუნიკაციების და კავშირგაბმულობის საშუალებები;
- სხვადასხვა საბანკო მომსახურების საავტომატიზაციო აპარატურა.

გამოთვლითი ტექნიკის საშუალებების გამოყენება ძირითადად ორიენტირებულია პერსონალურ კომპიუტერებზე. ინფორმაციული სისტემა იქმნება უფრო მძლავრ ცენტრალურ მინი-მანქანაზე და შედარებით იაფიან ტერმინალებზე. ცენტრალური მანქანად შეიძლება გამოყენებულ იყოს მრავალპროცესორული სისტემა. საბანკო სისტემების ტექნიკური ბაზების თანამედროვე ძირითადი მიმართულება – ლოკალური ქსელების ბაზაზე შექმნილი, განაწილებული სისტემები მაღალმწარმოებლური კომპიუტერით.

არსებულ საბანკო სისტემებში, სადაც სტანდარტულად გამოიყენება მხოლოდ ვინდოუსის პროგრამები (Windows Applications), ბევრ სირთულეებს, პრობლემებს ქმნის. ამ შემთხვევაში საჭიროა პროგრამების დაინსტალირება ყველა მომხმარებელის კომპიუტერზე. ხოლო როდესაც ადგილი აქვს პროგრამის განახლებას ან რაიმე პროგრამულ ცვლილებას, საჭიროა ხდება ცვლილელების შეტანა

თითოეულ კომპიუტერზე ცალ-ცალკე. რაც განსაკუთრებით შრომატევადია. გარდა ამისა ეს ძვირადღირებული პროცესია, რადგან ამ შემთხვევაში ორგანიზაციამ უნდა გასწიოს ხარჯი და შეინახოს პროგრამისტთა ჯგუფები სხვადასხვა ფილიალებსა თუ მომსახურე პუნქტებში, რათა მათ განახორციელონ მორიგი ცვლილებების შეტანა, დაინსტალირება, განახლება პროგრამებში. ქსელში მომხმარებლის მომსახურების რეჟიმები უნდა იყოს ორგანიზებული როგორც კლიენტ-სერვერის სისტემა.

ჩვენი აზრით ყველა ამ ზემოჩამოთვლილ პრობლემას ადვილად მოაგვარებს ვებ-პროგრამები. ვებ-პროგრამა არის პროგრამა შექმნილი კონკრეტული საჭიროებისათვის, რომელიც გამოიყენება ქსელში. ეს არის ვებ-გვერდი (ან ვებ-გვერდების ერთობლიობა), რომელიც საშუალებას აძლევს მომხმარებელს მიიღოს სასურველი ინფორმაცია, იყიდოს სასურველი ნივთები, მოუსმინოს ინტერნეტ რადიოს და სხვ. რასაც ქსელი გვთავაზობს. ვებ ინტერფეისის გამოიყენებისათვის მომხმარებელმა უნდა იცოდეს URL-მისამართი და, შესაძლებელია, დასჭრიდეს სახელი და პაროლი. სხვა გზა, რომლითაც შეიძლება ვებ-ინტერფეისი წარმოვიდგინოთ, არის ვებ-გვერდი, რომელიც გვთავაზობს ფართო შესაძლებლობებს.

ვებ-გვერდები ძალზე მძლავრი ხდება, როდესაც დაკავშირებულია მონაცემთა ბაზასთან. გვერდები წარმოიქმნება დინამიკურად და შეიცავს ყველაზე უახლეს ინფორმაციას, და ამისათვის არ არის საჭირო ვინმე აკეთებდეს ვებ-გვერდის განახლებას (update). გვერდებზე შესაძლებელია იყოს ახალი ამბები, პროდუქტის არსებობის თარიღები, ან ინფორმაცია სასურველი პროდუქტის ტრანსპორტირებაზე, ეს ყველაფერი ხორციელდება ინფორმაციის მიღებით ბაზიდან, როგორც კი სასურველ გეერდებზე მოთხოვნა გაიგზავნება. ვებ გვერდის ბაზასთან დაკავშირება რეალურად გადააქცევს მას მძლავრ ვებ-გვერდად.

**ინტრანეტი:** ორგანიზაციებს შეუძლია გაიყვანოს შიგა-ქსელი ე.წ. ინტარანეტი. ინტრანეტი იცავს ორგანიზაციას თავისი შიგა ფაილებისა და კონფიდენციალური ინფორმაციის წვდომისაგან გარე პირთათვის. ინტრანეტი ხშირად გამოიყენება ფაილების და მეილების

ორგანიზაციის წევრთათვის ერთობლივი წედომისათვის, და ამავე დროს, გარე პირთათვის იგივე ინფორმაციის ბლოკირებისათვის.

ვებ პროგრამა უფრო კარგად გამოიყენება ინტრანეტში ვიდრე ინტერნეტში. ეს ნიშნავს, რომ ის უფრო მისაწვდომია ინდივიდუალურად ვიზუალური გვერდისათვის. ვებ პროგრამები ინტერნეტში გამოყენებისას შეიძლება შეიზღუდოს სპეციალური მომხმარებლებით, და ზოგ შემთხვევებში კი სპეციფიკური IP მისამრთებით.

შიგა ვებ-პროგრამები, რომლებიც შეზღუდულია სპეციფიკური მომხმარებლისა თუ კომპიუტერებისთვის, დიდად გამოსაყენებელია ისეთ ფინანსურ ინსტიტუტებში, როგორიცაა კომერციული ბანკი.

**ATM ტექნოლოგია:** კომერციულ ბანკებში ფართოდ გამოიყენება Asynchronous Transfer Mode ტექნოლოგია. მას საფუძვლად უდევს ასინქრონული გადაცემის რეჟიმი და შეუძლია ნებისმიერი ინფორმაციის (მონაცემები, ვიდეოგამოსახულება და ხმა) გადაცემა. იგი სთავაზობს მომხმარებელს სხვადასხვა მასშტაბისა და აგებულების ქსელის აგებას.

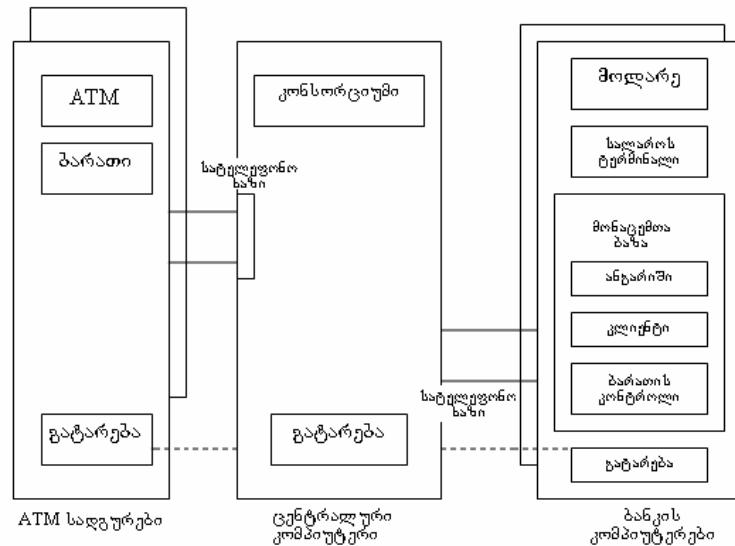
ბანკისათვის, რომელსაც გააჩნია ტერიტორიულად დაშორებული ობიექტები-განყოფილებები და რომლებც გამოიყენებს ერთიდამავე ინფორმაციულ სტრუქტურას, ATM ტექნოლოგია საკმარისად ეფექტურია ქსელის შექმნისათვის საჭირო ხარჯების ოპტიმიზაციის თვალსაზრისით. მისი მთავარი უპირატესობა მდგომარეობს იმაში, რომ მას შეუძლია გამოიყენოს ერთიანი უნიფიცირებული აპარატურა ყველა ტიპის კომუნიკაციისათვის: ლოკალური და გლობალური ქსელი, სატელეფონო, ვიდეოგამოსახულების გადაცემისათვის განკუთვნილი ქსელი (ვიდეოკონფერენციები, უსაფრთხოების სისტემა).

მაღალმწარმოებლურობის გარანტირებული ხარისხის ხარჯზე აუდიო და ვიდეო ინფორმაციის გადაცემა ATM არხებში შესაძლებელია რეალურ დროში, რაც ძალიან მნიშვნელოვანია ვიდეოკონფერენციის დროს. მონაცემები, რომლებიც გადაიცემა ATM ქსელში ვირტუალური არხით ორ კლიენტს შორის, ხელმისაწვდომია მხოლოდ მათვის. უსაფრთხოების უზრუნველყოფისა და კონფიდენციალური ინფორმაციის მიღების თვალსაზრისით ეს ქსელი გაცილებით უფრო დაცულია, ვიდრე საერთო გამოყენების პრინციპებზე აგებული ქსელები (მაგ.: Ethernet ან

Fast Ethernet, სადაც ინფორმაციის მიღება შესაძლებელია ქსელის სეგმენტის ნებისმიერ წერტილში).

საბანკო ქსელის მართვის სისტემის არქიტექტურა: განვიხილოთ საბანკო ქსელის მართვის სიტემა, რომელიც წარმოადგენს პიბრიდულ სისტემას. პირველ რიგში იგი არის სისტემა ინტერაქტიული ინტერფეისით, რომელშიც ინტერაქტიული მოქმედებები წარმოებს სალაროს ტერმინალების და ATM ტექნოლოგიების მეშვეობით, მეორეს მხრივ ეს არის ტრანსაქციების მართვის სიტემა, რადგან იგი უზრუნველყოფს გატარებების შესრულებას.

საბანკო ქსელის მართვის სიტემის არქიტექტურა მოცემულია 1.10 ნახაზზე. იგი შედგება 3 ქვესისტემისაგან: კონსორციუმის ქვესისტემა, ბანკისა და ATM მომსახურეობის ქვესისტემები.



ნახ.1.10. საბანკო ქსელის მართვის სიტემის არქიტექტურა

კლიენტის ანგარიშების მონაცემებისა და საბანკო დოკუმენტაციის საცავი მდებარეობს საბანკო ქვესისტემებში, რომლებიც რეალიზდება ბანკის კომპიუტერებში. ვინაიდან მნიშვნელოვანია მონაცემთა ერთობლიობის შენარჩუნება და რამდენიმე გატარების (ტრანსაქციების) პარალელური მომსახურების უზრუნველყოფა, ამიტომაც მონაცემთა საცავი რეალიზდება ამ ბანკების მონაცემთა ბაზის საფუძველზე.

ასინქრონული პარალელურობა წარმოიშობა რამდენიმე დამოუკიდებლად მომუშავე ATM-ის და სალაროს ტერმინალების

პარალელური მომსახურეობის აუცილებლობით. ყოველ ტერმინალს შეუძლია მოქმსახუროს მხოლოდ 1 გატარებას, რომელიც თავის მხრივ დაკავშირებულია კონსორციუმის ცენტრალურ და ერთ-ერთი ბანკის კომპიუტერებთან. როგორც ნახაზიდან ჩანს, თითოეული გატარება განაწილებულია სამ მოწყობილობაზე. პროგრამული უზრუნველყოფაც ასევე შედგება სამი ნაწილისაგან. ყოველი მათგანი რეალიზდება ცალკეული კლასის სახით.

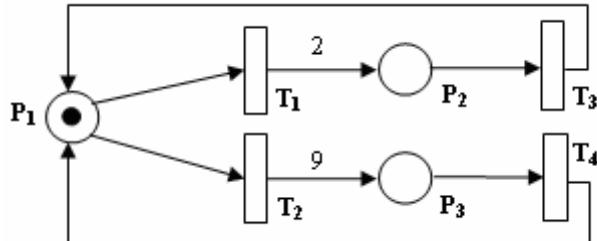
### **1.7. მდგომარეობათა და პროცესების მოდელირება პეტრის ქსელებით**

კლასიფიკაციის თვალსაზრისით გამოყოფენ დაბალი და მაღალი დონის პეტრის ქსელებს [11]. ძველი კლასიფიკაცია პოზიციებზე, გადასასვლელებსა და რკალებზე იყო ორიენტირებული და განასხვავებდა მათ ისეთ მახასიათებლებს, როგორიცაა მარკერთა მაქსიმალური რაოდენობა პოზიციაში, რკალების ჯერადობა და სხვა. ახალ კლასიფიკაციაში ყურადღება უშუალოდ მარკერთა სემანტიკაზეა გამახვილებული.

კერძოდ, დაბალი დონის პეტრის ქსელებში დაიშვება მხოლოდ „შავი” მარკერები ყოველგვარი შინაგანი სტრუქტურის გარეშე, ხოლო მაღალი დონის პეტრის ქსელები დამატებით წინასწარ განსაზღვრული სტრუქტურის „ფერად” მარკერებსაც შეიცავს, თუმცა უნდა აღინიშნოს, რომ ტერმინები „შავი” და „ფერადი” სიმბოლურია და ლიტერატურაში მათ ხშირად განსხვავებული სახელებით მოიხსენიებენ [12].

დაბალი დონის პეტრის ქსელების ქვეკლასებიდან შეიძლება დავასახელოთ ავტომატური პეტრის ქსელები, მარკირებული გრაფები, პეტრის ქსელები თავისუფალი არჩევანით, ელემენტარული სისტემური ქსელები, C/E-ქსელები, უსაფრთხო S/T ქსელები, S/T (კლასიკური) ქსელები და სხვა, ხოლო მაღალი დონის პეტრის ქსელების ყველაზე კარგად გამოკვლეულ და განსაზღვრულ ქვეკლასს სისტემური პეტრის ქსელები წარმოადგენს [13,14]. ჩვენს ნაშრომში მეტწილად გამოვიყენებთ S/T ტიპის კლასიკურ და სისტემურ პეტრის ქსელებს, ამოტომაც აქ მათ მოკლედ დავახასიათებთ.

S/T-ქსელები (State/Transition Nets) კლასიკური პეტრის ქსელების წარმომადგენელია (მდგომარეობა/გადასასვლელების ტიპის). იგი შედგება მსგავსი მარკერებისგან, რომელთა გრაფიკული ფორმა პატარა ჟავი წრეა პოზიციის ფარგლებში (ნახ.1.11).



ნახ.1.11. S/T - პეტრის ქსელის ფრაგმენტი

S/T-ქსელების პოზიცია შეიძლება ერთზე მეტ მარკერს შეიცავდეს, ხოლო მარკერების დიდი ოდენობის შემთხვევაში პოზიციაში მათი რაოდენობა რიცხობრივად ჩაიწერება. გადასასვლელის გაშვების აუცილებელი პირობაა ყველა შემაგალ პოზიციაში დამაკავშირებელი რკალის ჯერადობაზე არანაკლები ოდენობის მარკერების მოგროვება. რკალების ჯერადობა ნატურალური რიცხვით გამოისახება.

სისტემური პეტრის ქსელები (System Petri Nets) მაღალი დონის პეტრის ქსელის ქვეკლასს წარმოადგენს. მათი სტანდარტული ნოტაციის თანახმად, სისტემური პეტრის ქსელებისთვის განისაზღვრება კონსტანტები, ცვლადები და ფუნქციები, რომელთა ერთობლიობას სისტემური პეტრის ქსელის სტრუქტურა ეწოდება, ხოლო გადასასვლელებისთვის განისაზღვრება გახსნის პირობა, რომელსაც „გადასასვლელის დამცავი ფუნქცია“ ეწოდება [13].

სისტემური  $\Sigma$  პეტრის ქსელის უნივერსუმია  $A$ , რომელშიც თითოეული  $p \in P_\Sigma$  პოზიციისთვის აფიქსირებს მდგომარეობათა  $A_p$  სიმრავლეს, რომელსაც  $A$ -ში  $p$ -ს დომენი ეწოდება. მაშინ

1.  $\Sigma$ -ს მდგომარეობა  $a$  თითოეული პოზიციისთვის განსაზღვრავს სიმრავლეს  $a(p) \subseteq A_p$ ;
2. კოქვათ  $t \in T_\Sigma$ . ქმედება  $m$  თითოეული მოსაზღვრე  $f = (p, t)$  ან  $f = (t, p)$  რკალისთვის განსაზღვრავს სიმრავლეს  $m(f) \subseteq A_p$ .

ეს ნიშნავს, რომ სისტემური ქსელების პოზიციებში ცვლადის მნიშვნელობათა სიმრავლის მოთავსება შეიძლება, რკალსაც ელემენტარული პეტრის ქსელებისგან განსხვავებით ერთზე მეტი მნიშვნელობის გატარება შეუძლია, რაც მაღალი დონის პეტერის ქსელებს ახასიათებს. სისტემური პეტრის ქსელების შინაარსი სტრუქტურების ცნებაზეა დაფუძნებული. აღნიშნულ საკითხების გამოყენებას დეტალურად ჩვენ მომდევნო თავში დავუბრუნდებით.

### **1.8. დაპროგრამების UML ტექნოლოგია NET პლატფორმის ბაზაზე**

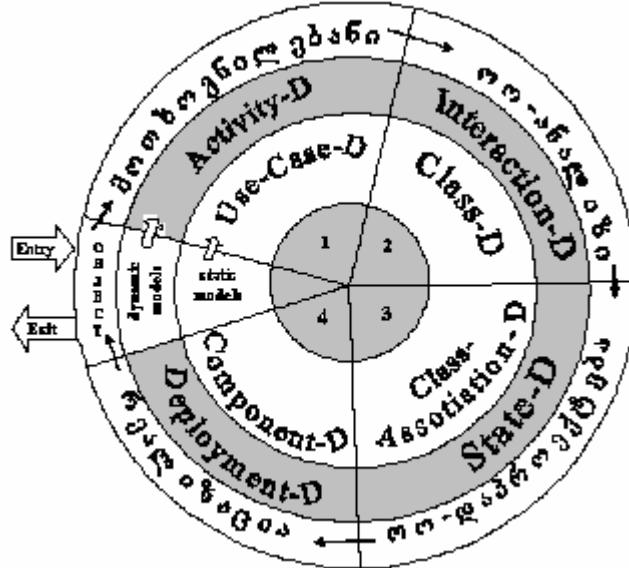
მაიკროსოფტის უახლესი პროგრამული .NET- პლატფორმა სულ უფრო ფართოდ იკიდებს ფეხს მსოფლიოს მოწინავე ქვეყნების საუნივერსიტეტო-სამეცნიერო და საწარმოო ფირმების ბიზნესის სფეროებში. იგი გამოიყენება Windows, Unix, და Linux ოპერაციული სისტემებისათვის. .NET-პლატფორმა შეიქმნა სპეციალურად განაწილებული გამოყენებითი სისტემების ასაგებად დიდი მოცულობის ინფორმაციის დასამუშავებლად კლიენტ-სერვერ არქიტექტურის ბაზაზე [15-20].

პროგრამული აპლიკაციების დაპროექტებისა და რეალიზაციის მეთოდოლოგია UML-ტექნოლოგიას ეყრდნობა. იგი გახდა საერთაშორისო სტანდარტი [2,17].

UML-ტექნოლოგია ობიექტ-ორიენტირებული მოდელირებისა და სტრუქტურული დაპროგრამების იდეოლოგიის მატარებელია, რომელიც თეორიული და პრაქტიკული ინფორმატიკის სამი ძირითადი მიმართულების (დაპროექტების ავტომატიზაცია, დაპროგრამების ავტომატიზაცია, მონაცემთა ბაზების დამუშავების ავტომატიზაცია) „გენერიკური“ მემკვიდრეა. UML არის პროგრამული მოდულების სპეციფიკაციის, კონსტრუირების, ვიზუალიზებისა და დოკუმენტირების ენა და აღნიშვნათა სისტემა. 1997 წლიდან Object Management Group-ს (OMG) მიერ იგი გამოცხადდა როგორც პროგრამული სისტემების აგების სტანდარტი.

UML არაა კონკრეტული მეთოდი. იგი არის ბაზა სხვადასხვა მეთოდებისათვის, გინაიდან მასში შემუშავებულია მოდელირების

კონსტრუქციების განსაზღვრული სიმრავლე აღწერის ერთიანი სისტემითა და სემანტიკით. 1.12 ნახაზზე. მოცემულია პროგრამული აპლიკაციების შექმნის 4 ეტაპი:

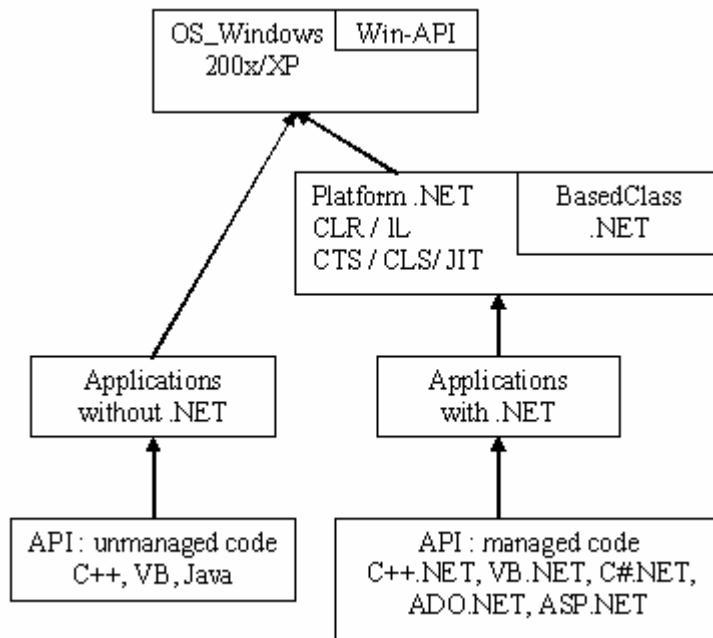


ნახ.1.12. UML-ტექნოლოგიის ეტაპები

საკვლევი ობიექტის ავტომატიზაციის მოთხოვნილებების დადგენა, მისი ობიექტ-ორიენტირებული (ოო) ანალიზი, ოოდაპროექტება (დეტალური დონე) და რეალიზაცია (პროგრამული კოდი). ეს ეტაპები სტატიკური და დინამიკური დიაგრამებით (D) ხორციელდება [20].

NET-პლატფორმა, რომელიც Visual Studio .NET 200\*- პაკეტებითაა გავრცელებული, არის გამოყენებითი პროგრამული დანართების ინტერფეისის (API - Application Programming Interface) სამუშაო გარემო, რომელიც შეთანხმებულად ფუნქციონირებს Windows-ოპერაციულ სისტემასთან [15].

1.13 ნახაზიდან ჩანს, რომ Windows-სისტემა უშუალოდ მუშაობს C++, VB, Java და სხვა ენებზე დაწერილ პროგრამულ API-დანართებთან, რომლებიც რეალიზებულია როგორც უმართავი კოდები (unmanaged code). ამასთანავე იგი მუშაობს C#.NET, C++.NET, VB.NET და ა.შ., ზოგადად .NET-პლატფორმის მიერ მართვად (managed code) პროგრამულ დანართებთან.



**ნახ.1.13. Windows და .NET-პლატფორმათა ურთიერთმოქმედება**

მართვაში იგულისხმება ის, რომ ეს კოდები ამჟავდება უშუალოდ .NET-ის მიერ, იმართება მათი პროცესებისა და მონაცემთა ნაკადები, მიეწოდება შესასრულებლად საჭირო დამხმარე რესურსები და ა.შ. პრინციპი, .NET-პლატფორმა ასრულებს „ოპერაციული სისტემის“ გარკვეულ ფუნქციებს და მოქნილად ფუნქციონირებს Windows-თან.

.NET-ის ქვებლოკი Based Class.NET არის ამ პლატფორმის საბაზო კლასების ბიბლიოთეკა (უმრავლესობა დაწერილია C#-ენაზე). იგი სრულად ობიექტ-ორიენტირებულია, შედგება ობიექტთა ერთობლიობისგან, რომელთაგანაც თითოეულ მი რეალიზებულია განსაზღვრულ მეთოდთა ჯგუფები. მაგალითად, ფანჯრებისა და ფორმების ასახვა (Windows GUI), მონაცემთა ფაილებთან ურთიერთობა (ADO.NET), ვებ-გვერდების ორგანიზება და ინტერნეტთან კავშირი (ASP.NET) და სხვ.

ამავე ბლოკში ნაჩვენებია .NET-runtime - პლატფორმის სამუშაო გარემო (რომელშიც სრულდება პროგრამა), ანუ CLR (Common Language Runtime) და მას შესრულების საერთო გარემოსაც უწოდებენ. ესაა პროგრამული უზრუნველყოფა მომხმარებელთა გამოყენებითი პროგრამების შესასრულებლად.

CTS საერთო ტიპების სისტემა (Common Type System), რომლის საფუძველზეც NET-პლატფორმა უზრუნველყოფს დაპროგრამების სხვადასხვა ენის თავსებადობას. ამასთანავე CTS აღწერს მომხმარებელთა კლასების განსაზღვრის წესებსაც.

IL შუალედური გარდაქმნის ენა (Intermediate Language). პროგრამები, რომელთა საწყისი კოდები დაწერილია, მაგალითად C#, C++, J++ ან VB ენებზე .NET-ში, კომპილატორი ამ მართვად კოდებს გადაიყვანს შუალედურ IL-ენაზე, რომელთაც შემდეგ CTS სწრაფად აკომპილირებს მანქანურ კოდში. ამგვარად, ობიექტური კოდები IL-ენის საშუალებით ისე მიიღება, რომ მათში არაა დაფიქსირებული, თუ რომელ ენაზე დაწერილი საწყისი კოდი.

CLS ენის საერთო სპეციფიკაცია (Common Language Specification), ანუ იმ სტანდარტების მინიმალური ერთობლიობა, რომელიც უზრუნველყოფს კოდებთან მიმართვას .NET-ის ნებისმიერი ენიდან. ამ ენების ყველა კომპილატორს გააჩნია CLS მხარდაჭერა.

JIT (Just-In-Time) ესაა შუალედური კოდის კომპილაციის ფაზა მანქანურ კოდში. სახელწოდება მიუთითებს იმაზე, რომ კოდის მხოლოდ იმ ცალკეული ნაწილების კომპილაცია ხდება, რომლებიც საჭიროა პროგრამის შესასრულებლად დროის მოცემულ მომენტში.

## 1.9. პირველი თავის დასკვნები

1. გაანალიზებულია კორპორაციული მართვის სისტემებში ბიზნეს-პროგრამების დამუშავებისა და შეფასების არსებული მეთოდები, ჩატარებულია მათი კლასიფიკაცია და გამოვლენილია არსებული პრობლემები და ამოცანები;

2. გამოკვეთილია ექსერტულ შეფასებათა გადაწყვეტილების მიღების მხარდაჭერი სისტემების უპირატესობა მონაცემთა საცავისა და მისი ოპერატორი ონლაინური ანალიზის პროცედურების ბაზაზე;

3. გამოკვლეულია საბანკო სისტემების თანამედროვე კომუნიკაციები და საინფორმაციო ტექნოლოგიები ბიზნეს-პროგრამების შეფასების თვალსაზრისით;

4. შემოთავაზებულია ასეთი პროცესების მოდელირებისათვის UML ტექნოლოგის და რეალიზაციისთვის .NET პლატფორმის გამოყენება. დისერტაციის მე-3 თავში ნაჩვენები იქნება დამუშავებული სისტემის სადემონსტრაციო ვერსიის რეალიზაციის ფრაგმენტები.

## II თავი. შედეგები და მათი განსჯა:

ბიზნეს-პროცესის მართვის ხელშემოყვარები კომაიუტერული  
სისტემის აბების მეთოდებისა და მოდელების შემუშავება

### 2.1. ბიზნეს-პროგრამების ექსპერტულ შეფასებათა ჯგუფური გადაწყვეტილების მიღების პროცესების მოდელირება UML დიაგრამებით

როგორც დისერტაციის 1-ელ თაგში აღნიშნეთ, კორპორაციული  
მართვის ობიექტებზე, განსაკუთრებით მცირე და საშუალო ბიზნესის  
ფირმებში, მაღალია „გაკოტრების რისკის“ სინდრომი მოულოდნელი,  
შემთხვევითი მოვლენებიდან გამომდინარე, თუ არასტაბილური,  
ეკონომიკურ-პოლიტიკური ვითარებით გამოწვეული. ფირმის  
ხელმძღვანელი, შეძლებისდაგვარად ყოველთვის მზად უნდა იყოს  
ოპერატიული მანევრირებისათვის, რათა სწორი ტაქტიკური ქმედებებით  
შეძლოს სტრატეგიული მიზნების მიღწევა [22,53].

ფირმის კრიზისულ სიტუაციაში ყოფნისას მნიშვნელოვანი  
ყურადღება უნდა მიექცეს მმართველობითი (მეცნიერული)  
კონსულტირების ფორმას, რომელსაც ფართოდ იყენებენ  
საზღვარგარეთის განვითარებულ თუ განვითარებად ქვეყნებში [1].  
გამოცდილი ექსპერტ-კონსულტანტები მეტად ძვირადდირებული  
სპეციალისტები არიან, რომელთაც გააჩნიათ არა მხოლოდ თეორიული  
განათლება და პრაქტიკული გამოცდილება, არამედ ინტუიცია და  
სწორი პროგნოზირების უნარი, რაც მეტად მნიშვნელოვანია კრიტიკულ  
სიტუაციაში.

ამგვარად, კონსულტანტები და ფირმის გამოცდილი  
სპეციალისტები, როგორც საპრობლემო სფეროს ექსპერტები,  
ერთობლივი ძალისხმევით, კონსულტაციებითა და კომპრომისებით  
ცდილობები მიიღონ ოპტიმალური გადაწყვეტილებანი ფირმის  
განვითარების სწორი კორპორაციული გეგმებისა და ტაქტიკურ  
ღონისძიებათა შესახებ.

დიდი ინფორმაციული ნაკადებისა და მათი დამუშავების მცირე  
დროის პირობებში (ან არასრული ინფორმაციისას) განსაკუთრებული  
მნიშვნელობა აქვს საინფორმაციო ტექნოლოგიების გამოყენებას. ჩვენი  
ნაშრომის მიზანი ექსპერტულ შეფასებათა ავტომატიზებული

დამუშავების პროგრამული პაკეტების შექმნაა, რომელიც ობიექტ-ორიენტირებული მეთოდების საფუძველზე უნდა აიგოს. განსაკუთრებით მნიშვნელოვანია აქ უნიფიცირებული მოდელირების ენის (UML-Unified Modeling Language) ტექნოლოგიის გამოყენება [3]. რთული პროგრამული პაკეტების დაპროექტებისა და რეალიზაციის CASE-მეთოდებით შესაძლებელია ჯგუფური გადაწყვეტილების მიღების მხარდამჭერი ავტომატიზებული სისტემების დამუშავება, რაც განსაკუთრებით აქტუალური ამოცანაა.

მართვის პროცესისთვის ჩვენ უნდა გამოვიყენოთ ექსპერტულ შეფასებათა მეთოდებით მოპოვებული ინფორმაციები და გადავამუშავოთ ისინი კომპიუტერის გამოყენებით. ეს გადამუშავება კი გულისხმობს სპეციალური პროგრამული პაკეტის ("Expert\_UML") დამუშავებას, ანუ უნდა შეიქმნას კომპიუტერული დიალოგური სისტემა ახალი, UML -ტექნოლოგიის საფუძველზე.

ჩვენს სისტემაში აქტიური მომხმარებლები იქნებიან ექსპერტები (საწარმოო ფირმის, კომერციული ბანკის ან სხვ. ხელმძღვანელები და მთავარი სპეციალისტები) და ერთი კონსულტანტი, რომელიც მეთოდურად ხელმძღვანელობს ექსპერტებს და წარმართავს როგორც მათი განსწავლის პროცესს, ასევე ინფორმაციის შეფასებას და გადამუშავებას. უნდა შეიქმნას კონსულტანტის ავტომატიზებული სამუშაო ადგილი. კომპიუტერში ჩაიდება მონაცემთა ბაზა ფირმის ხელმძღვანელებისა და სპეციალისტების წინადადებების ამსახველი, შეიქმნება პროგრამები, რომლებიც შეაჯერებენ ექსპერტულ შეფასებათა შედეგებს და გამოიმუშავებენ კოლექტიურ გადაწყვეტილებებს. მოხდება ამ გადაწყვეტილებათა ცხოვრებაში გატარების დონისძიებების შემუშავება და მათი დანახარჯების ანალიზი.

ამგვარად, ფირმის ხელმძღვანელობა და მთავარი სპეციალისტები თვითონ არიან ექსპერტები, თვითონ ეხმარებიან კონსულტანტს საბოლოო გადაწყვეტილებების მოქმედნაში. კონსულტანტს აქვს მეთოდიკა, თუ როგორ წარმართავს მუშაობას ექსპერტებთან. ეს მეთოდიკა გულისხმობს ობიექტის კვლევისათვის სისტემური მოდელირებისა და ექსპერტულ შეფასებათა მეთოდების გამოყენებას.

ექსპერტულ შეფასებათა ავტომატიზებული დამუშავების სისტემის ძირითადი მიზანია საწარმოო ფირმებისა ან ორგანიზაციების სტრატეგიული განვითარების გეგმების (კორპორაციული დაგეგმვა) შედგენის პროცესის ავტომატიზაცია.

ექსპერტულ შეფასებათა სხვადასხვა მეთოდი და ხერხი არსებობს. დღეისათვის ცნობილია მაგალითად, „გონიერივი შეტევის“ მეთოდი, რომელიც ემყარება ახალი იდეის ჩამოყალიბებას (რაღაც საკითხის გადასაწყვეტად) ყოველგვარი მეცნიერული დასაბუთების გარეშე. „ჯგუფური მეთოდები“ განსხვავდება მათი გამოყენების ხერხების მიხედვით, მაგალითად, „შეფასებათა შეთანხმების მეთოდი“, რომელიც ინდივიდუალურ მიდგომას ეყრდნობა: ყოველი ექსპერტი იძლევა საკუთარ შეფასებას (სხვისგან დამოუკიდებლად), ხოლო შემდეგ რომელიმე ხერხის მიხედვით ეს შეფასებები ერთიანდება ერთ განზოგადებულში ანუ „შეთანხმებულში“. „ჯგუფურ მეთოდს“ მიეკუთვნება ექსპერტების „ერთობლივი მუშაობის“ ხერხი, რომლის საფუძველზე მიიღება ჯამური შეფასება მთლიანი ჯგუფის მიერ. მესამე ხერხს ჯგუფურ მეთოდში წარმოადგენს ე.წ. „დელფის“ მეთოდი – ესაა ინდივიდუალურ შეფასებათა შედეგების თანმიმდევრობითი გაცნობა თითოეული ექსპერტის მიერ [21].

მეცნიერმა-პროგნოზისტებმა განსახილველ აზრთა კოლექტიური განსჯისა და შეთანხმების ხერხის საფუძველზე შექმნეს თანამედროვე ინსტრუმენტი – „დელფის“ მეთოდი, მეცნიერებისა და ტექნიკის განვითარების პერსპექტივების შეფასებისათვის.

„დელფის“ მეთოდის ძირითადი პრინციპები მდგომარეობს ექსპერტების ანონიმურ გამოკითხვაში მათი ურთიერთზეგავლენის მოხდენის გამორცხვის მიზნით. უკუკავშირი ექსპერტების ჯგუფის მართვის პროცესის განსახორციელებლად წარმოებს მათი შეფასებების გადამუშავების შედეგად მიღებული ინფორმაციის ერთად განხილვის საშუალებით. პროცესს უნდა წარმართავდეს კონსულტანტ-ანალიტიკოსი. ამ მეთოდის მიზანს არ შეადგენს ექსპერტთა შეხედულების სრული დამთხვევა. მომდევნო ეტაპის გამოკითხვის შედეგების ანალიზი უჩვენებს მათი შეხედულებების დაახლოებაზე,

მაგრამ, არა სრულ დამთხვევაზე, გამოკითხვა რამდენიმე ტურად მიმდინარეობს.

როგორც დავინახეთ „დელფის“ მეთოდი შედგება: გამოკითხვის-შედეგების ანალიზის – დისკუსიის-გამოკითხვის – . . . და ა.შ. ციკლებისაგან. იმისათვის, რომ შეფასდეს კოლექტიური აზრის მაჩვენებელი, საჭიროა დადგინდეს მათემატიკური (სტატისტიკურ-ალბათური) მეთოდების საფუძველზე მათი მნიშვნელობები. მაგალითად, თუ მოცემულია დაჯგუფებული ექსპერტული შეფასებები  $x_1, x_2, \dots, x_n$ , მაშინ უმარტივესი ხერხი განხოგადებული შეფასებებისა, როგორც ცნობილია მდგომარეობს მათი არითმეტიკული საშუალოს მოძებნაში:

$$\bar{x} = \sum_{i=1}^n x_i / n$$

ხშირად ყოველ ექსპერტულ შეფასებას მიეწერება განსაზღვრული წონა ( $v_i$ ), მათი მნიშვნელობის მიხედვით. ასეთ დროს შესაძლებელია გავიანგარიშოთ შეწონილი საშუალო არითმეტიკული:

$$\bar{x} = \sum_{i=1}^n x_i v_i / \sum_{i=1}^n v_i$$

თუ ექსპერტული შეფასებები წინასწარ დაჯგუფებულია (მოწესრიგებულია) რიგში, რომელსაც აქვს  $M$  ინტერვალი, მაშინ გამოსათვლელად იყენებენ ფორმულას:

$$\bar{x} = \sum_{i=1}^n x_i f_i / n$$

სადაც  $M$  არის ინტერვალების რაოდენობა;  $f_i$ -შეფასებათა რაოდენობა  $i$ -ერ ინტერვალში. ხშირ შემთხვევაში, ექსპერტთა შეფასებებში ადგილი აქვს „ძალიან დიდ“ ან „ძალიან მცირე“ მნიშვნელობებს, რომლებიც გავლენას ახდენს საშუალო არითმეტიკულზე და, ეს უკანასკნელი, უკვე ადარ შეიძლება ჩაითვალოს იმ შედარებით ზუსტ მონაცემად, რომელიც სასურველი იყო მისაღებად. ასეთ დროს იყენებენ მედიანის განსაზღვრის მექანიზმს. მედიანა წარმოადგენს მოწესრიგებულ (ზრდადობით ან კლებადობით) შეფასებებში შუაში შეფასების მნიშვნელობას. თუ შეფასებათა რიცხვი კენტია, მაგალითად,  $2*n + 1$ , მაშინ  $(n+1)$ -ე წევრის მნიშვნელობა იქნება

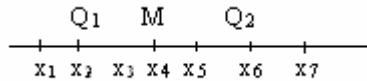
მედიანა; თუ შეფასებათა რიცხვი ლურჯია  $2^*n$ , მაშინ მედიანად მიიღება  $n$  და  $n+1$  წევრების მნიშვნელობათა საშუალო არითმეტიკული.

ექსპერტულ შეფასებათა ანალიზის დროს იყენებენ საშუალო შეფასებების ირგვლივ შეფასებათა მნიშვნელობების ვარიაციას. რაც ნაკლებია შეფასებების გაბნევა საშუალოდან, მით უფრო ზუსტად ასახავს ეს საშუალო შეფასება ჯგუფურ აზრს.

გაბნევის ამპლიტუდა გამოითვლება შეფასებათა მაქსიმალური და მინიმალური მნიშვნელობების სხვაობით.

$$R = x_{\max} - x_{\min}$$

ხშირად შეფასებათა მოსაწესრიგებლად სიმკრივისათვის გაითვლიან კვარტილებს ( $Q_1, Q_2, Q_3$  და ა.შ. ნახ. 2.1):



### ნახ. 2.1. ექსპერტულ შეფასებათა კვარტილების დერძი

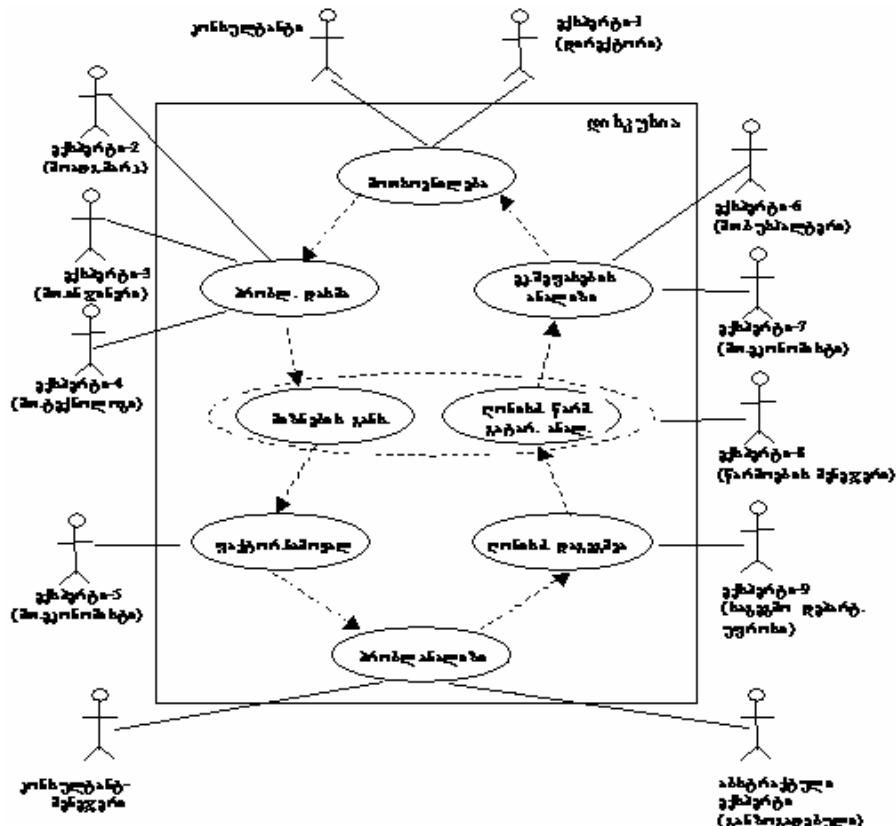
მაგალითად, თუ გვაქვს 7 ექსპერტული შეფასება, ისინი უნდა დალაგდეს მონოტონურად. შემდეგ საჭიროა ვიპოვოთ მედიანა (M). ჩვენს შემთხვევაში მედიანა ემთხვევა  $x_4$  შეფასებას. ახლა განვსაზღვროთ ქვედა და ზედა კვარტილები ( $Q_1$ ,  $Q_2$ ). მედიანა და კვარტილები ქმნის ოთხ ინტერვალს:  $< Q_1, Q_1M, MQ_2 \text{ და } Q_2 >$  ამათგან  $Q_1M$  და  $MQ_2$  ჩაითვლება მისაღებად, როგორც კოლექტიური აზრის თანმთხვევი. ექსპერტებმა, რომელთა შეფასებები არ მოთავსდა ( $Q_1, Q_2$ ) დიაპაზონში, უნდა დაასაბუთონ მიზეზები მათი აზრების კოლექტივისაგან განსაკუთრებული განსხვავებისა. ამ დასაბუთებებს და დასკვნებს (მათი ავტორების ვინაობის გაუმნებლად) გააცნობენ დანარჩენ ექსპერტებს.

„დელფის“ მეთოდი საშუალებას იძლევა მიღებულ იქნას ჯგუფური აზრის შედარებით საიმედო შეფასებები, ვიდრე უბრალოდ, მათ შეფასებათა გასაშუალების დროს. ნაკლად ითვლება ის ფაქტი, რომ მთლიანად ვერ აღმოიფხვრება ექსპერტთა ურთიერთზეგავლენა.

გარდა ზემოგანხილულისა, ცნობილია აგრეთვე სხვა ექსპერტული მეთოდებიც, როგორიცაა მაგალითად, დელფისა და

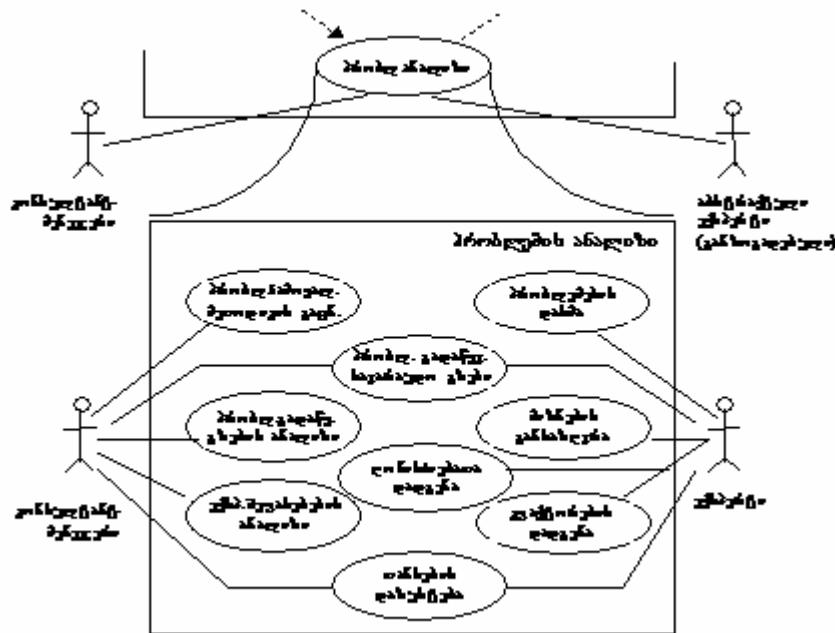
ქსელური გრაფიკის განზოგადოებისა და პერსპექტიული დაგეგმვის სელექციური მეთოდი, რომელსაც საფუძვლად უძვეს მიზნობრივი მიღებობა. აღნიშნული მეთოდის მსგავსია აგრეთვე პროგნოზირების ე.წ. „შეწონილ შეფასებათა“ მეთოდი, რომელიც იყენებს ასევე მიზნის ხეს და მატრიცებს. ცნობილია აგრეთვე პატერნ-მეთოდი, რომელიც ანალოგიურია მიზნობრივი სტრუქტურის გრაფის გამოყენებისა ა.შ. [21].

აღწერილი მეთოდების ალგორითმიზაცია და პროგრამული რეალიზაცია UML-ტექნოლოგიით ხორციელდება. მის პირველ ეტაპზე ჩვენ განვსაზღვრეთ ბიზნეს-პროცესების შინაარსს (Actions) და მათ შემსრულებლებს (Actors). ასეთი დიაგრამის ფრაგმენტი (Use Case Diagram) 2.2 ნახაზზეა მოცემული.



ნახ.2.2. ექსპერტულ შეფასებათა პროცესის Use Case დიაგრამა

აქ თვითონული პრეცედენტი (ოფალი) შეიძლება გაიშალოს დეტალურ დონეზე რამდენიმე ფუნქციის სახით. მაგალითად, 2.3 ნახაზზე ნაჩვენებია „პრობლემის ანალიზის“ შემთხვევა, სადაც ექსპერტ-კონსულტანტის როლების ტანდემისთვის მოცემულია შესასრულებელ ფუნქციათა ერთობლიობა.



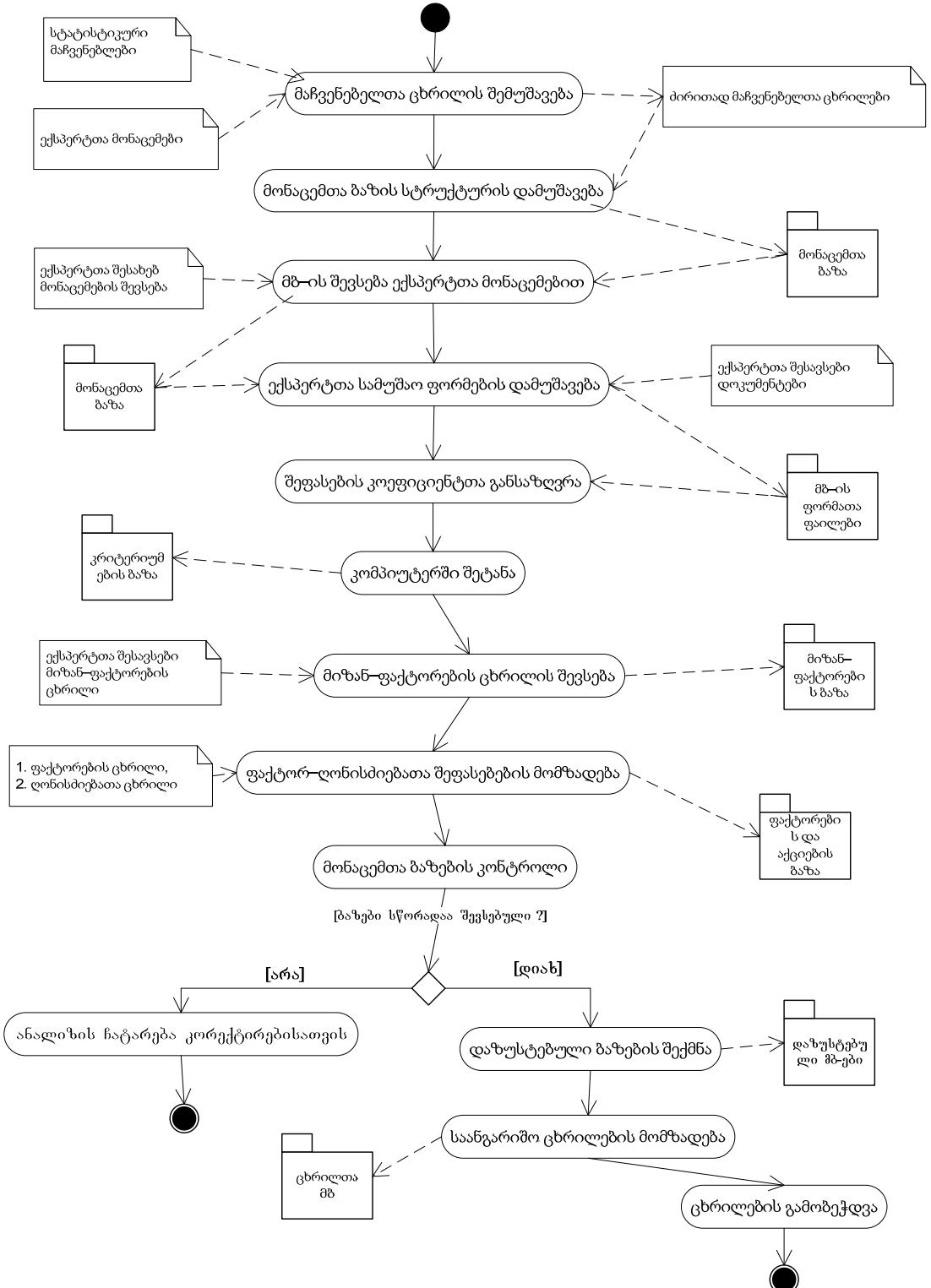
### **ნახ.2.3. „პრობლემის ანალიზის“ დეტალიზაციის დონე**

ნახაზზე მოცემულ დიაგრამასთან ერთად საჭიროა განისაზღვროს თითოეული გამოყენებითი შემთხვევის მოვლენები (Events) და სცენარები. მაგალითად, „პრობლემის ანალიზი“ გამოყენებითი შემთხვევის ძირითად მოვლენათა ნაკადი შეიძლება იყოს შემდეგი:

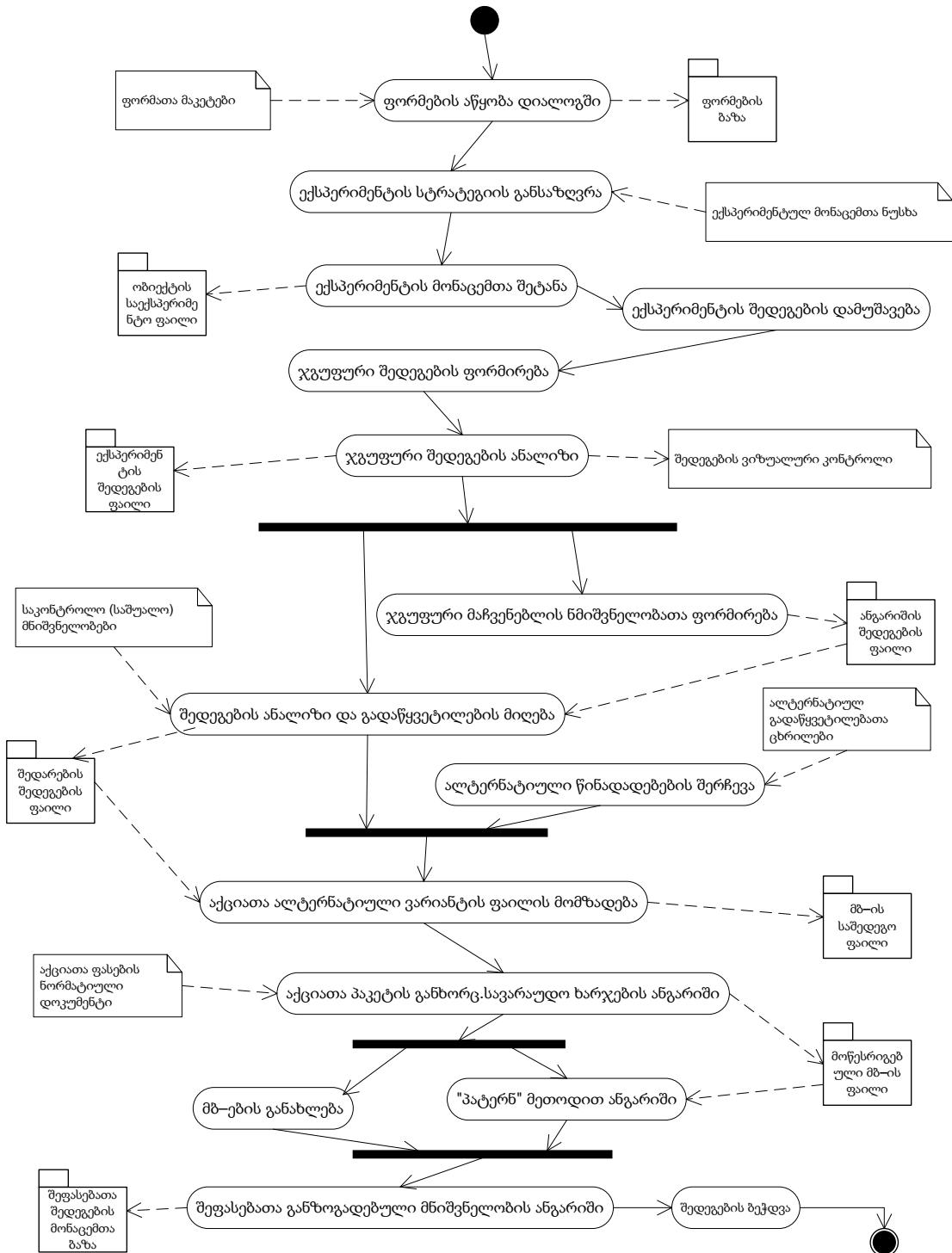
1. კონსულტანტი გახსნის ექსპერტთა მონაცემთა ბაზას და მათ ცხრილ-ფორმებს, დააფიქსირებს ექსპერტების საიდენტიფიკაციო ნომრებს;
  2. ექსპერტები თავიანთი ინდივიდუალური პაროლებით გახსნიან სამუშაო ცხრილებს და დააფიქსირებენ თავიანთ შეფასებებს;
  3. სისტემა გააერთიანებს ყველა ექსპერტის მონაცემებს ერთ საერთო ცხრილ-ბაზაში;
  4. სისტემა ჩაატარებს მონაცემების ანალიტიკურ-სტატისტიკურ დამუშავებას;
  5. სისტემა დააფიქსირებს რადიკალურად მოაზროვნე ექსპერტებს ანუ „ექსტრემისტებს“;
  6. კონსულტანტი შეადგენს მოქმედების გეგმას;
  7. სისტემა ამზადებს პრინტერზე ექსპერტთა მომდევნო ეტაპზე მუშაობის ცხრილებს;

8. განმეორებითი დისკუსიის შემდეგ ექსპერტები აფიქსირებენ ცხრილებში თავიანთ შეფასებებს;
9. კონსულტანტი ყველაფერს ინახავს მონაცემთა ბაზაში;
10. სისტემა განაახლებს ანალიზის შედეგებს;
11. კონსულტანტი სისტემის ანალიზური შედეგების საფუძველზე გადაწყვეტს დაბრუნდეს მე-5 მოვლენაზე თუ გადავიდეს ექსპერტულ შეფასებათა შემდეგ ეტაპზე;
12. სისტემა ანგარიშობს ექსპერტთა შეფასებების საშუალო მნიშვნელობებს;
13. კონსულტანტი აგებს საერთო (გასაშუალებული) შეხედულებათა მიზნის სტრუქტურულ გრაფს;
14. სისტემა ქმნის მიზნის სტრუქტურული გრაფის შესაბამის მატრიცას;
15. სისტემა ანგარიშობს „პატერნის“ მეთოდით დონისძიებათა დომინირებადობას მიზანზე;
16. კონსულტანტს შეაქვს თითოეულ დონისძიების შესრულების ფასები;
17. სისტემა ამ ფასების გათვალისწინებით მოაწესრიგებს დონისძიებებს;
18. სისტემა ბეჭდავს საბოლოო შედეგებს;
19. კონსულტანტი ამზადებს დასკვნით ანგარიშს;
20. კონსულტანტი და ექსპერტები დასკვნით შეხვედრაზე განიხილავენ საბოლოო შედეგებს;
21. დასკვნით დოკუმენტს მოეწერება ხელი ექსპერტების მიერ.

გამოყენებით შემთხვევათა დიაგრამების აგების შემდეგ განიხილება ბიზნეს-პროცესის ექსპერტულ შეფასებათა მსარდამჭერი კომპიუტერული სისტემის აგების ტექნოლოგიური პროცესის „AD4Expert“ აქტიურობათა დიაგრამა. იგი აგებულია Visual Studio.NET გარემოში, Ms Visio პაკეტით (ნახ.2.4).

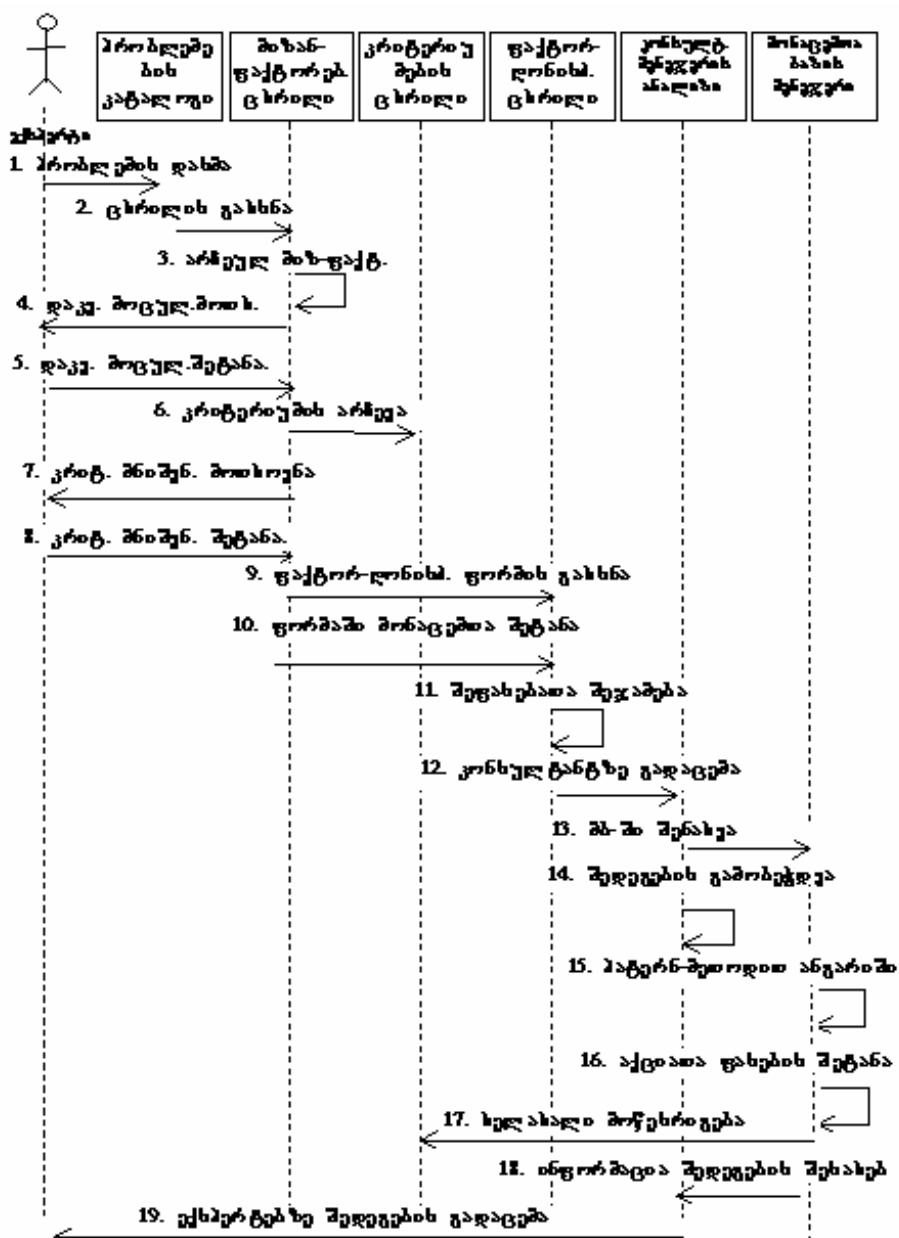


**ნახ.2.4. ექსპერტულ შეფასებათა მხარდაჭერი სისტემის აგების ტექნილოგირი პროცესის აქტიურობის დიაგრამა**

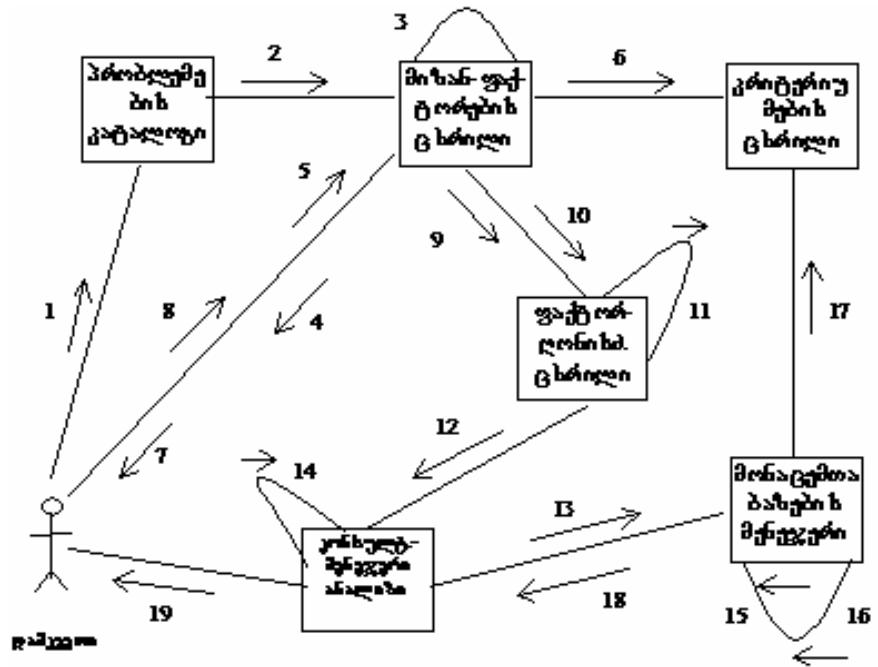


#### ნახ.2.4-ის გაგრძელება

სისტემის კლასთა ობიექტებს შორის ურთიერთოქმედების (Interaction) დიაგრამა. ამ მიზნით აიგება ორი სახის დიაგრამები: მიმდევრობითობის (Sequence) და თანამოქმედების (Collaboration). მათ საფუძველზე ხორციელდება ობიექტებს შორის შეტყობინებების გაცვლა დროის მიხედვით და შესაბამისი მეთოდების ამუშავება ინფორმაციული ნაკადების დასამუშავებლად და გადასაცემად [4]. 2.5 და 2.6 ნახაზებზე ნაჩვენებია მიმდევრობითობისა და თანამოქმედების დიაგრამათა ფრაგმენტები.



ნახ.2.5. მიმდევრობითობის დიაგრამის ფრაგმენტი



ნახ.2.6. თანამოქმედების დიაგრამის ფრაგმენტი

□ კლასი , → შეტყობინება

ჩვენი ამოცანის გადასაწყვეტად, ანუ UML-დიაგრამების ასაგებად გამოვიყენეთ მაიკროსოფტის ფირმის პროგრამული პაკეტი Ms Visio, რომელზეც დავაპროექტეთ აგრეთვე მონაცემთა ბაზის ობიექტური მოდელი და მისი შესაბამისი ლოგიკური სტრუქტურა. აღნიშნული პროცედურები იყენებს კომპონენტურ-ვიზუალურ ელემენტებს, რაც საშუალებს იძლევა ავტომატიზებულ რეჟიმში სწრაფად ავაგოთ სისტემის პროგრამული რეალიზაციის კოდი [15].

## 2.2. მონაცემთა ბაზაში ექსპერტთა ცოდნის ასახვის ობიექტ-როლური მოდელების შემუშავება კატეგორიალური მიდგომით

მოცემულ პარაგრაფში განიხილება განაწილებული ორგანიზაციული მართვის სისტემების უნიფიცირებული ბიზნეს-პროცესების (საქმის-წარმოების) დაპროექტებისა და პროგრამული რეალიზაციის საკითხები, თანამედროვე ობიექტ-ორიენტირებული მოდელირების მეთოდებისა და კლიენტ-სერვერული არქიტექტურის ბაზაზე [39,74-77]. ძირითადად ყურადღება გამახვილებულია შემდეგი საკითხების გადაწყვეტაზე:

- ბიზნეს-ობიექტის სტრუქტურის განსაზღვრა და მისი ფუნქციონირების წესების აღწერა არაფორმალიზებული მეთოდებით (ბუნებრივი სალაპარაკო ენის საშუალებით);

- ობიექტ-როლური მოდელების (ORM) ერთობლიობის განსაზღვრა ფორმალური გრამატიკის აპარატისა და ლოგოგურ-ალგებრული მეთოდების საფუძველზე [23];

- კონცეპტუალური მოდელის აგება კატეგორიალური მიდგომისა და ობიექტ-ორიენტირებული კლასთაშორისი დიაგრამების გამოყენებით [25];

- განაწილებული ბიზნეს-ობიექტების მონაცემთა ბაზის სტრუქტურის ფიზიკური რეალიზაცია კლიენტ-სერვერული არქიტექტურით და მომხმარებელთა ინტერფეისების დამუშავება;

აქ დასმული და გამოკვლეულია განაწილებული ბიზნეს-ობიექტების მონაცემთა ბაზების ავტომატიზებული დაპროექტების პროცესების ობიექტ-ორიენტირებული მოდელირებისა და ობიექტ-ორიენტირებული დაპროექტების დამუშავების ამოცანა, რაც მნიშვნელოვნად ამცირებს ინფორმაციული და პროგრამული უზრუნველყოფების შექმნის დროს და ამასათანავე, ორიენტირებულია გამოყენებითი სფეროს მომხმარებელზე.

ცოდნა, რომელიც აქვს საპრობლემო სფეროს მომხმარებელს, სპეციალური ინტერფეისების საშუალებით, რომელთა საფუძველია ფორმალურ გრამატიკათა აპარატის კატეგორიები და ლოგიკურ-ალგებრული მეთოდები, გადაუცემა ობიექტურ-როლური მოდელირების კომპიუტერულ პროგრამას. ამ ინსტრუმენტის დახმარებით აიგება მისი სემანტიკური სტრუქტურა ORM-დიაგრამების სახით. მომდევნო ეტაპზე, ავტომატიზებული პროცედურების გამოყენებით ფორმირდება საპრობლემო სფეროს კონცეპტუალური მოდელი, ანუ ER-დიაგრამები, შესაბამისი ცხრილებითა და ატრიბუტებით.

ახლა განვიხილოთ ობიექტ-ორიენტირებული ER-სქემის აგების მაგალითი კატეგორიალური მიდგომის გამოყენებით მონაცემთა ბაზებში [25,26].

კატეგორიები რომლებსაც აქ ვიხილავთ, არის ლოგიკური გრამატიკის შემადგენელი ნაწილი, რომელიც, როგორც ზოგადი ენის

სამირკველი, ბეგრად მდიდარია, ვიდრე ტრადიციული პრედიკატების ლოგიკა და ომელიც ცხადია დამოუკიდებელი უნდა იყოს ბუნებრივი ენის ემპირიული გრამატიკისგან. პრედიკატების ლოგიკა არის ლოგიკური გრამატიკა დანართების შინაარსის ასაგებად, მაგრამ ძალზე დარიბია და ნაკლებად განვითარებული - იგი არ ფლობს სინტაქსურ კატეგორიებს, რათა განასხვავოს ერთმანეთისგან მთავარი და არამთავარი პრედიკატები.

სალაპარაკო ენის წინადადება „პოპული არის კომერციული ფირმა“ პრედიკატების ენაზე ჩაიწერება ასე:

### **პოპული ε ფირმა ^ პოპული ε კომერციული**

ε—აღნიშვნა პრედიკატების ლოგიკაშიც და ზოგად ლოგიკურ ენაშიც არის კოპულა (კავშირი), ანუ დამხმარე საშუალება ენაში ასახვისათვის, რათა პრედიკატი (შემასმენელი) შეუსაბამოს არსს (საგანს, ობიექტს).

პრედიკატიზაცია არის ასიმეტრიული. კოპულას (ε) მარცხნივ ზის საკუთარი სახელი (პოპული) ან მახასიათებელი (აღნიშვნა). მარჯვნივ ზის პრედიკატორი (არსის დასახელება – განსხვავებული გამოსახულება).

საკუთარი სახელები და მახასიათებლები არის ნომინატორები.

ამგვარად, კატეგორია შეიძლება ჩავწეროთ სამეულით:

**Cat = <Nom, P, K>, სადაც**

**Nom** - ნომინატორებია,

**P** - პრედიკატორები და

**K**- კოპულაა.

განსხვავება მთავარ და დამხმარე პრედიკატორებს შორის პრედიკატულ ლოგიკაში წაშლილია, ვინაიდან პრედიკატულ-ლოგიკური გამოსახულება შეიძლება გულისხმობდეს წინადადებას „პოპული არის ფირმული კომერცია“.

„პოპული ε (კომერციული) (ფირმა)“, სადაც კომერციული - დამხმარე და პოპული - 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: „პოპული ყიდის“ გვექნება „პოპული ე გამყიდველი“, ან დამხმარე პრედიკატორით: „პოპული ყიდის სწრაფად“ ვდებულობთ:

„პოპული ე სწრაფი ჲ პოპული ე გამყიდველი“.

აქტიური კომპონენტები უნდა განვასხვავოთ სუფთა ობიექტებისაგან (არსებისგან). გამოსახულებაში:

„პოპული ჲ გაყიდვა“ ან

„პოპული ჲ (სწრაფი)(გამყიდველი)“

მოქმედების კოპულით ჲ (კეთება). ნაჩვენებია, რომ მთავარი პრედიკატორის სახით გამოდის მოქმედების პრედიკატორი და არა ობიექტის პრედიკატორი.

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

რთული წინადადება, მაგალითად, რამდენიმე მოქმედების პრედიკატორით π-ს შემდეგ, მიიღება მარტივი წინადაგების ლოგიკური ნაწილაკით შეერთებით.

პრედიკატული ლოგიკა უნდა განვიხილოთ, როგორც ლოგიკური გრამატიკის გაფართოება. მაგალითად, წინადადება:

„გაჭარაძე იდებს პროდუქტს საწყობიდან“.

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

„მზიური = i საწყობი“, სადაც = არის განსაზღვრების ნიშანი, ხოლო **Jota**

(i) არის მახასიათებლის ნიშანი.

ელემენტარული წინადადება ჩაიწერება ასე:

ვაჭარაძე π (აღება) (პროდუქტი) (i საწყობი)

[ნომინატორი] [მოქმედების [ობიექტის [მახასიათებელი]  
პრედიკატორი] პრედიკატორი]

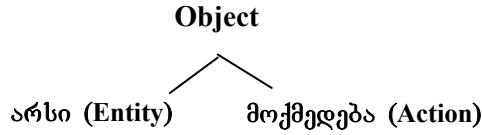
აქმდე განხილულ კატეგორიებს: ნომინატორი – პრედიკატორი; მთავარი პრედიკატორი – დამხმარე პრედიკატორი; ობიექტის პრედიკატორი – მოქმედების პრედიკატორი; არის კოპულა (ε) – კეთების კოპულა (π), ლოგიკურ გრამატიკაში ემატება სხვა მნიშვნელოვანი კატეგორიები, რათა შესაძლებელი იყოს კონცეპტუალური სქემის მეთოდურად აგება. ასეთ კატეგორიებს მიეკუთვნება უპირველეს ყოვლისა მსჯელობა ლოგიკური ენის დონეებზე, ანუ დიფერენცირება კონკრეტულ და აბსტრაქტულ დონეებზე და მათ ობიექტებზე (ტიპების ლოგიკა). შემოიტანება აგრეთვე კატეგორია „მთელი-ნაწილი“.

სქემების აგებისას დრო (ან დროის პუნქტი) არის ჩვეულებრივი პრედიკატორი, რომელიც „მანამ“, „შემდეგ“ და ა.შ. მიეთითება ორადგილიან პრედიკატორში „x, y ε before“ ან „x, y ε after“.

ამგვარად, შემოიტანება კიდევ შემდეგი კატეგორიები: აბსტრაქცია – კონკრეტიზაციის: (ენის სხვადასხვა დონეზე მისაღწევად); კომპოზიცია – დეკომპოზიციის: „მთელი - ნაწილი“; ობიექტების ენა – მეტაუნა: ობიექტების ენა ერთი ენაა (I დონე, მაგ., ინგლისური) რომლის შესახებაც საუბრობენ მეტაუნაზე (II დონე, მაგ., ქართული); სქემის დრო - მიმართვის (ათვლის) დრო.

სქემის დროსთვის იყენებენ ტერმინს „ინტენსიონალი“ (შინაარსი) 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). ამგვარად, ობიექტი ხასიათდება არსით და მოქმედებით (ქცევით – action).



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.

საბოლოოდ კატეგორია შეიძლება ჩავწეროთ როგორც

**Cat = < Nom, P, K , AC, CD, TP, LML, TIE, ERS>, სადაც**

**Nom** - ნომინატორები, **P** – პრედიკატორები,

**K = < ε, σ, π >** - „არის“, „აქვს“ და „პეოება“ კოპულებია.

**AC** - აბსტრაქტცია-კონკრეტიზაცია;

**CD** - კომპოზიცია – დეკომპოზიცია;

**TP** - მთელი - ნაწილი

**LML** - ენა – მეტაენა;

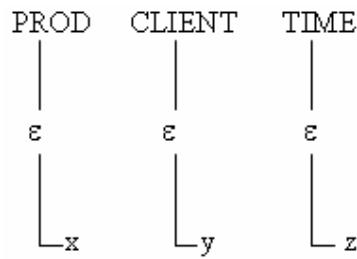
**TIE** - ინტენსიონალი- ექსტენსიონალი;

**ERS** - არსთა – დამოკიდებულებათა სქემა.

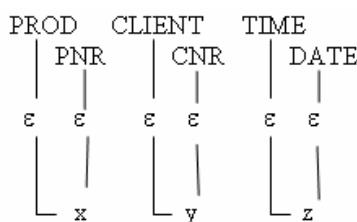
არსთა-დამოკიდებულების სქემის აგება საპრობლემო სფეროსთვის, ამ პროცესის პლევა და მისი სრულყოფა ეფექტური ალგორითმული სქემებით და პროგრამული გრაფო-ანალიზური ინსტრუმენტებით – არის ჩვენი ნაშრომის ერთ-ერთი მთავარი საკითხი.

კატეგორიალური მიდგომის საფუძველზე და ობიექტ-როლური მოდელირების ინსტრუმენტის გამოყენებით ახლა გადავიდეთ კონკრეტული ზემოაღწერილი ამოცანის გადაწყვეტის განხილვაზე.

პირველ რიგში მარტივ მაგალითზე ვაჩვენოთ, თუ როგორი სახე აქვს კატეგორიების საფუძველზე ჩაწერილ პრედიკატულ ფორმას გარკვეული სემანტიკური აზრის გამოსახატავად. მთავარი პრედიკატებია: პროდუქტი, კლიენტი, შეკვეთის-დრო (PROD, CLIENT, TIME) და ა.შ. მთავარ პრედიკატორებს საწყის ფაზაზე აქვს ბირთვის (საწყისის) მნიშვნელობა.



კატეგორიალური თვალსაზრისით, ჩვენ ვიმყოფებით სქემის-დროში (schma-time) და არა მიმართვის-დროში (referent-time). მაგალითად, „დოლიძე ე კლიენტი“ - ეს არაა განზოგადება, პირიქით „y ე კლიენტი“ არის y-ნომინატორული ცვლადით ყველა შემთხვევისთვის განზოგადებული. დოლიძე მხოლოდ კონკრეტულ მომენტი გააქტიურდება. ე. ი. y – ხდება დოლიძე. 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.



ახლა განვიხილოთ აბსტრაქცია, როგორც ენის ლოგიკური პროცესი. მაგალითად, ენაში სინონიმების არსებობა [CLIENT და CNR].

შემოიტანება „=>“ წესის ისარი (იმპლიკაციის მსგავსად, ან პროდუქციის წესი „თუ, მაშინ“), რომელიც ნიშნავს, რომ თუ მოცემულია მარცხენა მხარე (გამონათქვამი), მაშინ მარჯვენაც მართებულია:

$$\begin{aligned} y \in \text{CLIENT} &\Rightarrow y \in \text{CNR} \\ y \in \text{CNR} &\Rightarrow y \in \text{CLIENT}. \end{aligned}$$

მეორე მაგალითი: „**s ურეკავს კლიენტს**“, ჩაიწერება:  
**s π (დარეკვა)(CLIENT).**

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

**(CLIENT) ρ (CNR) →( A\_R(CLIENT )<→ A\_R(CNR )).**

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

**ρ**-აბსტრაქციის თვალსაზრისით არის ეკვივალენტობის რელაცია, რომლისთვისაც სამართლიანია რეფლექსურობის, სიმეტრიულობის და ტრანზიტულობის თვისებები.

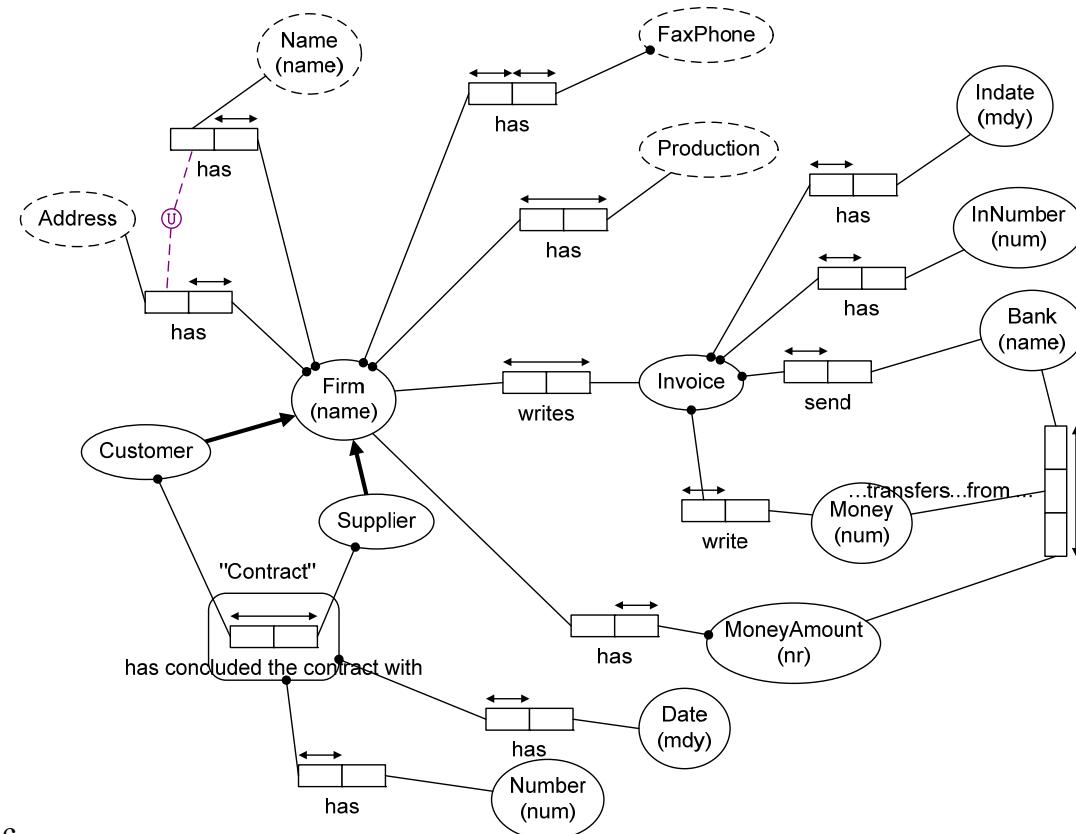
მიზანი ობიექტ-ორიენტირებული როლური მოდელების გამოყენებისა მდგომარეობს საპრობლემო სფეროს აღწერის მაქსიმალური სიზუსტის რეალიზაციაში. ყველაფერი, რაც ეხება ობიექტს, უნდა აღიწეროს როგორც არსთა-მოდელში, ასევე მის ყოფაქცევის-მოდელში. მსგავსი საკითხები განიხილება ფილოსოფიური დისკიპლინის ისეთ სფეროში, რომელიც დაპავშირებულია ენასა და შემეცნებასთან. მაგალითად, აქ შეიძლება გავიხსენით კანტის შრომები, რომელმაც პირველმა შემოიტანა განსხვავება სქემასა და მის გამოსახულებას შორის. ფრეგემ შემოიტანა აბტრაქტული თეორია ლოგიკური პრედიკატებისთვის, ხოლო რასელთან ერთად განსაზღვრებათა თეორია. კარნაპმა განავითარა ინტენსიონალ-ექსტენსიონალობის თემა და ა.შ. [26].

ობიექტ-როლური მოდელირება (ORM) – მონაცემთა ბაზის სტრუქტურების მოდელირებისა და ავტომატიზებული დაპროექტების მეთოდია კონცეპტუალირ დონეზე. ბუნებრივი ენის გრამატიკის გამოყენება, ისევე როგორც ინტუიციური დიაგრამებისა, რომელთა შევსება შესაძლებელია მაგალითებით, აგრეთვე ინფორმაციის გამოკვლევა, მარტივი ან ელემენტარული ფაქტების ტერმინებში, მნიშვნელოვნად ამარტივებს ავტომატიზებული დაპროექტების პროცესს. ORM განიხილავს საპრობლემო სფეროს როგორც ურთიერთ-

დაგავშირებულ ობიექტთა ერთობლიობას, რომლებიც თამაშობს გარკვეულ როლებს.

ზოგადი მაგალითისათვის აქ განიხილება ფირმა, რომლის ბიზნეს-პროცესები აღიწერება ელემენტარული ფაქტების დახმარებით, სადაც ასახულია ფირმის საქმიანობის კანონზომიერებანი. მაგალითად, „ფირმას, რომელიც არის შემკვეთი ან მიმწოდებელი, აქვს დასახელება, მისამართი, ფაქს-ტელეფონი, საბანკო-ანგარიში და ა.შ. ფირმა ახორციელებს პროდუქციის წარმოებას ან მომსახურებას (სერვისს). შემკვეთ და მიმწოდებელ ფირმებს შორის ფორმდება კონტრაქტები, რომელთაც აქვს კონტრაქტის–ნომერი, თარიღი. ყიდვა-გაყიდვა ფორმდება სპეციალური ანგარიშ-ფაქტურით, რომელსაც აქვს ნომერი, პროდუქციის (ან მომსახურების) დასახელება, მოცულობა, თანხა და თარიღი. კომერციული ბანკი გადარიცხავს შესაბამის თანხებს შემკვეთის ანგარიშიდან მიმწოდებელის ანგარიშზე, და ა.შ.“.

აღნიშნული პროცესის ORM-დიაგრამა, რომელიც აგებულია MsVisualStudio.NET+MsVisio ინსტრუმენტით, აქვს 2.7 ნახაზზე მოყვანილი სახე.



ნახ.2.7. ობიექტ-როლური მოდელის (ORM) ფრაგმენტი

დიაგრამაზე გამოსახულია ისეტი შეზღუდვებიც, რომლებიც დამახასიათებელია ობიექტ-როლური მოდელირებისთვის. ობიექტები გამოისახება ელიფსებით. წყვეტილ-კონტურიანი ელიფსი ასახავს ობიექტის მნიშვნელობას (value). პრედიკატები გამოსახულია მართკუთხედებით და ისინი ადწერს ობიექტთაშორის კავშირებს.

(U) - იძულებითი შეზღუდვა (მაგალითად, ყოველ ფირმას აქვს დასახელება);

↔ - შიგა უნიკალურობის შეზღუდვა (მაგალითად, კონტრაქტს აქვს მხოლოდ ერთი ნომერი);

• - გარე უნიკალურობის შეზღუდვა (მაგალითად, ფირმის უნიკალურობა განისაზღვრება მისი სახელით და მისამართით);

↑ - შეზღუდვა ობიექტის ქვეტიაზე (მაგალითად, შემკვეთი და მიმწოდებელი არის ზოგადად ფირმის ქვეტიაზე).

„Contract“ აისახება „ბუდის“ ტიპის საშუალებით, რომელიც უზრუნველყოფს პრედიკატის ტრანსფორმაციას შესაბამის ობიექტში [25].

მომდევნო პარაგრაფში განვიხილავთ აქ ჩამოყალიბებული მოსაზრებებისა და ეტაპების კონკრეტულ რეალიზაციას, ბიზნეს-პროგრამების ექსპერტულ შეფასებათა მხარდაჭერი სისტემის ობიექტ-როლური და კონცეპტუალური მოდელების ასაგებად.

### 2.3. ობიექტ-როლური მოდელების დაპროექტება და აგება აგტომატიზებულ რეჟიმში

ბიზნეს-ობიექტებზე განსაკუთრებული ყერადღება ექცევა მარკეტინგისა და მენეჯმენტის პროცესების სწორად და დროულად წარმართვას. ახალი ბიზნეს-პროგრამები, რომელთა განხორციელება, ერთის მხრივ სასიცოცხლოა თანამედროვე კონკურენციის პირობებში, ხოლო მეორეს მხრივ, ძნელადრეალიზებადია საკმაო მოცულობის ინვესტიციების მოზიდვის თვალსაზრისით, მოითხოვს ფირმის ხელმძღვანელებისა და ექსპერტ-სპეციალისტებისგან სწორი ეკონომიკური პოლიტიკის გატარებას გააზრებული ჯგუფური გადაწყვეტილების მიღების საფუძველზე [1,73].

ბიზნეს-პროგრამების ექსპერტულ შეფასებათა მხარდაჭერ გადაწყვეტილებათა მიღების მეთოდებისა და მოდელების შემუშავება,

აგრეთვე ასეთი მართვის ავტომატიზებული სისტემების ობიექტ-ორიენტირებული დაპროექტება და რეალიზაცია აქტუალური და მეტად მნიშვნელოვანია [2,3,52].

ალტერნატიული ბიზნეს-პროგრამების შეფასების პროცესი, რომელიც დიდი ინფორმაციული ნაკადების ანალიზს ეფუძნება, მოითხოვს აგრეთვე გამოცდილი ექსპერტ-კონსულტანტების ცოდნის, გამოცდილებისა და ინტეიციის ერთობლიობას, სწორი პროგნოზირების უნარს, რაც მეტად მნიშვნელოვანია კრიტიკულ სიტუაციაში და მცირე დროითი რესურსების პირობებში. ამგვარად, კონსულტანტები და ფირმის გამოცდილი სპეციალისტები, როგორც საპრობლემო სფეროს ექსპერტები, ერთობლივი ძალისხმევით, კონსულტაციებითა და კომპრომისებით ცდილობენ მიიღონ ოპტიმალური გადაწყვეტილებანი ფირმის განვითარების სწორი კორპორაციული გეგმებისა და ტაქტიკურ დონისძიებათა შესახებ, ბიზნეს-პროგრამების შერჩევის ნაკლები რისკით. 2.2 ნახაზზე მოცემული იყო დასაპროექტებელი სისტემის როლურ-ფუნქციური UseCase დიაგრამა UML ენაზე, რომლის განხორციელებაც განიხილება ქვემოთ.

ექსპერტთა შეფასებების მხარდაჭერი სისტემის დაპროექტება იწყება საკვლევი სფეროს ობიექტ-ორიენტირებული ანალიზით, რომელიც, როგორც ცნობილია, მოითხოვს სტატიკური და დინამიკური დიაგრამების: პრეცედენტების, აქტიურობის, მიმდევრობითობის, კლასების და ა.შ. მომზადებას [2,3].

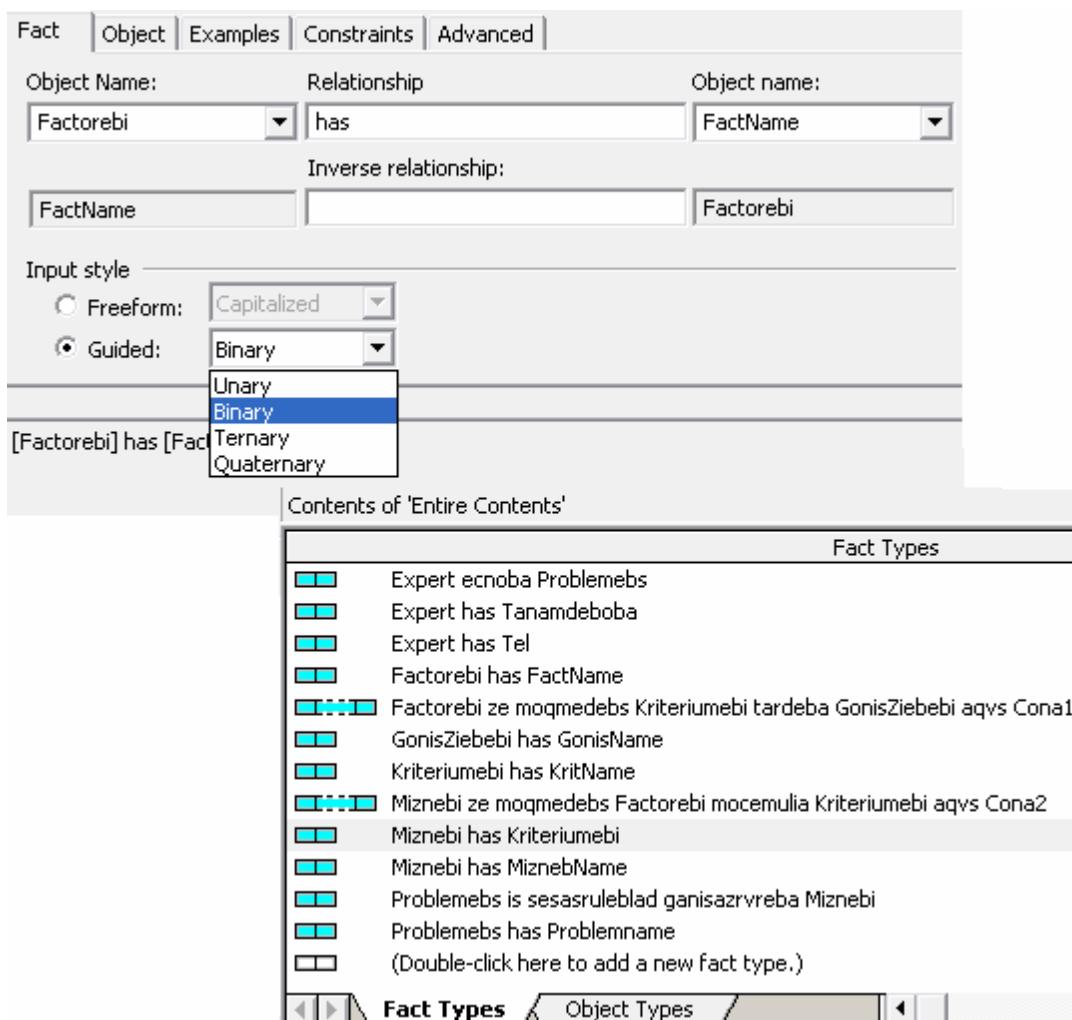
ამ მასალის საფუძველზე ექსპერტების ან კონსულტანტის მიერ შესაძლებელია კატეგორიალური მიდგომით გარკვეულ ფაქტების სიმრავლის ფორმირება [25,29]. ჩვენი ამოცანის შესაბამისად საკვლევი ობიექტის გარკვეული ცოდნა ასახულია 2.1 ცხრილში:

ცხრ.2.1

ც-№	ვაძლის ჩამონის ვრომა
f <sub>1</sub>	- ექსპერტს აქვს გვარი;
f <sub>2</sub>	- ექსპერტს აქვს ონამდებობა;
f <sub>3</sub>	- ექსპერტს აქვს ტელეფონი;
f <sub>4</sub>	- ექსპერტი ეცნობა პრობლემას;
f <sub>5</sub>	- პრობლემას აქვთ დასახელება;
f <sub>6</sub>	- პრობლემის შესასრულებლად განისაზღვრება მიზნები;
f <sub>7</sub>	- მიზანს აქვს დასახელება;
f <sub>8</sub>	- მიზანს აქვს კრიტერიუმი;
f <sub>9</sub>	- მიზანზე მოქმედებს ფაქტორები;
f <sub>10</sub>	- ფაქტორს აქვს დასახელება;
f <sub>11</sub>	- ფაქტორს აქვს კრიტერიუმი;

f <sub>12</sub> f <sub>13</sub> f <sub>14</sub> f <sub>15</sub> და ა.შ.	<ul style="list-style-type: none"> <li>- ფაქტორზე მოქმედებს დონისძიებები;</li> <li>- დონისძიებას აქვს დასახელება;</li> <li>- დონისძიებას აქვს კრიტერიუმი;</li> <li>- კრიტერიუმს აქვს წონა;</li> </ul>
---	---

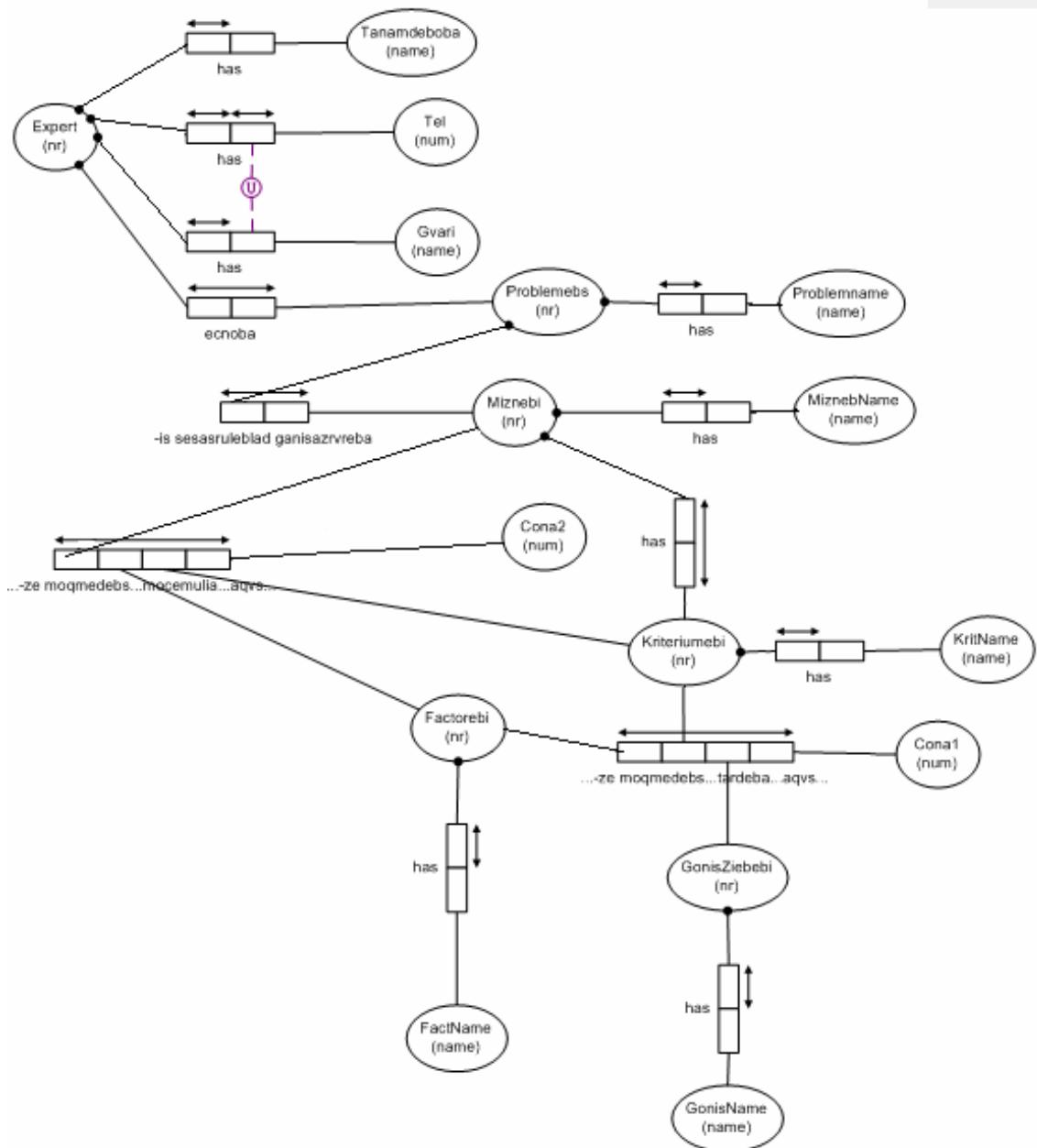
Visual Studio .NET გარემოში Ms Visio ინსტრუმენტის Database პატეგორიაში ORM Source Model- შაბლონით (Template) ვახორციელებთ ზემოაღწერილი ფაქტების ჩაწერას უნარული, ბინარული ან უფრო მაღალი რიგის რეალაციური დამოკიდებულებით (ნახ.2.8).



#### ნახ.2.8. ფაქტების ჩაწერის ინსტრუმენტი

დიაგრამიდან ჩანს, რომ საპრობლემო სფერო აღწერილია ელემენტარული ფაქტების საშუალებით, რაც საგრძნობლად ამარტივებს დაპროექტების პროცესს.

ფაქტების ჩაწერის ინსტრუმენტის გამოყენებით შესაძლებელი ხდება საპრობლემო სფეროს სემანტიკური ინფორმაციის გადატანა მოდელში, რომელიც ობიექტ-როლური მოდელის სახელითაა ცნობილი და, ჩვენს შემთხვევაში, ექნება 2.9 ნახაზზე მოცემული სახე.



**ნახ.2.9. ORM მოდელის ფრაგმენტი ექსპერტულ შეფასებათა სისტემისთვის**

ობიექტები გამოისახება ელიფსებით, პრედიკატები  
მართვულებით, მნიშვნელობის ტიპი გამოისახება წყვეტილი

ელიფსით. იმ შემთხვევაში თუ განისაზღვრება ობიექტის მხოლოდ ერთი თვისება, საქმე გვაქვს ერთ ადგილიან პრედიკატან (unary fact). პრედიკატს შეიძლება ჰქონდეს (1,2,3,...) ოპერანდი, თუ პრედიკატი ელემენტარულია 3-4 ოპერანდზე მეტი იშვიათად გვხვდება, თუმცა არსებობს n-არული პრედიკატიც.

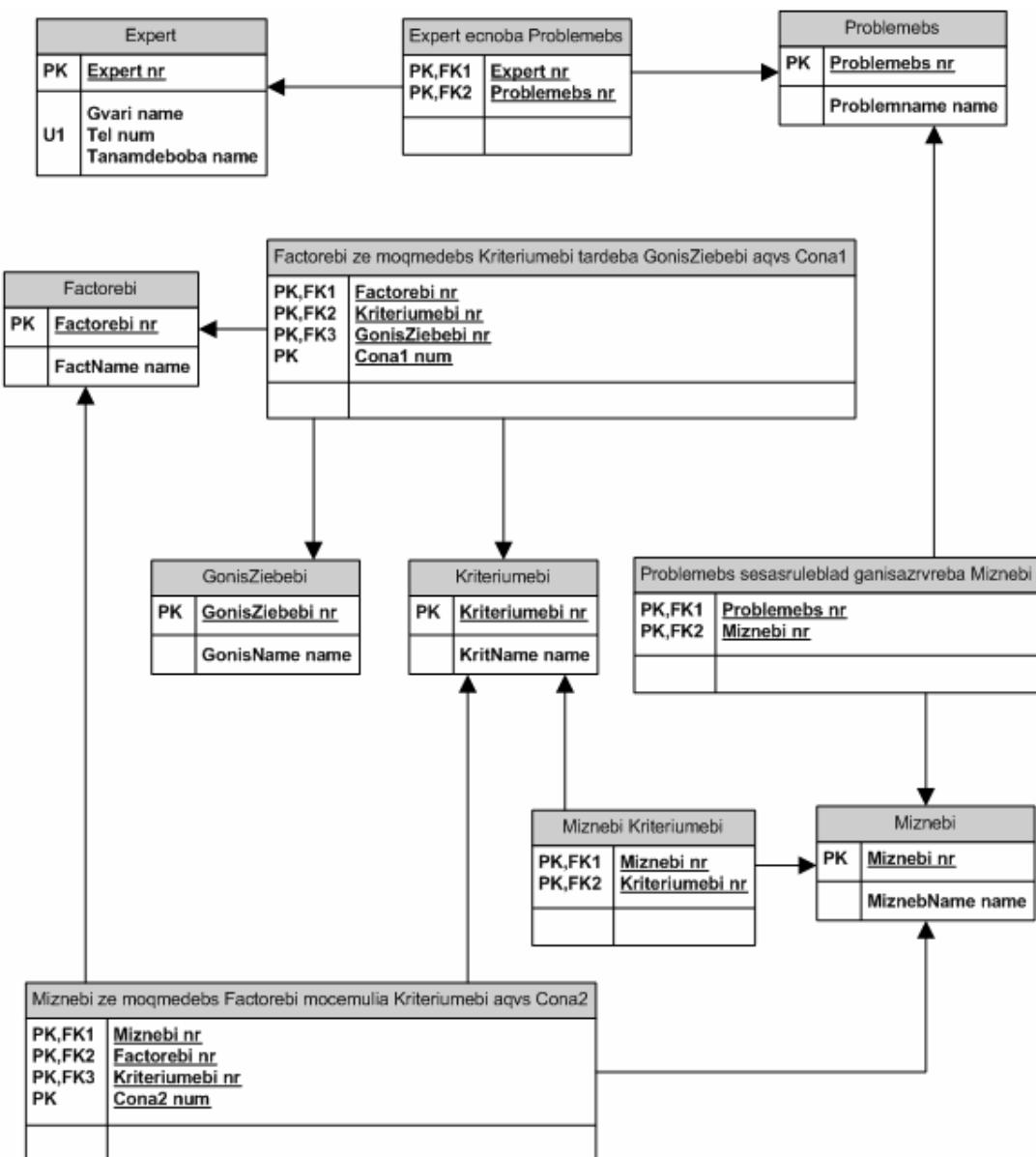
ობიექტ-როლური მოდელირებაში გამოყენება სხვადასხვა შეზღუდვები. ჩვენს შემთხვევაში გამოყენებულია შემდეგი სახის შეზღუდვები.

- - იძულების შეზღუდვა გვიჩვენებს, რომ ექსპერტი გალდებულია ჰქონდეს გვარი
- ↔ - შიგა უნიკალურობის შეზღუდვა გვიჩვენებს, რომ რომ ექსპერტს აქვს მხოლოდ ერთი თანამდებობა.
- (U) - გვიჩვენებს, რომ მონაცემთა ბაზაში ექსპერტი უნიკალურედ განისაზღვრება გვარითა და ტელეფონის ნომრით.

## 2.4 კონცეპტუალური მოდელებისა და კლასთა ურთიერთკავშირების სქემის დაპროექტება ობიექტ-როლური მოდელების ბაზაზე

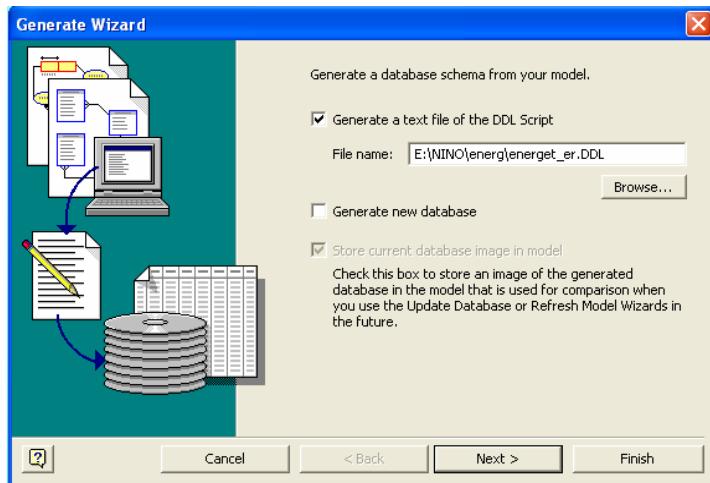
2.10 ნახაზზე მოცემულია ERM-დიაგრამა, რომელიც ავტომატიზებულ რეჟიმში აიგო ORM-დიაგრამის საფუძველზე. ამისათვის გამოყენებულ იქნა Ms Visio ინსტრუმენტის Database კატეგორიაში ERM მოდელის გენერაციის პროცესში.

მიღებული არსთა-დამოკიდებულების ER-მოდელი (ცხრილები და ცხრილთაშორისი კავშირები) ფორმირებულია ORM-მოდელის აგების პროცესში ჩაწერილი სემანტიკური ინფორმაციის საფუძველზე, რაც საშუალებას იძლევა ავტომატიზებულ რეჟიმში განისაზღვროს არა მხოლოდ ცხრილთაშორისი კავშირები, არამედ პირველადი (PK) და მეორეული (FK) გასაღებური, ინდექსური (I) და უნიკალურობის (U) ატრიბუტების ბაზაზე.



ნახ.2.10. ERM მოდელის ფრაგმენტი, აგებული ავტომატიზებულ  
რეჟიმში ORM მოდელის საფუძველზე

მესამე ეტაპზე ER-მოდელის საფუძველზე ვახდენთ კოდის სახით  
მონაცემთა აღწერის .DDL ფაილის ავტომატურ გენერირებას (ნახ.2.11).



ნახ.2.11. კოდის გენერირების ვიზარდ-პროგრამა

ქვემოთ, 2.2 ცხრილში მოცემულია DDL ფაილის კოდის ფრაგმენტი ჩვენი ამოცანისათვის :

#### ცხრ.2.2

```
/*
This SQL DDL script was generated by Microsoft Visual Studio (Release Date:
LOCAL BUILD).
*/
/* Driver Used : Microsoft Visual Studio - Microsoft SQL Server Driver. */
/* Create expert database. */
use master
go
use "expert"
go
alter table "Miznebi Kriteriumebi"
    add constraint "Miznebi_Kriteriumebi_PK" primary key ("Miznebi_nr",
"Kriteriumebi_nr")
..
...
go
alter table "Expert_ecnoba_Problemebs"
    add constraint "Problemebs_Expert_ecnoba_Problemebs_FK1" foreign
key (
    "Problemebs_nr")
    references "Problemebs" (
    "Problemebs_nr") on update no action on delete no action
```

ამგვარად, ახალი ინფორმაციული ტექნოლოგიის გამოყენებით და  
განაწილებული მონაცემთა ბაზების სტრუქტურების ამოცანის  
გადაწყვეტის შესაბამისი ალგორითმული სქემების დაპროექტებით,  
ობიექტ-ორიენტირებული მეთოდებისა და ინსტრუმენტების საფუძველზე  
შესაძლებელია დაპროექტების და დაპროგრამების პროცესების

ავტომატიზაცია, რაც მნიშვნელოვნად ამცირებს სისტემების შექმნის დროს და ამაღლებს მის ხარისხს. მონაცემთა ბაზების სტრუქტურები და ცხრილთაშორისი კაგშირები, ჩვენს შემთხვევაში მზადაა კონკრეტული მონაცემების შესატანად MsSQL Server გარემოში. აუცილებლობის შემთხვევაში შესაძლებელია მიღებული მონაცემთა ბაზის სტრუქტურის ავტომატიზებული მოდიფიკაცია და მისი შემდგომი რეალიზაცია (სადემონსტრაციო ვერსია იხ. დანართში).

## **2.5. განაწილებული ბიზნეს-პროცესების ობიექტ-ორიენტირებული მოდელირება და ანალიზი პეტრის ქსელებით**

საწარმოო, ადმინისტრაციულ და ორგანიზაციულ სისტემებში მეტად აქტუალურია საქმიანი პროცესების ტექნოლოგიური ციკლების თანმხლები საინფორმაციო ნაკადების მართვის მექანიზმების დახვეწა, უნიფიცირებული დოკუმენტებისა და დოკუმენტრუნვის პრობლემათა გათვალისწინებით, ახალი საკანონმდებლო აქტების და ინტერნეტ-ინტრანეტის პირობებში.

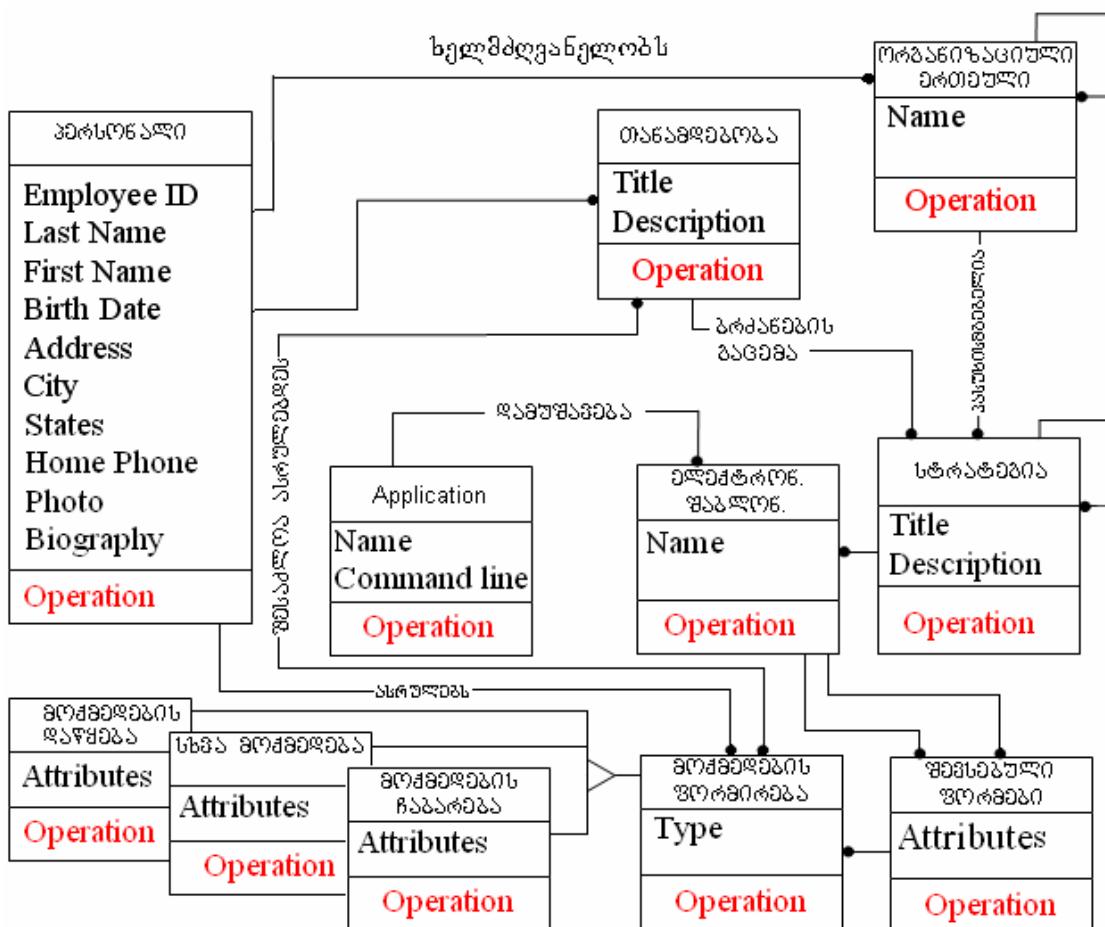
ბიზნეს-ობიექტი, ზოგადად წარმოადგენს ფიზიკურად განაწილებულ ბიზნეს-პროცესებს, მათი ლოგიკურად ერთიანი მონაცემთა საცავის (ბაზების ერთობლიობის) და ადმინისტრაციული მართვის კლასიკური მოდელის ინტეგრირებულ სტრუქტურას, უახლესი ინფორმაციული, მოდელურ-პროგრამული, ტექნიკურ-ტექნოლოგიური, ლინგვისტურ-ინტერფეისული, იურიდიული და ორგანიზაციულ-მეთოდურ უზრუნველყოფათა ერთობლიობით.

ყოველ ორგანიზაციაში მიმდინარეობს მისთვის დამახასიათებელი ტექნიკური, ტექნოლოგიური და ინფორმაციული პროცესები. წინამდებარე ნაშრომში ჩვენ გვაინტერესებს ის საერთო, ზოგადი, რაც ახასიათებს ამ ოფისებს. მაგალითად, მართვის ამოცანები: დაგეგმვა, ადრიცხვა, ანალიზი, გადაწყვეტილების მიღება, და რაც მთავარია, მათი ინფორმაციული უზრუნველყოფის თვალსაზრისით.

განაწილებულ ბიზნეს-სისტემებში მიმდინარე პროცესების ავტომატიზაციის ინფრასტუქტურის ობიექტ-ორიენტირებული ანალიზური მოდელი ნაჩვენებია 2.12-ელ ნახაზზე. ანალიზის შედეგად ოფის-ობიექტებში გამოიკვეთა შემდეგი ძირითადი კლასები: პერსონალი, თანამდებობა, ორგანიზაციული ერთეული, სტრატეგია, აპლიკაცია,

ელექტრონული ფორმის შაბლონი, ელექტრონული ფორმის ეგზემპლარი, ფორმის მოქმედება. კლასთაშორის ასოციაციები ასახულია რელაციური, მემკვიდრეობითი და აგრეგატული კავშირებით [3].

მოცემული სქემის ფუნქციონირების სემანტიკური აღწერის მიზნით განვიხილოთ მისი ელემენტებისა და კავშირებისათვის დამახასიათებელი არაფორმალიზებული (ტექსტური) ინფორმაცია, რომელშიც ასახული იქნება მოქმედებათა ზოგადი წესები.



ნახ.2.12. საქმიანი პროცესების ზოგადი ანალიზური მოდელი

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

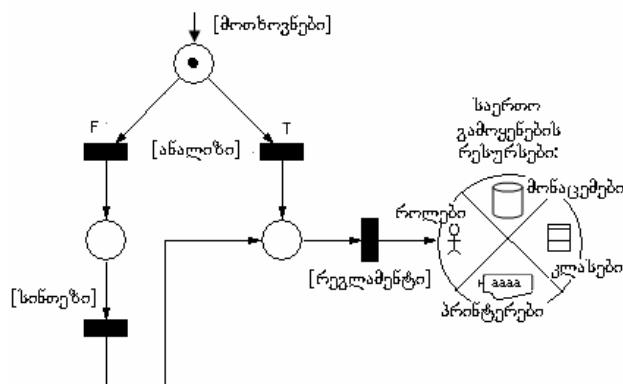
თითოეული თანამდებობა (როლი) მიმაგრებულია კონკრეტულ პირზე, თუმცა ზოგიერთმა თანამშრომელმა შეიძლება შეითავსოს

რამდენიმე. 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 იქნას 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) და ა.შ.

განაწილებულ ოფის-სისტემებში ფუნქციური ამოცანების ეფექტური გადაწყვეტის მიზნით, სწრაფი დოკუმენტუზრუნველყოფის თვალსაზრისით საერთო გამოყენების რესურსების პირობებში, შესაძლებელია გარკვეული ალგორითმული სქემების შემუშავება და მათი ანალიზი პეტრის ქსელების ბაზაზე [12,28,51].

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

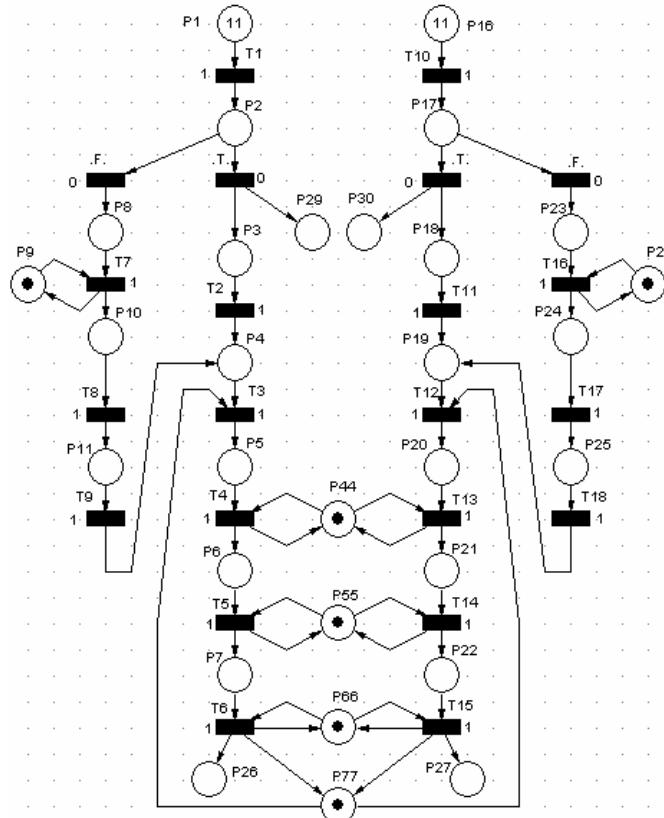


**ნახ.2.13. განაწილებულ სისტემაში მოთხოვნების დამუშავების ზოგადი მოდელი**

მარგერი საწყის პოზიციაში მიუთითებს მოთხოვნის არსებობაზე, რომელიც ანალიზის ეტაპზე განშტოვდება ორ კონფლიქტურ (T. ან F.) გადასასვლეს 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 ორგანიზაციის კოლექტიური გამოყენების თვალსაზრისით.

აღნიშნული პროცესის მართვის მოდელის დეტალიზება და მისი ეფექტურობის ანალიზი გარკვეული რაოდენობრივი კრიტერიუმებით, შესაძლებელია პეტრის ქსელის შემდგომი დაზუსტებით. 2.14 ნახაზზე ნაჩვენებია ასეთი სქემის მაგალითი, ხოლო 2.3-2.4 ცხრილებში მოცემულია მისი შესაბამისი პოზიციების და გადასასვლელთა აღწერა. მოდელი შეიძლება განზოგადებულ იქნას **n**—მომხმარებლისთვის.



ნახ.2.14. დეტალიზებული პეტრის ქსელის ფრაგმენტი

### ცხრ.2.3

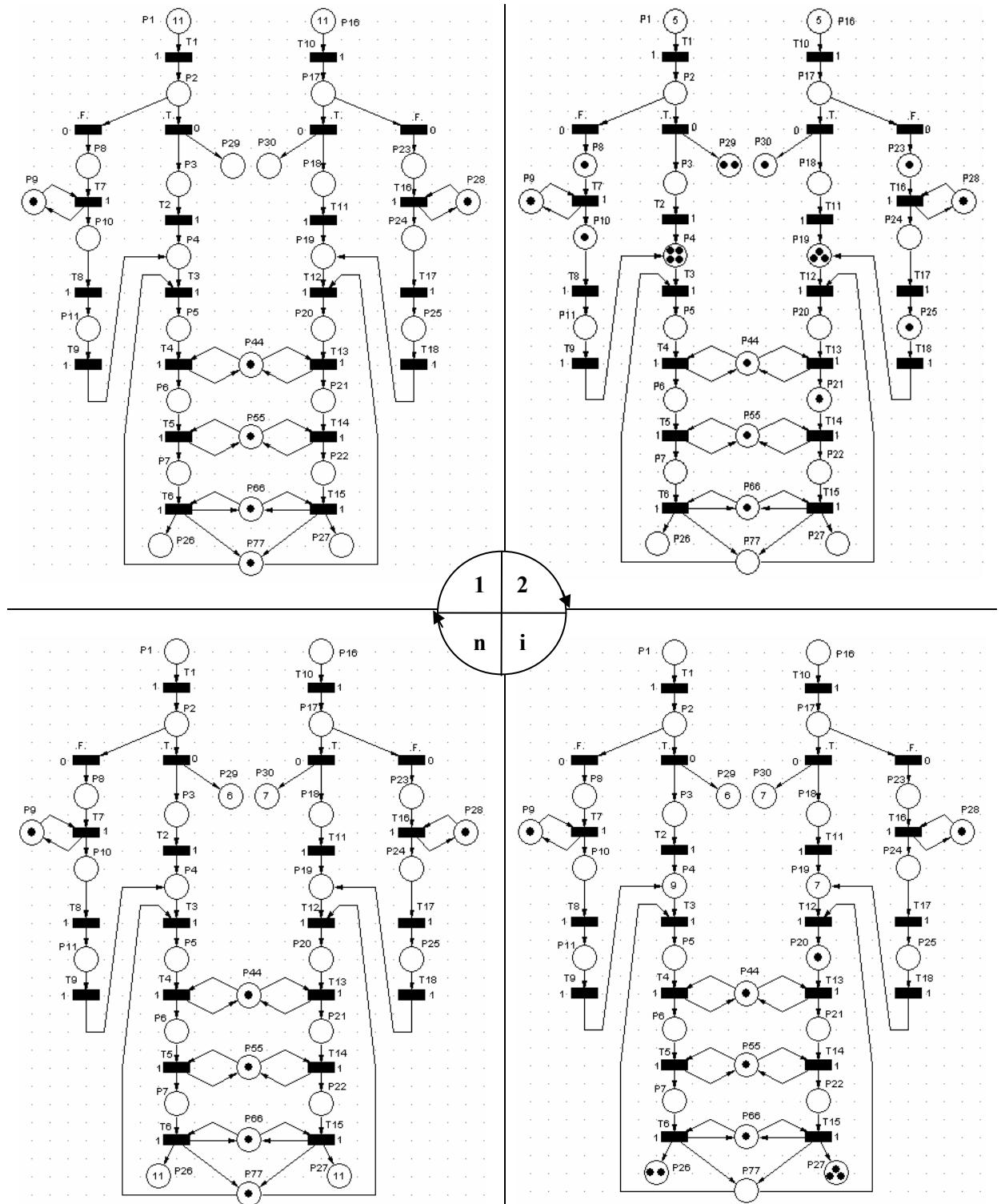
პოზიციების ცხრილი	
P <sub>j</sub>	პოზიცია (მარკერებით)
P <sub>1, 16</sub>	მოთხოვნა არის (თუ მარკერია)
P <sub>2, 17</sub>	დოკუმენტის ფორმა (კოდი)
P <sub>3, 18</sub>	ცნობილი დოკუმენტის ფორმა
P <sub>4, 19</sub>	დასამუშავებელი დოკუმენტები
P <sub>5, 20</sub>	დოკუმენტი როლთან
P <sub>6, 21</sub>	დოკუმენტი მონაცემთა 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.2.4

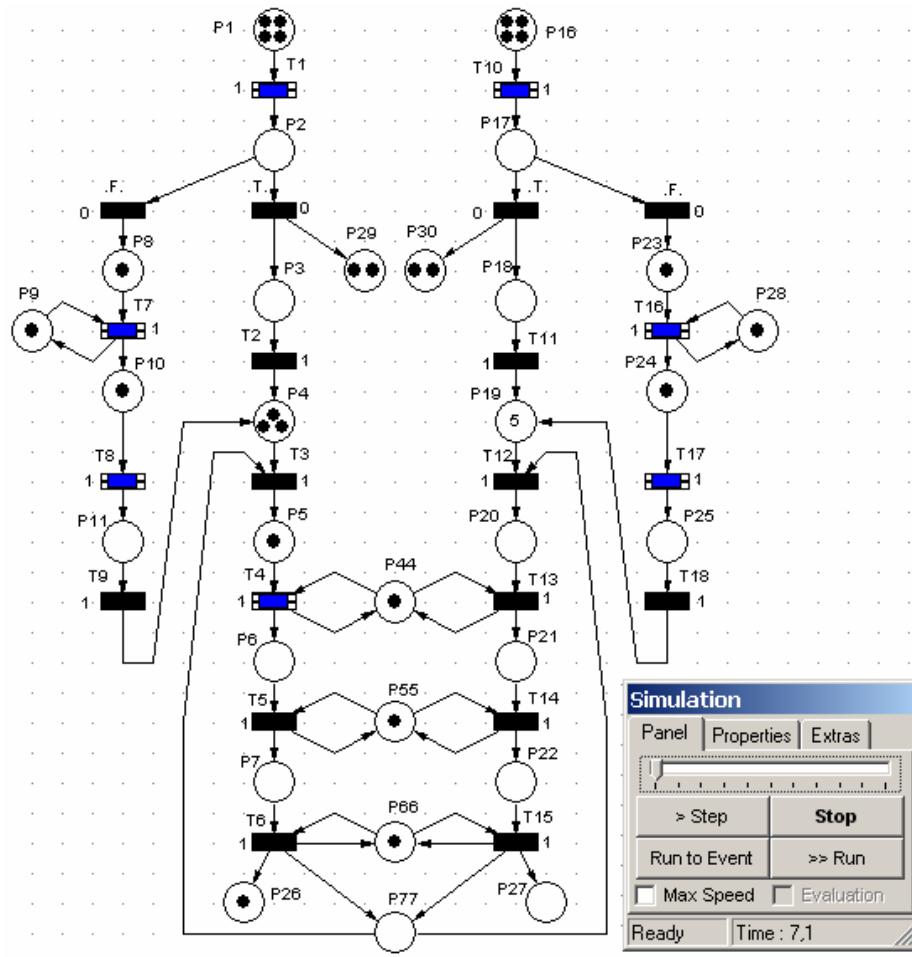
გადასასვლელების ცხრილი	
T <sub>i</sub>	გადასასვლელის ფუნქცია
T <sub>1, 10</sub>	მოთხოვნის ანალიზი
.T. / .F.	ფორმა ცნობილია/არაა ცნობილი
T <sub>2, 11</sub>	დოკუმენტის ფორმის არჩევა
T <sub>3, 12</sub>	ადამიანური რესურსით ფორმის შევსების წესის განსაზღვრა
T <sub>4, 13</sub>	მონაცემთა რესურსის მიღება
T <sub>5, 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

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

2.16 ნახაზზე ნაჩვენებია იმიტაციური პროცესის მართვის აპარატი (Run, Stop და სხვა დილაკებით) და თვით პეტრის ქსელის ერთ-ერთი შუალედური მდგომარეობა, რომელზეც კარგად ჩანს მარკერების განაწილება და პარალელურად მომუშავე გადასასვლელები (მონიშნულია). ამ მომენტში შერსულებულია მხოლოდ ერთი დავალება (26 – ში ერთი მარკერია).



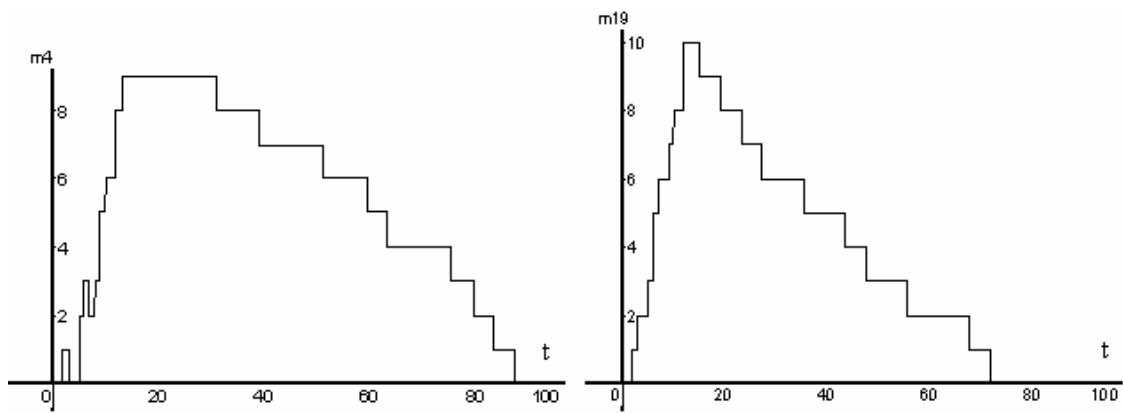
### **ნახ.2.15. პეტრის ქუსოვის სიმულატორით მიღებული ეტაპობრივი შედეგები**



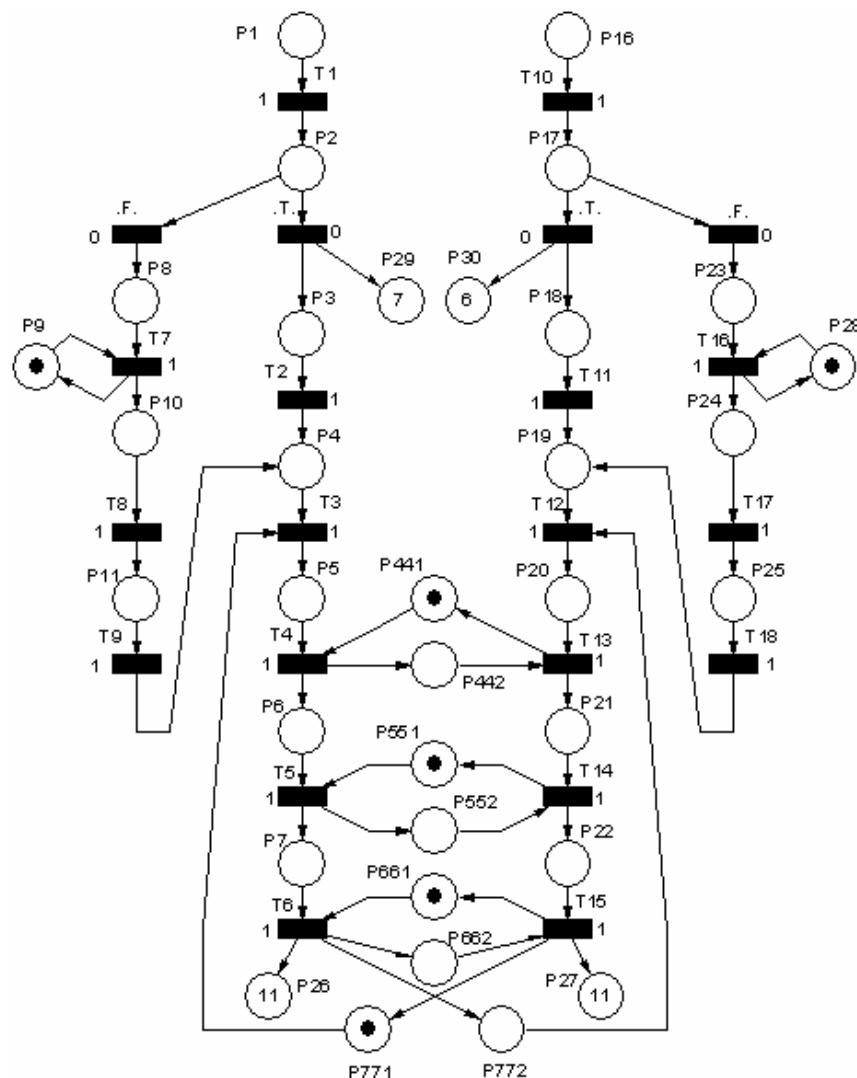
ნახ.2.16. იმიტაციის პროცესის ფრაგმენტი პეტრის ქსელით

2.15 ნახაზის ბოლო ეტაპზე  $P_{26}$  და  $P_{27}$  ჩანს, რომ ყველა მოთხოვნა შერულებულია.  $P_{29}$  და  $P_{30}$  პოზიციები მიუთითებს, რომ 1-ელ კვანძში 11 შემოსული მოთხოვნიდან 6 იყო დეტერმინირებულ ამოცანათა კლასიდან, ხოლო მეორეში – 7.

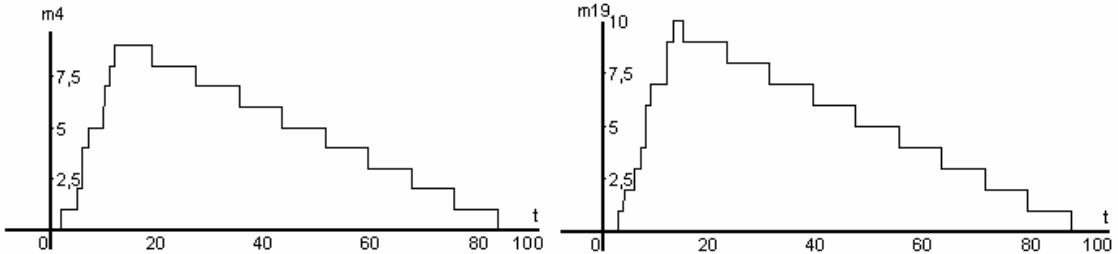
$P_4$  და  $P_{19}$  ის პოზიციებია, სადაც კონკრეტულ როლთან (ადამიანის რესურსი) თავს იყრის სხვადასხვა მოთხოვნები და საიდანაც უნდა მოხდეს მათი უზრუნველყოფა მონაცემთა ბაზებით, პროგრამებით და საბეჭდი მოწყობილობით. სქემის აღნიშნულ ფრაგმენტს განსაკუთრებული მნიშვნელობა აქვს მრავალმომხმარებლურ რეჟიმში მუშაობისას, როდესაც სისტემის ადამიანური და სერვერული რესურსები ეფექტურად უნდა იმართოს. 2.17 ნახაზზე მოცემულია  $P_4$  და  $P_{19}$  პოზიციების მდგომარეობები დროის მიხედვით (მარკერთა შესაბამისი ცვლადების ცვლილების დიაგრამები).



ნახ.2.17.  $P_4$  და  $P_{19}$  პოზიციების მდგომარეობათა ცვლილებები დროში



ნახ.2.18. რესურსების ტრიგერული მართვის ფრაგმენტი

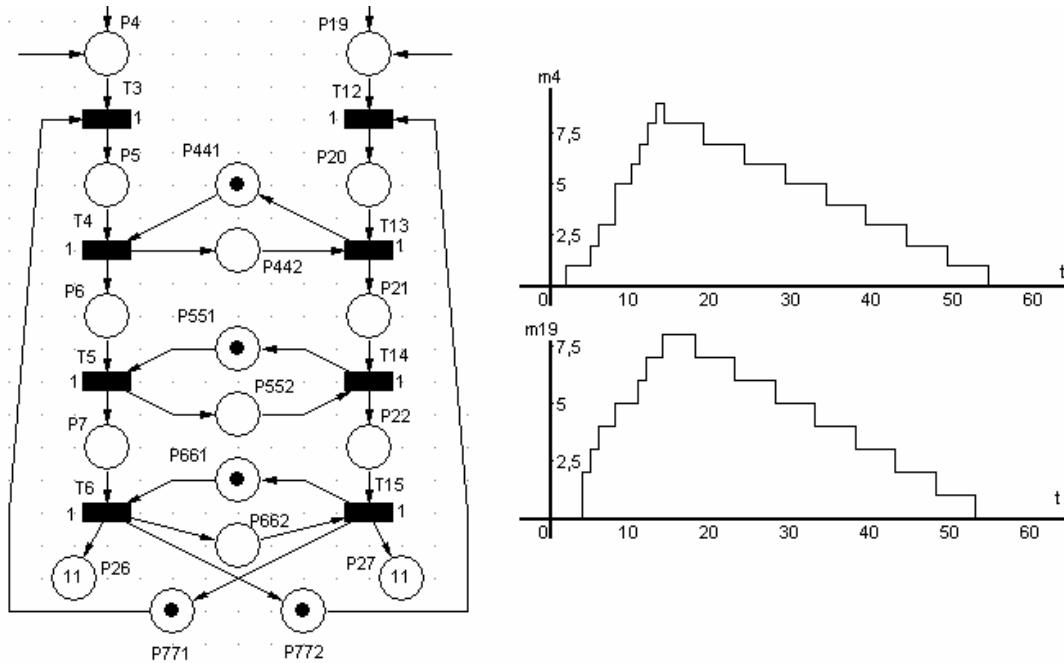


**ნახ.2.19.  $P_4$  და  $P_{19}$  პოზიციების მდგომარეობათა ცვლილებები დროში  
ტრიგერული მართვისას**

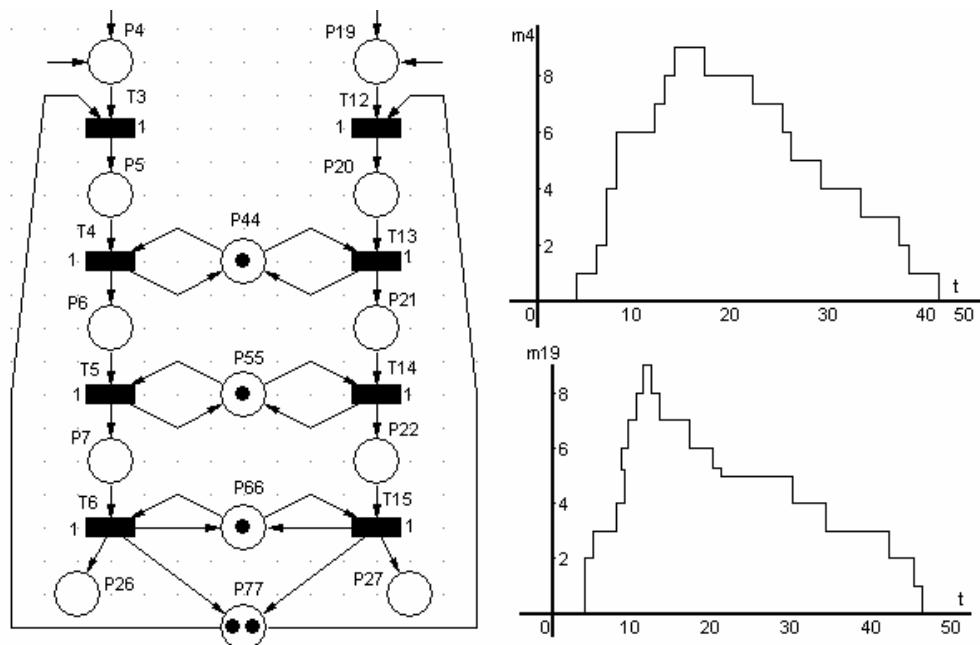
ტრიგერული მართვისას რესურსები მიმდევრობით გადაეცემა მომხმარებლებს ყოველგვარი პრიორიტეტის გარეშე. ამიტომაც აქ მოთხოვნების დამუშავება სინქრონულად მიმდინარეობს.

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

მომდევნო მაგალითებში ნაჩვენები გვაქვს რესურსების ექსტენსიონალური (ნახ.2.20) და ინტენსიონალური (ნახ.2.21) გაფართოებით მოთხოვნების დამუშავების დროითი დიაგრამები. ნახაზებზე კარგად ჩანს მოთხოვნების დროის შემცირების ტენდენცია.



**ნახ.2.20.  $P_4$  და  $P_{19}$  პოზიციების მდგომარეობათა ცვლილებები დროში  
ტრიგერული მართვისას გაორკეცებული ადამიანური რესურსით**



ნახ.2.21.  $P_4$  და  $P_{19}$  პოზიციების მდგომარეობათა ცვლილებები დროში ადამიანური რესურსის გაორკეცხული სწრაფქმედებით

## 2.6. მეორე თავის დასკვნები

1. ბიზნეს-პროგრამების ექსპერტულ შეფასებათა ჯგუფური გადაწყვეტილების მიღების პროცესების მოდელირება განხორციელდა UML-ენის სტატიკური და დინამიკური დიაგრამებით, რაც უზრუნველყოფს დასაპროექტებელი სისტემის სემანტიკურ სისრულეს;
2. ბიზნეს-პროგრამების შეფასების ესქემების ცოდნის მოდელირებისათვის შერჩეულ იქნა კატეგორიალური მიდგომა, რაც უზრუნველყოფს სისტემის დაპროექტების შემდგომ ეტაპებზე რელაციური საცავებისა და ბაზების თავსებადობას;
3. ობიექტ-როლური მოდელების აგება განხორციელებულია ექსპერტთა ცოდნის ვიზუალურ რეჟიმში ასახვის ინსტრუმენტის საშუალებით, რაც მნიშვნელოვნად ამცირებს დაპროექტების დროს პროცესების ავტომატიზაციის საფუძველზე;
4. კონცეპტუალური მოდელების ბაზაზე შემუშავდა კლასთა ურთიერთკავშირების ავტომატიზებული დაპროექტების მექანიზმი, რომელშიც რეალიზებულია ობიექტ-ორიენტირებული მიდგომის პრინციპი;
5. კორპორაციული სისტემების განაწილებული ბიზნეს-პროცესების დოკუმენტური უზრუნველყოფის და რაციონალური დოკუმენტბრუნვის დასაპროექტებლად მნიშვნელოვანი ინსტრუმენტია კლასებსა და პროცესებზე ორიენტირებული საპრობლემო სფეროს მოდელირებისა და ანალიზის მეთოდები, რომელთა რეალიზაცია და კვლევა ხორციელდება პეტრის ქსელებით საერთო რესურსების პირობებში.

### III თავი. შეღებების პრაქტიკული რეალიზაცია:

#### 06 ვორმაციულ-აროგრამული უზრუნველყოფის დამუშავება

##### 3.1. განაწილებული მონაცემთა ბაზების აგების ავტომატიზაცია .NET გარემოში

თანამედროვე კორპორაციული კომერციული ობიექტები ხასიათდება დიდი მოცულობის ინფორმაციული ნაკადებით, რომელთა მოძრაობისა და აქტუალიზების დინამიკა განსაკუთრებით მაღალი ტემპებით გამოირჩევა. მათი ხელმძღვანელების საპრობლემო ამოცანა ამ ინფორმაციის კომპლექსური ანალიზისა და სწორი, ოპტიმალური გადაწყვეტილების გამომუშავებაში მდგომარეობს. აქტუალურია ასეთი პრობლემების გადასაწყვეტად კომპიუტერული მსარდამჭერი სისტემების აგება, რომელთა გული მონაცემთა საცავია, მრავალგანზომილებიანი, რელაციური მოდელის საფუძველზე დაპროექტებული, ხოლო გონება – მათი ოპერატიული ანალიზის სპეციალური მეთოდები (პროგრამები), რეალიზებული ობიექტ-ორიენტირებული დაპროგრამების UML-ტექნიკებით [3,20,67]. სტატიაში განიხილება კომერციულ-საფინანსო ბანკების მაგალითზე მონაცემთა საცავის დაპროექტებისა და მისი მონიტორინგის სისტემის აგების ამოცანები.

ჩვენი კვლევის ობიექტი საინფორმაციო მონიტორინგის კომპიუტერული სისტემაა, რომელიც ტერიტორიულად განაწილებული კორპორაციული საბანკო-საფინანსო ორგანიზაციისა და მისი ფილიალების ერთიანი მონაცემთა საცავის აგებასა და მისი ოპერატიული ანალიზის მექანიზმების სრულყოფას ემსახურება. ადნიშნული მართვის ავტომატიზებული სისტემა მიეკუთვნება გადაწყვეტილებათა მიღების ხელშემწყობ კომპიუტერულ სისტემათა კლასს [8].

კომერციულ ბანკებში, განსაკუთრებული მნიშვნელობა აქვს არსებული ოპერატიული ინფორმაციის ანალიზს, მაგალითად მისი ფინანსური მდგომარეობის შესახებ, კლიენტთა შემოსავლების დინამიკის შესახებ, ან ბანკის შემდგომი განვითარების მიმართულებათა პროგნოზის კვლევის საკითხებს და ა.შ. ასეთი ამოცანების ეფექტურად გადაჭრა მხოლოდ მონაცემთა საცავების საფუძველზეა

მოსახერხებელი, რომელშიც თავმოყრილია არა მხოლოდ მიმდინარე, ოპერატორიული ინფორმაცია, არამედ მასში ინახება აგრეთვე ისტორიული მონაცემებიც (შესაძლებელია მისი დაარსების დღიდან არსებული ინფორმაციული არქივების სახით). ამასთანავე მონაცემთა საცავის პროგრამული უზრუნველყოფა მოიცავს ოპერატორიული ანალიზის OLAP-ინსტრუმენტსაც, რომელიც ფართოდ გამოიყენება საზღვარგარეთის ბანკებში ფინანსური ანალიზის ამოცანების გადასაწყვეტად [29].

საბანკო ოპერაციების ავტომატიზაცია ითხოვს მრავალი მაჩვენებლის ერთ მთლიან სისტემაში გაერთიანებას და მათ შორის კავშირების განსაზღვრას. მთავარი როლი უჭირავს კლასიფიკაციას და კოდირებას, რომელიც ამცირებს ინფორმაციის მოძიების დროს და მოცულობას, ამარტივებს ინფორმაციის დამუშავების პროცესს. საბანკო საქმიანობაში ინფორმაციის სისტემატიზაციისათვის გამოიყენება სხვადასხვა ტიპის კლასიფიკატორი. ესენია, როგორც დარგობრივი: ბანკოტების, მონეტების, ჩეკების, აქციების, ვალუტის, ბანკების და ა.შ. კოდები, ასევე ლოკალური: საბანკო ანგარიშების, სარეგისტრაციო ნომრების, კლიენტთა ნომრების კლასიფიკატორები და ა.შ.

თანამედროვე საინფორმაციო საბანკო სისტემის ინფორმაციული სტრუქტურა, რომელიც მომხმარებელს სთავაზობს ოპერაციულ-აღმრიცხველი სამუშაოების ავტომატიზაციის გარდა, ანალიტიკური და ასევე საკრედიტო დაწესებულების მდგომარეობის პროგნოზირების შესაძლებლობას, აგებულია მონაცემთა საცავის (data warehouse) საფუძველზე. მონაცემთა საცავი არის ლოგიკურად ინტეგრირებულ მონაცემთა ნაკრები, რომელიც უზრუნველყოფს ანალიზისათვის და გადაწყვეტილების მიღებისათვის საჭირო ინფორმაციაში მაქსიმალურად სწრაფად და ეფექტურად შეღწევას. მონაცემთა საცავი ხასიათდება შემდეგი თვისებებით:

- მოიცავს მხოლოდ იმ ინფორმაციას, რომელიც შეიძლება სასარგებლო იყოს გადაწყვეტილების მიღების პროცესისათვის;
- ინფორმაციის დაცვა, რომელიც ითვალისწინებს მონაცემების უცვლელობას და არამოდიფიცირებას და მათთან შეღწევას მხოლოდ წაკითხვის რეჟიმში;

- სხვადასხვა წყაროებიდან მოწოდებული მონაცემების ინტეგრაცია, მათი შემოწმება, შეჯამება და ერთიან ფორმატში დაყვანა;
- აგრეგაცია, რომელიც ითვალისწინებს მოწესრიგებული და განსაზღვრული სახით გადამუშავებული, ანუ შეჯამებული ინფორმაციის შენახვას;
- ოპერატიული გადამუშავებისა და ანალიტიკური ამოცანების გადსაწყვეტად საჭირო მონაცემთა ნაკრების დაყოფა.

მონაცემთა საცავის შევსება წარმოებს პერიოდულად მონაცემთა დამუშავების ოპერატიული ე.წ. OLTP სისტემებიდან და ასევე სხვა გარეშე წყაროებიდან [8].

მონაცემთა საცავს აქვს აგრეგირებულ მონაცემთა სივრცის მრავალწახნაგოვანი კუბის სტრუქტურა. კუბის განზომილება – არის ერთი ტიპის მონაცემთა სიმრავლე, რომელიც წარმოადგენს ამ კუბის ერთ-ერთ წახნაგს. განზომილების დეტალიზაციის მაგალითად შეიძლება მოვიყვანოთ: ტერიტორიული ნიშანთვისებებით გაერთიანებული ფილიალები; მნიშვნელობათა თარიღი – ესაა კონკრეტული თარიღი (ან ინტერვალი), რომელშიც ტარდებოდა გამოთვლები; კლიენტები – ესაას რომელიმე ნიშანთვისებით დეტალიზებული კლიენტთა კოდები (მაგალითად, იურიდიული და ფიზიკური პირები, რეზიდენტი და არარეზიდენტი ბანკები და ა.შ.).

ოპერაციები მრავალგანზომილებიან კუბზე წარმოადგენს ინფორმაციის ანალიზის პროცედურებს. ესენია, მაგალითად, ბრუნვა, ჭრილის ფორმირება, დეტალიზაცია, პროექცია, აგრეგაცია. მონაცემთა საცავის განმასხვავებული თვისება არის მეტამონაცემთა არსებობა, რომლებიც შეიცავს ცნობებს პირველად მონაცემთა წყაროებზე, მათგან ინფორმაციის ჩატვირთვის თაობაზე, გადამუშავების ალგორითმზე და ა.შ.

საბანკო სისტემის მომხმარებლებს უნდა შეეძლოთ ერთმანეთში ინფორმაციის გაცვლა, ანუ მომხმარებელი უნდა იყოს ერთდროულად როგორც მონაცემების გადამცემი, ასევე მათი მიმღებიც. კლიენტი გადასცემს ბანკს ინფორმაციას თვის ოპერაციებზე და ღებულობს საშედეგო დოკუმენტებს და მოთხოვნებზე პასუხებს. ფილიალები

უგზავნის თავიანთ ანგარიშებს ცენტრალურ ბანკს და ღებულობს ინსტრუქციულ მასალას და ინფორმაციულ ბაზაში შეღწევის უფლებას.

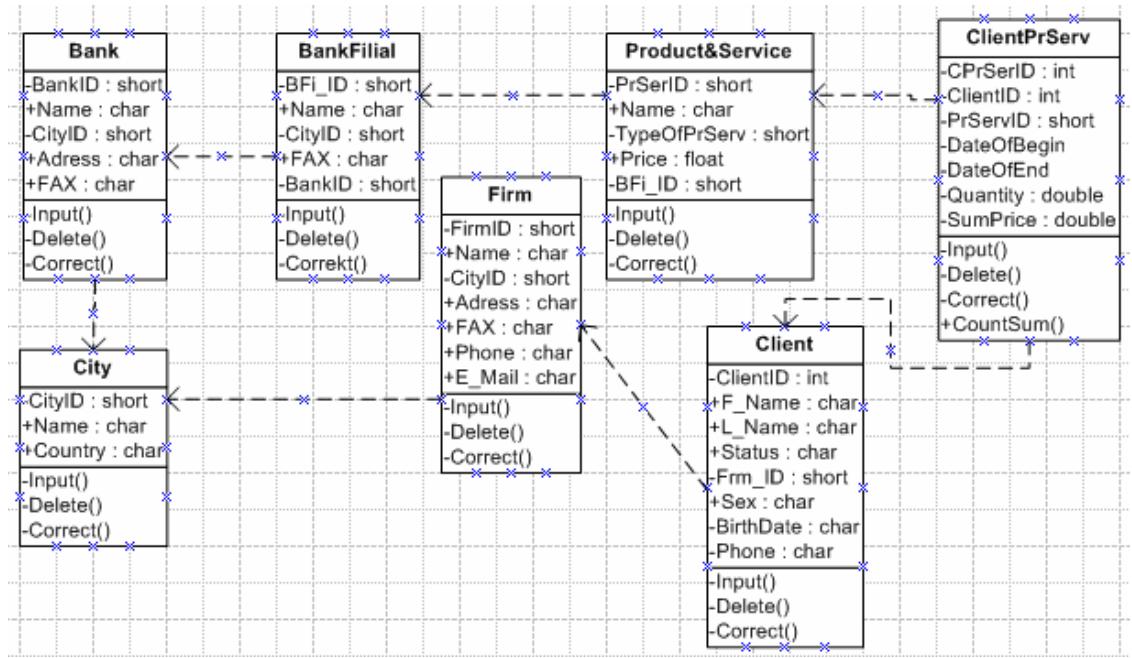
ბანკის ინფორმაციულ სისტემაში მუდმივმა ცვლილებებმა, მოქმედების არის გაფართოებამ, სიახლეების დანერგვამ საბანკო სფეროს პროგრამულ-აპარატულ უზრუნველყოფაში და ინფორმაციული ტექნოლოგიების საშუალებებში, ბანკთაშორისი კონკურენციის გაზრდამ და გლობალური კომპიუტერული ტელეკომუნიკაციური ქსელების (პირველ რიგში Internet-ის) სწრაფმა განვითარებამ ხელი შეუწყო კლიენტების ელექტრონული მომსახურეობის სპეციალის გაფართოებას.

საბანკო ტექნოლოგიის ძლიერ და შედარებით ახალ ინფორმაციულ უზრუნველყოფას წარმოადგენს ხელოვნური ინტელექტის სისტემის, კერძოდ ექსპერტული სისტემის გამოყენება. ასეთი სისტემების ერთ-ერთი კლასია ნეიროქსელები.

მათი ძირითადი დანიშნულებაა სპეციალური ალგორითმის საფუძველზე ინფორმაციის განზოგადება მონაცემებს შორის კავშირის დადგენით. დღესდღეისობით ნეიროპაკეტები უფრო მეტად გამოიყენება საბანკო სფეროში ფასიანი ქაღალდების მართვისათვის, ფინანსური ანალიზის ამოცანების გადასაწყვეტად და დასაგეგმად, იპოთეკური დაკრედიტების, საკრედიტო რისკის შეფასებისათვის და ა.შ. [29].

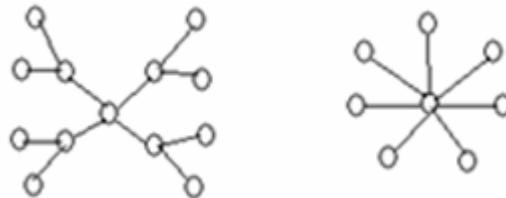
თანამედროვე პროგრამული აპლიკაციების აგების UML-ტექნოლოგიის გამოყენებით საბანკო სისტემის მოთხოვნილებათა განსაზღვრისა და ობიექტ-ორიენტირებული ანალიზის ეტაპებზე დავადგინეთ მომხმარებელთა როლების, ფუნქციების, ინტერაქტიული პროცესების, კლასებისა და ობიექტების, მოდგომარეობათა დიაგრამებისა და კლასთა ასოციაციების (ნახ.3.1) სქემების ერთობლიობანი.

მონაცემთა საცავის დასაპროექტებლად ვიყენებთ ობიექტ-როლური მოდელის ORM-დიაგრამას, რომლის საფუძველზეც აიგება არსთა-დამოკიდებულების ER-მოდელი. ასეთი სისტემების საფუძველი იყო და კვლავაც რჩება მონაცემთა რელაციური მოდელები და ბაზები [46-:49]. განსაკუთრებით ეფექტურია ასეთი მოდელების გამოყენება მარკეტინგული ამოცანების გადაწყვეტისას [50].



ნახ.3.1. კლასთა-ასოციაციის დიაგრამა

ასეთი კონცეპტუალური სქემის დაპროექტება ჩვენი კლასების დიაგრამისათვის შესაძლებელია „ფიფქის“ ან „გარსკვლავის“ ტოპოლოგიით (ნახ.3.2).

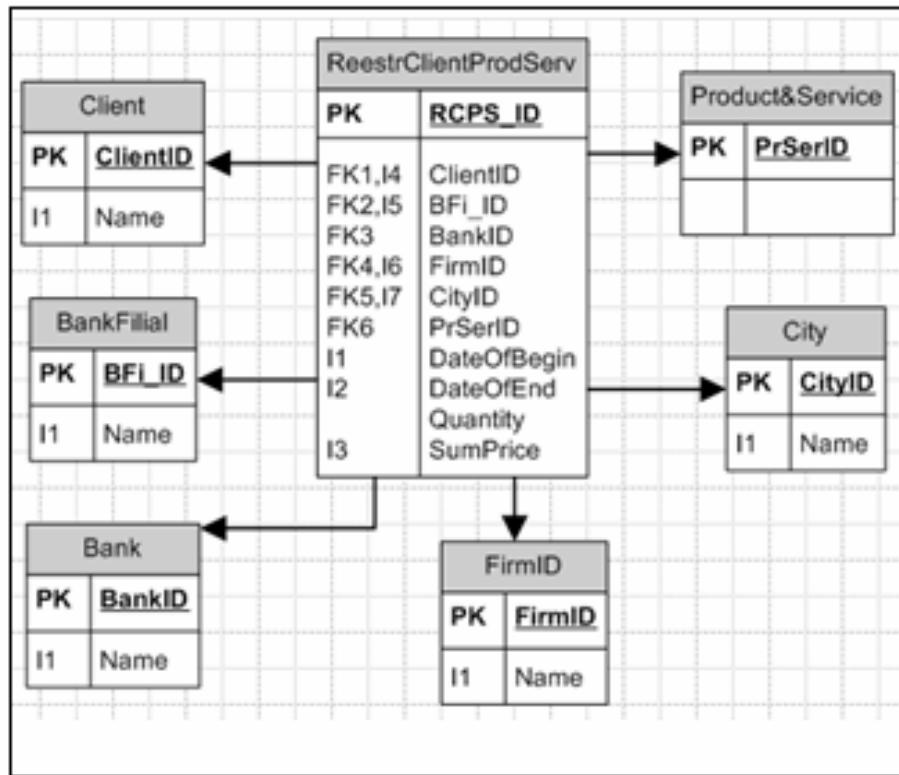


ნახ.3.2. ტოპოლოგია: ფიფქი და გარსკვლავი

### 3.3 ნახაზე მოცემულია

#### MsVisio->Database->ModelDiagram

ინსტრუმენტით აგებული ვარსკვლავური ER-სქემა. აქ მითითებულია პირველადი (PK) და მეორეული (FK) გასაღებები და ინდექსური (I) ატრიბუტები. ისრებით გამოსახულია კლასთა შორის ასოციაციური კავშირები.



ნახ.3.3. ER-დიაგრამის ფრაგმენტი

ვისარგებლოთ ბანკში იური კლიენტის შემოსავლების ანგარიშის მეთოდით [30]. ჩვენი მოდელისთვის მას ექნება ასეთი სახე:

$$E_i = P_i + C_i + K_i + T_i + N_i, \text{ სადაც}$$

$E_i$  - ეკონომიკური მოგება კლიენტიდან;

$P_i$  - პირდაპირი შემოსავლების სალდო კლიენტიდან;

$C_i$  - სარეზერვო მოძრაობის სალდო კლიენტის საკრედიტო პოზიციების მიხედვით;

$K_i$  - საკომისიო შემოსავლების და დანახარჯების სალდო, მიღებული კლიენტიდან;

$T_i$  - სატრანსფერო შემოსავლების და დანახარჯების სალდო კლიენტის რესურსების მიხედვით;

$N_i$  - ზედდებული ხარჯები კლიენტის მომსახურებიდან.

თითოეული კლიენტის შემოსავლების შეფასების ადგენატურობისათვის მიღებული მნიშვნელობა შეუდარდება მოულოდნელი ზარალის რისკს j-ოპერაციის მიხედვით:

$$R_{ij} = E_i / R * 100\%, \text{ სადაც}$$

$R$ - საერთო რისკია იური კლიენტის ყველა ოპერაციისათვის.

თანამედროვე პროგრამული UML-ტექნოლოგიების გამოყენებით შესაძლებელია კორპორაციული მართვის ავტომატიზებული სისტემების დაპროექტების პროცესების ავტომატიზაცია, რაც საგრძნობლად ამცირებს სისტემის შექმნის დროს და ხარჯებს, ქმნის პირობას სტანდარტული პაკეტების ასაგებად დაპროგრამების უახლეს ენებზე (C#, C++, VB, Java). ნაშრომში წარმოდგენილია კომერციული ბანკის მაგალითზე მონაცემთა საცავის აგებისა და პროგრამული სისტემის კლასთა დიაგრამების დაპროექტების ფრაგმენტები.

### **3.2. მონაცემთა საცავის დაპროექტება გადაწყვეტილებათა მიღების მხარდაჭერი სისტემისათვის**

მართვის კომპიუტერული სისტემების პროგრამული უზრუნველყოფის აგების პროცესების სრულფასოვანი ავტომატიზაცია კომპონენტურ-ვიზუალური დაპროგრამების სახელწოდებით დამკვიდრდა და იგი მოდელების გრაფო-ანალიზურ წარმოდგენას ეყრდნობა. ასეთი ინსტრუმენტები ფლობს როგორც პირდაპირ (გრაფიკიდან პროგრამული კოდისაკენ), ასევე რეგერსიულ (კოდიდან გრაფიკისაკენ) ტექნოლოგიას [43,63,64].

შეიძლება აღვნიშნოთ, რომ UML-ტექნოლოგია დაპროგრამების ენებისა და მონაცემთა ბაზების მართვის სისტემების ინტეგრირებული გამოყენებისა და მის საფუძველზე შექმნილი ინსტრუმენტების (Rational Rose, Ms Visio, ParadigmPlus და სხვ.) მეთოდოლოგიას წარმოადგენს [15,18]. ფორმალურად იგი შეიძლება განვიხილოთ როგორც „წარმოებული კლასი“ დაპროგრამების ენისა და მონაცემთა ბაზების მართვის სისტემის „საბაზო კლასებიდან“, რომელსაც გააჩნია როგორც „მშობლების“ თვისებები, ასევე ახალი, მძლავრი ვიზუალური მახასიათებლები.

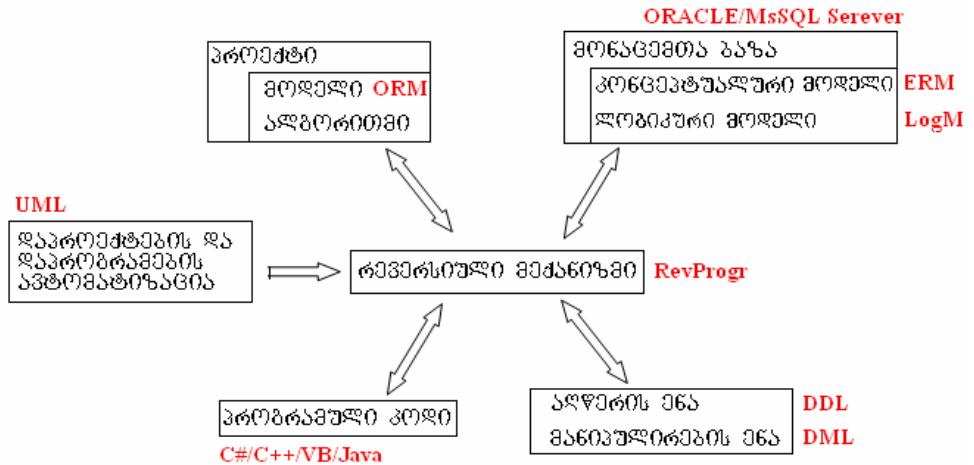
რელაციურ მონაცემთა ბაზების მართვის სისტემები დღესაც აქტუალურ მიმართულებად ითვლება (მაგალითად, Oracle, MsAccess, MsSQLServer, InterBase, MySQL და სხვ.). ამ სისტემებში ერთ-ერთი მნიშვნელოვანი კომპონენტია საპრობლემო სფეროს კონცეპტუალური სქემა ანუ ER-მოდელები (Entity-Relations Model), რომლის საფუძველზეც

დაპროექტდება შემდგომში მონაცემთა ბაზის ლოგიკური სტრუქტურა [31].

დაპროგრამების ენებთან შედარებით მგმს იყო პირველი ცდა პროგრამულ კოდში მონაცემთა სტრუქტურების აგების ავტომატიზაციისათვის. ობიექტ-ორიენტირებულ დაპროგრამების ენებში და UML-ტექნოლოგიაშიც ეს კონცეპტუალური ER-მოდელები, „კლასების დიაგრამების“ სახით, მნიშვნელოვანი ვიზუალური კომპონენტია. მომხმარებელი Use Case (გამოყენებითი შემთხვევა) დიაგრამებიდან ააგებს კლასების, შემდეგ კი კომპონენტების დიაგრამებს, რომლებიც საბოლოოდ ფიზიკური განლაგების დიაგრამებში აისახება [15].

90-ანი წლებიდან მნიშვნელოვნად განვითარდა სტრუქტურული დაპროგრამებისა და ობიექტ-ორიენტირებული მოდელირების კონცეფციები (C++, Java), ხოლო 2000 წლიდან დღემდე ეს ენები კომპონენტური და ვიზუალური თვისებებით გამდიდრდა. C#, XML, Visual-C++, Visual Basic, J++ და სხვ. კომპიუტერული დაპროგრამების ის ინტეგრირებული პაკეტებია რომლებიც დღეისათვის ყველაზე პოპულარული და აქტუალურია ამჟრიკისა და ევროპის თითქმის ყველა უნივერსიტეტსა და ბიზნესის მართვის სფეროში [54-:60].

განსაკუთრებით საყურადღებოა ამ თვალსაზრისით უახლესი საინფორმაციო ტექნოლოგია, რომელიც .NET პლატფორმითაა ცნობილი. იგი ადჭურვილია ისეთი სპეციალური მედიატორული თვისებებით, რომლებიც უზრუნველყოფს ზემოხამოთვლილ დაპროგრამების ენებს შორის სრულ თავსებადობას [74-77,79,82,83]. წინამდებარე სტატიის მიზანი სწორედ ამ ტექნოლოგიაზე მონაცემთა რელაციური ბაზების აგების (ORM/ERM – მოდელების) ავტომატიზაციის საკითხები არის წარმოდგენილი (ნახ.3.4).

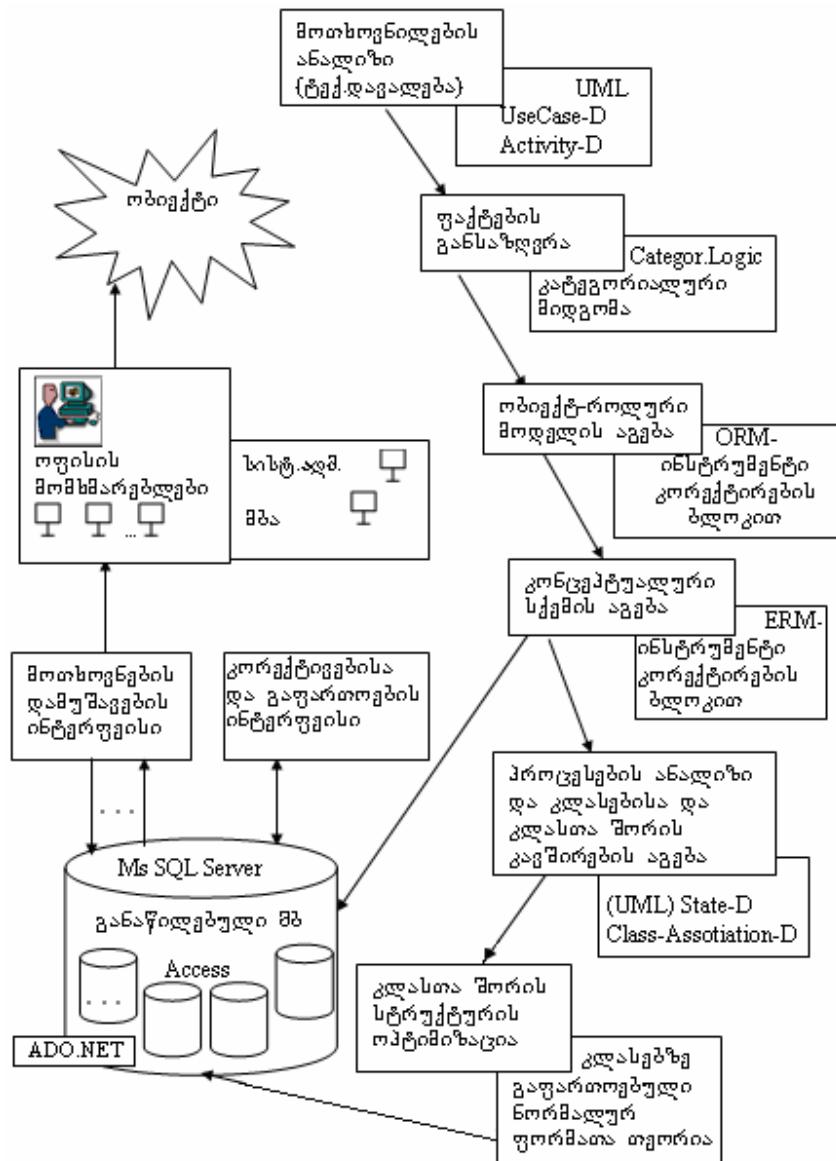


### ნახ.3.4. მბ-ის დაპროექტების პროცესის რევერსიული მექანიზმის კომპონენტები

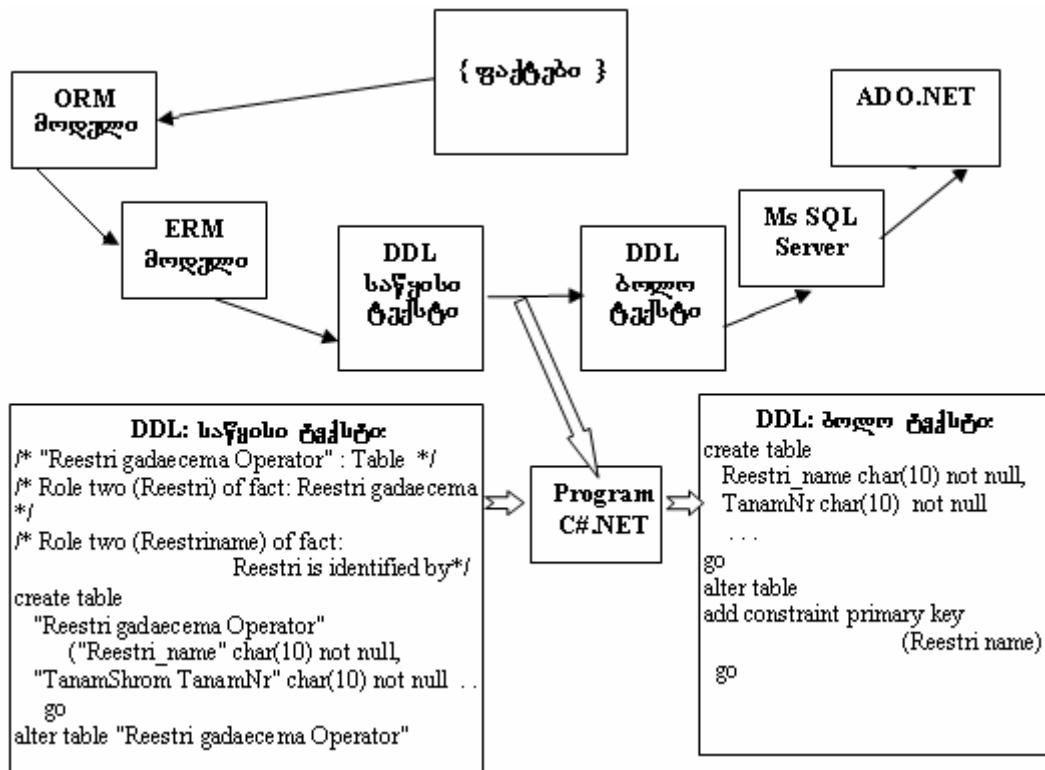
განიხილება განაწილებული ოფის-ობიექტების მონაცემთა ბაზების ობიექტ-ორიენტირებული მოდელირებისა და ობიექტ-ორიენტირებული დაპროექტების ავტომატიზებული პროცესების დამუშავების ამოცანა, რაც საგრძნობლად ამცირებს სისტემების ინფორმაციული და პროგრამული პაკეტების აგების დროს. 3.5 ნახაზზე მოცემულია საპრობლემო სფეროს მოდელირებისა და მონაცემთა ბაზაში აგტომატიზებული ასახვის ამოცანის ძირთადი ეტაპებისა და მათი რეალიზაციის ინსტრუმენტული საშუალებების სქემა.

ცოდნა, რომელიც საპრობლემო სფეროს შესახებ გააჩნია მომხმარებელს, სპეციალური ინტერფეისების საშუალებით, რომელთა საფუძველს ფორმალური ენის გრამატიკის კატეგორიები და ლოგიკურ-ალგებრული მეთოდები შეადგენს, გადაეცემა ობიექტ-როლური მოდელირების კომპიუტერულ პროგრამას [25].

3.6 ნახაზზე მოცემულია ERM-მოდელის შესაბამისი DDL-ფაილის ავტომატიზებული კორექტირების პროცედურები.



ნახ.3.5. მგბ-ის დაპროექტების პროცესის ავტომატიზაციის ზოგადი სქემა



ნახ.3.6. DDL ფაილების ავტომატიზებული კორექცია

ამგვარად, განაწილებული სისტემების მონაცემთა ბაზების დაპროექტება/აგების ამოცანების გადაწყვეტა დაპროგრამების უნიფიცირებული, ვიზუალურ-კომპონენტებური, ობიექტ-ორიენტირებული მეთოდებითა და შესაბამისი ინსტრუმენტებით საგრძნობლად ამცირებს სისტემების რეალიზაციის დროს და ამაღლებს მისი ფუნქციონირების ხარისხს და მოქნილობას.

### 3.3. გადაწყვეტილების მიღების მხარდაჭერ სისტემაში მრავალფაქტორული ამოცანის გადაწყვეტის პროგრამული რეალიზაცია

წინამდებარე პარაგვაფში განხილულია ბიზნეს-პროცესების ოპერატიული ანალიზის OLAP-ისტრუმენტის გამოყენების საკითხი [8]. შემოთვაზებულია განაწილებული მონაცემთა ბაზებიდან შერჩეული მონაცემების ერთიან გადაწყვეტილების მიღების მხარდაჭერ სისტემაში სტრუქტურიზებული ორგანიზება და შესაბამისი პროგრამული პაკეტის რეალიზაცია ობიექტ-ორიენტირებული დაპროგრამების C++ ენის Decision

Cube კომპონენტის გამოყენებით მრავალფაქტორული ანალიზის ამოცანებისთვის [19,61].

OLAP (Online Analyzing Processing) – ტექნოლოგიის ძირითადი მოთხოვნები და მისი რეალიზაციის პრინციპები შემოთავაზებული იყო ცნობილი ამერიკელი მეცნიერის, ედგარ კოდის მიერ 1993 წელს [32]. როგორც ცნობილია, ის იყო რელაციური მოდელებისა და ბაზების პირველი ავტორი და ამ მიმართულების დამაარსებელი [33]. დღეისათვის ეს ორი საკითხი მონაცემთა საცავში გადაიკვეთა, რომ იგი უნდა იყოს რელაციური ბაზების საფუძველზე აგებული და მასში გამოიყენებოდეს ონლაინური ოპერატიული ანალიზის სისტემა.

OLAP ტექნოლოგიის გამოყენებით შესაძლებელია სხვადასხვა განაწილებული მონაცემთა ბაზებიდან მონაცემთა მოპოვება და მათზე ანალიზის ჩატარება. ამ ტექნოლოგიის არსი მდგომარეობს ინფორმაციის მრავალგანზომილებიანი კუბით წარმოდგენაში, რომელშიც მოსახერებებებით ინფორმაციის მანიპულირება.

იმისათვის, რომ კარგად გავაანალიზოთ, თუ რატომ არის ხელსაყრელი ინფორმაციის წარმოდგენა კუბის სახით და მისი ანალიზი, განვიხილოთ იგი ერთ-ერთ ორგანიზაციაში პროდუქციის მიმოქცევის მაგალითზე.

პირველ ეტაპზე ხდება ინფორმაციის თავმოყრა, რომელიც შეეხება საქონლის შემოსვლას, დატვირთვას, შეკვეთას, ანგარიშსწორებას და ა.შ. ამის შემდეგ უნდა შესრულდეს დოკუმენტბრუნვასთან დაკავშირებული ყველა პრობლემის გადაწყვეტა, ბოლოს კი დგება საკითხი, სისტემიდან საჭირო ინფორმაციის დროულად მისადებად.

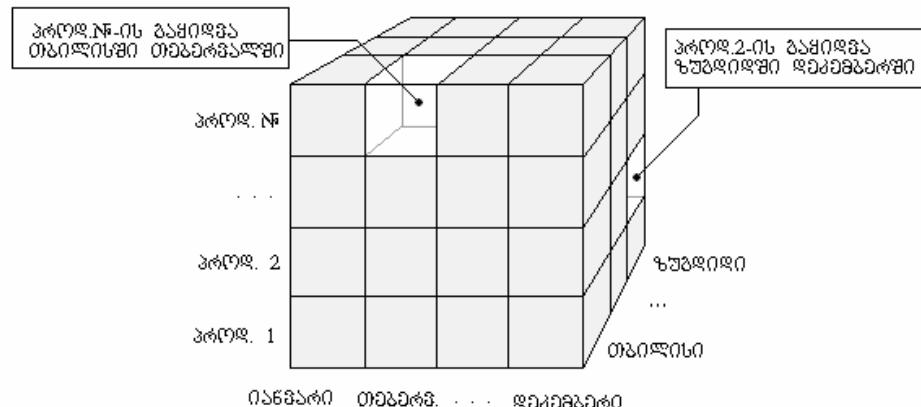
სისტემის მიმართ მოთხოვნა შესაძლოა გაუჩნდეს ორგანიზაციის ნებისმიერ თანამშრომელს, მაგალითად, თუ რა რაოდენობის პროდუქციაა დარჩენილი სარეალიზაციოდ, როგორ არის დაჯგუფებული არსებული მონაცემები, რა მოგება დაგვიტოვა გაყიდულმა პროდუქციამ და ა.შ. მუშაობის პროცესში წარმოქმნილ ყველა შესაძლო მოთხოვნის დასაკმაყოფილებლად, ერთ-ერთი ოპტიმალური ვარიანტია **OLAP** ტექნოლოგიის გამოყენება და არსებული ინფორმაციის კუბში წარმოდგენა. **OLAP** სისტემის მუშაობის ზოგადი

პრინციპი საკმაოდ მარტივია, პირველ ეტაპზე ანგარიში წარმოვადგინოთ 3.1 ცხრილის სახით, რომლშიც არსებული მონაცემები განთავსდება სამგანზომილებიან კუბში:

ცხრ.3.1

ქალაქი	პროდ. დასახელ	იანვარი	თებერვალი	მარტი	აპრილი
თბილისი	უთო	10	22	15	47
	მტვერსასრუტი	2	7	5	14
	ჩაიდანი	17	34	20	71
ჯამი		29	63	40	132
გორი	მაცივარი	2	0	3	5
	ჩაიდანი	5	6	3	14
	ტელეფონი	12	22	7	41
ჯამი		19	28	13	60
ზუგდიდი	უთო	7	7	5	19
	ტელეფონი	10	12	15	37
	მტვერსასრუტი	2	3	0	5
ჯამი		19	22	20	61

3.7 ნახაზზე წარმოდგენილია სამგანზომილებიანი კუბი. განზომილებათა რაოდენობა შეიძლება შევირჩეს ნებისმიერი.



ნახ.3.7. მონაცემები სამგანზომილებიან კუბში

თუ მნიშვნელობათა განსაზღვრას ვაწარმოებთ ვერტიკალურად, მაშინ მივიღებთ ანგარიშთა 3.2 ცხრილს:

**ცხრ.3.2**

ქაღაპი	იავგარი	თებერვალი	მარტი	აპრილი
თბილისი	29	63	40	132
გორი	19	28	13	60
ზუგდიდი	19	22	20	61
ჯამი	67	113	73	253

კუბთან მუშაობა საშუალებას გვაძლევს დავაჯგუფოთ მონაცემები და წარმოვადგინოთ სხვადასხვა ჭრილში. ეს არის პრობლემათა გადაჭრის ერთ-ერთი მთავარი პრინციპი. მონაცემთა ასეთი სახით წარმოდგენა აიოლებს ინფორმაციასთან მუშაობას.

იმისათვის, რომ მივიღოთ კარგი შედეგი, აუცილებელია ეკრენზე გამოვიდეს არა მთლიანად კუბი არამედ მხოლოდ ის ნაწილი, სადაც წარმოდგენილია ჩვენთვის საჭირო ინფორმაცია, თუ არ გვაინტერესებს კონკრეტული ინფორმაცია მაგ. რომელიმე ქალაქის შესახებ, სადაც აწარმოებენ პროდუქციის გაყიდვას თავიდანვე ამოვაგდებთ განზომილებას „ქალაქი“.

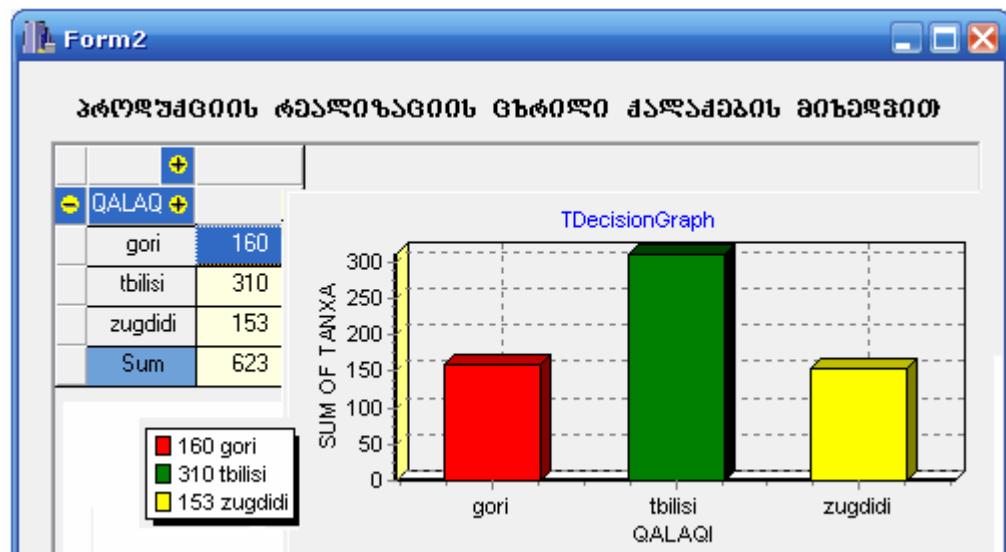
OLAP- სისტემასთან მუშაობა მომხმარებლისათვის უნდა იყოს იოლი და მინიმალური დროის განმავლობაში მიეწოდოს მაქსიმალურად ამომწურავი პასუხი დაყენებულ მოთხოვნაზე.

3.8 და 3.9 ნახაზებზე მოცემულია ჩვენი მაგალითისათვის რეალიზებული მომხმარებლის ინტერფეისის ფრაგმენტები პროდუქციის რეალიზაციის ჯამური თანხებისა და ამ თანხების ქალაქების მიხედვით გადანაწილების შესახებ. პროგრამირების სამუშაო გარემოდ გამოყენებულია გიზუალური, ობიექტ-ორიენტირებული დაპროგრამების პაკეტი Borland C++ Builder. „+“ სიმბოლო საშუალებას იძლევა ვმართოთ მრავალფაქტორიანი ანალიზის ცხრილები. მაგალითად, 3.10 ნახაზზე წარმოდგენილია თანხების განაწილება მხოლოდ თვეების მიხედვით, ხოლო 3.11 ნახაზზე ქალაქებისა და თვეების მიხედვით.

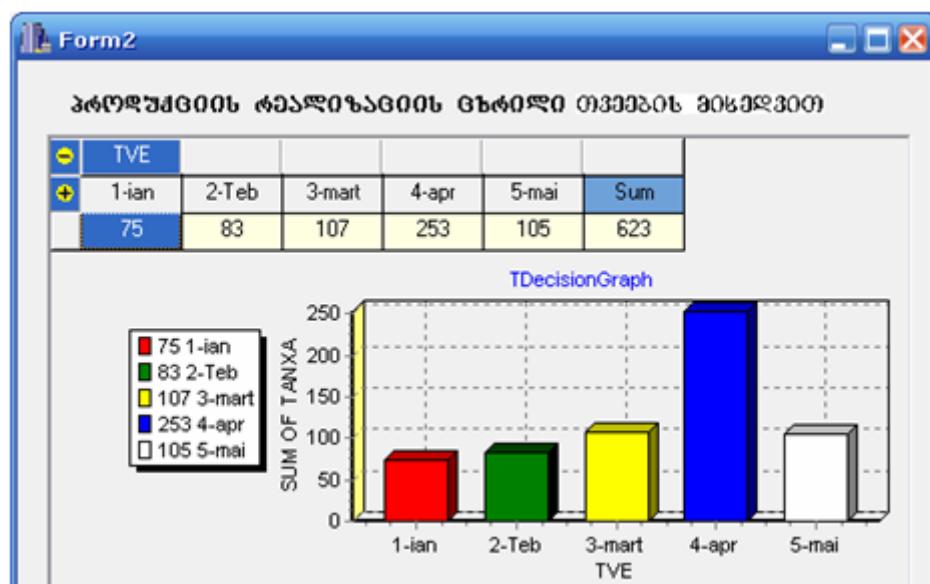
### პროცესის რეალიზაციის ცხრილი

	სულ:
	623 ათასი ლარი

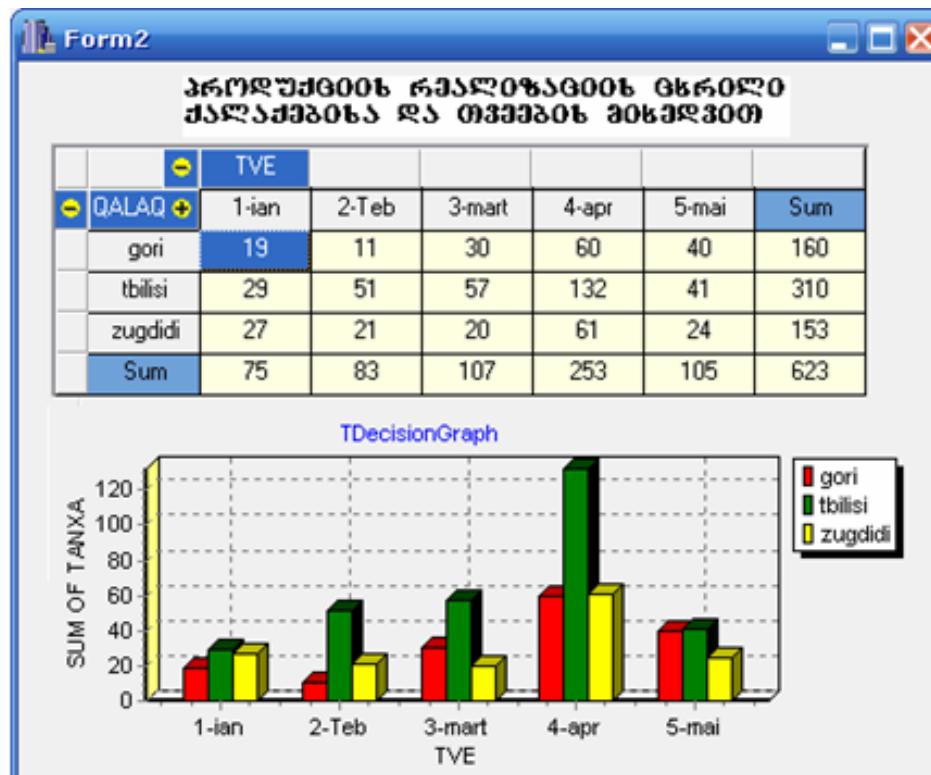
ნახ.3.8. ჯამური რეალიზაციის თანხები



ნახ.3.9. თანხები ქაღაპების მიხედვით



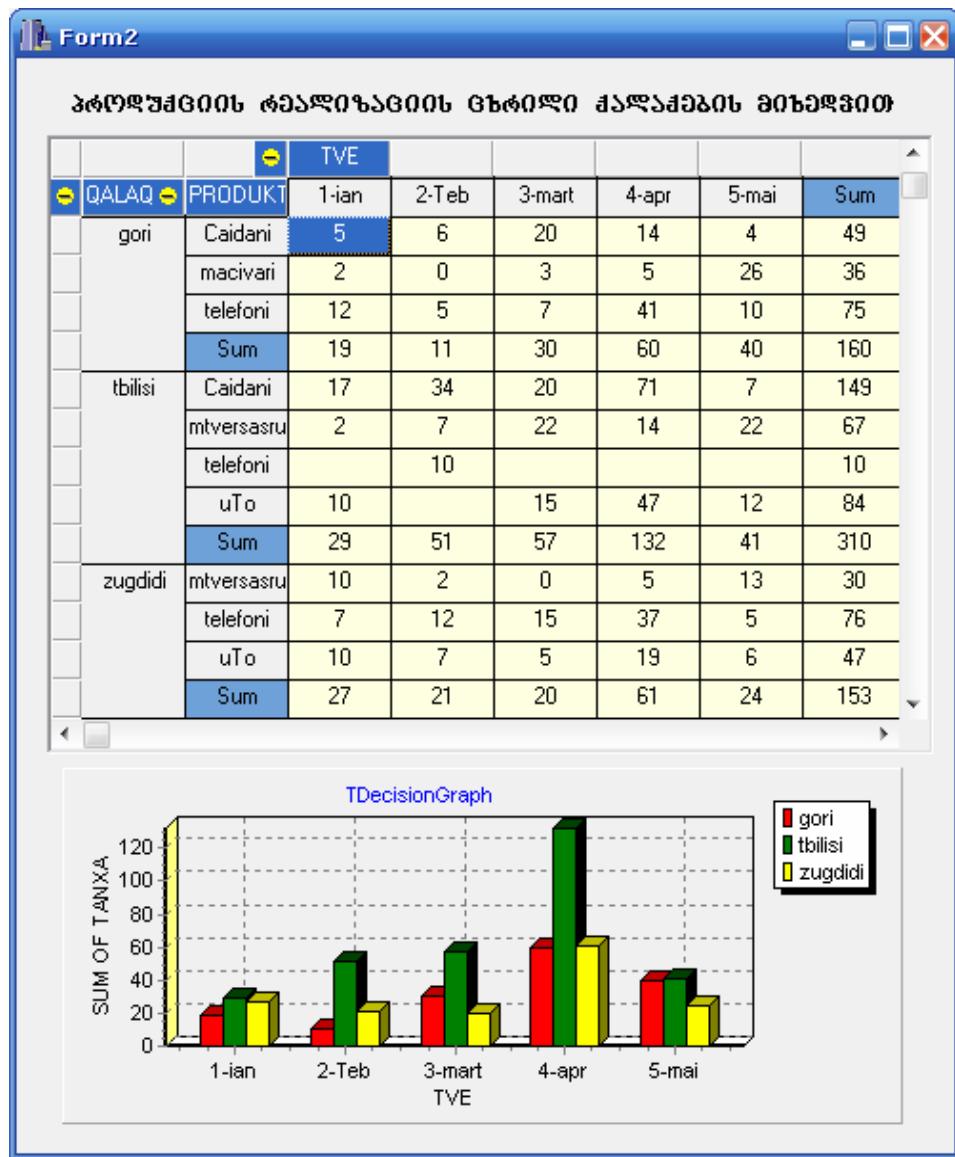
ნახ.3.10. პროდუქციის რეალიზაცია თვეების მიხედვით



ნახ.3.11. პროდუქციის რეალიზაცია ქალაქებისა და თვეების მიხედვით

3.12 ნახაზზე მოცემულია პროდუქციის რეალიზაციის სურათი სამივე ფაქტორის გათვალისწინებით: ქალაქი, პროდუქცია და თვეები. ამასთანავე, ცხრილის სტრიქონებსა და სვეტებში მოცემულია ჯამური თანხების (Sum) მნიშვნელობები ქალაქების და თვეების მიხედვით.

ამგვარად, OLAP-კონცეფცია არის ინფორმაციის მოპოვების ინსტრუმენტი, რომლის გამოყენება შესაძლებელა მრავალფაქტორიული ანალიზისათვის. ამ კონცეფციის რეალიზაცია ვიზუალური, ობიექტორიენტირებული დაპროგრამების ინსტრუმენტების საშუალებით იძლევა მოქნილი და ეფექტური ინტერაქციები აგების საშუალებას მრავალფაქტორული ამოცანების გადასაწყვეტილ, რომელთა ინფორმაცია განთავსებულია რელაციურ მონაცემთა ბაზებში.



ნახ. 3.12. ცხრილი სამი ფაქტორის გათვალისწინებით

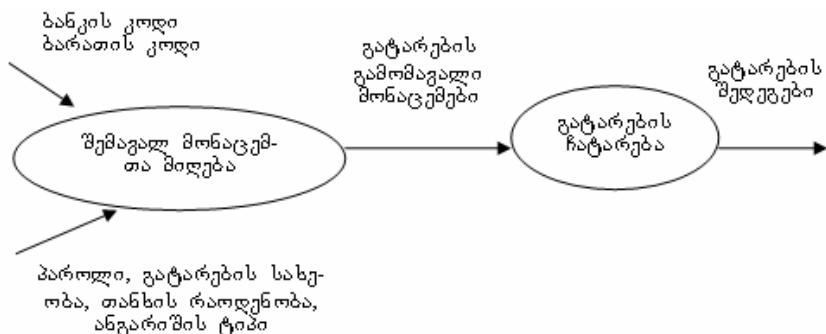
### 3.4. ინფორმაციული ნაკადების დიაგრამები ავტომატიზებული საბანკო სისტემისთვის

ფუნქციონალური მოდელი წარმოადგენს მონაცემთა ნაკადების დიაგრამების ნაკრებს (მნდ), რომელიც აღწერს ოპერაციის არსეს [29]. მნდ ასახავს სისტემაში არსებულ მნიშვნელობების ფუნქციურ დამოკიდებულებას, შემავალი და გამომავალი მნიშვნელობების ჩათვლით. მნდ – არის გრაფიკი, რომელზეც მოცემულია მონაცემთა მოძრაობა მათი წარმოშობის წყაროებიდან მომხმარებლამდე, გარკვეული პროცესების გავლით. მნდ ასახავს პროცესებს, რომლებიც

გარდაქმნის მონაცემებს, მათ საცავს, ნაკადებს და მათი წარმოშობის ობიექტებს.

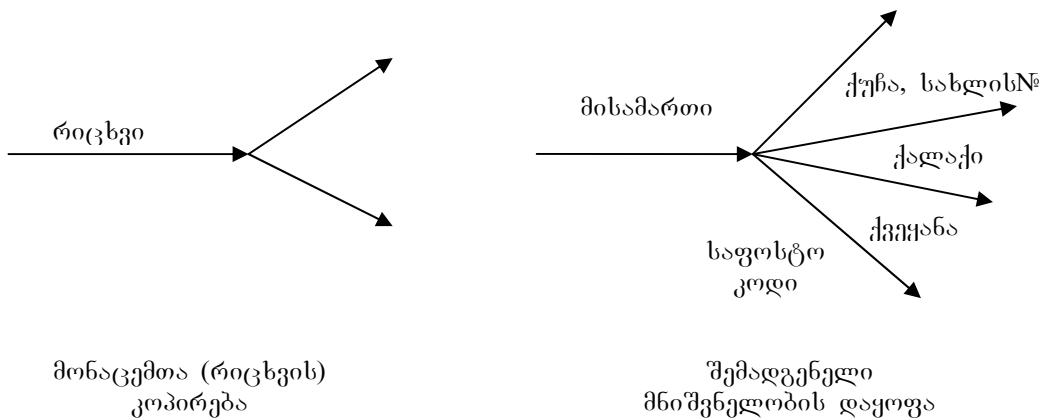
**პროცესები:** როგორც ზემოთ აღვნიშნეთ, პროცესები გარდაქმნის მონაცემთა მნიშვნელობებს. ყველაზე ქვედა დონის პროცესები წარმოადგენს ფუნქციებს გარე ეფექტების გარეშე. ასეთი ფუნქციების მაგალითად შეიძლება მოვიყვანოთ: „ორი რიცხვის ჯამის გამოთვლა”, „საბანკო ბარათით ჩატარებული გატარების საკომისიო ნაერთის გამოთვლა”. მონაცემთა ნაკადის მთლიანი გრაფი ასევე წარმოადგენს პროცესს, თუმცა უფრო მაღალი დონისას. პროცესს შეიძლება გააჩნდეს გარე ეფექტები, თუ იგი შეიცავს არაფუნქციურ კომპონენტებს, როგორიცაა მონაცემთა საცავი ან გარეშე ობიექტები.

მონაცემთა ნაკადების დიაგრამაზე პროცესი გამოსახულია ელიფსის სახით, რომლის შიგნითაც თავსდება პროცესის სახელი. ყოველ მათგანს აქვს შემავალი და გამომავალი მონაცემების ფიქსირებული რაოდენობა (ნახ.3.13).



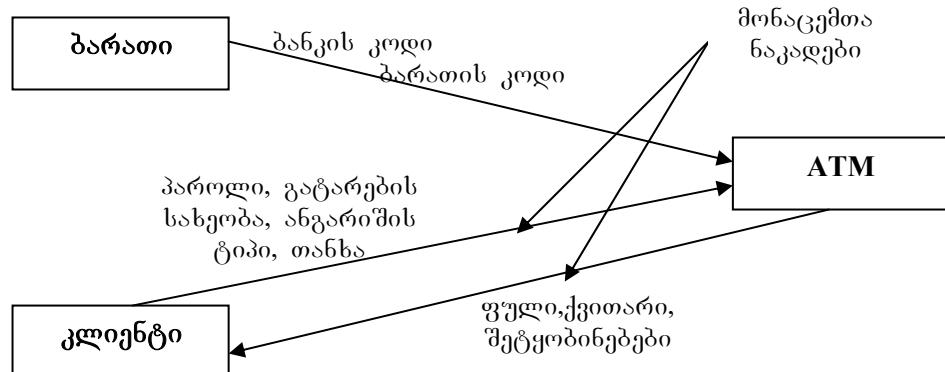
### ნახ.3.13. პროცესთა ნაკადების დიაგრამის ფრაგმენტი

**მონაცემთა ნაკადები:** მონაცემთა ნაკადები აკავშირებს ერთ ობიექტს (ან პროცესს) მეორესთან. მათი მაგალითები მოცემულია 3.14 ნახ.აზზე. ისინი გამოისახება ისრების სახით და აკავშირებს ერთმანეთთან მონაცემების მწარმოებლებსა და მომხმარებლებს. პირველ მაგალითზე ნაჩვენებია მონაცემთა კოპირება, ორ მაგალითზე ერთიდაიმავე მნიშვნელობის გადაცემისას, მეორეზე – სტრუქტურის გელებად დაყოფა, სხვადასხვა მაგალითზე სხვადასხვა გელების გადაცემისას.



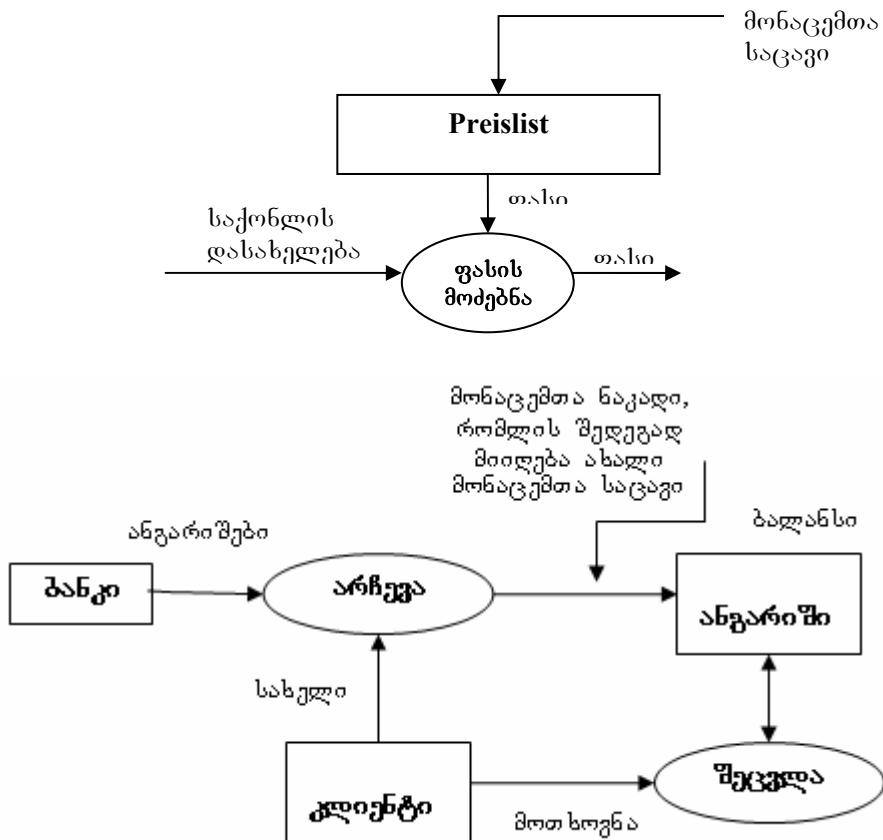
### ნახ.3.14. მონაცემთა ნაკადების დიაგრამის ფრაგმენტი

**აქტიური ობიექტები:** აქტიური ობიექტი ეწოდება ისეთ ობიექტს, რომელიც უზრუნველყოფს მონაცემთა მოძრაობას, მათ მიწოდებას და მოხმარებას. როგორც წესი, აქტიური ობიექტები დაკავშირებულია მონაცემთა ნაკადების დიაგრამების შესასვლელებთან და გამოსასვლელებთან. 3.15 ნახაზზე ისინი გამოსახულია მართვულებით.



### ნახ.3.15. აქტიური ობიექტების დიაგრამის ფრაგმენტი

**მონაცემთა საცავი:** მონაცემთა საცავი არის პასიური ობიექტი მონაცემთა ნაკადების დიაგრამაზე, რომელშიც მონაცემები ინახება შემდგომ გამოყენებამდე. მისი მაგალითები მოცემულია 3.16 ნახაზზე. მონაცემთა აგრეგატული საცავები, როგორიცაა მაგალითად სიები და ცხრილები, უზრუნველყოფს მონაცემთა შეღწევას იგივე თანმიმდევრობით, როგორც ისინი იქნენ მიღებული.

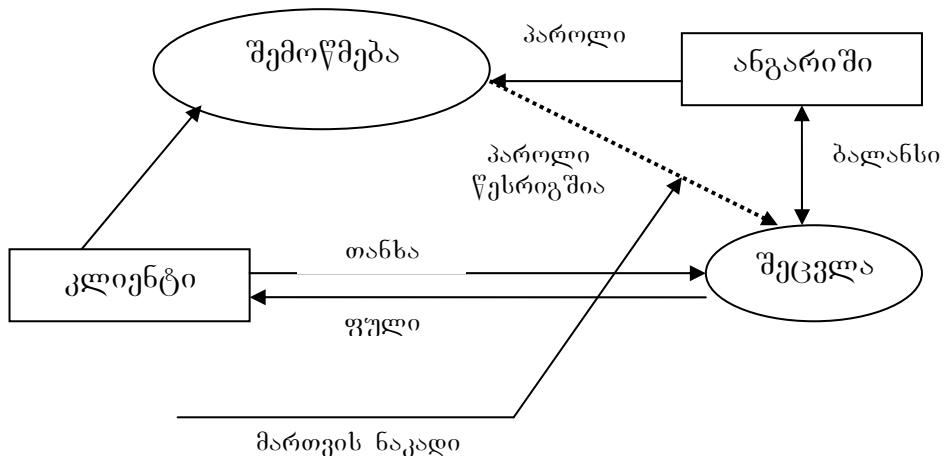


ნახ.3.16. მონაცემთა საცავის დიაგრამის ფრაგმენტები

**მართვის ნაკადები:** მონაცემთა ნაკადების დიაგრამა გვიჩვენებს მნიშვნელობების გამოთვლის ყველა გზას, მაგრამ არ გვიჩვენებს გამოთვლის თანმიმდევრობას.

გამოთვლის თანმიმდევრობის გადაწყვეტილება დაკავშირებულია პროგრამის მართვასთან და მიიღება სპეციალური ფუნქციებით, ე.წ. პრედიკატებით. ისინი ქმნის შესაბამის პირობებს პროცესების ჩასატარებლად. როგორც წესი, პრედიკატების ჩართვა ფუნქციურ მოდელში აუცილებელი არაა. ფუნქცია, რომელიც ღებულობს გადაწყვეტილებას პროცესის გაშვებაზე წარმოქმნის მართვის ნაკადს და მონაცემთა ნაკადების დიაგრამაზე გამოისახება წყვეტილი ისრით.

3.17 ნახაზზე გამოსახულია მართვის ნაკადის მაგალითი: კლიენტს, რომელსაც სურს ანგარიშიდან ფულის მოხსნა, შეაქვს ATM-ში პაროლი და თანხის რაოდენობა.



ნახ.3.17. მართვის ნაკადის დიაგრამის ფრაგმენტი

მიუხედავად იმისა, რომ მართვის ნაკადების ჩართვა მონაცემთა ნაკადების დიაგრამაზე სასარგებლოა, გასათვალისწინებელია, რომ ეს იწვევს მის დინამიკურ მოდელში შემავალი ინფორმაციის დუბლირებას.

პროცესები მონაცემთა ნაკადების დიაგრამაზე საბოლოოდ რეალიზებულ უნდა იქას როგორც ობიექტის ოპერაციები. ამასთან ზედა დონის პროცესების რეალიზაცია შეიძლება განსხვავდებოდეს მათი წარმოდგენისგან, რადგან მათი რეალიზაციისას წარმოებს მათი ოპტიმიზაცია.

ყველა ოპერაცია შეიცავს მის სიგნატურას, ანუ ოპერაციის სახელს, რაოდენობას, თანმიმდევრობას, მისი პარამეტრებისა და მის მიერ გაცემული მნიშვნელობების ტიპებს და მისი ეფექტების აღწერას. ოპერაციის ეფექტების აღწერისათვის შეიძლება გამოვიყენოთ: მათემატიკური ფორმულები, ცხრილისებური ფუნქციები, განტოლებები, რომლებიც აკავშირებენ შემავალ და გამომავალ მნიშვნელობებს, ოპერაციის აქსიომატური გადაწყვეტილება, გადაწყვეტილების მიღების ცხრილები, ფსევდოკოდი და ბუნებრივი ენა.

ოპერაციის აღწერის მაგალითი, რომელშიც მისი ეფექტებია ასახული ბუნებრივი ენით, მოცემულია 3.18. ნახ.აზზე. ეფექტის აღწერისას გამოყენებულია ოპერაციები: „გატარების შეწყვეტა”, „მოთხოვნის გაცემა”, „ფულის გაცემა”, „ანგარიშის დებეტირება”, „ანგარიშის დაკრედიტება”.

**ოპერაცია:** “ანგარიშის შეცვლა” (ანგარიში, თანხა, გატარების ტიპი)  
**შედეგი:** ფული, ქვითარი

**თუ** თანხა უნდა მოიხსნას ანგარიშიდან და იგი აღემატება ანგარიშის  
ბალანსს, მაშინ - “გატარების შეწყვეტა”.

**თუ** თანხა იხსნება და იგი ნაკლებია ბალანსზე, მაშინ  
“თანხის დებეტირება” და “ფულის გაცემა”.

**თუ** ანგარიშზე ფული შეიტანება, მაშინ “ანგარიშის დაკრედიტება”.

**თუ** შეტანილია მოთხოვნა, მაშინ “მოთხოვნის გაცემა”.

ნებისმიერ შემთხვევაში ქვითარი უნდა შეიცავდეს:  
ATM-ის ნომერს, თარიღს, დროს, ანგარიშის ნომერს, გატარების ტიპს,  
თანხას, ანგარიშის ახალ ბალანსს.

### ნახ.3.18. ოპერაცის აღწერის მაგალითი

ინფორმაციული ტექნოლოგიის ყველა ოპერაცია შეიძლება  
დაიყოს სამ კატეგორიად: მოთხოვნები, მოქმედებები და აქტიურობები.  
მოთხოვნად ჩაითვლება ოპერაცია ობიექტის გარეშე.  
მახასიათებლებისათვის უკურეაქციის გარეშე.

მოქმედებად იწოდება ოპერაცია, რომელსაც გააჩნია სისტემის  
ობიექტები მოქმედი გვერდითი მოვლენები. ყოველი მოქმედება გა-  
ნისაზღვრება ობიექტის ატრიბუტების და კავშირების ცვლილებებით.

აქტიურობა ეწოდება ობიექტის მიერ ან ობიექტზე წარმოებულ  
ოპერაციას, რომლის შესრულებას სჭირდება გარკვეული დრო. მას  
ახასიათებს გარეშე მოვლენები. აქტიურობები შეიძლება გააჩნდეს  
მხოლოდ აქტიურ ობიექტებს, რადგან პასიური ობიექტები უბრალოდ  
მონაცემთა შემნახველებია.

### 3.5. კორპორაციული მართვის სისტემის Web-აპლიკაციის დამუშავება Internet-Intranet გარემოში .NET-პლატფორმაზე

განიხილება კორპორაციული მართვის სისტემების ვებ-  
აპლიკაციების დაპროექტების და რეალიზაციის საკითხები. საფინანსო  
ბანკის კლიენტთა მომსახურების მაგალითზე ილუსტრირებულია მათი  
ინტერფეისული კომპონენტების აწყობისა და მონაცემთა სერვერული  
ბაზების ორგანიზების ამოცანები. სისტემა დამუშავებულია .NET-

პლატფორმაზე, C#, ASP.NET, ADO.NET და SQL Server ობიექტ-ორიენტირებული ინსტრუმენტების გამოყენებით [45].

კორპორაციული სისტემები და მათი მართვის მექანიზმები ხასიათდება განსაკუთრებული სირთულით, დიდი ინფორმაციული ნაკადების ოპერატორულად დამუშავებისა და გადაწყვეტილების მიღების მცირე დროის არსებობის თვალსაზრისით, რაც აუცილებლად მოითხოვს ამ ორგანიზაციაში თანამედროვე საინფორმაციო ტექნოლოგიების დანერგვას.

ნაშრომში განსაკუთრებული ყურადღება ექცევა საფინანსო ბანკებში (და არა მხოლოდ აქ) საინფორმაციო-მომსახურების სისტემების დაპროექტებას WEB ტექნოლოგიით, ნაცვლად სტანდარტული Windows დანართებისა (აპლიკაციებისა). აქტუალურად მიგვაჩნია ასეთი სისტემების შემუშავება .NET-პლატფორმაზე, C#, ADO.NET და ASP.NET დაპროგრამების ვიზუალური, ობიექტ-ორიენტირებული, ინტეგრირებული ინსტრუმენტების გამოყენებით [8,15].

დასაპროექტებელი მართვის ინფორმაციული სისტემის ძირითადი მოთხოვნები ასე განისაზღვრა:

- ინფორმაციის უსაფრთხოების და დაცვის მაღალი დონე;
- სისტემასთან ურთიერთობის გამარტივებული და მისი მომსახურების სიადვილე;
- სისტემის მომხმარებლებთან ურთიერთობის მეგობრული ინტერფეისის არსებობა.

ნაშრომში შემოთავაზებულია საბანკო სისტემაში ვებ-ტექნოლოგიაზე დაფუძვნებული ინფორმაციული სისტემების დაპროექტების და მისი შემდგომი რეალიზაციის საკითხები.

ისეთ საფინანსო ორგანიზაციის ფილიალებში, როგორიცაა ბანკი, ინტერნეტ- ინტრანეტის პირობებში გაიზრდება უსაფრთხოება და სისტემის დაცვა. ინტრანეტი იცავს ორგანიზაციას თავისი შიდა ფაილებისა და კონფიდენციალური ინფორმაციის წვდომისაგან გარე პირთათვის. იგი ხშირად გამოიყენება ფაილების და მეილების ორგანიზაციის წევრთათვის ერთობლივი წვდომისათვის, და ამავე დროს გარე პირთათვის იგივე ინფორმაციის ბლოკირებისათვის.

კორპორაციის შიგა ვებ-პროგრამები, რომელიც შეზღუდულია სპეციფიკური მომხმარებლისა თუ კომპიუტერებისათვის, დიდად გამოსაყენებელია ფინანსურ ბანკებში, რამეთუ შიგა ქსელის დამოწმებით და ინტრანეტის გაყვანით, ყველა აუცილებელი ოპერაცია სრულდება და ამავდროულად ვირუსების, ტროიანების და ა.შ. საფრთხე მკვეთრად მცირდება.

ჩვენ ვიხილავთ კლიენტ-სერვერულ სისტემას, სადაც გვაქვს მხოლოდ ერთი დიდი სერვერი, რომელიც მოთავსებულია ბანკის (პირობითად) სათაო ოფისში. დანარჩენ ფილიალებში მოთავსებული კომპიუტერები კი წარმოადგენს კლიენტებს. ანუ საქმე გვაქვს ცენტრალიზებულ სისტემასთან, რაც მკვეთრად ამცირდებს ადმინისტრირების ხარჯებს.

ვინდოუსის-პროგრამების შემთხვევაში სისტემის „მწყობრიდან გამოსვლის“ სიხშირე გაცილებით მაღალია, ამიტომაც საჭიროებს მუდმივი თვალყურის დევნებას ადმინისტრატორის მიერ. ამ შემთხვევაში აუცილებელია პროგრამული უზრუნველყოფის დაყენება ყველა მომხმარებლის კომპიუტერზე. ინსტალირების პროცესი მოითხოვს დროს, ხოლო ახალი ვერსიების გამოსვლა კი ხელს უწყობს ამ დროს გაზრდას. რადგან როდესაც პროგრამული უზრუნველყოფის ახალი ვერსია გამოდის საჭიროა მისი ყოველ კომპიუტერზე ხელახლი დაინსტალირება, რაც თავისთვად გამოიწვევს დროს ხარჯვას.

ვებ-პროგრამის შემთხვევაში ყველა ეს პრობლემა იხსნება. რადგან ვებ პროგრამა ჩაწერილია სერვერზე, და პროგრამისტთა ერთი ჯგუფიც კი საკმარისია, რათა ყველა ხარვეზი ადგილზევე, სერვერზევე აღმოიფხვრას.

როდესაც ვინდოუსის პროგრამებთან გვაქვს საქმე, ისიც უნდა გავითვალისწინოთ, რომ იმ კომპიუტერზე, სადაც ინსტალირდება ეს პროგრამა, უნდა იყოს დაინსტალირებული მთელი რიგი სხვა პროგრამები რათა ამ უკანასკნელმა იფუნქციონიროს. ჩვენს მიერ შემოთავაზებული ვებ-პროგრამისთვის კი სულერთია კლიენტის კომპიუტერზე რომელი ოპერაციული სისტემა იქნება დაინსტალირებული, ის ერთნაირად კარგად იმუშავებს, როგორც Windows-ის, ასევე Linux და Unix ოპერაციულ სისტემებში [34].

ვებ-პროგრამა აგებული გვაქვს Visual Web Developer 2005 Express-ის საშუალებით, რომელიც არის Microsoft Visual Studio 2005 ოჯახის წევრი და ASP.NET-ზე ვებ პროგრამების დაწერის საუკეთესო საშუალებას იძლევა. როგორც Express გამოცემა, ის წარმოადგენს Visual Studio Standard-ის უფრო გაუმჯობესებულ ვარიანტს [35].

Visual Web Developer სპეციალურად აწყობილია ვებ პროგარმების დასაწერად და იყენებს ახალ ვებ-პროფილს, რომელიც მენიუს და ფანჯრის ვებ-პროგრამირებისთვის ოპტიმიზირებულ ვარიანტებს გვთავაზობს. პროგრამირების გარემო შეიცავს HTML კოდის რედაქტორს, ვებ გვერდების გაუმჯობესებულ დიზაინერს, ახალ საპროექტო სისტემას, მონაცემებთან მუშაობის უკეთეს მხარდაჭერას და XHTML-ის მთლიან მხარდაჭერას. ერთიანად ეს თვისებები აძლევს პროგრამისტს საშუალებას სწრაფად, ადვილად და ეფექტურად გამოიმუშაოს ვებ პროგრამა.

ვებ პროგრამის კოდის ძირითადი ნაწილი დაწერილია C#-ზე და ASP.NET-ზე. კოდის დინამიკური ნაწილი ASP-ზე, ხოლო სტატიკური ნაწილი HTML-ზე. მონაცემთა სერვერ-ბაზად შერჩეულია MS SQL Server, ხოლო დაპროგრამების ენად C# .NET. MS SQL Server პაკეტი კორპორაციული სისტემებისთვის დღეისათვის მეტად ეფექტურია მასშტაბურობის, სწრაფქმედების და მწარმოებლურობის თვალსაზრისით [36,65,66].

ახლა განვიხილოთ კონკრეტული საილუსტრაციო მაგალითი ჩვენი სისტემიდან, კერძოდ კლიენტებისათვის საბანკო ანგარიშზე თანხის შეტანის (ან გატანის) მაგალითისათვის. იმისათვის, რომ სისტემა დაცული იყოს არასანქციური შეღწევებისაგან, გამოვიყენოთ აუტენტიფიკაციის მექანიზმი, რისთვისაც პირველ რიგში მომხმარებელი შეიტანს თავისი იდენტიფიკატორს და პაროლს.

სწორი აუტენტიფიკაციის შემდეგ მომხმარებელი მიიღებს სისტემის მთავარ გვერდს, რომელზეც გამოსახულია გალუბის გაცვლის კურსი, მენიუ და ღილაკები (ნახ.3.19).

Address http://localhost:1891/Bank/Admin/main.aspx

[თურქული კურსები](#) [ანგლიური კურსები](#)

you are logged in  
[Logout](#) [თურქული კურსები](#) ►

[სასამართლო თურქული კურსები](#) ►

[კონვერტაციები](#) ►

Currency	OfficialCourses	Buy	Sell
USD/GEL	1.7820	1.7950	1.7720
EUR/GEL	2.1200	2.0000	2.3500
EUR/USD		1.1000	1.2550
GBP/GEL	3.0862	3.0000	3.2100
RUR/GEL	0.0622	0.0550	0.0720

### ნახ.3.19. ინტერფეისის მთავარი გვერდი

„ინფორმაცია კლიენტზე” ღილაკით ოპერატორი გადავა ვებ-გვერდზე, რომელზეც მას შეუძლია იხილოს ინფორმაცია კონკრეტულ კლიენტზე. ღილაკით „ინფორმაცია ანგარიშზე” გამოყენებისას ოპერატორი მოხვდება გვერდზე, რომელზეც მას შეუძლია ნახოს კლიენტის ანგარიშები.

თუ ოპერატორს სჭირდება ინფორმაცია კლიენტის ანგარიშების შესახებ, მას შეუძლია გადავიდეს გვერდზე „ინფორმაცის ანგარიში”, შესაბამისი სახელწოდების ღილაკით. კლიენტის მონაცემების (სახელი და გვარი) შეტანით, სპეციალურად განკუთვნილ ტექსტურ ველებში, ის იხილავს ინფორმაციას კლიენტის ანგარიშების შესახებ. გამოტანილი იქნება შემდეგი ინფორმაცია: სახელი, გვარი, ანგარიშის ნომერი, ანგარიშის ტიპი, ბალანსი, ოვერდრაფტი, კრედიტი და ვალუტის ტიპი (ნახ.3.20).

[თურქული კურსები](#)

Name	Surname	AccountNumber	TypeofAccount	Balance	Overdraft	Loan	Currency
Irma	Berdzenishvili	333010525	Current	1400.0000	0	0	GEL

სახელი  გვარი

### ნახ.3.20. გვერდი: „ინფორმაცია ანგარიშზე”

თუ კლიენტს სურს თანხის შეტანა ანგარიშზე, ამ შემთხვევაში ოპერატორი უნდა გადავიდეს გვერდზე „თანხის შეტანა”. აქ ოპერატორს აქვს საშუალება გადაამოწმოს კლიენტის ანგარიშები, რათა დადასტურდეს, რომ ის ანგარიში, რომელზეც კლიენტს შემოაქვს ფული, არსებობს. ეს შესაძლებელია ანგარიშის ნომრის მითითებით და შემდეგ ღილაკით „ანგარიშის ნახვა” (ნახ.3.21). გამოტანილი იქნება ინფორმაცია მითითებული ანგარიშის შესახებ. თანხის შეტანისას ველში “თანხა” ოპერატორმა უნდა მიუთითოს თანხის მნიშვნელობა და დანიშნულება.

The screenshot shows a web-based banking application interface. At the top, there is a header bar with the address 'http://localhost:1891/Bank/Tanxis%20Setana.aspx'. Below the header, there are two main menu items: 'საფულრმაცხა კლიენტზე' (Customer Client) and 'საფულრმაცხა ანგარიშზე' (Account Record). The second item is highlighted. A sub-menu below it contains 'ანგარიშის ნომერი' (Account Number), 'თანხა' (Transaction), and 'დანიშნულება' (Value). On the left side, there is a message 'you are logged in' and a 'Logout' link. The bottom navigation bar includes links for 'სალაროს თავრიაცევა', 'თანხის შეტანა', 'თანხის გატანა', and 'კონფერტაცევა'.

ნახ.3.21. გვერდი: „თანხის შეტანა”

თანხის შეტანის დროს კლიენტის ანგარიშზე არსებული ბალანსი იცვლება. ოპერატორმა უნდა მოახდინოს ბალანსის რედაქტირება, რაც შესაძლებელია Edit ღილაკით. შესატანი თანხის მითითებით და update ღილაკით ოპერატორი ახორციელებს ცვლილებების შეტანას მონაცემთა ბაზაში კლიენტის ბალანსის შესახებ, კერძოდ, შეტანილი თანხა ემატება არსებულ ბალანსს (ნახ.3.22).

ანგარიშმაცია ანგარიშები									
ანგარიშის ნომერი	333010525			ანგარიშის ნაწილი					
თანხა	300								
დანიშნულება	ანგარიშები თანხის შეტანა								
AccountNumber	TypeofAccount	Balance	Overdraft	Loan	Currency	Name	Surname		
<a href="#">Edit</a> 333010525	Current	1700.0000	0	0	GEL	Irma	Berdzenishvili		

ნახ.3.22. თანხის შეტანის საბოლოო შედეგი

იგივე ხდება თანხის გატანისას, ოდონდ საპირისპირო ნიშნით. არსებულ ბალანსს აკლდება გამოტანილი თანხა.

დასასრულ, შეიძლება დაგასკვნათ, რომ ვებ-დანართების აგების პროცესების ავტომატიზაციით მიიღება მაღალი ხარისხის საიმედო პაკეტები. განსაკუთრებით ახალი საინფორმაციო ტექნოლოგიების, როგორიცაა NET-პლატფორმა და C#, ASP.NET.

### 3.6. ავტომატიზებული ანალიზის სისტემის დაპროექტება სატენდერო კომისიის ექსპერტებისთვის

მოცემულ პარაგრაფში აღწერილია სატენდერო პროცესის ხელშემწყობი კომპიუტერული სისტემა, რომლის მიზანია სატენდერო კომისიის ექსპერტებისათვის აუკილებელი საინფორმაციო ბაზისა და მისი თკერატიული ავტომატიზებული ანალიზის ჩატარების განხორციელება. კონცეპტუალური მოდელი დაპროექტებულია ORM-დიაგრამისა და მისი შესაბამისი ER-მოდელის საშუალებით [44]. მონაცემთა ბაზა რეალიზებულია Ms SQL სერვერზე.

დღეს, მეტად აქტუალური გახდა ტენდერის ჩატარება ნებისმიერ სფეროში, მშენებლობა იქნება ეს, სარემონტო სამუშაოები, მომარაგება, ამა თუ იმ პროდუქციის შესყიდვა, თუ რომელიმე სხვა პროექტის განხორციელება. ტენდერს აცხადებს ორგანიზაცია, რომელსაც სურს

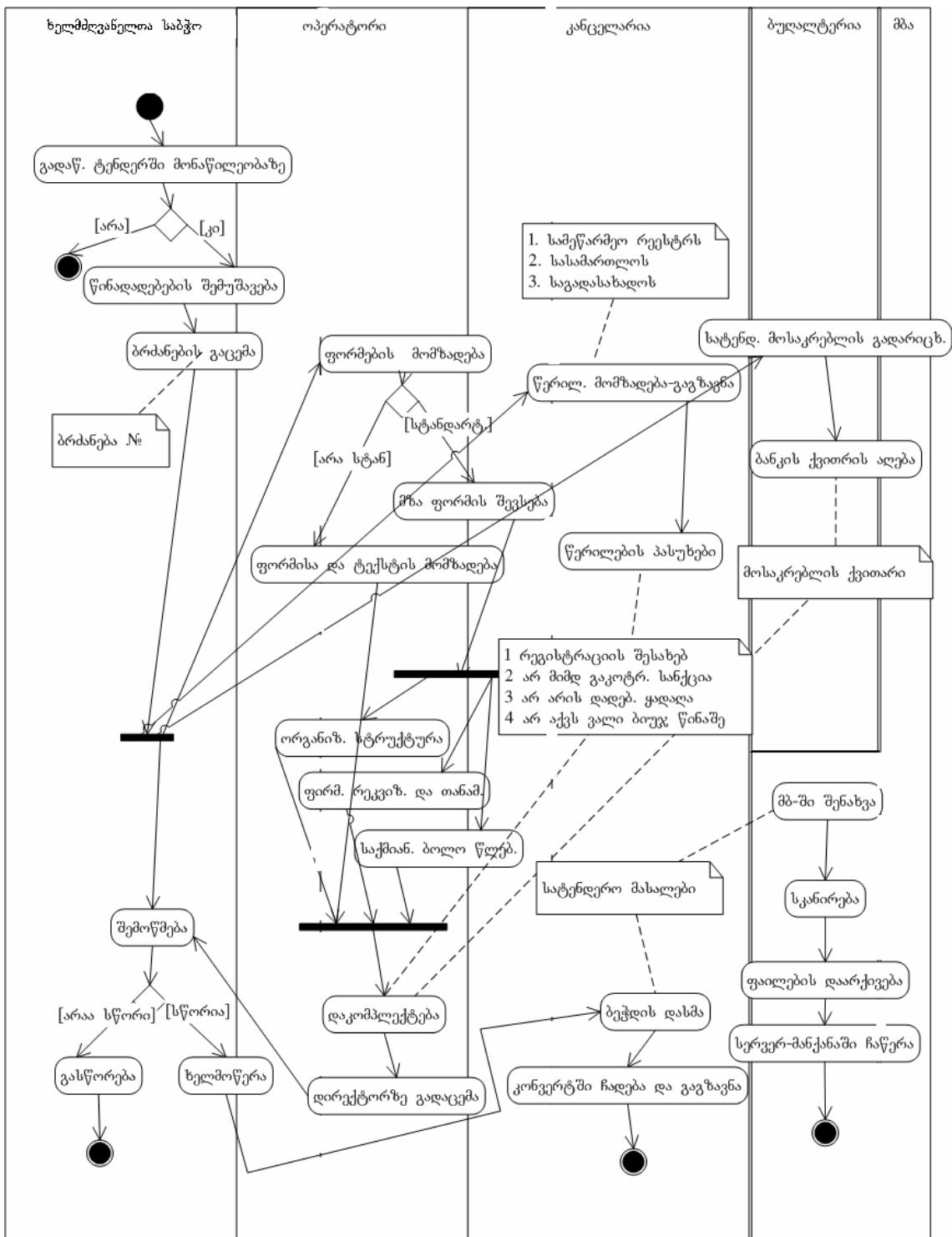
გარკვეული მოცულობის ამა თუ იმ სამუშაოს შესრულება. ფირმები, რომელთაც სურს ამ სამუშაოს შესრულება, აკეთებენ ოფიციალურ განაცხადს ტენდერში მონაწილეობის მისაღებად.

უმეტესად ასეთი საქმიანობით სხვადასხვა ტიპის სპეციალიზებული ფირმებია დაინტერესებული. ტენდერში მონაწილეობის მისაღებად აუცილებელია საჭირო იურიდიული დოკუმენტაციის შეკრება სხვადასხვა დაწესებულებებიდან, ასევე აუცილებელია სატენდერო მოსაკრებლის გადახდა ტენდერის მოთხოვნებში მითითებულ ბანკის ანგარიშზე. ტენდერში მონაწილეობის მიღების მიზნით შესასრულებელი სამუშაო დეტალურადაა წარმოდგენილი UML Activity-დიაგრამაზე (ნახ.3.23).

კონცეპტუალური მოდელის დაპროექტებისათვის გამოყენებულია ობიექტ-როლური მოდელირება (ORM), რომელიც კონცეპტუალური მოდელირების განვითარებულ ტექნიკას წარმოადგენს. ობიექტ-როლური მოდელირება მიახლოებულია ბუნებრივ სალაპარაკო ენასთან. ესაა მოდელირება ფაქტების საფუძველზე, სადაც საპრობლემო არ ე განიხილება, როგორც ობიექტების ერთობლიობა, რომლებიც თამაშობს გარკვეულ „როლებს“. ობიექტ-როლური მოდელირების მეთოდი აქტუალურია და ფართოდ გამოიყენება საზღვარგარეთაც [24,37]. აღნიშნული ინსტრუმენტული საშუალებანი ემსახურება მონაცემთა ბაზის დაპროექტების ავტომატიზაციას.

UML-ტექნოლოგიის საფუძველზე პროგრამული პროდუქტების შექმნა მოითხოვს საკვლევი ობიექტის მოთხოვნილებათა განსაზღვრის, ობიექტ-ორიენტირებული ანალიზის, ობიექტ-ორიენტირებული დაპროექტებისა და რეალიზაციის (ობიექტ-ორიენტირებული დაპროგრამების) ეტაპების განხორციელებას.

ობიექტ-როლური მოდელირების მეთოდი გამოიყენება საგნობრივი სფეროს კვლევის ეტაპზე და ახორციელებს ექსპერტების (დამპროექტებლების) მიერ გარკვეული ცოდნის, ფაქტების ფიქსირებას, რომლებიც აუცილებელია ORM-დიაგრამის (ობიექტ-როლური მოდელის) ასაგებად.



ნახ.3.23. სატენდერო პროცესის მომზადების აქტიურობის დიაგრამა

სატენდერო ამოცანების გადაწყვეტის ხელშემწევი კომპიუტერული სისტემის აგების მიზნით შეიძლება შემდეგი ზოგადი ფაქტების ჩამოყალიბება, რაც შეესაბამება 3.23 ნახაზე მოცემულ აქტიურობათა დიაგრამას:

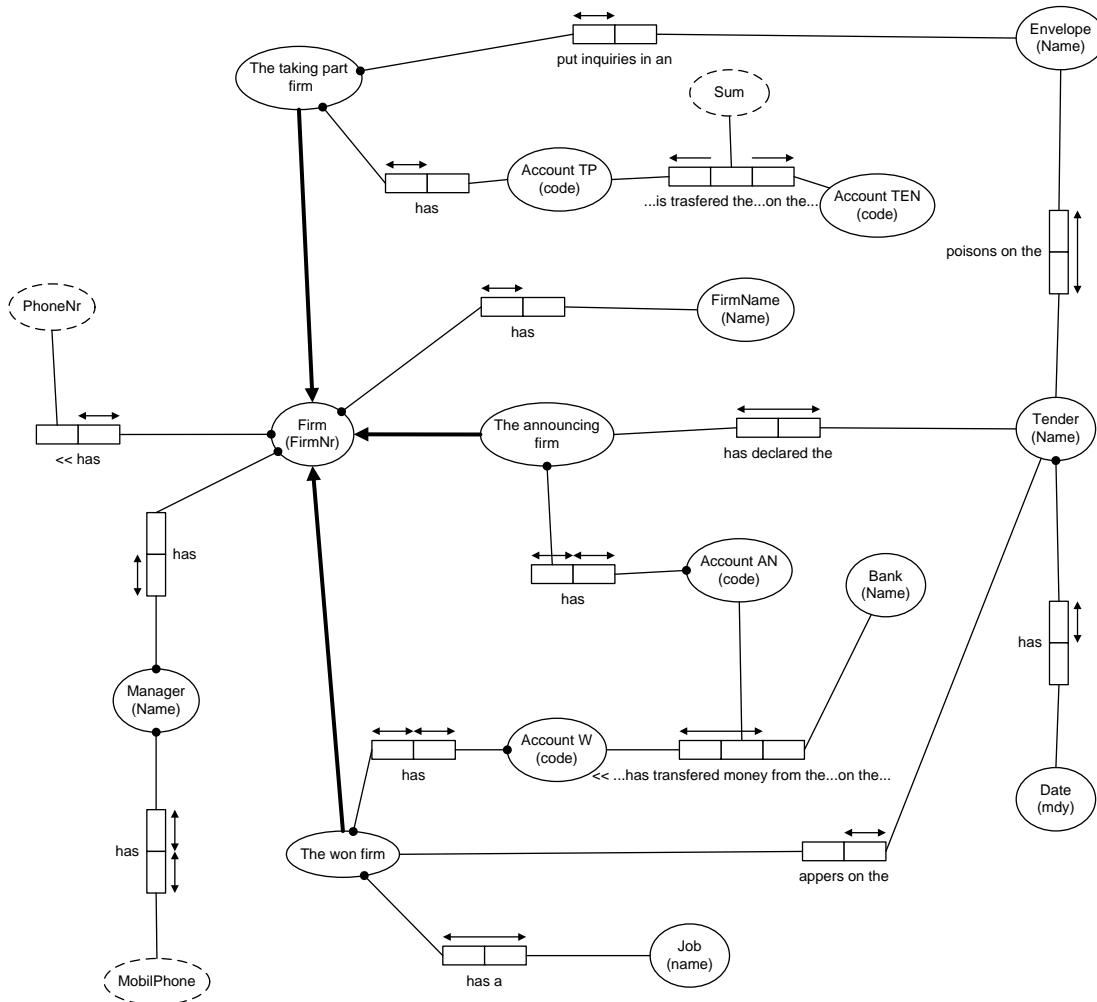
„არსებობს ფირმა რომელიც აცხადებს ტენდერს, აგრეთვე ფირმა (ფირმები), რომელიც მონაწილეობს ტენდერში, და ბოლოს - გამარჯვებული ფირმა. ფირმას აქვს რეკვიზიტები და პყავს მენეჯერი, აქვს საკუთარი ანგარიშის ნომერი ბანკში. ტენდერში მონაწილეობის მისაღებად საჭირო დოკუმენტები თავსდება კონვერტში, ილუქტი და იგზავნება ტენდერის მოთხოვნებში მითითებულ მისამართზე. აქვა მითითებულია ტენდერის ჩატარების თარიღი და დრო.

ტენდერის ჩატარების დროს ყველა მონაწილე ორგანიზაციის წარმომადგენელი, ვალდებულია გამოცხადდეს მითითებულ მისამართზე. სატენდერო კომისიის წევრები საჯაროდ გახსნიან დალუქტულ კონვერტებს, შეამოწმებენ იურიდიულად საჭირო დოკუმენტაციის არსებობას და შეადგენენ ოქმს ტენდერის დაწყებისა და მასში მონაწილე ორგანიზაციების შესახებ. თუ დოკუმენტაციას აკლია რომელიმე აუცილებელი დოკუმენტი, ან არასწორადაა წარმოდგენილი, ასეთი ფირმა მოიხსნება ტენდერიდან.

მომდევნო პერიოდში ტენდერის კომისიის წევრები დეტალურად გაეცნობიან შემოთავაზებულ წინადადებებს, გადაამოწმებენ საბუთებს, მსჯელობის შემდეგ ამოირჩევენ საუკეთესო (მისაღებ) წინადადებას და შემდგომ დაასახელებენ ტენდერში გამარჯვებულ ორგანიზაციას. დამარცხებულ ფირმებს უფლება აქვს სასამართლოში გაასაჩივროს ტენდერის ჩამტარებელი ფირმის გადაწყვეტილება, რაც ხდება შემდგომი იურიდიული დავის საგანი.

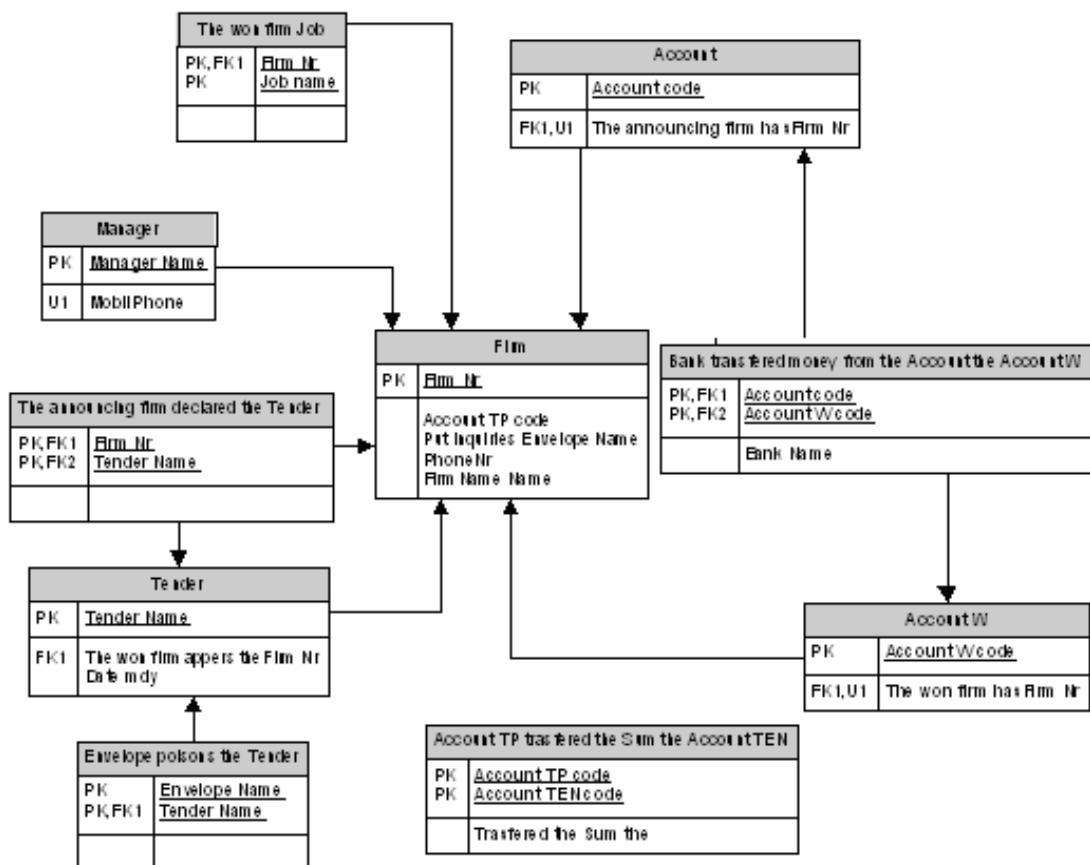
გამარჯვებული ფირმის ანგარიშზე გადაირიცხება სამუშაოს დაწყებისათვის აუცილებელი თანხა და ა.შ.“

ასეთი არაფორმალიზებული აღწერიდან ფორმალიზებულზე გადასასვლელად არის საჭირო სწორედ ფაქტების გამოკვეთა, რომლებიც შემდგომ ობიექტ-როლური დიაგრამისთვის გამოგვადგება. 3.24 ნახაზე მოცემულია „ტენდერის ჩატარების“ პროცესის ფაქტების გადატანა ORM-დიაგრამაზე.



ნახ.3.24. სატენდერო პროცესის ORM-დიაგრამა

Ms Studio.Net პროგრამული პაკეტი, კერძოდ Ms Visio საშუალებას გვაძლევს ORM-დიაგრამიდან ავტომატურად ავაგოთ საპრობლემო სფეროს კონცეპტუალური მოდელი, ER-დიაგრამა, რომელიც ჩვენი შემთხვევისთვის მოცემულია 3.25 ნახაზზე. ERM (Entity Relation Modeling) – ის საფუძველზეც აიგება რელაციურ მონაცემთა ბაზების დოკუმენტი სტრუქტურა. მონაცემთა ბაზის დაპროექტების მიზნით ვიყენებთ SQL-სერვერს [31,38,83].



ნახ.3.25. „ტენდერის“ საპრობლემო სფეროს ER-მოდელი

3.26 ნახაზზე მოცემულია ტენდერის ჩატარების ხელშემწყვბი კომპიუტერული სისტემის TENDER-მონაცემთა ბაზის ცხრილების ფრაგმენტი.

**Design Table 'Firm participating in the tender' in 'tender'**

	Column Name	Data Type	Length	Allow Nulls
▶	firmcode	int	4	
	saxeli	char	10	✓
	qalaqi	char	10	✓
	misamarTi	char	53	✓
	tel	char	10	✓
	faqsi	char	19	✓

**Design Table 'List of firm' in 'tender' on 'USER-EOAD098E2'**

	Column Name	Data Type	Length	Allow Nulls
▶	firmcode	int	4	✓
	dasaxeleta	char	10	✓
	fasi	money	8	✓

**Design Table 'Prize list' in 'tender' on 'USER-EOAD098E2F'**

	Column Name	Data Type	Length	Allow Nulls
▶	dasaxeleta	char	10	
	modeli	char	10	
	raodenoba	char	10	✓
	fasi	money	8	✓

### ნახ.3.26. ტენდერის მონაცემთა ბაზის ფრაგმენტის ცხრილების სტრუქტურა

ამგვარად, დასმული და გადაწყვეტილია ამოცანა საწარმოო ფირმათა სატენდერო პროცესის ხელშემწყობი კომპიუტერული სისტემის საინფორმაციო ბაზის ავტომატიზებული დაპროექტებისათვის მონაცემთა ობიექტ-როლური მოდელირებისა და UML-ტექნოლოგიის საფუძველზე. ასეთი პროგრამული პროდუქტის ინტეგრირებული გამოყენების საფუძველზე შესაძლებელია საპრობლემო სფეროს ობიექტ-ორიენტირებული მოდელირებისა და დაპროგრამების პროცესების ავტომატიზაცია, რაც საგრძნობლად ამცირებს სისტემების აგების დროს და აუმჯობესებს მის სარისხს.

### 3.7. მესამე თავის დასკვნები

1. გადაწყვეტილებათა მიღების მხარდამჭერი სისტემისთვის რეალიზებულია მონაცემთა რელაციური ბაზების დაპროექტების

ავტომატიზაციის ამოცანა NET პლატფორმაზე, ობიექტ-როლური მოდელირების საფუძველზე, რაც უზრუნველყოფს მონაცემთა საცავის ლოგიკურ მთლიანობას;

2. რეალიზებულია გადაწყვეტილების მიღების მხარდამჭერი სისტემის გრაფო-ანალიზური პროგრამული აპლიკაცია მრავალფაქტორული ამოცანებისათვის;

**3. realizebilija sabanko korporaciuli sistemis Web-aplikacia Internet-Intranet-garemoSi  
NET-platformaze, rac uzrunvelyofs moqnili klient-serveruli sistemis funqcionirebas.**

## დასკვნა

ჩატარებული თეორიულ და ექსპერიმენტულ გამოკვლევათა საფუძველზე მიღებული შედეგების ბაზაზე შეიძლება შემდეგი დასკვნების ჩამოყალიბება:

1. გაანალიზებულია ორგანიზაციული მართვის სისტემებში კორპორაციული დაგეგმვის პროცესების პრობლემები და დასმულია ამოცანა მათი შემდგომი სრულყოფისათვის მმართველობითი კონსულტირების ბიზნეს-პროცესების ავტომატიზაციის საფუძველზე.
2. დამუშავებულია ბიზნეს-პროგრამების ექსპერტულ შეფასებათა მხარდამჭერი გადაწყვეტილებების მიღების ობიექტ-ორიენტირებული მეთოდები და მოდელები.
3. დასმული და გადაწყვეტილია მონაცემთა საცავის აგებისა და ოპერატიული ინფორმაციის ანალიზის თანამედროვე სისტემების გამოყენების ამოცანა ორგანიზაციული მართვის ობიექტებზე. შემუშავებულია მონაცემთა მრავალგანზომილებიანი, რელაციური არსთა-დამოკიდებულების ვარსკვლავური სქემის დაპროექტების ვიზუალურ-ანალიზური მოდელი ობიექტ-როლური დიაგრამების ინსტრუმენტის საფუძველზე.
4. გადაწყვეტილია ბიზნეს-პროცესების ოპერატიული ანალიზის OLAP-ისტრუმენტის გამოყენების საკითხი. შემოთავაზებულია განაწილებული მონაცემთა ბაზებიდან შერჩეული მონაცემების ერთიან გადაწყვეტილების მიღების მხარდამჭერ სისტემაში სტრუქტურიზებული ორგანიზება და შესაბამისი პროგრამული პაკეტის რეალიზაცია ობიექტ-ორიენტირებული დაპროგრამების C++ ენის Decision Cube კომპონენტის გამოყენებით მრავალფაქტორული ანალიზის ამოცანებისთვის.
5. დამუშავებულია ბიზნეს-პროგრამების ექსპერტულ შეფასებათა პროცესების ობიექტ-ორიენტირებული მოდელირების, დაპროექტებისა და რეალიზაციის საკითხები, კატეგორიალური და ლოგიკურ-ალგებრული მეთოდების საფუძველზე.

6. პროგრამულად რეალიზებულია ბიზნეს-პროცესების ობიექტ-როლური მოდელირების პროცედურები UML/ORM გარემოში და კლიენტ-სერვერ არქიტექტურის განაწილებული სისტემის მონაცემთა ბაზის სტრუქტურის (ERM) ავტომატიზებული დაპროექტებისათვის. შედეგები ადაპტირებულია Ms SQL Server მონაცემთა ბაზების მართვის სისტემაში.

7. კლიენტ-სერვერული არქიტექტურის ბიზნეს-პროცესების მოდელირებისა და იმიტაციური ანალიზისათვის აგებულია პეტრის ქსელის გრაფების ალტერნატიულ (ეკვივალენტურ) სქემათა ერთობლიობა, გამოკვლეულია მათში მიმდინარე პროცესების დროითი მახასიათებლები და დადგენილია შედარებით ეფექტური, მისაღები მოდელების ერთობლიობა, საერთო რესურსების ეფექტურად გამოყენებისა და მოთხოვნების დამუშავების დროის შესამცირებლად.

8. ორგანიზაციული მართვის სისტემებში ექსპერტულ შეფასებათა გადაწყვეტილების მიღების მხარდამჭერი ვინდოუს-პროგრამების გარდა დამუშავებულია ვებ-აპლიკაციების დაპროექტების და რეალიზაციის საკითხები. კერძოდ საფინანსო ბანკის კლიენტთა მომსახურების მაგალითზე ილუსტრირებულია მათი ინტერფეისული კომპონენტების აწყობისა და მონაცემთა სერვერული ბაზების ორგანიზების ამოცანები. სისტემა დამუშავებულია .NET-პლატფორმაზე, C#, ASP.NET, ADO.NET და MsSQL Server ობიექტ-ორიენტირებული ინსტრუმენტების გამოყენებით.

## გამოყენებული ლიტერატურა

1. Абрамсон Р., Халсет У. Повышение эффективности работы предприятия с помощью планирования. Пер. с англ., Тбилиси, 1987.
2. გ. სურგულაძე, ი. ვაჭარაძე, ბ. ფოლადაშვილი. ექსპერტულ შეფასებათა პროცესების ავტომატიზაცია ობიექტ-ორიენტირებული მეთოდებით. სტუ-ს შრ.კრ. „მას“ №1, 2006. გვ. 175-178.
3. Booch G., Jacobson I., Rumbaugh J. Unified Modeling Language for Object-Oriented Development. Rational Software Corporation, Santa Clara, 1996.
4. Рамбо Д., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. Изд. "Питер", 2006.
5. <http://accoona.ru/referat/ref13161.html> - გადამოწმებულია 20.05.08
6. Inmon W.H., Hackathorn R.D. Using the Data Warehouse. John Wiley & Sons, ISBN 0-471-05966-8
7. [http://www.b-eye-network.com/blogs/drewek/archives/2005/03/\\_data\\_warehouse.php](http://www.b-eye-network.com/blogs/drewek/archives/2005/03/_data_warehouse.php) - გადამოწმებულია 20.05.08
8. სურგულაძე გ., პეტრიაშვილი ლ. მონაცემთა საცავის აგების ტექნიკური ინტერნეტული ბიზნესის სისტემებისათვის. სტუ. თბილისი, 2005.
9. Merz M., Electronic commerce: Marktmodelle, Anwendungen und Technologien. dpunkt-Verlag, Germany, 1999.
10. Codd E.F., Codd S.B., Salley C.T.: Providing OLAP to User-Analysts: An IT Mandate. Codd & Associates, Ann Arbor/Michigan 1993.
11. Reisig W., Rosenberg G. Lectures on Petri Nets I: Basic Models. Berlin ; Heidelberg ; New York et al : Springer, 1998
12. სურგულაძე გ., გულუა დ. განაწილებული სისტემების ობიექტ-ორიენტირებული მოდელირება უნიფიცირებული პეტრის ქსელებით. სტუ. თბ., 2005.
13. Reisig W. Elements of Distributed Algorithms : Modeling and Analysis with Petri Nets. Berlin ; Heidelberg ; New York et al : Springer, 1998
14. Питерсон Дж. Теория Сетей Петри и моделирование систем. Перевод с английского. Москва, «Мир», 1983
15. სურგულაძე გ. დოლიძე თ., ყვავაძე ლ. კომპონენტურ-ვიზუალური დაპროგრამება. სტუ. თბილისი, 2006.
16. სურგულაძე გ. ობიექტ-ორიენტირებული დაპროგრამების მეთოდი. სტუ. თბილისი, 2007.
- 17.. ჩოგოვაძე გ., გოგიჩაიშვილი გ., სურგულაძე გ., შეროზია თ., შონია თ. მართვის ავტომატიზებული სისტემების დაპროექტება და აგება (თეორიულ-პრაქტიკული ინფორმატიკა). სტუ. თბილისი, 2001.
18. რეისიგი ვ., სურგულაძე გ., გულუა დ. ვიზუალური, ობიექტ-ორიენტირებული დაპროგრამების მეთოდები. სტუ. თბილისი, 2002.
19. Архангельский А. Программирование в BorlandC++Builder. Москва, 2001.
20. სურგულაძე გ. დაპროგრამების ვიზუალური მეთოდები და ინსტრუმენტები: UML, Ms Visio, Borland C++Builder . სტუ. თბილისი, 2000.
21. ბუკია გ., სურგულაძე გ., დოლიძე თ., შარაშიძე ბ., შონია თ. ექსპერტთა შეფასებების დამუშავება პერსონალურ კომპიუტერზე. სმმი, თბილისი, 1990.
22. სურგულაძე გ., ვაჭარაძე ი., ფოლადაშვილი ნ., ტყეშელაშვილი თ. ექსპერტულ შეფასებათა პროცესების ავტომატიზაცია ობიექტ-ორიენტირებული მეთოდებით. სტუ შრ.კრ. „მას“, №1, 2006. 175-178 გვ.

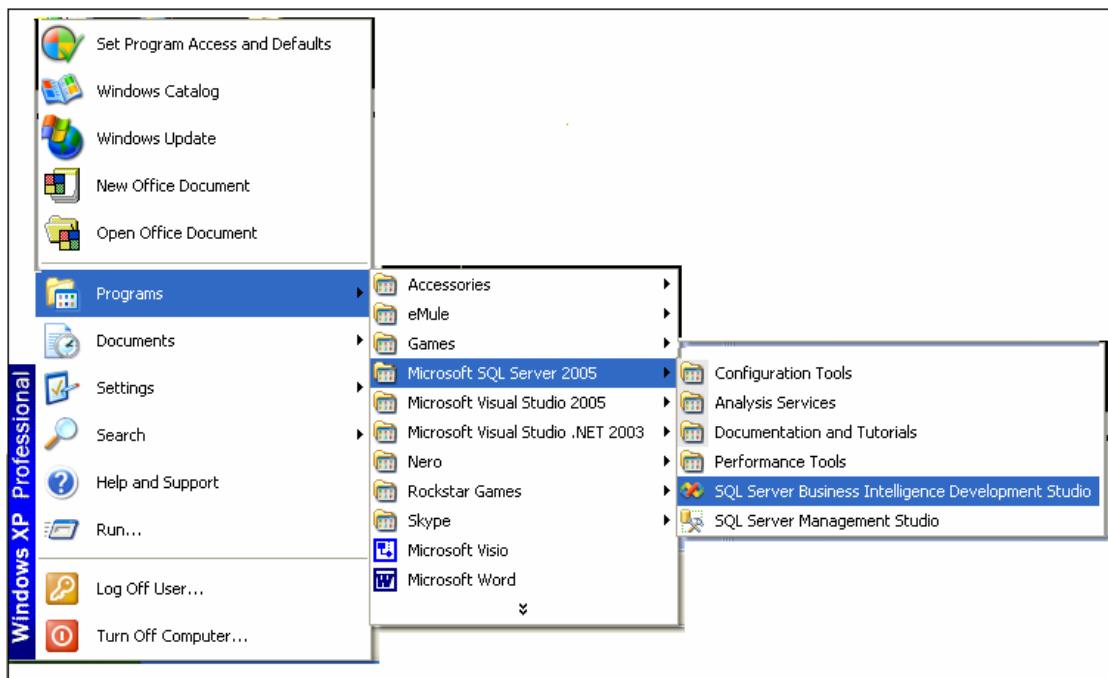
23. Halpin, T.A. 2004, ‘Object-Role Modeling (ORM/NIAM)’, Handbook on Architectures of Information Systems, Bernus, P., Mertins, K. & Schmidt (eds), Springer, Heidelberg, Ch. 4. (online at [www.orm.net](http://www.orm.net)).
24. Овчинников В.В., Повышение управляемости больших концептуальных моделей // Информационные технологии, №10, 2004.
25. სურგულაძე გ., კვდეკინბი პ., თოფურია ნ. განაწილებული ოფის-სისტემების მონაცემთა ბაზების დაპროექტება და რეალიზაცია UML-ტექნოლოგიით. მონოგრაფია. სტუ, თბილისი. 2006.
26. Wedekind H. Objektorientierte Schemaentwicklung. Ein kategorialer Ansatz fuer Datenbanken und Programmierung. Wissenschaftsverlag, Manheim/Wien/Zuerich. 1991.
27. Halpin T.A., Information Modeling and relational Databases, Morgan Kaufmann Publishers, SanFrancisco,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).
28. სურგულაძე გ., თოფურია ნ., პეტრიაშვილი ლ., ვაჭარაძე ი. ბიზნეს-პროგრამების ექსპერტულ შეფასებათა პროცესების ობიექტ-ორიენტირებული დაპროექტება და რეალიზაცია. სტუ. შრ.კრ. 2008
29. Широков Л.А. Информатизация банковской деятельности. Москва. МГИУ, 2002.
30. Ашхинадзе А. Практика финансового управления: расчет доходности клиентов. М., Банки и Технологии, №2, 2005.
31. სურგულაძე გ., შონია რ., ყვავაძე ლ. მონაცემთა ბაზების მართვის სისტემები: Ms Access, SQL Server, InterBase, Oracle, Corba. სტუ, თბილისი, 2004.
32. Codd E.F, Codd S.B., Salley C.T. Providing OLAP to User-Analysts: An IT Mandate, Codd & Associates, Ann Arbor/Michigan, 1993.
33. Codd E.F. A relational model of data for large shared data banks. Communications of the ACM. 1970.
34. ბოტჰე პ., სურგულაძე გ., დოლიძე. თანამედროვე პროგრამული პლატფორმები და ენები. სტუ. თბილისი, 2003.
35. <http://msdn.microsoft.com/vstudio/express/vwd/> გადამოწმებულია 10.04.08
36. <http://msdn.microsoft.com/vstudio/express/sql/> გადამოწმებულია 10.04.08
37. Halpin T.A., Microsoft's new database modeling tool. *Journal of Conceptual Modeling* [www.orm.net](http://www.orm.net)., 2002 . გადამოწმებულია 10.04.08
38. გ. სურგულაძე, ნ. თოფურია, ვ. ქაჩიბაია, ი. ილდიზი. კონცეპტუალური მოდელის დაპროექტება UML-ტექნოლოგიით უნივერსიტეტის მონაცემთა ბაზების აგებისას. სტუ-ს შრ.კრ. „მას“, №1, 2006.
39. Сургуладзе Г.Г., Топурия Н.Ш., Вачарадзе И.В. Автоматизация Проектирования распределенных офис-систем на базе UML /ORM –технологии. *Georgian Engineering News*, № 3. Тб., 2007.
40. სურგულაძე გ., თოფურია ნ., ვაჭარაძე ნ. განაწილებულ ოფის-სისტემებში საქმიანი პროცესების ობიექტ-ორიენტირებული მოდელირება და ანალიზი პეტრის ქსელებით. *Georgian Electronic Scientific Journal*, 2006, N3
41. პეტრიაშვილი ლ., ვაჭარაძე ი., ბასილაძე გ. გადაწყვეტილების მიღების მხარდამჭერ საინფორმაციო სისტემებში OLAP კონცეფციის ერთი რეალიზაციის შესახებ. სტუ შრ.კრ. "მას" N1(4), 2008. 103-107 გვ.
42. ვაჭარაძე ი. მონაცემთა საცავის დაპროექტება გადაწყვეტილებათა მიღების მხარდამჭერი სისტემისათვის კომერციული ბანკის მაგალითზე. სტუ შრ.კრ. "მას", N1(2), 2007. 207-210 გვ.
43. სურგულაძე გ., თოფურია ნ., ყვავაძე ლ., ვაჭარაძე ი. განაწილებული მონაცემთა ბაზების აგების ავტომატიზაცია .NET გარემოში. სტუ შრ.კრ. "მას", N1(2), 2007. 105-108 გვ.

44. სურგულაძე გ., თოფურია ნ., ვაჭარაძე ნ. ავტომატიზებული ანალიზის სისტემის დაპროექტება სატენდერო კომისიის ექსპერტებისთვის. სტუ შრ.კრ. N4(463), 2007. 19-23 გვ.
45. სურგულაძე გ., ბერძენიშვილი ი., ვაჭარაძე ი., ხელაძე ნ., ბულია ი. კორპორაციული მართვის სისტემის Web-აპლიკაციის დამუშავება Internet-Intranet გარემოში .NET-პლატფორმაზე. სტუ შრ.კრ. "მას", N1, 2006. 159-162გვ.
46. ჩოგოვაძე გ., სურგულაძე გ., შონია ო. მონაცემთა და ცოდნის ბაზების აგების საფუძვლები. თბილისი, განათლება, 1996.
47. სურგულაძე გ. მონაცემთა ბაზების სამაგიდო სისტემები: MsAccess. თბ., სტუ, 2004.
48. ჯავაშვილი ა., ებრაელი ა., ნიშანიშვილი ა. ბერძენიშვილი და მონაცემთა ბაზების საფუძვლები. თბილისი, განათლება, 1996.
49. ჩოგოვაძე გ., სურგულაძე გ. რელაციური ალგებრის ოპერაციების შესრულების ეფექტური პროცედურის აგების ერთი ინსტრუმენტის შესახებ მონაცემთა ბაზებში. საქ.მეცნ.აკად. „მოამბე“, 148-№3, 1993.
50. სურგულაძე გ., დოლიძე თ. საწარმოო ფირმებში მარკეტინგული პროცესების მართვის ინფორმაციული სისტემის დაპროექტება და რეალიზაცია UML-ტექნილოგიით. სტუ-ს შრ.კრებ., №4(437) თბ., 2001.
51. რეისიგი გ., სურგულაძე გ., გულუა დ. კოზუალური ობიექტ-ორიენტირებული დაპროგრამების მეთოდები. თბ., სტუ, 2002.
52. Boggs W., Boggs M. Mastering UML with Rational Rose. Copyright 1999 SyBEX, California.
53. კოტლერი ფ. მარკეტინგის საფუძვლები. თარგ.ინგლ. მაცნე, თბ., 1993.
54. ბოტე კ., სურგულაძე გ., დოლიძე თ., შონია ო. თანამედროვე პროგრამული პლატფორმები და ენები. სტუ-ს შრ.კრ., თბ., 2003.
55. ებრაელი ა., დაენაიდო ე. Borland C++Builder: ყიდვების დამატებითი მეთოდები. თბ., სტუ, 2001.
56. იობაე ს. ენდო ს. SQL Server 2000. წ-ლაბადა, 2002.
57. იაბე ე. ლენა ა. MS ACCESS 2000: მარკეტინგის დამატებითი მეთოდები. წ-ლაბადა, 2002.
58. Landy M., Siddiqui S., Swisher J, et al. . Borland JBilder. Developer's Guide. Williams Publ., 2004.
59. აბე ა. ებრაელი ი. Oracle8/8i Server. ყიდვების დამატებითი მეთოდები. თბ., 2000.
60. იობაე ა. Oracle 9/9i Server. ყიდვების დამატებითი მეთოდები. თბ., 2004.
61. სურგულაძე გ., პეტრიაშვილი ლ. კუბში მონაცემთა აგრეგაცია გრაფების თეორიის გამოყენებით. ჟურნალი „ინტელექტი“ №3(20), თბილისი, 2004
62. Albrecht, J., Hummer,W., Lehnern W., Using Semantics for Query Derivability in Data Warehouse Applications. (FQAS, 2000 Warschau)
63. Bothe K., Reverse Engineering: the Challenge of Large-Scale Real-World Educational Project, Conference on Software Engineering Education and Training, Charlotte, USA, Febr. 2001.
64. Bothe K. Reverse engineering projects: approaching real-world conditions in educational environments. Trans. of the GTU, 2001, 4(437).
65. Brown S., Wilde N., Carlin J. A Software Maintenance Process Architecture, 9th Conference Software Engineering Education and Training, Daytona Beach, 1996.
66. Bruegge B. From Toy Systems to Real Software Development: Improvements in Software Engineering Education “SEUH” 94.
67. Bauer A. Management of multidimensional Aggregates for efficient online Analytical Processing// Montreal, Canada 1999, S. 156-164.

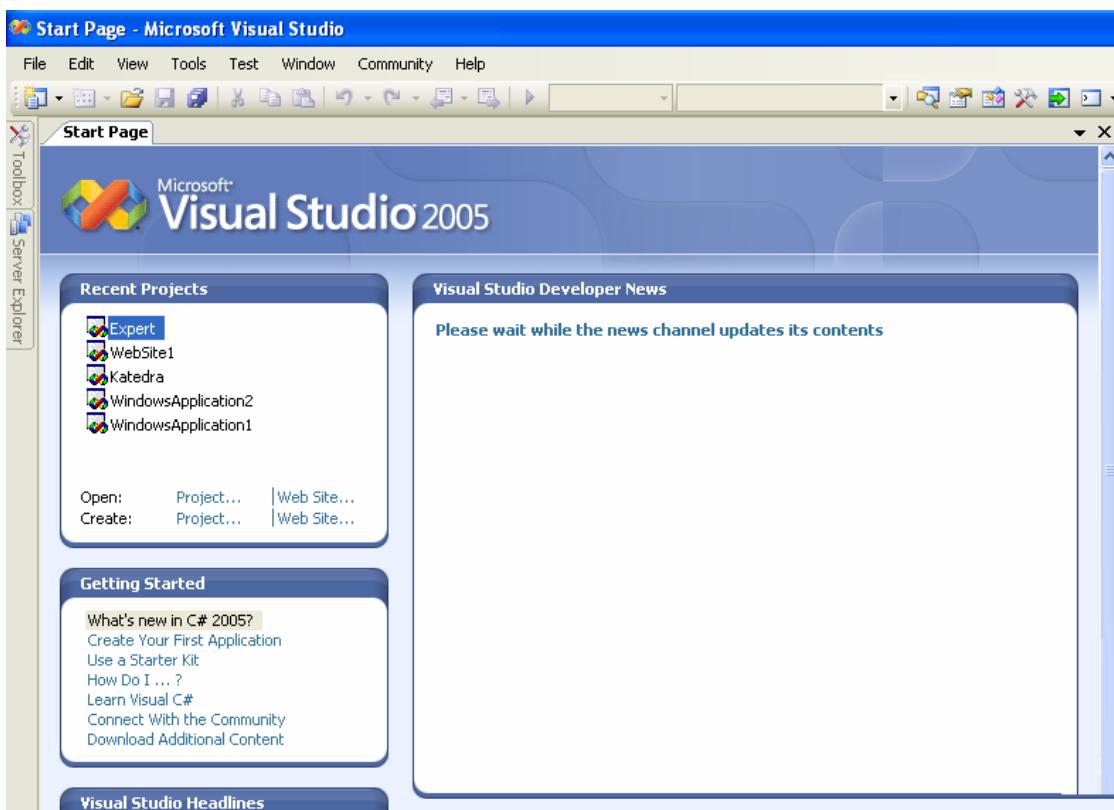
68. [http://mf.grsu.by/other/lib/olap/bd\\_wh/doc01.htm](http://mf.grsu.by/other/lib/olap/bd_wh/doc01.htm) DWH: გადამოწმებულია 10.05.08
69. [http://mf.grsu.by/other/lib/olap/bd\\_wh/doc03.htm](http://mf.grsu.by/other/lib/olap/bd_wh/doc03.htm) OLAP: გადამოწმებულია 10.05.08
70. [http://mf.grsu.by/other/lib/olap/bd\\_wh/doc09.htm](http://mf.grsu.by/other/lib/olap/bd_wh/doc09.htm) OLAP: გადამოწმებულია 10.05.08
71. [http://mf.grsu.by/other/lib/olap/bd\\_wh/doc10.htm](http://mf.grsu.by/other/lib/olap/bd_wh/doc10.htm) RelationalDBS to Multi-Dimensional: გადამოწმებულია 10.05.08
72. [http://mf.grsu.by/other/lib/olap/bd\\_wh/doc18.htm](http://mf.grsu.by/other/lib/olap/bd_wh/doc18.htm) DWH: გადამოწმებულია 10.05.08
73. სურგულაძე გ., ოურქია ვ. ბიზნეს-გებმის ავტომატიზებული დამუშავების პროცესის სამუშაო ნაკადების მართვის სისტემა. სტუ-ს შრომები, № 8(424), თბილისი, 1998.
74. Цаленко М. Моделирование Семантики в базах данных. Наука, Москва, 1989.
75. Цаленко М. Итоги науки и техники Информатика Том 9. Мир, Москва, 1985.
76. Гольдблат Р. Топосы Категорный анализ логики. Мир, Москва, 1983.
77. Биргкоф Т.Б. Современная прикладная Алгебра , 1976.
78. Организация и автоматизация документооборота -<http://www.termica.ru/dou/resh/avtomatiz.html> - გადამოწმებულია 15.03.08
79. Barker R. CASE\*Method. Entity-Relationship Modelling. Copyright Oracle Corporation UK Limited, Addison-Wesley Publishing Co., 1990.
80. Brackett J., C. McGowan. Applying SADT to Large System Problems. SofTechTechnical Paper TP059,January 1977.
81. Document Object Model (DOM) Level 1 Specification <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/> გადამოწმებულია 15.03.08
82. ჟავა ე. სტარტოვი ე. ელექტრონული კომუნიკაციების კონსტრუქციების და მოდელირებაში. Georgian Electronic Scientific Journal: Computer Science and Telecommunications No.1(8), 2006. [http://gesj.internet-academy.org.ge/gesj\\_articles/1212.pdf](http://gesj.internet-academy.org.ge/gesj_articles/1212.pdf) - გადამოწმებულია 18.03.2008.
83. Robinson S., Cornes O., Glynn J., Harvey B., McQueen C., Moemeka J., Nagel C., Skinner M., Watson K. Professional C#. Birmingham, WroxPress, 2001.
84. ვერულავა დ., ფრანგიშვილი ა., ვერულავა ი., გასიტაშვილი ზ. კოგნიტური მიდგომა საქართველოს სათბობ-ენერგეტიკული კომპლექსის კვლევასა და მოდელირებაში. Georgian Electronic Scientific Journal: Computer Science and Telecommunications No.1(8), 2006. [http://gesj.internet-academy.org.ge/gesj\\_articles/1212.pdf](http://gesj.internet-academy.org.ge/gesj_articles/1212.pdf) - გადამოწმებულია 18.03.2008.
85. ჯავახაძე გ. ბიზნეს-პროგრამების მართვისას გადაწყვეტილებათა მიღების მხარდამჭერი მოდელები და მეთოდები : ავტორეფ. ტექნ. მეცნ. დოკტ. 05.13.06. სტუ - თბ., 2003 - 47გვ.
86. Гогичаишвили Г.Г. Автоматизация принятия решений в системах управления. Тбилиси. 1985.
87. გოგიჩაიშვილი გ. სიტუაციური მართვა ავტომატიზებულ სისტემებში. სტუ-ს მრ.კრებ. მას-N1, 2006.
88. Козлов Л.А. Когнитивный подход к исследованию информационных процессов на ранних стадиях проектной деятельности. «Системы автоматизированного проектирования», №5. 1986.

## დანართი:

ექსპერტულ შეფასებათა დამუშავების ავტომატიზებული სისტემის “Expert” სადემონსტრაციო ვერსიის ფრაგმენტი შერულებულია Ms SQL Server 2005 პაკეტის გამოყენებით Visual Studio.NET პლატფორმაზე. ქვემოთ წარმოდგენილია ამ სისტემის ინტერფეისები, ბაზები და პერიფერიული კოდის ლისტინგები:



გახ. დ1: SQL Server 2005 → SQL Server Business Intelligence Development Studio



**ნახ. დ2: სასტარტო გვერდი  
ლისტინგი-დ1**

```
// Form2.cs --- ექსპერტის ინტერფეისი - ფაქტორის შეფასების შეტანა -----
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Expert
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                DataSet1TableAdapters.QueriesTableAdapter ins = new
Expert.DataSet1TableAdapters.QueriesTableAdapter();

                ins.SpAddRelation(Convert.ToInt32(CBEXPERT.SelectedValue),
Convert.ToInt32(CBMizani.SelectedValue),
Convert.ToInt32(CBKriteriomı.SelectedValue),
Convert.ToInt32(CbFaqtori.SelectedValue),
Convert.ToInt32(textBox1.Text));
                this.Close();
            }
            catch (Exception ex)
            {

                MessageBox.Show("chawera ver ganxorcielda, sheamowmet
shevsebuli gaqvT Tu ara yvela veli");
            }
        }

        private void bindningexpert()
        {

        }
        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'dataSet1.Kriteriumebi' table. You can move, or remove it, as needed.

this.kriteriumebiTableAdapter.Fill(this.dataSet1.Kriteriumebi);
            // TODO: This line of code loads data into the
'dataSet1.Factorebi' table. You can move, or remove it, as needed.
        }
    }
}
```

```

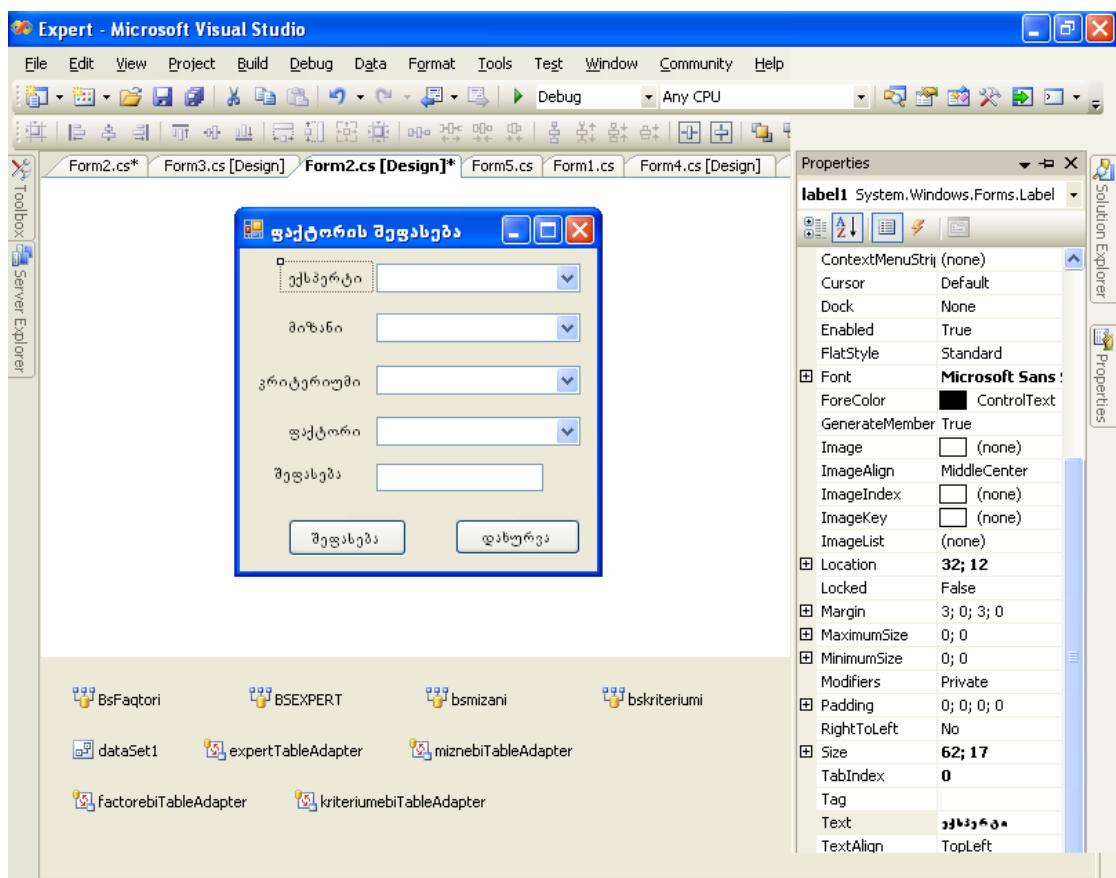
        this.factorebiTableAdapter.Fill(this.dataSet1.Factorebi);
        // TODO: This line of code loads data into the
        'dataSet1.Miznebi' table. You can move, or remove it, as needed.
        this.miznebiTableAdapter.Fill(this.dataSet1.Miznebi);
        // TODO: This line of code loads data into the
        'dataSet1.Expert' table. You can move, or remove it, as needed.
        this.expertTableAdapter.Fill(this.dataSet1.Expert);
        // TODO: This line of code loads data into the
        'dataSet1.Expert' table. You can move, or remove it, as needed.
        this.expertTableAdapter.Fill(this.dataSet1.Expert);

    }

    private void BsFaqtori_CurrentChanged(object sender,
EventArgs e)
{
}

}
}

```



ნახ. დ3-1: Form2.C# - პროგრამის შედეგი

ნახ. დ3-2: Form2.C# - პროგრამის შედეგი. ამორჩევის შესაძლებლობები

## ლისტინგი-დ2

```
// Form1.cs --- ექსპერტის ინტერფეისი - ფაქტორის მიხედვით
// ღონიამიებათა შეფასების შეტანა -----
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Expert
{
    public partial class Form1 : Form
    {
        public Form1()
        {
```

```

        InitializeComponent();
        this.Refreshgrid();
    }

private void toolStripButton1_Click(object sender, EventArgs e)
{
    this.Refreshgrid();
}

private void Refreshgrid()
{
    DataSet1TableAdapters.SpGetRelationTableAdapter ad = new
Expert.DataSet1TableAdapters.SpGetRelationTableAdapter();
    spGetRelationDataTableBindingSource.DataSource =
ad.GetData();
}
private void newToolStripButton_Click(object sender,
EventArgs e)
{
    Form2 f = new Form2();
    f.ShowDialog();
    this.Refreshgrid();
}

private void openToolStripButton_Click(object sender,
EventArgs e)
{
    DataSet1.SpGetRelationRow row =
((DataRowView)spGetRelationDataTableBindingSource.Current).Row as
DataSet1.SpGetRelationRow;
    Form3 fr = new Form3(row.RowId, false, 0, 0, 0);
    fr.ShowDialog();
    this.bindingdetail(row.RowId);
}
private void bindingdetail(int HedId)
{

DataSet1TableAdapters.SpGetGonisdziebaRelationTableAdapter gon = new
Expert.DataSet1TableAdapters.SpGetGonisdziebaRelationTableAdapter();
    spGetGonisdziebaRelationDataTableBindingSource.DataSource =
gon.GetData(HedId);
}

private void
spGetRelationDataTableBindingSource_CurrentItemChanged(object sender,
EventArgs e)
{
    DataSet1.SpGetRelationRow row =
((DataRowView)spGetRelationDataTableBindingSource.Current).Row as
DataSet1.SpGetRelationRow;
    this.bindingdetail(row.RowId);
}

private void toolStripButton2_Click(object sender, EventArgs e)
{
    if (dataGridView2.RowCount == 0)
    {
        MessageBox.Show("monishnet gasakoretirebeli veli");
        return; }
}

```

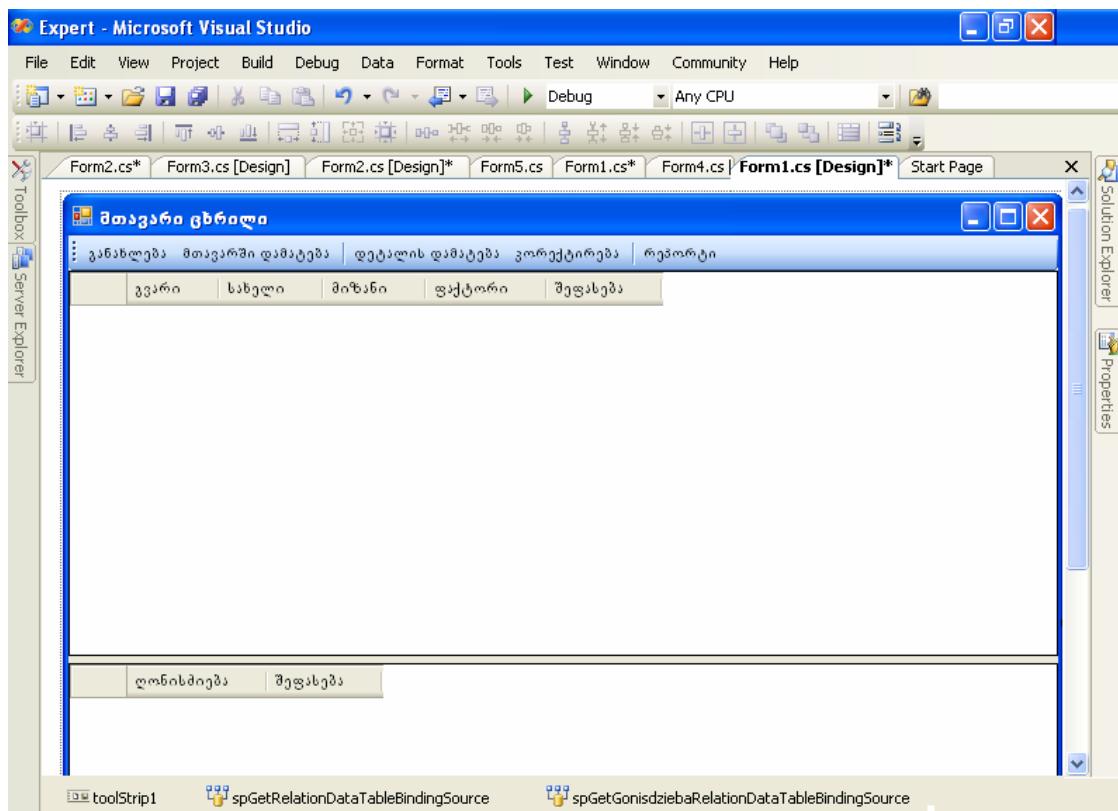
```

        DataSet1.SpGetRelationRow row =
((DataRowView)spGetRelationDataTableBindingSource.Current).Row as
DataSet1.SpGetRelationRow;
        DataSet1.SpGetGonisdziebaRelationRow row1 =
((DataRowView)spGetGonisdziebaRelationDataTableBindingSource.Current)
.Row as DataSet1.SpGetGonisdziebaRelationRow;
        Form3 fr = new Form3(row.RowId, true, row1.GonisdziebaId,
row1.shefaseba, row1.RowId);
        fr.ShowDialog();
        this.bindingdetail(row.RowId);
    }

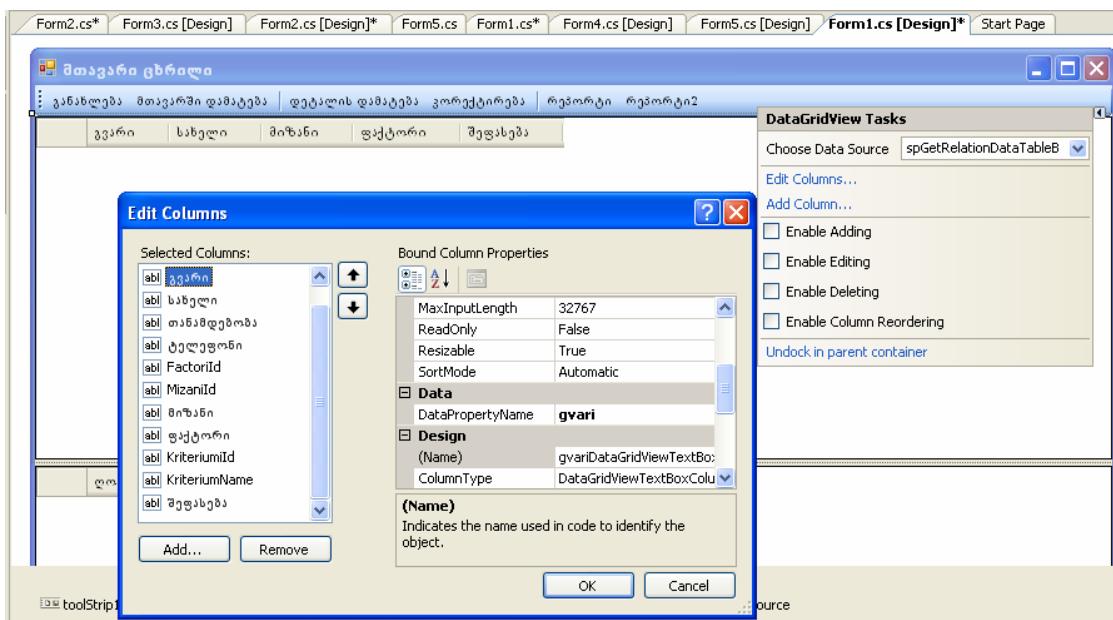
private void toolStripButton3_Click(object sender, EventArgs e)
{
    Form4 fr = new Form4();
    fr.ShowDialog();
}

private void toolStripButton4_Click(object sender, EventArgs e)
{
    Form5 fr = new Form5();
    fr.ShowDialog();
}
}
}

```



**ნახ. დ4: Form1.C# - პროგრამის შედეგი**



ნახ. დ5-1: ცხრილის სვეტებში ველების რედაქტორი

გვარი	სახელი	მიზანი	ფაქტორი	შეფასება
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	მატერიალურ-ტექნიკური შაზა	20
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	პიტურიენტები	10
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	ვადრები	20
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	სასწავლო პროცესის ორგანიზაცია	20
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	ლექტორთა პირობები	10
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	მეცანეობების ხარისხი	20

ლონისმიერა	შეფასება
ახალი ტექნიკის შემცნება	40
ახალი ტექნოლოგიის დანერგვა	40
აუდიტორიის აღჭურვა მულტიმედიალური ტექნიკით	20

ნახ. დ5-2: ა-ვ) ფაქტორით ღონისძიებების არჩევისა და შეფასებების შეტანის ინტერფეისები

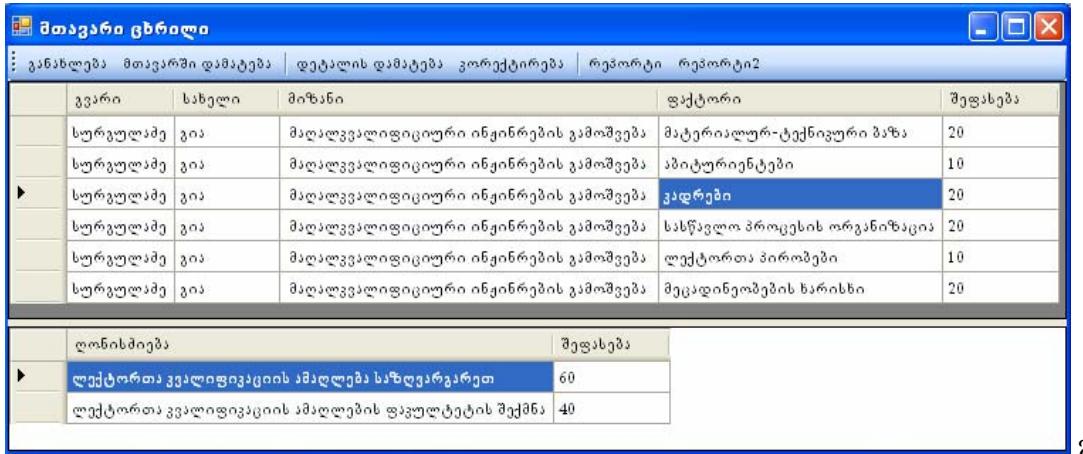
გვარი	სახელი	მიზანი	ფაქტორი	შეფასება
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	მატერიალურ-ტექნიკური შაზა	20
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	პიტურიენტები	10
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	ვადრები	20
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	სასწავლო პროცესის ორგანიზაცია	20
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	ლექტორთა პირობები	10
სურგულაძე	გია	მაღალგვალიფიციური ინჟინერის გამოშვება	მეცანეობების ხარისხი	20

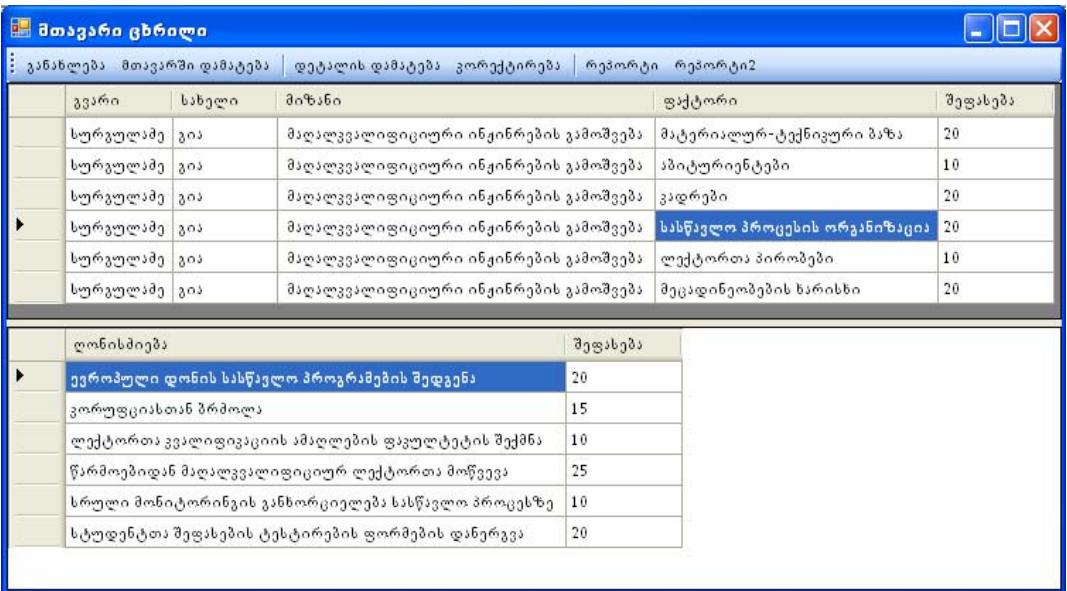
ლონისმიერა	შეფასება
მაღალულიანი აბიტურიენტების მოზიდვა	70
აბიტურიენტების საწყისი ტესტირების ჩატარება	30

ა)

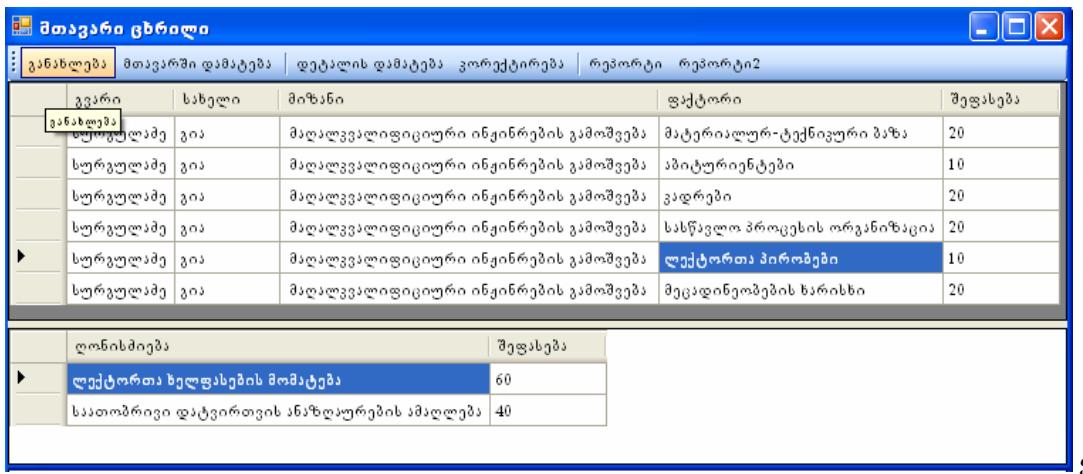
ბ)



3)



g)



3)

ნახ. დ5-2: ა-ვ) ინტერფეისების გაგრძელება

**გთავარი ცხრილი**

განაზღება მთავარი დამატება | დეტალის დამატება კორექტირება | რეპორტი რეპორტი2

გვარი	სახელი	მიზანი	ფაქტორი	შეფასება
სურგულაძე	გია	მაღალგვალიფიციური ინგინერის გამოშვება	მატერიალურ-ტექნიკური სახა	20
სურგულაძე	გია	მაღალგვალიფიციური ინგინერის გამოშვება	პიტურიენტები	10
სურგულაძე	გია	მაღალგვალიფიციური ინგინერის გამოშვება	კაფები	20
სურგულაძე	გია	მაღალგვალიფიციური ინგინერის გამოშვება	სასწავლო პროცესის ორგანიზაცია	20
სურგულაძე	გია	მაღალგვალიფიციური ინგინერის გამოშვება	ლექტორთა პირობები	10
სურგულაძე	გია	მაღალგვალიფიციური ინგინერის გამოშვება	შეცადინეობების ზრისისი	20

ლონისძიება	შეფასება
შეცადინეობების საწყილ-შეთოდური დანის შეღლება	45
ლექტირთა შეცადინეობებზე ურთიერთდასწრება	15
სტუდენტთა მიერ ლექტირთა ანონიმური შეფასების დაწერვა	15
სტუდენტთა შეფასების ტესტირების ფორმების დაწერვა	25

3)

### ნახ. დ5-2: ა-ვ) ინტერფეისების გაგრძელება

#### ლისტინგი-დ3

```
// Form3.cs --- ექსპერტის ინტერფეისი - ღონიაძიებათა შეფასების შეტანა -----
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Expert
{
    public partial class Form3 : Form
    {
        int relid;
        bool editstatus = false;
        int _gon;
        int shef;
        int _rowid;
        public Form3(int relationid, bool status, int gonisdieba, int shefaseba, int RRowid)
        {
            InitializeComponent();
            relid = relationid;
            editstatus = status;
            _gon = gonisdieba;
            shef = shefaseba;
            _rowid = RRowid;
        }

        private void Form3_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
            'dataSet1.Gonisdziebebi' table. You can move, or remove it, as
            needed.

            this.gonisdziebebiTableAdapter.Fill(this.dataSet1.Gonisdziebebi);
        }
    }
}
```

```

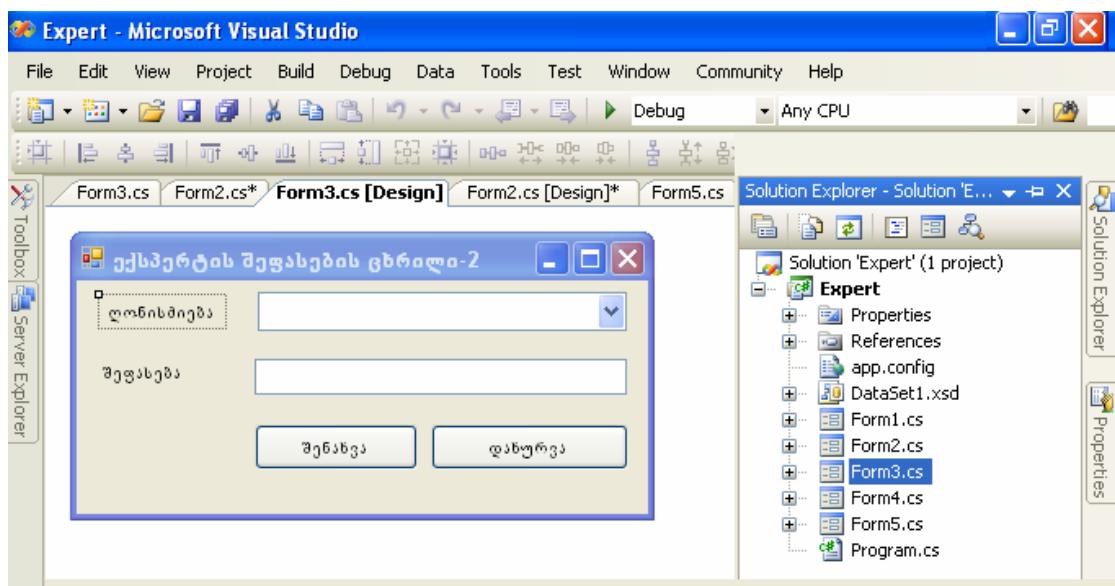
        if (editstatus)
        {
            comboBox1.SelectedValue = _gon;
            textBox1.Text = shef.ToString();
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            DataSet1TableAdapters.QueriesTableAdapter q = new
Expert.DataSet1TableAdapters.QueriesTableAdapter();
            if (editstatus)

                q.SpEditGonisdziebaRelation(Convert.ToInt32(comboBox1.SelectedValue),
Convert.ToInt32(textBox1.Text), _rowid);
            else
                q.SpaddGonisdziebaRelation(relid,
Convert.ToInt32(comboBox1.SelectedValue),
Convert.ToInt32(textBox1.Text));
            this.Close();
        }
        catch (Exception)
        {

            MessageBox.Show("chawera ver ganxorcielda, sheamowmet
shevsebuli gaqvT Tu ara yvela veli");
        }
    }
}

```



ნახ. დ6-1: ღონისძიების შეფასებათა შეტანის ინტერფეისი

ნახ. დ6-2: ღონისძიებების შეფასებათა შეტანის (ა,ბ) და კორექტირების (გ) ინტერფეისები

#### ლისტინგი-დ4

```
// Form4.cs --- ექსპერტის ინტერფეისი - შედეგების გამოტანის ცხრილი ----

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Expert
{
    public partial class Form4 : Form
    {
```

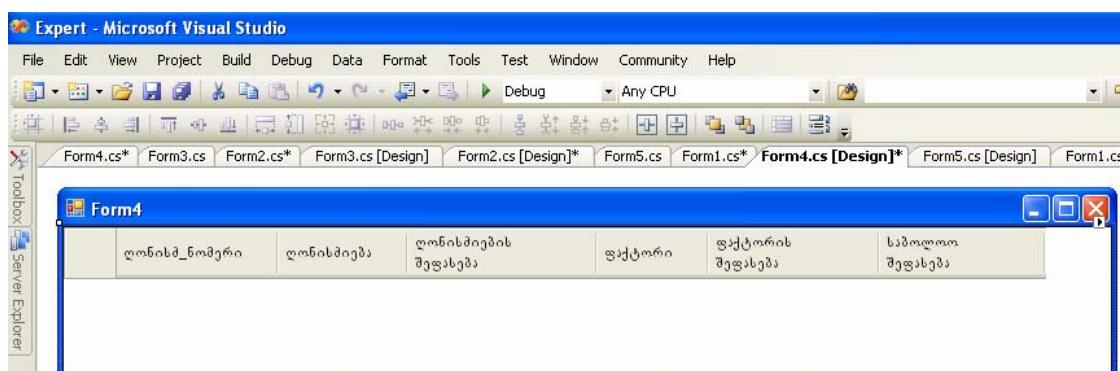
```

public Form4()
{
    InitializeComponent();
    DataSet1TableAdapters.SpGetGonisdziebafaqtoriTableAdapter
f = new
Expert.DataSet1TableAdapters.SpGetGonisdziebafaqtoriTableAdapter();
    spGetGonisdziebafaqtoriDataTableBindingSource.DataSource
= f.GetData();
}

private void Form4_Load(object sender, EventArgs e)
{
}

}
}

```



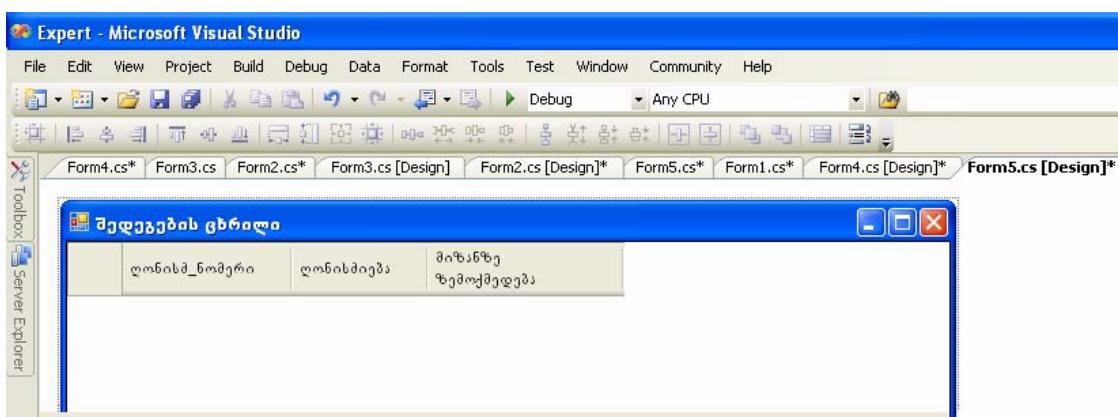
ნახ. დ7-1: შედეგების გამოტანის ინტერფეისი

ლონისძნომერი	ლონისმიერა	ლონისმიების შეფასება	ფაქტორი	ფაქტორის შეფასება
1	ახალი ტექნიკის შემწილი	40	მატერიალურ-ტექნიკური პარა	20
2	ახალი ტექნიკური დანერგვა	40	მატერიალურ-ტექნიკური პარა	20
3	აუტომატიზირებული ტექნიკის განვითარება	20	მატერიალურ-ტექნიკური პარა	20
6	მაღალურიანი პირუტინების მიზანდება	70	პირუტინები	10
7	პირუტინების სტაციონარული ტექნიკის განვითარება	30	პირუტინები	10
4	ლექტორთა კვალიფიკაციის ამაღლება საზღვარგარეთ	60	კადრები	20
5	ლექტორთა კვალიფიკაციის ამაღლების ფაკულტეტის შექმნა	40	კადრები	20
8	კვალიფიცირებული დანის სასწავლო პროცესების შედეგები	20	სასწავლო პროცესის ორგანიზაცია	20
13	კოროფიციალური მიზანი	15	სასწავლო პროცესის ორგანიზაცია	20
5	ლექტორთა კვალიფიკაციის ამაღლების ფაკულტეტის შექმნა	10	სასწავლო პროცესის ორგანიზაცია	20
11	წარმოებითან მაღალვალიფიციურ ლექტორთა მოწვევა	25	სასწავლო პროცესის ორგანიზაცია	20
12	სრული მოიცოდონების განხორციელება სასწავლო პროცესზე	10	სასწავლო პროცესის ორგანიზაცია	20
15	სტუდენტთა შეფასების ტექსტინგის ფორმების დაწერვა	20	სასწავლო პროცესის ორგანიზაცია	20
9	ლექტორთა ხელფასების მომტებები	60	ლექტორთა მირობები	10
10	საათობრივი დატერმინის ანაზღაურების ამაღლება	40	ლექტორთა მირობები	10
14	მეცანიკონის სასწავლო-მეთოდური დონის ამაღლება	45	მეცანიკონის ხარისხი	20
16	ლექტორთა მეცანიკონის ურთიერთდაწერება	15	მეცანიკონის ხარისხი	20
17	სტუდენტთა მიერ ლექტორთა ანონიმური შეფასების დაწერვა	15	მეცანიკონის ხარისხი	20
15	სტუდენტთა შეფასების ტესტირების ფორმების დაწერვა	25	მეცანიკონის ხარისხი	20

ნახ. დ7-2: შედეგების გამოტანის ინტერფეისი

## ლისტინგი-დ5

```
// Form5.cs --- ექსპერტის ინტერფეისი - საბოლოო შედეგების
// გამოტანის ცხრილი -----  
  
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;  
  
namespace Expert
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();  
  
DataSet1TableAdapters.SpGetGonisdziebafaqtoriDistinctTableAdapter
ta=new
Expert.DataSet1TableAdapters.SpGetGonisdziebafaqtoriDistinctTableAdap
ter();  
  
spGetGonisdziebafaqtoriDistinctDataTableBindingSource.DataSource =
ta.GetDataTable();
        }
  
        private void Form5_Load(object sender, EventArgs e)
        {
        }
    }
}
```



ნახ. დ8-1: საბოლოო შედეგების გამოტანის ცხრილი

	ლონისმ_ნომერი	ლონისმიერა	მიზანზე ზემოქმედება
▶	4	ლექტორთა კვალიფიციის ამაღლება საზღვარგარეთ	1200
	5	ლექტორთა კვალიფიციის ამაღლების ფავულტეტის შექმნა	1000
	14	მეცნადინეობების სასწავლო-მეთოდური დონის ამაღლება	900
	15	სტუდენტთა შეფასების ტესტირების ფორმების დანერგვა	900
	1	ახალი ტექნიკის შემცნა	800
	2	ახალი ტექნოლოგიის დანერგვა	800
	6	მაღალქულიანი პიტტორიენტების მოზიდვა	700
	9	ლექტორთა ხელფასების მომატება	600
	11	წარმოებითან მაღალგალიფიციურ ლექტორთა მოწვევა	500
	10	საათობრივი დატვირთვის ანაზღაურების ამაღლება	400
	3	სუდიტორიების აღჭურვა მულტიმედიალური ტექნიკით	400
	8	ეკროპული დონის სასწავლო პროგრამების შეფასენა	400
	13	კორუფციასთან შრმოლა	300
	16	ლექტინრთა მეცნადინეობებზე ურთიერთდასწრება	300
	7	პიტტორიენტების საწყისი ტესტირების ჩატარება	300
	17	სტუდენტთა მიერ ლექტორთა ანონიმური შეფასების დანერგვა	300
	12	სრული მონიტორინგის განხორციელება სასწავლო პროცესზე	200

ნახ. დ8-2: საბოლოო შედეგების ცხრილის ფრაგმენტი

### მონაცემთა ბაზის ცხრილები

dbo.Expert: Table(gia.Expert)		
Column Name	Data Type	Allow Nulls
▶ ExpertId	int	<input type="checkbox"/>
gvari	nvarchar(30)	<input checked="" type="checkbox"/>
Saxeli	nvarchar(20)	<input checked="" type="checkbox"/>
Tanamd	nvarchar(30)	<input checked="" type="checkbox"/>
Telefoni	nvarchar(20)	<input checked="" type="checkbox"/>
staJI	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

ნახ. დ9-1: სტრუქტურა ექსპერტები

Expert - Microsoft Visual Studio

The screenshot shows the Microsoft Visual Studio interface with the title bar "Expert - Microsoft Visual Studio". The menu bar includes File, Edit, View, Project, Build, Debug, Data, Query Designer, Tools, Test, Window, Community, and Help. The toolbar has icons for file operations like Open, Save, and Print. The status bar shows "Any CPU". The main area displays a grid of data from the "Expert" table. The columns are ExpertId, gvari, Saxeli, Tanamd, Telefoni, and staJi. The data rows are as follows:

	ExpertId	gvari	Saxeli	Tanamd	Telefoni	staJi
▶	1	სორიანაძე	გია	მიმართულების ხელმისაწვდომი	37-37-37	35
	2	ვანია	ომარი	სრ, კრონებითი	55-55-55	30
	3	გამგებავა	ომარი	ასოც, კრონებითი	25-25-25	25
	4	ოლეგრია	ნინო	ასოც, კრონებითი	39-39-39	15
	5	სოსახველი	თემურაზი	ასოც, კრონებითი	99-99-99	30
	6	გოგიაშვილი	გიორგი	ლეპარტამენტის უფროსი	32-32-32	37
	7	გრიგორიშვილი	არჩილი	რეკრიუს მოადგილი	95-95-95	25
*	NULL	NULL	NULL	NULL	NULL	NULL

ნაბ. დ9-2: ექსპერტები

dbo.Miznebi: Table(gia.Expert) | dbo.Expert: Table(gia.Expert)

The screenshot shows the Microsoft Visual Studio interface with the title bar "Expert - Microsoft Visual Studio". The main area displays the structure of the "Miznebi" table. The columns are MizaniId and dasaxeleba. MizaniId is of type int and Allow Nulls is checked. dasaxeleba is of type nvarchar(MAX) and Allow Nulls is checked.

	Column Name	Data Type	Allow Nulls
▶	MizaniId	int	<input type="checkbox"/>
	dasaxeleba	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

ნაბ. დ10-1: სტრუქტურა მიზნები

Miznebi: Query(gia.Expert) | dbo.Ronisdieba...ble(gia.Expert)

The screenshot shows the Microsoft Visual Studio interface with the title bar "Expert - Microsoft Visual Studio". The main area displays the data in the "Miznebi" table. The columns are MizaniId and dasaxeleba. The data rows are as follows:

	MizaniId	dasaxeleba
▶	1	მათალევალიანი ინიციატივის გამოშვება
	2	სწავლების ხარისხის აზაღურება კრონები
	3	ლეპარტა სოციალური საკითხების გაუმჯობესება
*	NULL	NULL

Below the table, there are navigation buttons (back, forward, first, last, etc.) and a message: "Cell is Read Only."

ნაბ. დ10-2: მიზნები

Column Name	Data Type	Allow Nulls
FactorId	int	<input type="checkbox"/>
FactorName	nvarchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

ნაბ. დ11-1: სტრუქტურა ფაქტორები

	FactorId	FactorName
▶	1	მატერიალურ-ტექნიკური გარე
	2	კადრები
	3	ენერგეტიკული
	4	სასწავლო პროცესის ორგანიზაცია
	5	ლიკელობის პირობები
*	6	მეცნიერებების სარისხი
	NULL	NULL

ნაბ. დ11-2: ფაქტორები

Column Name	Data Type	Allow Nulls
KriteriumId	int	<input type="checkbox"/>
KriteriumName	nvarchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

ნაბ. დ12-1: სტრუქტურა კრიტერიუმები

	KriteriumId	KriteriumName
▶	1	ეკონომიკური
	2	ტექნიკური
	3	სოციალური
	4	ეროვნული
*	NULL	NULL

ნაბ. დ12-2: კრიტერიუმები

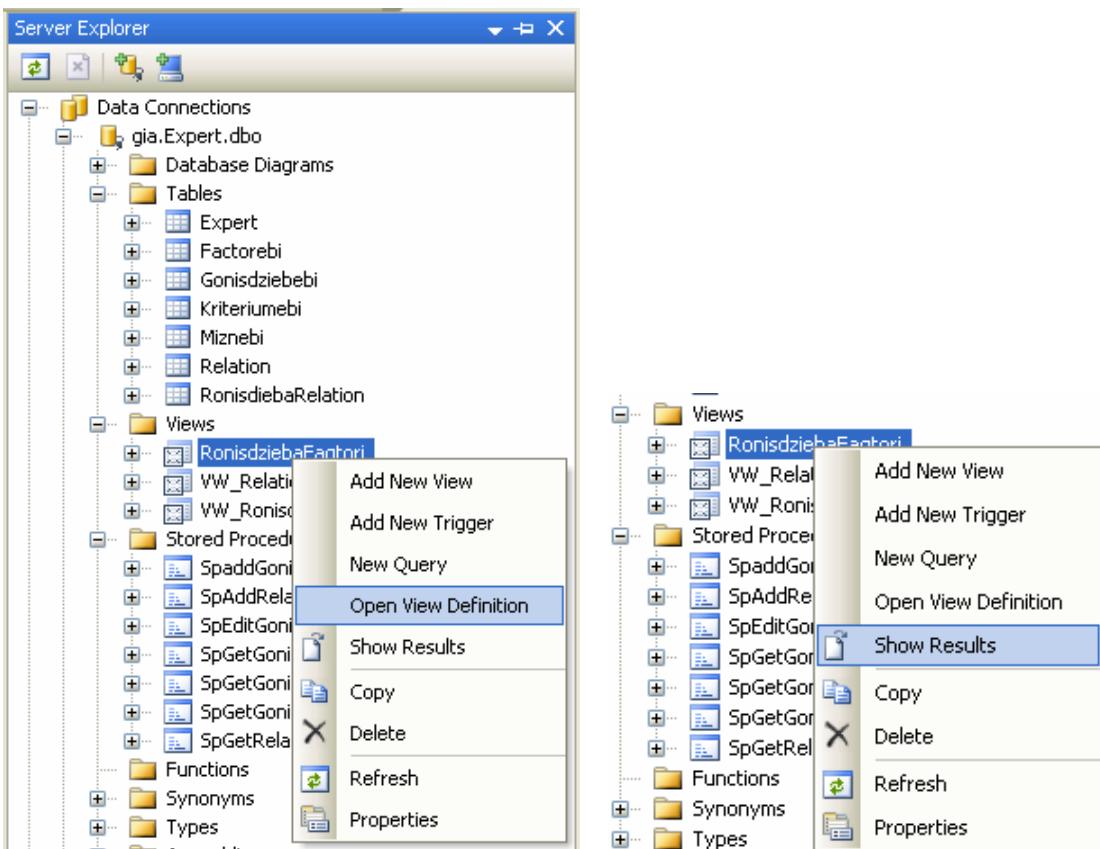
	Column Name	Data Type	Allow Nulls
▶	GonisdziebaId	int	<input type="checkbox"/>
	GonisdziebaName	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

ნაბ. დ13-1: სტრუქტურა ღონისძიებები

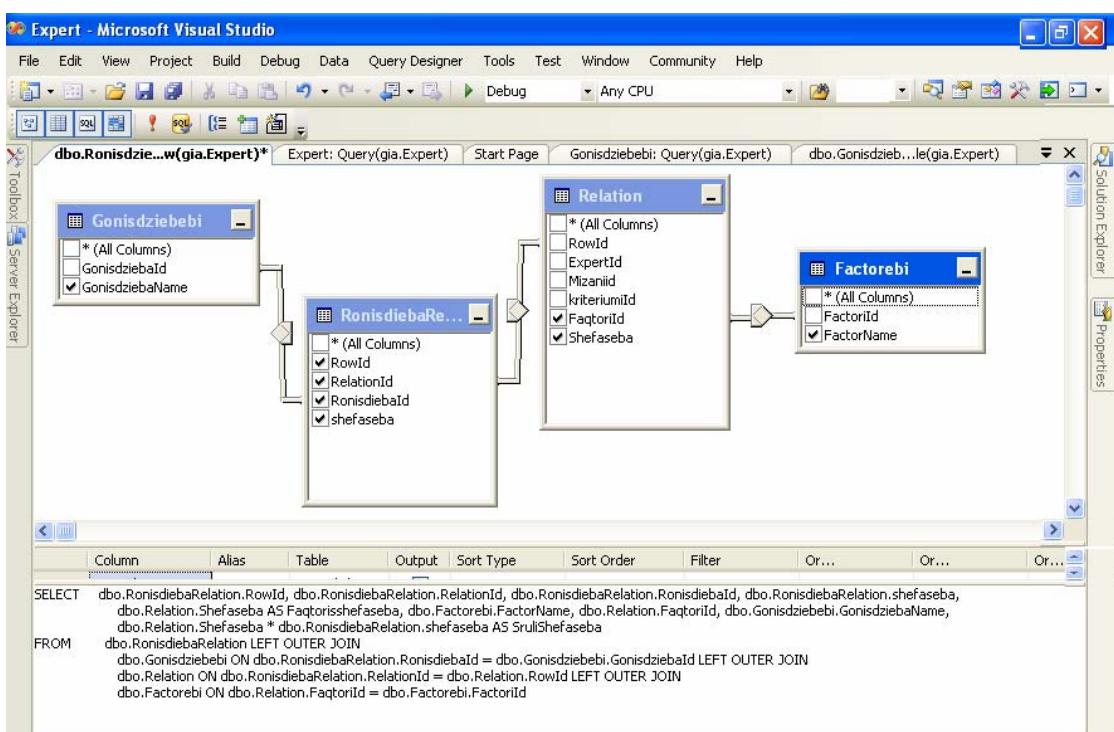
	GonisdziebaId	GonisdziebaName
▶	1	ახალი ცეკვის შემება
	2	ახალი ცეკვის დანერგვა
	3	აუთიფირის აღწერა მულტიმედიალური ცეკვისთ
	4	ლექტორის კვალიფიკაციის ამაღლება საზოგადოებრივ
	5	ლექტორის კვალიფიკაციის ამაღლების ფაკულტეტის შექმნა
	6	მაღალქულიანი აშერტებულების მოზიდვა
	7	აშერტებულების საწყისი ტესტირების ჩატარება
	8	კრიპული დონის სასწავლო პროექტების შედეგები
	9	ლექტორის ხელფასების მომატება
	10	საათობრივი დაცვითი თვის ანაზღაურების ამაღლება
	11	წარმოებით მაღალკვალიფიციურ ლექტორის მოწვევა
	12	სრული მონიკრონების განხილვება სასწავლო პროცესზე
	13	კორუფციასთან მომთავრებელი
	14	მეცადინეობების სასწავლო-მეთოდური დონის ამაღლება
	15	სტუდენტთა შეფასების ტესტირების ფორმების დანერგვა
	16	ლექტორის მეცადინეობებზე ურთიერთდასწრება
	17	სტუდენტთა მიერ ლექტორის ანონიმური შეფასების დანერგვა
*	NULL	NULL

ნაბ. დ13-2: ღონისძიებები

## Expert dbo მონაცემთა ბაზის Views წარმოდგენები SQL ტექსტებით



**ნაბ. დ14-1:** View ცხრილთაკავშირებისა და შესაბამისი SQL-ის გამოტანის  
საშუალებები



**ნაბ. დ15-1:** View ღონისძიება-ფაქტორი SQL-ით

Expert - Microsoft Visual Studio

File Edit View Project Build Debug Data Query Designer Tools Test Window Community Help

Change Type

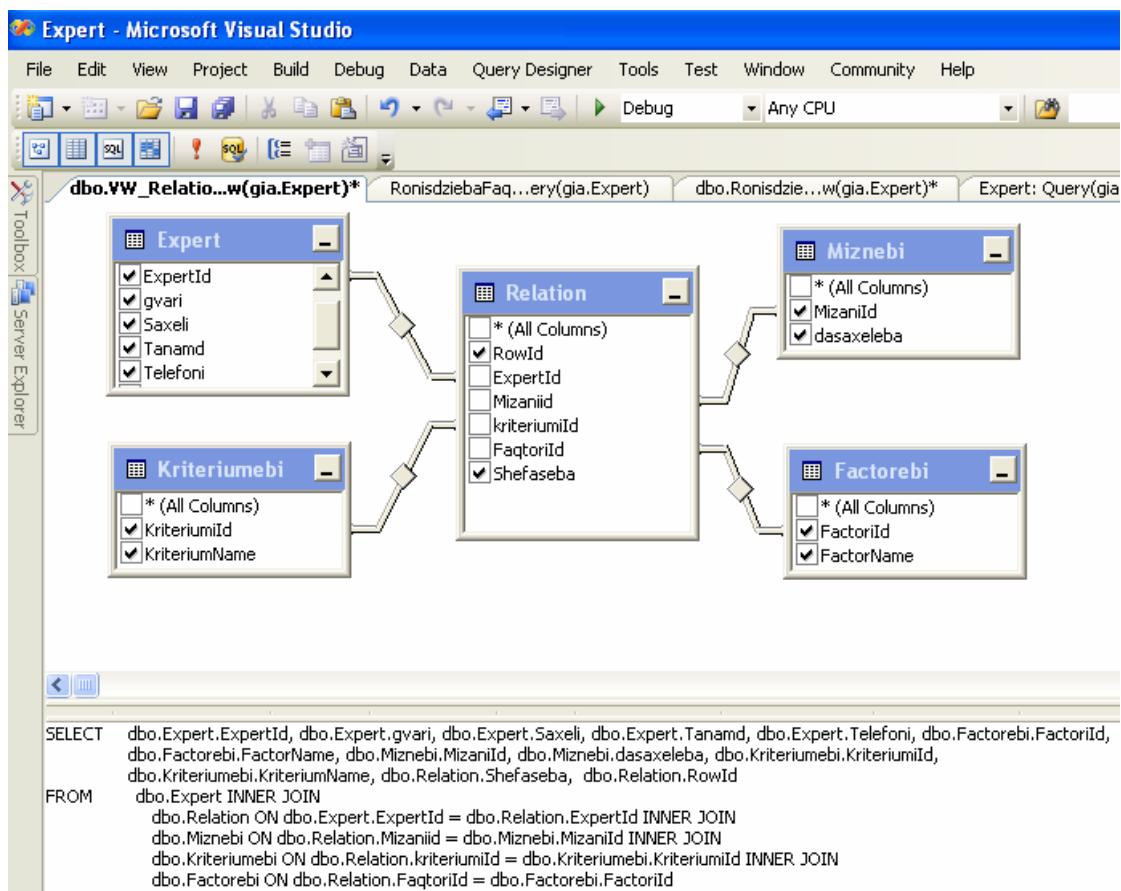
RonisdziebaFaq...ery(gia.Expert) dbo.Ronisdzie...w(gia.Expert)\* Expert: Query(gia.Expert) Start Page Gonisdziebi: Query(gia.Expert)

RowId RelationId Ronisdziebatd shefaseba Factorishefas... FactorName Faq... GonisdziebiName SruliShefaseba

4	1	1	40	20	გაცემის დროულობა განახ.	1	ახალი ტექნიკის შექმნა	800
5	1	2	40	20	გაცემის დროულობა განახ.	1	ახალი ტექნიკის შექმნა	800
6	1	3	20	20	გაცემის დროულობა განახ.	1	აუთიფიციანის აუმჯობესებულობრივ ტექნიკა	400
7	10	6	70	10	აუთიფიციანი	3	მიზნური მიზნების მიზნური მიზნები	700
8	10	7	30	10	აუთიფიციანი	3	აუთიფიციანის საფუძვლის აუმჯობესებულობრივ ტექნიკა	300
9	11	4	60	20	კადეტი	2	ლეგიტიმის კადეტის აუმჯობესებულობრივ ტექნიკა	1200
10	11	5	40	20	კადეტი	2	ლეგიტიმის კადეტის აუმჯობესებულობრივ ტექნიკა	800
11	12	8	20	20	სასწავლის მიზნები	4	კომუნიკაციის სასწავლის აუმჯობესებულობრივ ტექნიკა	400
12	12	13	15	20	სასწავლის მიზნები	4	კომუნიკაციის სასწავლის აუმჯობესებულობრივ ტექნიკა	300
13	12	5	10	20	სასწავლის მიზნები	4	ლეგიტიმის კადეტის აუმჯობესებულობრივ ტექნიკა	200
14	12	11	25	20	სასწავლის მიზნები	4	ტერმინური მიზნური მიზნების დამტკიცება	500
15	12	12	10	20	სასწავლის მიზნები	4	სისული მონიტორინგის კადეტის აუმჯობესებულობრივ ტექნიკა	200
16	12	15	20	20	სასწავლის მიზნები	4	სტატისტიკური კადეტის აუმჯობესებულობრივ ტექნიკა	400
17	13	9	60	10	ლეგიტიმის კადეტი	5	ლეგიტიმის კადეტის აუმჯობესებულობრივ ტექნიკა	600
18	13	10	40	10	ლეგიტიმის კადეტი	5	სასწავლის დამტკიცებულობრივ ტექნიკა	400
19	14	14	45	20	მიზანური მიზნები	6	მიზანური მიზნების სასწავლის აუმჯობესებულობრივ ტექნიკა	900
20	14	16	15	20	მიზანური მიზნები	6	ლეგიტიმის მიზანური მიზნების აუმჯობესებულობრივ ტექნიკა	300
21	14	17	15	20	მიზანური მიზნები	6	სტატისტიკური მიზანური მიზნების აუმჯობესებულობრივ ტექნიკა	300
22	14	15	25	20	მიზანური მიზნები	6	სტატისტიკური მიზანური მიზნების აუმჯობესებულობრივ ტექნიკა	500
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

1 of 19

ნახ. დ15-2: შედეგი - View ღონისძიება-ფაქტორი SQL-ით



ნახ. დ16-1: View ექსპერტი-ღონისძიება-ფაქტორი კრიტერიუმი SQL-ით

**VW\_Relation: Query(gia.Expert)**

ExpertId	gvvari	Saxeli	Tanamd	Telefoni	FactorId	FactorName	MizaniId
1	სურალაძე	გია	მიმართულების ხელმისაწვდომი	37-37-37	1	მაკრისალურ-ტექნიკური გარა	1
1	სურალაძე	გია	მიმართულების ხელმისაწვდომი	37-37-37	3	აპიკარისტები	1
1	სურალაძე	გია	მიმართულების ხელმისაწვდომი	37-37-37	2	კატრუზი	1
1	სურალაძე	გია	მიმართულების ხელმისაწვდომი	37-37-37	4	სასწავლო პროცესის ორგანიზაცია	1
1	სურალაძე	გია	მიმართულების ხელმისაწვდომი	37-37-37	5	ლექციარის პიროვნები	1
1	სურალაძე	გია	მიმართულების ხელმისაწვდომი	37-37-37	6	მეცნიერებების სარისხი	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**RonisdziebaFaq...ery(gia.Expert)**

dasaxeleba	KriteriumId	KriteriumName	Shefaseba	RowId
მაღალვალიფიციური ინჟინერების გამოშვება	1	კვანძობელური	20	1
მაღალვალიფიციური ინჟინერების გამოშვება	1	კვანძობელური	10	10
მაღალვალიფიციური ინჟინერების გამოშვება	1	კვანძობელური	20	11
მაღალვალიფიციური ინჟინერების გამოშვება	1	კვანძობელური	20	12
მაღალვალიფიციური ინჟინერების გამოშვება	1	კვანძობელური	10	13
მაღალვალიფიციური ინჟინერების გამოშვება	1	კვანძობელური	20	14
NULL	NULL	NULL	NULL	NULL

ნახ. დ16-2: შედეგი View ექსპერტი-ღონისძიება-ფაქტორი კრიტერიუმი SQL-ით

**Expert - Microsoft Visual Studio**

**dbo.VW\_Ronisd...w(gia.Expert)\***   **VW\_Relation: Query(gia.Expert)**   **dbo.VW\_Relatio...w(gia.Expert)\***   **RonisdziebaFaq...ery(gia.Expert)**

**Gonisdziebebi**

**RonisdziebaRelation**

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...
GonisdziebaId	Gonisdziebebi	Gonisdziebebi	<input checked="" type="checkbox"/>					
GonisdziebaName	Gonisdziebebi	Gonisdziebebi	<input checked="" type="checkbox"/>					
shefaseba	Ronisdzieba...	Ronisdzieba...	<input checked="" type="checkbox"/>					
RelationId	Ronisdzieba...	Ronisdzieba...	<input checked="" type="checkbox"/>					

```

SELECT dbo.Gonisdziebebi.GonisdziebaId, dbo.Gonisdziebebi.GonisdziebaName, dbo.RonisdziebaRelation.shefaseba, dbo.RonisdziebaRelation.RelationId,
       dbo.RonisdziebaRelation.RowId
FROM dbo.Gonisdziebebi INNER JOIN
      dbo.RonisdziebaRelation ON dbo.Gonisdziebebi.GonisdziebaId = dbo.RonisdziebaRelation.RonisdziebaId
  
```

ნახ. დ17-1: View ღონისძიებათა შეფასებების კავშირი ფაქტორებთან SQL-ით

Expert - Microsoft Visual Studio

File Edit View Project Build Debug Data Query Designer Tools Test Window Community Help

Change Type

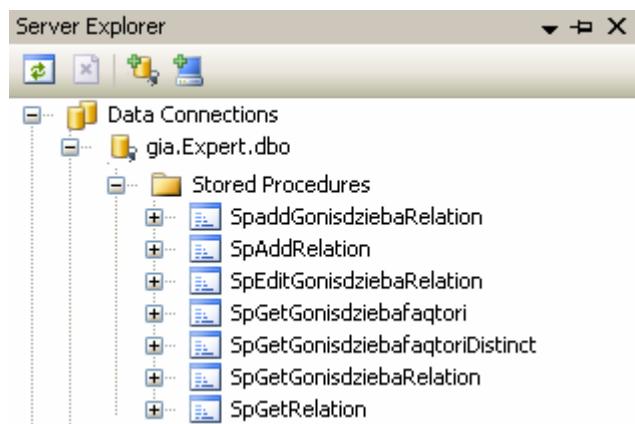
VW\_Ronisdzieb...ry(gia.Expert) | dbo.VW\_Ronisd...w(gia.Expert)\* | VW\_Relation: Query(gia.Expert) | dbo.VW\_Relatio...w(gia.Expert)\*

Toolbox Server Explorer

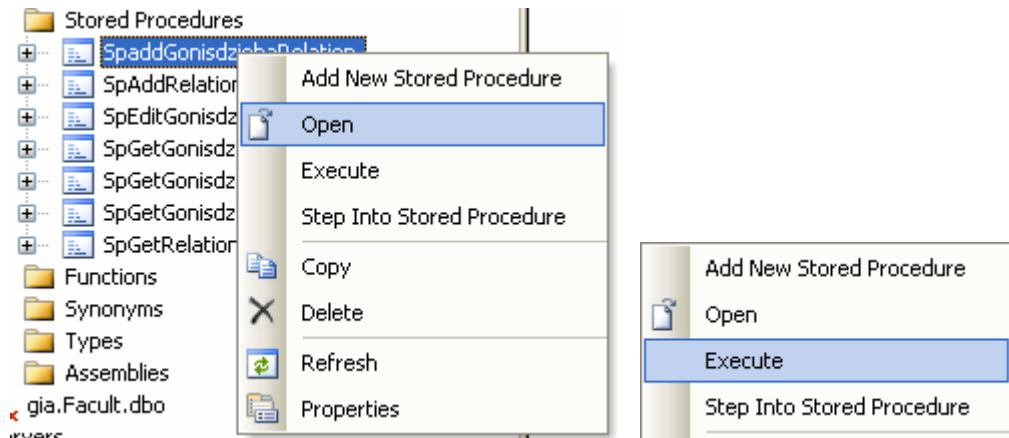
	GonisdziebaId	GonisdziebaName	shefaseba	RelationId	RowId
1	1	ახალი კექნივის შეძენა	40	1	4
2	2	ახალი ტექსტოდოკუმენტის დაწერვა	40	1	5
3	3	აუთომატური აღმურავა მულტიმედიალური კექნივის	20	1	6
4	4	ლექტორის კვალიფიკაციის ამაღლება საზოგადოებრივ	60	11	9
5	5	ლექტორის კვალიფიკაციის ამაღლების ფაკულტეტის შექმნა	40	11	10
6	6	ლექტორის კვალიფიკაციის ამაღლების ფაკულტეტის შექმნა	10	12	13
7	7	მაღალი კოდირებულის მონიტორი	70	10	7
8	8	აშენებული საწყისი კესტინგის ჩატარება	30	10	8
9	9	აშენებული დონის სასწავლის პროგრამების შეღება	20	12	11
10	10	აშენებული დონის სასწავლის პროგრამების ამაღლება	40	13	18
11	11	აშენებული მონიტორინგის განვითარებული ლექტორის მოწვევა	25	12	14
12	12	სრული მონიტორინგის განვითარებული სასწავლო პროცესზე	10	12	15
13	13	კორეფიციანი მომთავრებელი	15	12	12
14	14	მეცნიერებების სასწავლო-მუსიკური დონის ამაღლება	45	14	19
15	15	სტუდენტთა შეჯასვის კესტინგის ფორმების დაწერვა	20	12	16
16	16	სტუდენტთა შეჯასვის კესტინგის ფორმების დაწერვა	25	14	22
17	17	სტუდენტთა მეცნიერებების ურთიერთდასწრება	15	14	20
*	NULL	NULL	NULL	NULL	NULL

ნახ. დ17-2: შედეგი View ღონისძიებათა შეფასებები

## Expert.dbo მონაცემთა ბაზის Stored Procedures



ნახ. დ18-1: შენახვადი პროცედურების სია



ნაბ. დ18-2: ა) Open

ბ) Execute:

შენახვადი პროცედურების ლისტინგები:

#### ლისტინგი 1:

```
-- SpaddGonisdziebaRelation
ALTER PROCEDURE [dbo].[SpaddGonisdziebaRelation]
@relationID int,
@gonisdziebaId int,
@shefaseba int
AS
BEGIN
--დონისძიების შეფასების შეტანა.
    INSERT INTO [Expert].[dbo].[RonisdziebaRelation]
        ([RelationId]
        ,[RonisdziebaId]
        ,[shefaseba])
    VALUES
        (@relationID ,
        @gonisdziebaId,
        @shefaseba )
END
```

#### ლისტინგი 2:

```
-- SpAddRelation
ALTER PROCEDURE [dbo].[SpAddRelation]
@ExpertId int,
@Mizaniid int,
@kriteriumiId int,
@FaqtoriId int,
@Shefaseba int
AS
BEGIN
--ფაქტორის შეფასების შეტანა
    INSERT INTO [Expert].[dbo].[Relation]
        ([ExpertId]
        ,[Mizaniid]
        ,[kriteriumiId]
```

```

        , [FaqtoriId]
        ,[Shefaseba])
VALUES
    (@ExpertId
    ,@Mizaniid
    ,@kriteriumiId
    ,@FaqtoriId
    ,@Shefaseba)
END

```

### ლისტინგი 3:

```

-- SpEditGonisdziebaRelation
ALTER PROCEDURE [dbo].[SpEditGonisdziebaRelation]
@gonisdziebaId int,
@shefaseba int,
@RowId int
AS
BEGIN
--ღონისძიების შეფასების კორექტირება
    UPDATE [Expert].[dbo].[RonisdziebaRelation]
    SET [RonisdziebaId] = @gonisdziebaId
        ,[shefaseba] = @shefaseba
    WHERE RowId=@RowId
END

```

### ლისტინგი 4:

```

-- SpGetGonisdziebafaqtori
ALTER PROCEDURE [dbo].[SpGetGonisdziebafaqtori]
AS
BEGIN
--პირველი რეპორტი
    select *,(FaqtorisShefaseba*shefaseba) as SruliShefaseba
from dbo.RonisdziebaFaqtori
END

```

### ლისტინგი 5:

```

-- SpGetGonisdziebafaqtoriDistinct
ALTER PROCEDURE [dbo].[SpGetGonisdziebafaqtoriDistinct]
AS
BEGIN
--საბოლოო რეპორტი
    select RonisdziebaId,GonisdziebaName, sum(SruliShefaseba)
    as SruliShefaseba from dbo.RonisdziebaFaqtori
        group by RonisdziebaId,GonisdziebaName order by
    sum(SruliShefaseba) desc
END

```

### **ლისტინგი 6:**

```
--  
ALTER PROCEDURE [dbo].[SpGetGonisdziebaRelation]  
@relationID int  
AS  
BEGIN  
--დეტალების წამოღება  
    select * from dbo.VW_RonisdziebaRelation where  
RelationId=@relationID  
END
```

### **ლისტინგი 7:**

```
-- SpGetRelation  
ALTER PROCEDURE [dbo].[SpGetRelation]  
AS  
BEGIN  
--პედერის(მთავარის) წამოღება  
    select * from dbo.VW_Relation  
END
```