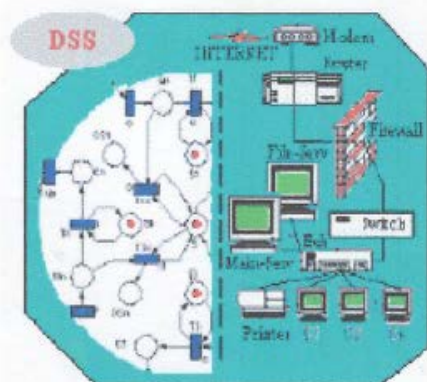


არჩილ ფრანგიშვილი, ბიკ სურგულაძე,
ირინა ვაჭარაძე

ბიზნეს-პროგრამების ექსპერტულ შეფასებებში გადაწყვეტილებათა მიღების მხარდაჭერი მეთოდები და მოდელები



„ტექნიკური უნივერსიტეტი“

საქართველოს ტექნიკური უნივერსიტეტი

არჩილ ფრანგიშვილი, გია სურგულაძე,
ირინა ვაჭარაძე

**ბიზნეს-პროგრამების ექსპერტულ
შეფასებებში გადაწყვეტილებათა
მიღების მხარდაჭერი მეთოდები
და მოდელები**



რეგისტრირებულია
სტუ-ს სარედაქციო-
საგამომცემლო
საბჭოს მიერ

თბილისი – 2009

უაკ 681.3.06

გადმოცემულია კორპორაციული მართვის სისტემებში ბიზნეს-პროგრამების შემუშავების პროცესების ავტომატიზაცია ექსპერტულ შეფასებათა მეთოდებისა და უნიფიცირებული, ობიექტ-ორიენტირებული მოდელირების, ანალიზის, დაპროექტების და პროგრამული რეალიზაციის საფუძველზე. შემოთავაზებულია ორგანიზაციული მართვის საინფორმაციო სისტემებში კოლექტიური მოხმარების რესურსების გამოყენების ეფექტიანობის ამაღლების ამოცანის გადაწყვეტა. UML-ტექნოლოგიის, კატეგორიული მიდგომის, ობიექტ-როლური მოდელირებისა და პეტრის ქსელების ინსტრუმენტული საშუალებით განსორციელებულია ექსპერიმენტული შედეგების კვლევა. მონოგრაფია განკუთვნილია მართვის საინფორმაციო სისტემების სპეციალისტების, მაგისტრანტების, დოქტორანტებისა და სტუდენტებისთვის, რომლებიც მოღვაწეობენ ინფორმაციული ტექნოლოგიების მენეჯმენტის სფეროში.

რეცენზენტები:

ტ.მ.დ., პროფ. ზ. გასიტაშვილი

ტ.მ.კ., ასოც.პროფ. ე. თურქია

ტ.მ.დ., პროფ. გ. სურგულაძის რედაქციით

© საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, 2009

ISBN 978-9941-14-450-9

<http://www.gtu.ge/publishinghouse/>



ყველა უფლება დაცულია. ამ წიგნის არც ერთი ნაწილი (იქნება ეს ტექსტი, ფოტო, ილუსტრაცია თუ სხვა) არანაირი ფორმით და საშუალებით (იქნება ეს ელექტრონული თუ მექანიკური), არ შეიძლება გამოყენებულ იქნას გამომცემლის წერილობითი ნებართვის გარეშე.

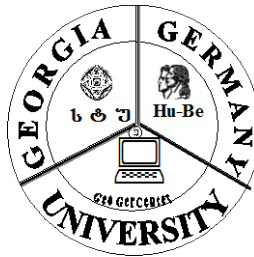
საავტორო უფლებების დარღვევა ისჯება კანონით.

Georgian Technical University

**ARCHIL PRANGISHVILI, GIA SURGULADZE,
IRINA VACHARADZE**

**METHODS AND MODELS OF
SUPPORT OF DECISION-MAKING IN
EXPERT ESTIMATIONS FOR
BUSINESS PROGRAMS**

Supported by DAAD
(Germany)



© Publication House "Technical University", Tbilisi, 2009
ISBN 978-9941-14-450-9

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.

წინასიტყვაობა

მონოგრაფიაში განიხილება ორგანიზაციული მართვის სისტემების 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, რელაციური 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-ინსტრუმენტის გამოყენების საკითხი. შემოთავაზებულია განაწილებული მონაცემთა ბაზებიდან შერჩეული მონაცემების ერთიან გადაწყვეტილების მიღების მხარდამჭერ სისტემაში სტრუქტურული 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, კატეგორიული 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 მოდელირება განხორციელებულია უნიფიცირებული მოდელირების ენის (UML) სტანდარტებით და პეტრის ქსელის 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-დამოკიდებულების მოდელის (ERM) და მონაცემთა აღწერის ენის (DDL) ურთიერთთავსებადობის შესახებ. შემოთავაზებულია რეალიზაციის შესაბამისი ალგორითმული სქემები MsSQL Server და ADO.NET გარემოში სამუშაოდ. ორგანიზაციული მართვის სისტემებში ექსპერტულ შეფასებათა გადაწყვეტილების მიღების მხარდამჭერი Windows-პროგრამების გარდა შემოთავაზებულია Web-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.

პროფ. ა. ფრანგიშვილი

შინაარსი

წინასიტყვაობა	6
შესავალი	11
1 თავი. კორპორაციითა ბიზნეს-პროგრამების დაგეგმვის პრობლემები და მათი სრულყოფის გზები საინჟინერო ტექნოლოგიებით	14
1.1. კორპორაციული ობიექტების მართვის მექანიზმების სრულყოფის მიზნები და ამოცანები	14
1.2. საკვლევი ობიექტის დიაგნოსტიკური ანალიზის ოთხეტაპიანი მოდელი	18
1.3. მიზნის მისაღწევად შესასრულებელი ამოცანები	23
1.4. გადაწყვეტილების მიღების ხელშემწყობი სისტემები	26
1.5. მონაცემთა საცავი ოპერატიული ანალიზის ინსტრუმენტით	27
1.6. საბანკო სისტემების კომუნიკაციები და საინფორმაციო ტექნოლოგიები	35
1.7. მდგომარეობათა და პროცესების მოდელირება პეტრის ქსელებით	41
1.8. დაპროგრამების UML2 ტექნოლოგია	53
1.9. .NET პლატფორმა და პროგრამული ინსტრუმენტები	57
1.10. პირველი თავის დასკვნები	66
2 თავი. ბიზნეს-პროგრამების მართვის ხელშეწყობის კომპიუტერული სისტემის აგების მეთოდებისა და მოდელის შემუშავება	67
2.1. ბიზნეს-პროგრამების ექსპერტულ შეფასებათა გადაწყვეტილების მიღების პროცესების მოდელირება UML-ის სტატიკური და დინამიკური დიაგრამებით	67
2.2. მონაცემთა ბაზაში ექსპერტთა ცოდნის ასახვის ობიექტ-როლური მოდელის შემუშავება კატეგორიული მიდგომით	79
2.3. ობიექტ-როლური მოდელის დაპროექტება და აგება ავტომატიზებულ რეჟიმში	88

2.4 კონცეპტუალური მოდელებისა და კლასთა ურთიერთკავშირების სქემის დაპროექტება ობიექტროლური მოდელების ბაზაზე	92
2.5. განაწილებული ბიზნეს-პროცესების ობიექტორიენტირებული მოდელირება და ანალიზი პეტრის ქსელებით	95
2.6. მეორე თავის დასკვნები	106
3 თავი. ინფორმაციული უზრუნველყოფის დამუშავება	
MySQL Server პაკეტი	107
3.1. განაწილებული მონაცემთა ბაზების აგების ავტომატიზაცია NET გარემოში	107
3.2. მონაცემთა საცავის დაპროექტება გადაწყვეტილებათა მიღების მხარდამჭერი სისტემისათვის	113
3.3. Ms SQL Server სისტემის არქიტექტურა და ობიექტები	117
3.3.1. ცხრილები	121
3.3.2. ატრიბუტები (სვეტები)	122
3.3.3. ინდექსები	123
3.3.4. წარმოდგენები	125
3.3.5. მთლიანობის შეზღუდვები	126
3.3.6. შენახვადი პროცედურები	127
3.4. მოთხოვნების დამუშავების ინსტრუმენტი Query Analyzer	128
3.5. მესამე თავის დასკვნები	130
4 თავი. პროგრამული უზრუნველყოფის დამუშავება	131
4.1. გადაწყვეტილების მიღების მხარდამჭერ სისტემაში მრავალფაქტორული ამოცანის გადაწყვეტის პროგრამული რეალიზაცია	131
4.2. ავტომატიზებული საბანკო სისტემის პროცესები და ინფორმაციული ნაკადების დიაგრამები	137
4.3. კორპორაციული მართვის სისტემის Web-აპლიკაციის დამუშავება Internet-Intranet გარემოში .NET-პლატფორმაზე	142
4.4. ავტომატიზებული ანალიზის სისტემის დაპროექტება სატენდერო კომისიის ექსპერტებისთვის.	147
4.5. მეოთხე თავის დასკვნები	154

5	თავი. მქსამრტულ შმფასნებატა ავტომატიზებული ღამუშაჲმბის სისტემის საღმონსტრაციო ვერსია	155
5.1.	სამუშაო გარემო: Ms Visual Studio .NET პლატფორმა	177
5.2.	მონაცემთა ბაზის ცხრილები	181
5.3.	Expert.dbo მონაცემთა ბაზის View წარმოდგენები SQL-ტექსტებით	187
5.4.	Expert.dbo მონაცემთა ბაზის Stored Procedures	188
5.5.	შენახვადი პროცედურების ლისტინგები	190
5.6.	მეხუთე თავის დასკვნები	191
	დასკვნა	193
	ლიტერატურა	199
	დანართი	

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 ოპერატიული ინფორმაციის არსებობისას. ამ ფაქტორმა გამოიწვია ორგანიზაციული სისტემების 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-სპეციალისტებისაგან სწორი ეკონომიკური პოლიტიკის გატარებას გააზრებული ჯგუფური გადაწყვეტილების მიღების საფუძველზე [1].

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

აღნიშნული მიზნის მისაღწევად მონოგრაფიაში 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 კომერციული 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 საბაზრო 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 კატეგორიულური მიდგომით, აგრეთვე ამ პროცესების ავტომატიზებულ რეჟიმში გადაწყვეტის პროცედურები ვიზუალურ-კომპონენტური დაპროგრამების საფუძველზე; წარმოდგენილია განაწილებული ბიზნეს-პროცესების ობიექტ-ორიენტირებული მოდელებისა 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.

მეოთხე თავი ეხება ბიზნეს-პროგრამების ექსპერტულ შეფასებათა მხარდამჭერი კომპიუტერული სისტემის პროგრამული უზრუნველყოფების დამუშავებას და რეალიზაციას.

მეხუთე თავში მოცემულია ექსპერიმენტული შედეგები, რომლებიც განხორციელდა პროდუქციის საწარმოო ფირმის და კომერციულ-საფინანსო ბანკის ობიექტებზე. შემოთავაზებულია მომხმარებელთა ინტერფეისების სადემონსტრაციო პაკეტი Web-აპლიკაციის სახით, კომერციული ბანკის სერვისის ამოცანების გადასაწყვეტად.

I ტაში

კორპორაციატა ბიზნეს-პროგრამების დაგეგმვის პრობლემები და მათი სრულყოფის ამოცანები საინფორმაციო ტექნოლოგიების ბაზაზე

1.1. კორპორაციული ობიექტების მართვის მექანიზმების სრულყოფის მიზნები და ამოცანები

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

კორპორაციული ობიექტების მართვის ავტომატიზებული სისტემა მოიცავს დიავნოსტიკური ანალიზის, ექსპერტული შეფასებების, ბიზნეს-პროგრამების 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 ექცევა საზღვარგარეთ [1].

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

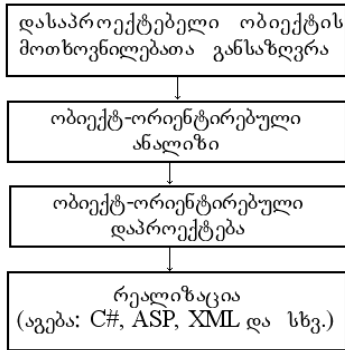
ამგვარად, დისერტაციაში დასმული ამოცანების გადაწყვეტის მიზანი ფირმის სასიცოცხლო ციკლთან – არსებობის საკითხთანაა დაკავშირებული და ამგვარად მისი აქტუალურობა და მნიშვნელობა ეჭვს არ იწვევს.

კორპორაციული ობიექტების მართვის პროცესისთვის ჩვენ უნდა გამოვიყენოთ 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 სამეურნეო-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.

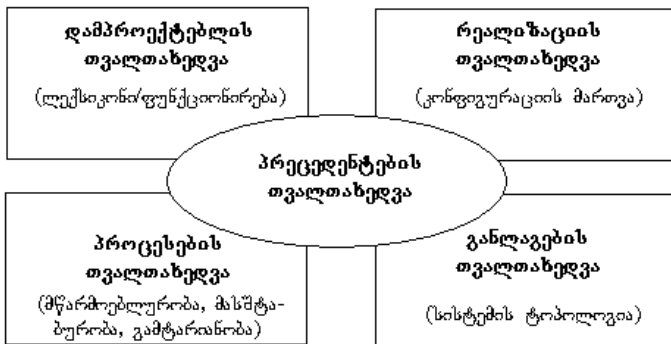
UML ტექნოლოგია ეყრდნობა ობიექტ-ორიენტირებული მოდელირებას და რამდენიმე ეტაპისაგან შედგება:



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 ყოფაქცევა სხვებთან მიმართებაში და ა.შ. პროგრამული სისტემის არქიტექტურა შეიძლება კარგად აისახოს ხუთი ურთიერთდაკავშირებული ელემენტით (ნახ.1.2). თითოეული თვალთახედვა დამოუკიდებელია, 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, მულმივი 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.

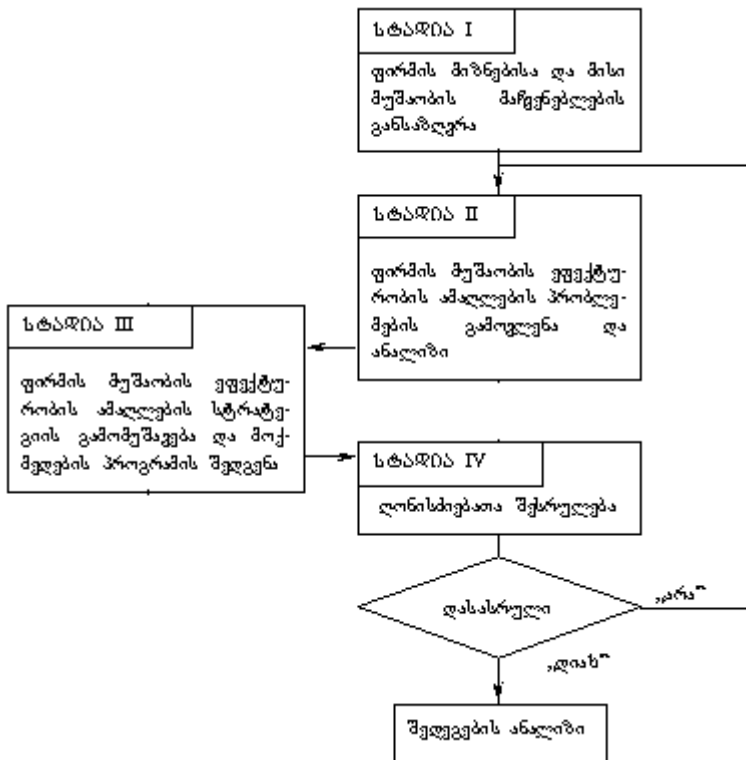
აქედან გამომდინარე, სისტემის ძირითადი მოთხოვნები შეიძლება ასე ჩამოვაყალიბოთ: - შეიქმნას კონსულტანტის ავტომატიზებული სამუშაო ადგილი (სერვერზე, ან ცალკე კლიენტის სახით); - აიგოს მონაცემთა განაწილებული ბაზა ფირმის ხელმძღვანელებისა და სპეციალისტების ცოდნის ამსახველი ინფორმაციით (კლიენტების სახით); - შეიქმნას ალგორითმი და პროგრამული პაკეტი, რომლებიც შეაჯერებს ექსპერტულ შეფასებათა შედეგებს და გამოიმუშავებს კოლექტიურ გადაწყვეტილებებს; - მოხდება ამ გადაწყვეტილებათა ცნობრებაში 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 მეთოდოლოგია, თუ

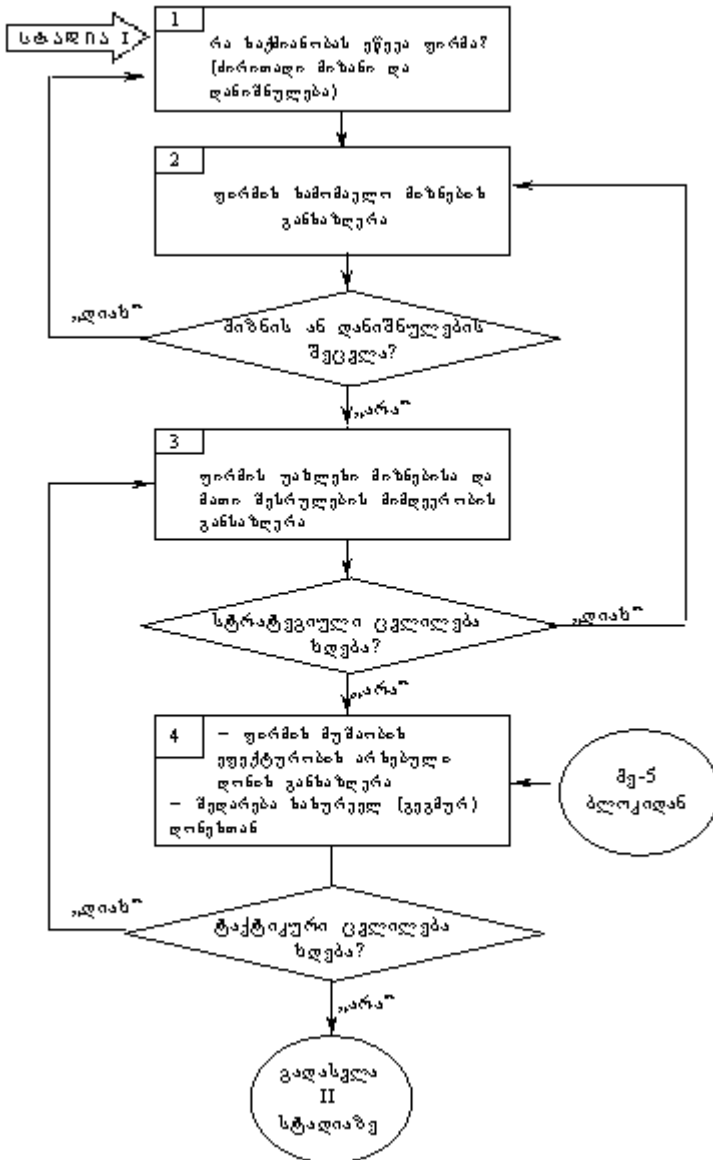
როგორ წარმართოს მუშაობა ექსპერტებთან. ეს მეთოდთა გულისხმობს ობიექტის კვლევისათვის სისტემური მიდგომისა და ექსპერტულ შეფასებათა მეთოდების გამოყენებას. ჩვენ მათ შემდგომში განვიხილავთ.

1.2. საკვლევი ობიექტის დიაგნოსტიკური ანალიზის ოთხეტაპიანი მოდელი

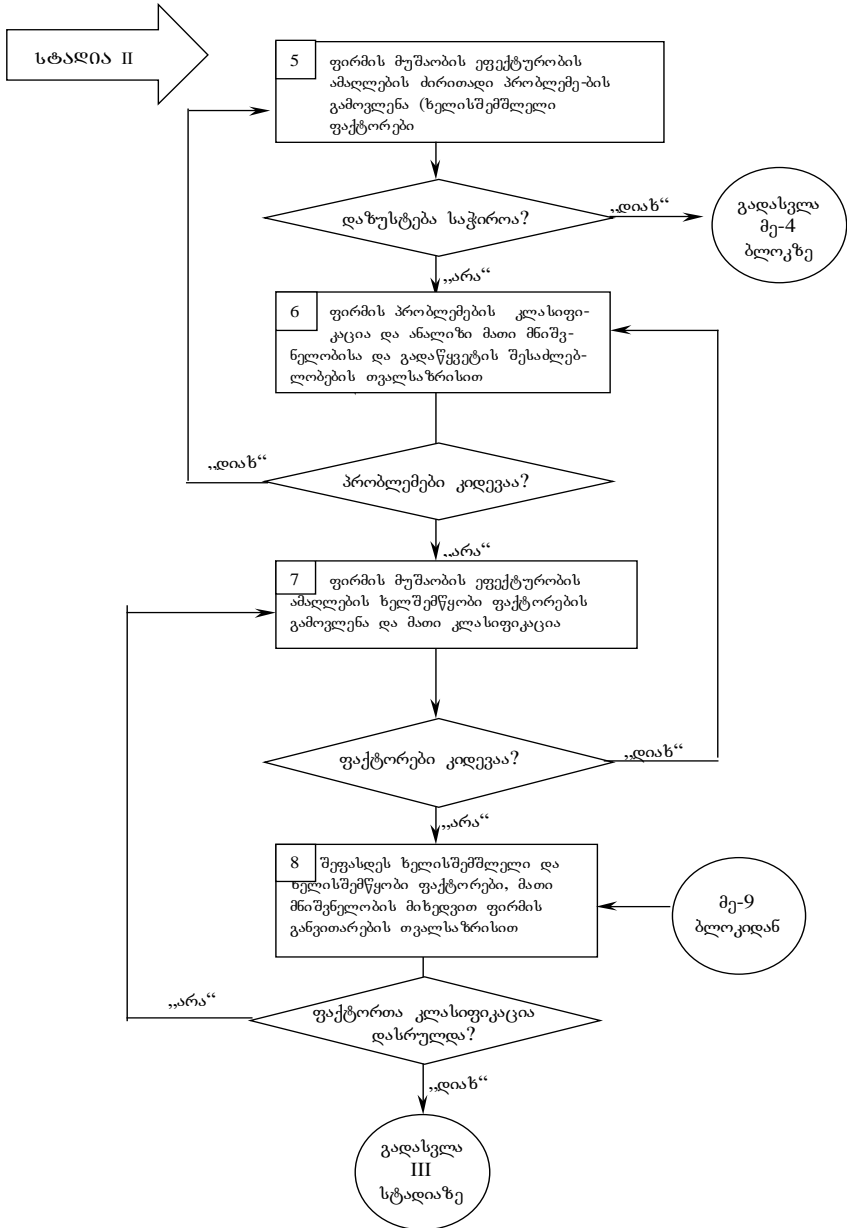
ამერიკელი მეცნიერები ეკონომიკური დაგეგმვისა და ფირმების მუშაობის კონსულტირების სფეროში რობერტ აბრამსონი და უოლტი ჰალსეტი გთავაზობენ ოთხეტაპიან მოდელს ორგანიზაციის გამოსაკვლევად და სტრატეგიული ღონისძიებების შესადგენად (ნახ. 1.3÷1.7) [1].



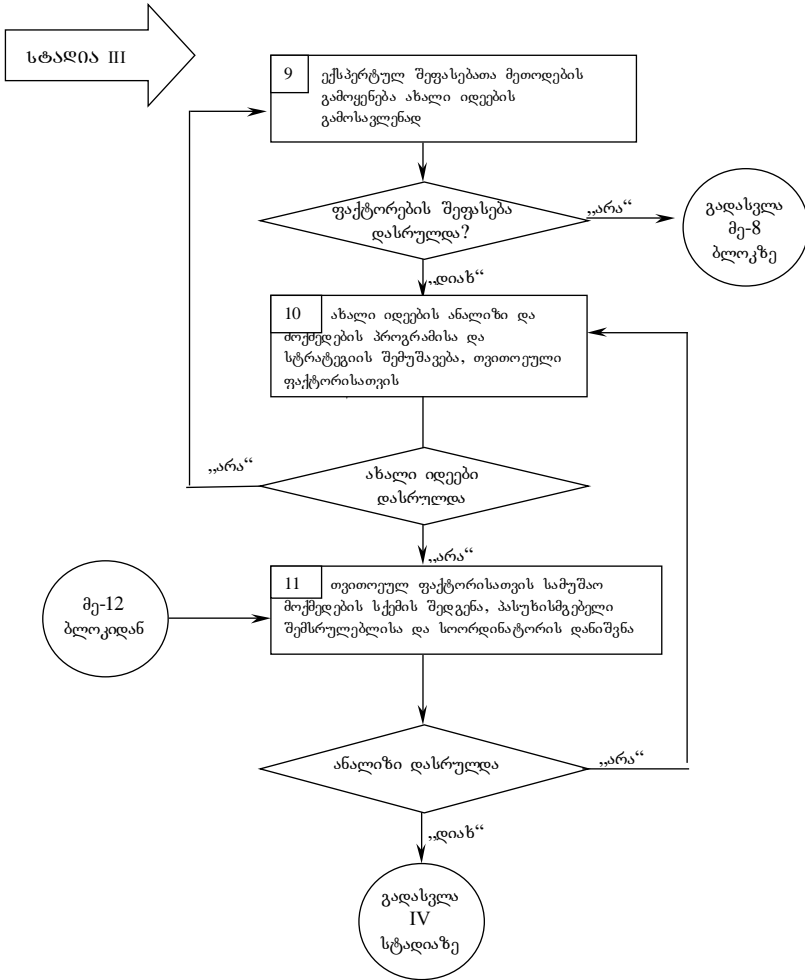
ნახ.1.3 სტრატეგიული დაგეგმვის ოთხეტაპიანი სქემა



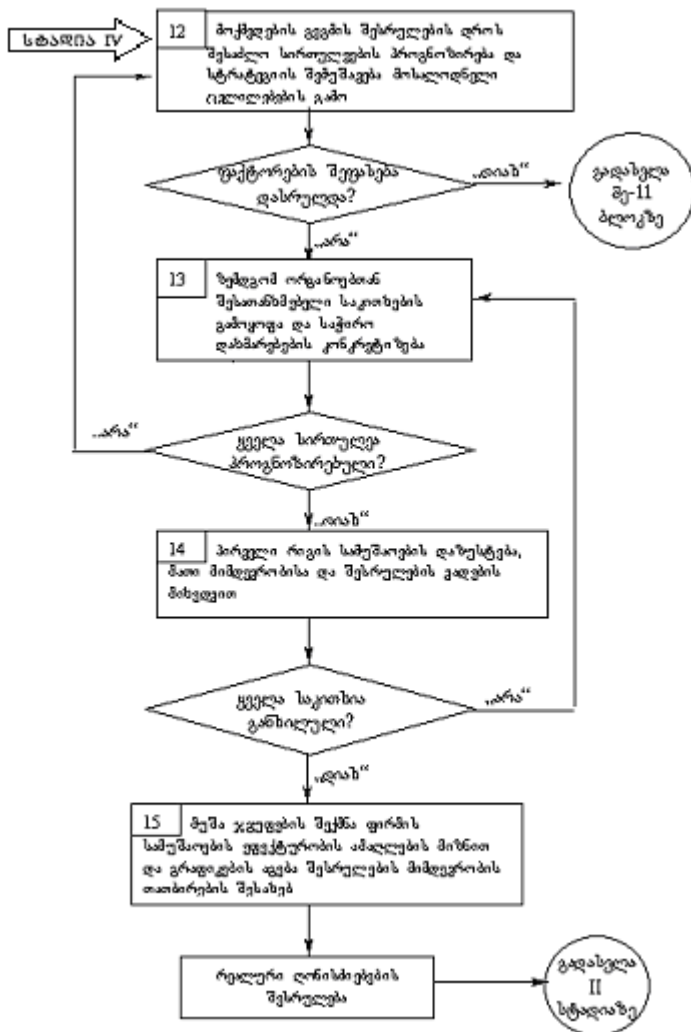
ნახ. 1.4. პირველი სტადიის ეტაპები



ნახ.1.5. 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 კონსულტანტის 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 – ოპერატიული 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 ორგანიზაციის განვითარების სტრატეგიული გეგმაა, 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.

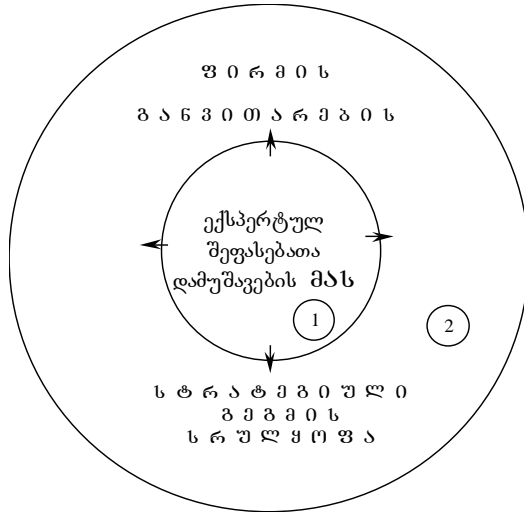
მომავალში ასეთი ინფორმაციული ბაზები შეიძლება გამოყენებულ იქნეს დაგეგმვის განყოფილებაში ე.წ. ოპერატიული (1 წლიანი) გეგმების შესადგენად. ექსპერტთა შეფასებების დამუშავების ავტომატიზებული სისტემა შეიძლება გამოყენებულ იქნეს არა მხოლოდ ფირმის (ზედა) დონეზე, არამედ ფირმის ქვეგანყოფილებებსა და საამქროებში, თუ 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 შევვხებით.

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). ამავარად, 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 მიდგომით;
- 2) ექსპერტულ შეფასებათა მეთოდების ანალიზი და მათი შერჩევა დასმული მიზნის მისაღწევად;
- 3) კონსულტანტის მეთოდის ფორმალიზება და ფუნქციონალური ბლოკ-სქემის აგება;
- 4) ინფორმაციული უზრუნველყოფის საკითხების დამუშავება;
- 5) დიალოგური პროცედურების აგება კონსულტანტისა და ექსპერტებისათვის;
- 6) პროგრამული პაკეტის დამუშავება და ტესტირება;
- 7) მომხმარებლის ინსტრუქციების ჩამოყალიბება.

1.4. გადაწყვეტილების მიღების ხელშემწყობი სისტემები

როგორც აღვნიშნეთ, დიდი კორპორაციების ეფექტურად 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 სისწრაფე, მიმდინარე და ისტორიული მონაცემების ანალიზის შესაძლებლობა. ასეთ შესაძლებლობათა მქონე სისტემებს უწოდებენ გადაწყვეტილების მიღების მხარდამჭერ სისტემებს (გმმს) [5]. ისინი ფართოდ გამოიყენება სხვადასხვა დარგებში, როგორცაა, 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 კატეგორიული ანალიზისა და კონკრეტული მოდელირების შემეცნებითი 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 გმმს-ს აქვს მონაცემებთან მიმართვის სერვისის სხვადასხვა 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.

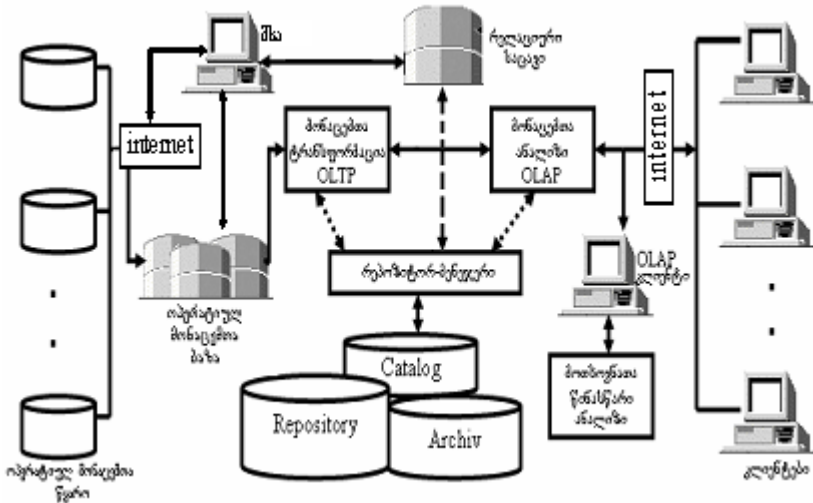
კომერციულ ბანკებში ჩატარებულმა ანალიზმა აჩვენა, რომ საბაზო ტრანზაქციების და სასურველ მონაცემებზე არსებული ინფორმაციის მოცულობა ძალიან დიდია. პირველადი ანალიზისათვის უნდა მომზადდეს გაფართოებული მონაცემები და მიეცეს ინდექსაცია. ამისათვის საჭიროა ფართო კომპიუტერული რესურსები, რომელიც საშუალებას იძლევა მცირე დროის განმავლობაში შესრულდეს ძებნა რამდენიმე ცხრილიში, რომელიც ათობით მილიონ 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 ჰიპერლინკური 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.1.9. მონაცემთა საცავის 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 სტანდარტად, ამიტომაც ინფორმაციული საცავებისათვის მიზანშეწონილია რელაციური პლატფორმა.

მონაცემთა საცავი ორიენტირებულია განსაზღვრულ საგნობრივ სფეროზე და ორგანიზებულია მონაცემთა ოპერატიული ბაზიდან შემოსულ სტრუქტურულად გადამუშავებულ მონაცემთა ქვესიმრავლეების 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 მენსიერების 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 გადასავსავენად. 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, როგორცაა 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

ორიენტირებისათვის. 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 დეფექტური ტრანზაქციის შედეგად მოხდეს მონაცემთა „დაზიანება“. ამ შემთხვევაში ამოქმედდება მონაცემთა დაცვის სისტემა, რაც უზრუნველყოფს დეფექტების აღმოფხვრას და არასასურველი ინფორმაციის განადგურებას.

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

ანალიზური დამუშავების OLAP–ტექნოლოგია: მას მონაცემთა მრავალგანზომილებიანი, კომპლექსური ანალიზის ტექნოლოგიას უწოდებენ. **OLAP (Online Analytical Processing)** ნიშნავს „მონაცემთა ოპერატიული ანალიზს“. იგი პირველად მონაცემებს წარმოადგენს ინფორმაციის სახით, რომლის დახმარებითაც შესაძლებელი ხდება საწარმოს მოცულობის შესახებ ვიქონიოთ რეალური წარმოდგენა. ამავე დროს იგი უნიკალური ინსტრუმენტია, რომელიც საშუალებას გვაძლევს სხვადასხვა ანალიზური ჭრილით ჩავატაროთ ინფორმაციის მრავალგანზომილებიანი ანალიზი.

მონაცემთა საცავების დაპროექტებისა და მისი ფუნქციონირებისათვის, მეთოდური თვალსაზრისით ჩვენ ვიყენებთ ეფ. კოდის მიერ ჩამოყალიბებულ პრინციპებს [10]. ესაა ის 12 წესი,

რომელსაც უნდა აკმაყოფილებდეს ნებისმიერი განაწილებული სისტემა მონაცემთა საცავით, რათა ჩატარდეს საინფორმაციო ბლოკების სრულფასოვანი ოპერატიული 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) აიოლებს მოდელირებას, ანალიზს და გამოთვლებს;

2. გამჭვირვალობა. 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 ემყარება 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 სისტემამ 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 შეუძინეველი და არ უნდა აისახებოდეს საწარმოო პროცესების წარმადობის შემცირებაზე;

5. კლიენტ სერვერის – არქიტექტურა. მონაცემთა დიდი ნაწილი, რომელიც მოითხოვს ოპერატიულ ანალიზურ გადაამუშავებას, 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 გამჭვირვალებას და საშუალებას იძლევა აიგოს საერთო კონცეპტუალური, ლოგიკური და ფიზიკური სქემები;

6. მრავალგანზომილება. გაზომვის ყოველი მცდელობის დროს გამოყენებული უნდა იყოს მიუკერძოებელი 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 ინტერფეისებს. 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 ელასტიური. 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, ოპერატიული ანალიზური დამუშავების თეორიულ ბაზისად. როგორც უკვე აღვნიშნეთ **OLAP-**ში ძვეს 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

კომერციულ 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 სთავაზობდა ბანკს ყველა სფეროს მომსახურებას, რომელიც საჭირო იყო რუტინული სამუშაოს შესამცირებლად და ბანკის ფინანსური მდგომარეობის 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 შევაყვასოთ რამდენიმე ფორმა, და ასევე საკმაოდ ინტელექტუალური კომპლექსები, რომლებიც წყვეტს ბანკის მართვის ამოცანებს.

მართვის ინფორმაციული სისტემა უნდა უზრუნველყოფდეს 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;
- ტელეკომუნიკაციების 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 მომხმარებლის კომპიუტერზე. ხოლო როდესაც ადგილი აქვს პროგრამის განახლებას ან რაიმე პროგრამულ ცვლილებას, საჭიროა ხდება ცვლილებების შეტანა თითოეულ კომპიუტერზე ცალ-ცალკე. რაც განსაკუთრებით შრომატევადია. გარდა ამისა ეს ძვირადღირებული პროცესია, რადგან ამ შემთხვევაში ორგანიზაციამ უნდა გასწიოს ხარჯი და შეინახოს პროგრამისტა 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 გვერდებზე მოთხოვნა გაიგზავნება. ვებ გვერდის ბაზასთან დაკავშირება რეალურად გადააქცევს მას მძლავრ ვებ-გვერდად.

ინტრანეტი: ორგანიზაციებს შეუძლია გაიყვანოს შიგა-ქსელი ე.წ. ინტრანეტი. ინტრანეტი იცავს ორგანიზაციას თავისი შიგა ფაილებისა და კონფიდენციალური ინფორმაციის წვდომისაგან გარე პირთათვის. ინტრანეტი ხშირად გამოიყენება ფაილების და მეილების ორგანიზაციის წევრთათვის ერთობლივი წვდომისათვის, და ამავე დროს, გარე პირთათვის იგივე ინფორმაციის ბლოკირებისათვის.

ვებ პროგრამა უფრო კარგად გამოიყენება ინტრანეტში ვიდრე ინტერნეტში. ეს ნიშნავს, რომ ის უფრო მისაწვდომია ინდივიდთა პატარა ჯგუფისათვის. ვებ პროგრამები ინტერნეტში გამოყენებისას შეიძლება შეიზღუდოს სპეციალური მომხმარებლებით, და ზოგ შემთხვევებში კი სპეციფიკური IP მისამართებით.

შიგა ვებ-პროგრამები, რომლებიც შეზღუდულია სპეციფიკური მომხმარებლისა თუ კომპიუტერებისთვის, დიდად გამოსაყენებელია ისეთ ფინანსურ ინსტიტუტებში, როგორცაა 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 ერთდროულად ინფორმაციულ სტრუქტურას, ATM ტექნოლოგია საკმარისად ეფექტურია ქსელის შექმნისათვის საჭირო ხარჯების ოპტიმიზაციის თვალსაზრისით. მისი მთავარი უპირატესობა მდგომარეობს იმაში, რომ

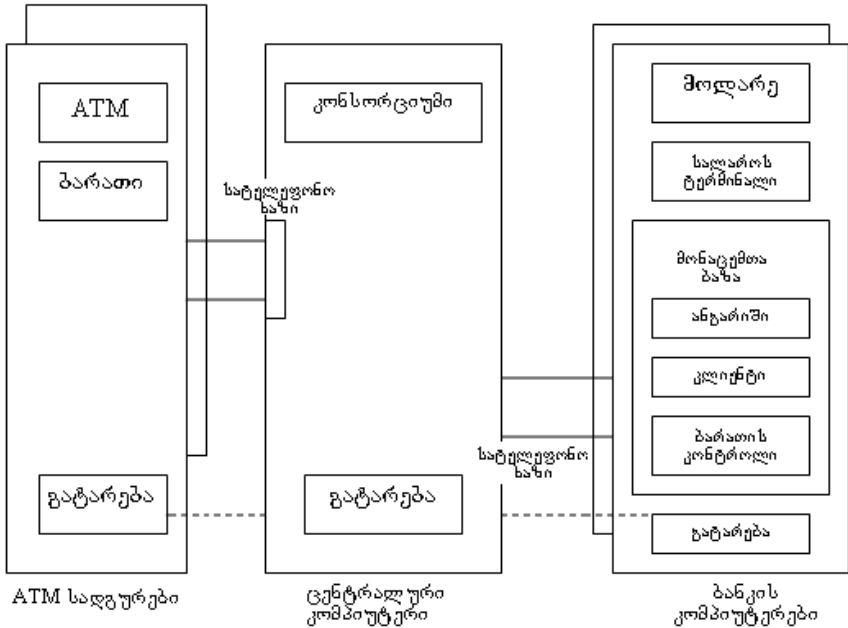
მას შეუძლია გამოიყენოს ერთიანი უნიფიცირებული აპარატურა ყველა ტიპის კომუნიკაციისათვის: ლოკალური და გლობალური ქსელი, სატელეფონო, ვიდეოგამოსახულების გადაცემისათვის განკუთვნილი ქსელი (ვიდეოკონფერენციები, უსაფრთხოების სისტემა).

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 მათთვის. 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 ჰიბრიდულ 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 მომსახურების ქვესისტემები.

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.



ნახ.1.10. საბანკო ქსელის მართვის სიტემის არქიტექტურა

ასინქრონული პარალელურობა წარმოიშობა რამდენიმე დამოუკიდებლად მომუშავე ATM-ის და სალაროს ტერმინალების პარალელური მომსახურების აუცილებლობით.

ყოველ ტერმინალს შეუძლია მოემსახუროს 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, როგორცაა მარკერთა მაქსიმალური რაოდენობა პოზიციაში, რკალების ჯერადობა და სხვა. ახალ კლასიფიკაციაში ყურადღება უშუალოდ მარკერთა სემანტიკაზე გამახვილებული.

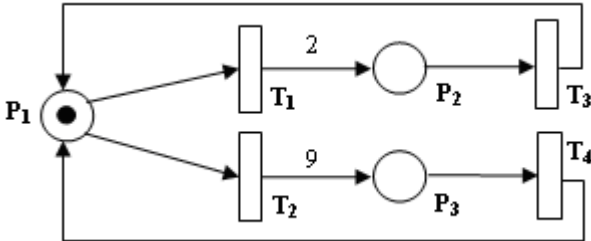
კერძოდ, დაბალი დონის პეტრის ქსელებში დაიშვება მხოლოდ „შავი“ მარკერები ყოველგვარი შინაგანი სტრუქტურის გარეშე, ხოლო მაღალი დონის პეტრის ქსელები დამატებით წინასწარ განსაზღვრული სტრუქტურის „ფერად“ მარკერებსაც შეიცავს, თუმცა უნდა აღინიშნოს, რომ ტერმინები „შავი“ და „ფერადი“ სიმბოლურია და ლიტერატურაში ისინი ხშირად განსხვავებული სახელებით მოიხსენიება [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, ამოტომატ აქ მათ მოკლედ დავასახელებთ.

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

S/T-ქსელებში პოზიცია შეიძლება ერთზე მეტ მარკერს შეიცავდეს, ხოლო მარკერების დიდი ოდენობის შემთხვევაში პოზიციაში მათი რაოდენობა რიცხობრივად ჩაიწერება. გადასასვლელის გაშვების აუცილებელი პირობაა ყველა შემავალ პოზიციაში დამაკავშირებელი რკალის ჯერადობაზე არანაკლები ოდენობის

მარკერების მოგროვება. რკალების ჯერადობა ნატურალური რიცხვით გამოისახება.



ნახ.1.11. S/T - პეტრის ქსელის ფრაგმენტი

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 Σ 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. Σ -ს მდგომარეობა a თითოეული პოზიციისთვის განსაზღვრავს სიმრავლეს $a(p) \subseteq A_p$;

2. ვთქვათ $t \in T_{\Sigma}$. ქმდება 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 მოთავესება შეიძლება, რკალსაც ელემენტარული პეტრის ქსელებისგან განსხვავებით ერთზე მეტი მნიშვნელობის გატარება შეუძლია, რაც მაღალი დონის პეტრის ქსელეს ახასიათებს. სისტემური პეტრის ქსელების შინაარსი სტრუქტურების ცნებაზეა დაფუძნებული. აღნიშნულ საკითხების გამოყენებას დეტალურად ჩვენ მომდევნო თავებში დავუბრუნდებით.

განსაკუთრებით საყურადღებოა პეტრის ქსელების გამოყენება პარალელური პროცესების მქონე რთულ ობიექტებში, რომლებშიც პროცესები მიმდინარეობს გარკვეულ მიზეზ–შედეგობრივი კავშირებით. როგორც ცნობილია სისტემების მოდელირებისა და ანალიზის ამოცანების გადასაწყვეტად ფართოდ გამოიყენება ისეთი მექანიზმები, როგორებიცაა მასობრივი მომსახურების და იმიტაციური მოდელირების თეორიები. ამასთანავე შეიძლება აღინიშნოს, რომ მათი გამოყენების ეფექტურობა, გამომსახველობითი და ანალიტიკური მხარეები მკვეთრად განსხვავებულია და დამოკიდებულია, როგორც თვით ინსტრუმენტის შესაძლებლობაზე, ასევე ობიექტის სირთულეზე (პარამეტრების რიცხვზე და სხვა).

სისტემური პეტრის ქსელების ერთ-ერთ კლასიკური მაგალითია ფერადი პეტრის ქსელი (Coloured Petri Net), რომელშიც კარგადაა შერწყმული პეტრის ქსელებისა და დაპროგრამების თეორია (იერარქიულობა, მოდულურობა – დიდი სისტემების მოდელირებისთვის), რაც მის დიდ პრაქტიკულ ღირებულებასაც განაპირობებს თანამედროვე ინფორმაციულ ტექნოლოგიათა გამოყენების მრავალ სფეროში, განსაკუთრებით ბიზნესისა და მარკეტინგის მენეჯმენტის ამოცანების გადასაწყვეტად [89,90].

უპირატესობა:

- შეიძლება წარმოდგენილი იქნას, როგორც გრაფიკული, ასევე ანალიტიკური ფორმით;

- უზრუნველყოფს ავტომატიზებული ანალიზის შესაძლებლობას;

- აქვს საკუთარი მოდელირების ენა (CPN_ML: www.smlnj.org), რომელზეც შესაძლებელია ახალი ფუნქციების შექმნა;

- იძლევა სისტემის აღწერის ერთი დეტალიზაციის დონიდან სხვაზე გადასვლის საშუალებას.

ნაკლი:

- ინსტრუმენტის ინტერფეისი რთულია და მოითხოვს მომხმარებლისგან დროს მასში გასარკვევად;

- CPN-ის ძირითად ბირთვს არ აქვს მოდელირებადი სისტემის დროითი მახასიათებლების აგების და გრაფიკული გაფორმების საშუალება, მაგრამ იგი ადვილად იყენებს არსებულ პაკეტებს (მაგალითად, ორ- და სამგანზომილებიან გრაფიკას [91].

ახლა განვიხილოთ საილუსტრაციო მაგალითი პროდუქციის წარმოების ინფორმაციულ-ტექნოლოგიური პროცესების მოდელირების მიზნით CPN-ის გამოყენებით [92].

წარმოდგენილია პროდუქციის წარმოების ინფორმაციულ-ტექნოლოგიური პროცესების კვლევა სისტემური ანალიზისა და საწარმოს მარკეტინგული უზრუნველყოფის თვალსაზრისით. განიხილება ინტეგრირებული მართვის ავტომატიზებული სისტემის აგების კონცეფცია UML სტანდარტებით და კლიენტ-სერვერ არქიტექტურით. შემოთავაზებულია პროდუქციის წარმოების პროცესის იმიტაციური მოდელის აგების მაგალითი. გამოკვლეულია ამ პროცესის უზრუნველყოფის რესურსების ოპტიმალური რაოდენობა, რომელიც იძლევა მაქსიმალურ მოგებას და განსაზღვრავს დამზადებული პროდუქციის მოცულობას [93].

პროდუქციის წარმოების ობიექტების საწარმოო ინფორმაციულ-ტექნოლოგიური პროცესების გამოკვლევის საფუძველზე, რომელიც ტარდება მათი სისტემური ანალიზისა და ექსპერტ-სპეციალისტთა შეფასებების გათვალისწინებით, დადგინდა, რომ ესაა დაგეგმვა-წარმოება-რეალიზაციის (ანუ მარკეტინგის) ტექნოლოგიური პროცესების ინფორმაციული ასახვის, გადაამუშავებისა და მართვის პროცესების ავტომატიზაციის მრავალდონიანი, კომპლექსური სისტემა. მასში რეალიზებულ უნდა იქნას ოპერატიული მართვის, ტექნოლოგიური, დაპროექტებისა და კონსტრუირების, დაგეგმვის, აღრიცხვის, კონტროლის და სხვ. პროცესების ავტომატიზაცია. ასეთი ინტეგრირებული მართვის ავტომატიზებული სისტემის აგება მომხმარებელთა ფუნქციური სამუშაო ადგილების კომპიუტერიზაციით, ანუ კლიენტ-სერვერ არქიტექტურითაა მიზანშეწონილი და მისი პროგრამული უზრუნველყოფა უნდა დაპროექტდეს თანამედროვე ობიექტ-ორიენტირებული სტანდარტების, ანუ UML-ტექნოლოგიის მიხედვით. ცალკეული ბიზნეს-პროცესების გამოსაკვლევად გამოიყენება იმიტაციური მოდელირების მეთოდი [92]. აქ განიხილება პროდუქციის საწარმოო ფირმის ბიზნეს-პროცესების წარმოების ზოგადი ინფორმაციულ-ტექნოლოგიური სქემა, ჩაწერილი უნიფიცირებული მოდელირების ენის აქტიურობათა დიაგრამის სახით, რომლის შინაარსი შემდეგში მდგომარეობს:

სამომხმარებლო, საერთო ბაზრის არსებობის პირობებში თითქმის უმართავი ხდება ნებისმიერი პროდუქციის წარმოებისა და რეალიზაციის ოპტიმალური გეგმების შედგენისა და მათი შესრულების კონტროლის პროცესები. შეუძლებელია როგორც ამ გეგმების, ასევე დაგეგმვის მექანიზმების ეფექტურობის შეფასება. შესაძლებელი უნდა იყოს წარმოების ტექნოლოგიური მომზადების პროცესის სწრაფი მართვა.

მარკეტინგული პროცესების მართვისას აუცილებელია განვიხილოთ წარმოებისა და საბაზრო სისტემების ურთიერთკავშირი, პროდუქციაზე მოთხოვნისა და სარეკლამო სამსახურის კუთხითაც.

რეკლამა ხელს უწყობს პროდუქციის კლიენტურის მაქსიმალურად დიდი აუდიტორიის შექმნასა და მიზიდვას. რეკლამის მიზანია საბაზრო გარემოსა და მომხმარებელთა მსყიდველობით ქცევაზე პირველადი ინფორმაციის შეგროვება სპეციალური პროცედურების გატარებით. სპეციალური პროცედურები ესაა რეკლამის ძირითადი სახეობები, მაგალითად, ინფორმაციული, ბეჭდვითი, აუდიო-ვიზუალური, ინტერნეტული, საგამოფენო (გამოფენა-გაყიდვები), სუვენირული, გარე, შიგა და ა.შ. სარეკლამო პროცესის დაგეგმვაში გასათვალისწინებელია რეკლამის სხვადასხვა სახეობებისა და საშუალებების ოპტიმალური კომბინაციის მოძებნა, რეკლამის ეფექტურობის შეფასება და რეკლამისთვის ბიუჯეტის განსაზღვრა. რეკლამის სახეობის და რეკლამის სახეობათა კომბინაციის შერჩევას გასათვალისწინებელია შემდეგ ფაქტორთა კვლევა: მომხმარებელთა აქტივობა, რეკლამის სახეობის ეფექტურობა, გამოყენების პერიოდი, დამზადების პერიოდი და ხარჯი რეკლამის თითოეული სახეობის დამზადებაზე. სარეკლამო ღონისძიებების გატარება, გარდა პროდუქციის მიზიდვისა და გაცნობისა მნიშვნელოვან ასახვას ჰპოვებს მარკეტინგული საქმიანობის წარმართვაზე, რაც შესაბამისად გავლენას ახდენს პროდუქციის საწარმოო პროცესის დაგეგმვისა და მართვის ასპექტებზე. პრაქტიკულად, რეკლამა საშუალებას იძლევა ჩატარდეს წინასწარი კვლევა მომხმარებელთა პროდუქციის სავარაუდო მსყიდველობით რეაქციაზე.

ამ თვალსაზრისით, სარეკლამო ღონისძიებების ერთ-ერთი ეფექტური სახეობა გამოფენა-გაყიდვა, რაც საკმაოდ სწრაფი

საშუალებაა საბაზრო სიტუაციის შესასწავლად და ახალ საექსპორტო ბაზარზე გასასვლელად პროდუქციის მხოლოდ მინიმალური რაოდენობის ან პროდუქციის მინი-სახეობების დამზადების ხარჯზე. რეკლამის ყველა სხვა სახეობებისგან განსხვავებით გამოფენები შეიძლება მივაკუთვნოდ მარკეტინგული პროცესების საერთო კომპლექსური ღონისძიებების გატარების ნაწილს, სადაც შესაძლებელია ერთდროულად გადაწყდეს მარკეტინგული კვლევის საკმარის ფართო დიაპაზონი: მაგალითად, საერთო მარკეტინგული ინფორმაციის შეგროვება (პროდუქციის სხვადასხვა მახასიათებლების კვლევისთვის და გაუმჯობესებისთვის - ფასი, ხარისხი და ა. შ.), გარიგებებისა და გაყიდვების ორგანიზება, პოტენციალური მომხმარებლების წრის გამოვლენა, საქმიანი ურთიერთობების გამყარება, გაყიდვების მოცულობის გაზრდა, კონკურენტუნარიანი საწარმოების კონტროლი და დაკვირვება, დარგის მდგომარეობის შესწავლა, გამოცდილების გაცვლა, თანამშრომლობის განვითარება, საქონელზე და ფასებზე რეაქციის შემოწმება, გასაღების არხებისა და ქსელის განვითარება და ა.შ.

ყველაზე მნიშვნელოვანი და მაქსიმალური შედეგი, რაც შეიძლება მიიღოს კომპანიამ გამოფენა-გაყიდვებიდან ეს არის კონტრაქტები. ამ შემთხვევაში კონტრაქტები შეიძლება ჩაითვალოს ორგანიზაციისთვის პირველადი, რეალური გასაღების არხების წყაროდ და შესაბამისად წარმოების დაგეგმვის საფუძვლად.

კონტრაქტების არსებობის შემთხვევაში, მარკეტინგული პროცესების წარმოების მრავალმხრივი და რთული სისტემა, დეკომპოზიციის მხრივ შესაძლებელია დავყოთ ხუთ ძირითად განზოგადებულ ეტაპად – ბაზარი (მიზნობრივი კლიენტურის მოთხოვნილებათა განსაზღვრა კონტრაქტების მიხედვით), წარმოების დაგეგმვა (საწარმოო სიმძლავრეების გათვალისწინება ორიენტირებული რეალურ დაკვეთაზე), წარმოება, გაგზავნა და მიწოდება.

ამ შემთხვევაში მარკეტინგული კვლევით კომპანიამ უნდა მიიღოს რეკომენდაციები, დროის რა პერიოდში არის შესაძლებელი მიღებული კონტრაქტების შესრულება, რა მატერიალური ხარჯით. მარკეტინგული სამსახურის მიზანია მიღებული კონტრაქტების საფუძველზე, საწარმოო სიმძლავრეების გათვალისწინებით ორგანიზაციის მატერიალური და არამატერიალური რესურსების

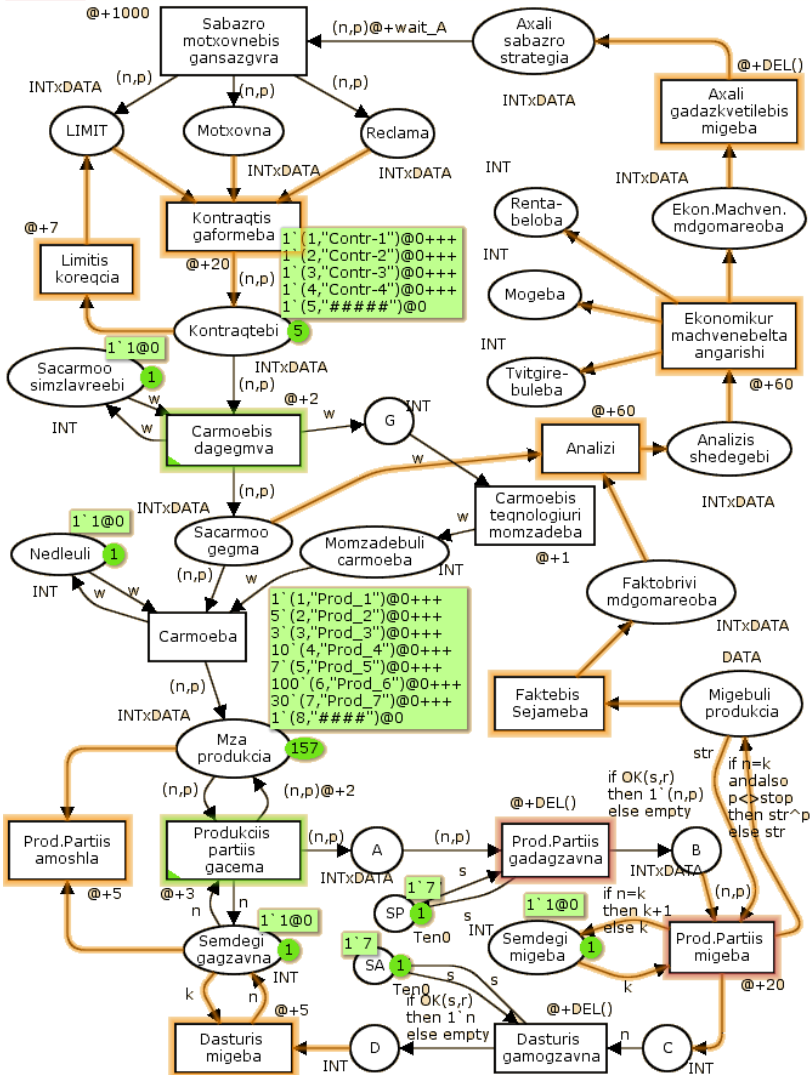
ოპტიმალურად გამოყენების (ხარჯების შემცირების, პროდუქციის თვითღირებულების ფასის შესაძლო დაწვევა, ახალი პოტენციალების გამოყენება და ა.შ.), მიწოდების ეფექტური გზების მოძებნის (ადგილის, დროის და მიწოდების ხერხების არჩევა) აპექტების კვლევა დროითი ეფექტურობის მიმართებაში.

ამგვარი მოდელის მარკეტინგული კვლევის ისტრუმენტად ეფექტურია პეტრის-ქსელის გამოყენება, რაც შესაძლებლობას იძლევა პროცესების მიზეზ-შედეგობრივი და მიმდევრობით-პარალელური შესრულების საფუძველზე გაანალიზებულ იქნას სისტემის მახასიათებლები.

1.12 ნახაზზე მოცემულია ფერადი პეტრის ქსელის ფრაგმენტი საწარმოო ფორმის მარკეტინგული პროცესების გამსხვილებული მოდელირებით [93]. აქ გადასასვლელების ბლოკებში ნაჩვენებია, მაგალითად, საბაზრო მოთხოვნილებათა განსაზღვრის, კონტრაქტების ფორმირების, წარმოების დაგეგმვის, წარმოების ტექნოლოგიური მომზადების, პროდუქციის წარმოების, დამკვეთებზე გადაცემა-გაგზავნის, შედეგების ანალიზის და ახალი გადაწყვეტილებების მიღების პროცესები.

თითოეული მათგანი უნდა გაიშალოს დამოუკიდებელი პეტრის ქსელით და მოხდეს მათი ანალიზი, ამასთანავე შეიქმნება ერთიანი იერარქიული სისტემა ჩადგმული პეტრის ქვექსელებით. დანიელი მეცნიერ-ინჟინრების მიერ შექმნილი ფერადი პეტრის ქსელების გრაფო-ანალიზური CPN-ინსტრუმენტი იყენებს ობიექტ-ორიენტირებული, ვიზუალური დაპროგრამების პრინციპებს, მისი ენა CPN საშუალებას იძლევა აღიწეროს ქსელის ფერადი კომპონენტები (მარკერები), ცვლადები, კონსტანტები და თვით პოზიციების, გადასასვლელებისა და რკალების ტექსტური აღწერები, რაც ერთგვარ კომფორტს ქმნის ქსელის წასაკითხად და გასაგებად.

1.13 ნახაზზე ნაჩვენებია ამ ინსტრუმენტის ფუნქციების, დახმარების, ოფციებისა და აღწერის (Declarations) ჩვენი ქსელის მაგალითზე. მოვიყვანთ ზოგიერთ განმარტებებს CPN-ის წასაკითხად. ქსელის ყოველ პოზიციას გააჩნია მინიმუმ ორი ჭდე: სახელი, რომელიც აღმნიშვნელი წრის ან ელიფსის შიგნით იწერება და მარტივი ან შედგენილი ტიპი (პოზიციის გვერდით, კურსივით, საკვანძო სიტყვა type, color ან string).



ნახ.1.12. მარკეტინგული პროცესის ფრაგმენტი CPN-ის გარეშე

მაგალითად, პოზიცია „კონტრაქტები“ INTxDATA ტიპისაა, რომელიც წინასწარ განსაზღვრული INT და DATA ტიპების დეკარტული ნამრავლით წარმოიქმნება. ფერადი პეტრის ქსელი შეიცავს „ფერად“ მარკერებს, რომლებიც კონკრეტული ტიპის შესაძლო მნიშვნელობათა სიმრავლე ან მულტისიმრავლეა.

```

CPN Tools (Version 1.4.0 - May 2005)
▼ Tool box
  ▶ Auxiliary
  ▶ Create
  ▶ Hierarchy
  ▶ Net
  ▶ Simulation
  ▶ State space
  ▶ Style
  ▶ View
▶ Help
▶ Options
▼ Marketing-4.cpn
  Step: 0
  Time: 0
  ▶ History
  ▼ Declarations
    ▼ colset INT = int timed;
    ▼ colset DATA = string;
    ▼ colset INTxDATA = product INT*DATA timed;
    ▼ var n, k, w, d, wait: INT;
    ▼ var p, str: DATA;
    ▼ val wait_A=8760;
    ▼ val stop = "#####";
    ▼ colset Ten0 = int with 0..12;
    ▼ colset Ten1 = int with 1..12;
    ▼ var s: Ten0;
    ▼ var r: Ten1;
    ▼ fun OK(s:Ten0, r:Ten1) = (r<=s);
    ▼ colset NetDelay = int with 25..75;
    ▼ fun DEL() = NetDelay.ran();
  Marketing
  
```

ნახ.1.13. CPN-სამუშაო გარემო

„კონტრაქტი-2“), 1'(3, „კონტრაქტი-3“), 1'(4, „კონტრაქტი-4“), 1'(5, „#####“) }. აქ ბოლო, მე-5 ელემენტი შეესაბამება დასასრულის იდენტიფიკაციას - stop.

საყურადღებოა „1“-იანი ყოველი ელემენტის დასაწყისში (მას კოეფიციენტი ეწოდება), რომელიც მიუთითებს, რომ პოზიციაშია არაუმეტეს 1 ცალი მოცემული ფერის მონაცემი (ანუ არსებობს მხოლოდ ერთი კონტრაქტი ნომრით „კონტრაქტი-1“, რომლის ფერია

ნახაზზე ნაჩვენებია სისტემაში კონსტანტების (საკვანძო სიტყვა val), ცვლადების (var) და ფუნქციების (fun) აღწერა. სხვადასხვა ტიპის მონაცემთა შორის კავშირების ასახვისთვის გამოიყენება სიმრავლეთა და კომპლექტების თეორიის ელემენტები. გარდა მონაცემთა ტიპისა, ყოველი პოზიციის გვერდით შეიძლება აისახოს მოცემულ მომენტში შემავალი ფერადი მარკერები.

საინიციალიზაციო მარკირება ხაზგასმული ტექსტის სახით გამოითანება. მაგალითად, საწყის მდგომარეობაში პოზიცია „კონტრაქტები“ შეიცავს INTxDATA ტიპის ფერად მარკერთა 5-ელემენტიან სიმრავლეს

(საინიციალიზაციო მარკირება): {1'(1, „კონტრაქტი-1“), 1'(2,

- რიგითი ნომერია 1). ამ შემთხვევაში გვაქვს მონაცემთა ელემენტების სიმრავლე.

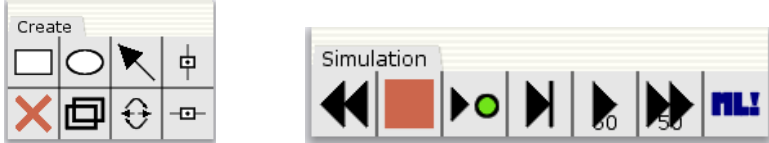
მეორე მაგალითი, პოზიცია „მზა პროდუქცია“ შედგება 157 ელემენტისგან ($1+5+3+10+7+100+30+1$), რომლებიც 7 სხვადასხვა (მარკერების ფერის) დამზადებული პროდუქტის რაოდენობას, ანუ მულტისიმრავლეს ასახავს.

პროცესების შესრულების დრო (დაყოვნება) აისახება გადასავლელთან სიმბოლოს და დროის ერთეულის (მაგალითად, @+7, @+wait) მითითებით, სადაც wait წინასწარ განსაზღვრული კონსტანტაა.

ამავე ნახაზზე ასახულია არადეტერმინირებული ლოგიკური გამოსახულება (პირობის ბლოკი) ფერადი პეტრის ქსელის რკალებზე, რომელიც გადასასვლელთა გაშვების სხვადასხვა პირობებს და შედეგებს ასახავს, ანუ ლოგიკური პირობის ჭეშმარიტებისას გადასასვლელს განსხვავებული მნიშვნელობა მიეწოდება (ან გადასასვლელიდან განსხვავებული მნიშვნელობა გამოვა), მცდარობისას – განსხვავებული. მაგალითად, გადასასვლელს „პროდუქციის პარტიის გადაგზავა“ გამოსასვლელ რკალზე აქვს ლოგიკური პირობა - თუ გამოგზავნილი პროდუქციის ნომერი (n) ემთხვევა კლიენტის კონტრაქტით მისაღებ პროდუქციის ნომერს (k), მაშინ გვაქვს „true“, წინააღმდეგ შემთხვევაში „false“, რაც იმას ნიშნავს, რომ საჭირო პროდუქცია არაა მოსული. თუ ყველაფერი წესრიგშია, მაშინ მიძღები უგზავნის მწარმოებელს შეტყობინებას გადასასვლელით „დასტურის გამოგზავნა“. პროდუქციის და შეტყობინების გადაცემათა ქსელში შემთხვევითი პროცესის არსებობა განპირობებულია დაყოვნების ცვლადი დროის გამო, რაც აისახება `colset NetDelay=int with 25..75, fun DEL() =NetDelay.ran() random`-ფუნქციით. ლოგიკური პირობის მნიშვნელობა სხვადასხვა შემთხვევებში სხვადასხვანაირად განისაზღვრება. ინტერაქტიულ სიმულატორებში ჭეშმარიტება-მცდარობას თავად მომხმარებელი განსაზღვრავს, ავტომატური სიმულაციისას – შემთხვევით სიდიდეთა გენერატორი.

1.14 ნახაზზე ნაჩვენებია CPN-გარემოში პეტრის ქსელის აგებისა და იმიტაციური მოდელირების ვიზუალური კომპონენტები. სიმულაციის მე-3 ლილაკი (მწვანე რგოლით) საშუალებას იძლევა იტერაციულად, ხელით ავამუშავოთ ჩვენთვის საჭირო გადასასვლელი

(აირჩევა რამდენიმე ალტერნატიულიდან). მე-ნ ლილაკი იძლევა საბოლოო მარკირების სურათს. 1-ელი ლილაკი – კი აღადგენს საწყის მარკირებას, ექსპერიმენტის თავიდან ჩასატარებლად.



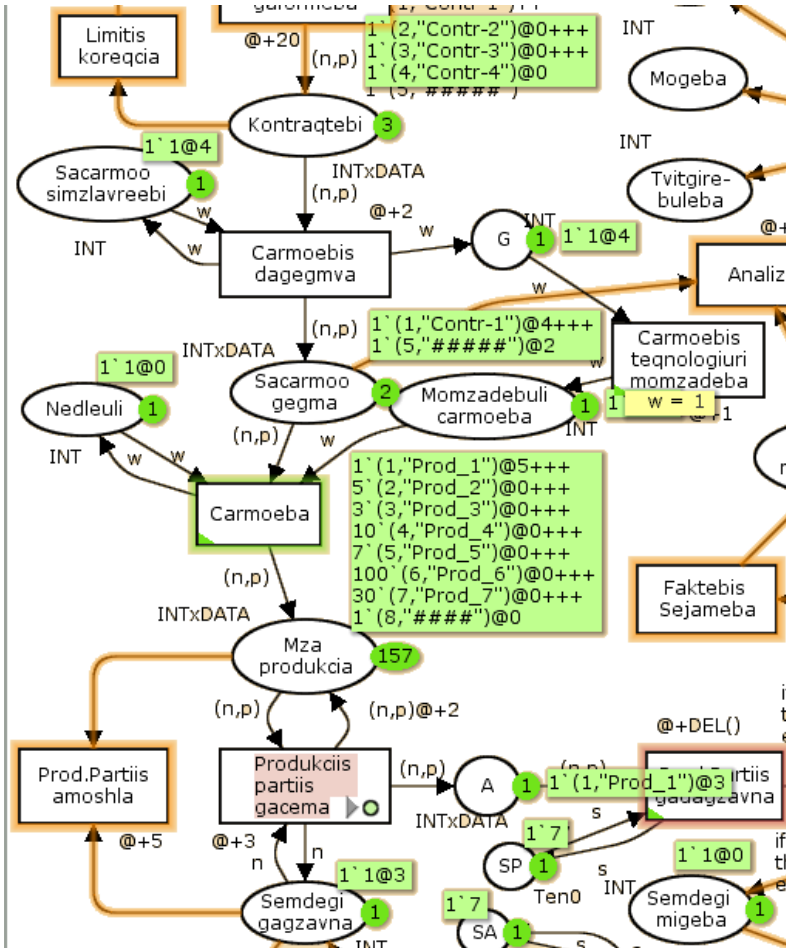
ნახ.1.14. პეტრის ქსელის შექმნისა და იმიტაციური მოდელირების ინსტრუმენტები

1.15 ნახაზზე ნაჩვენებია გვაქვს ჩვენი ქსელის ფრაგმენტი რამდენიმე ბიჯის შემდეგ, სადაც ჩანს მარკირების შეცვლილი მდგომარეობა. თავიდან გაიშვება გადასასვლელი „წარმოების დაგეგმვა“ (იხ. ნახ.1.12), ვინაიდან მის შესასვლელ პოზიციებში „კონტრაქტები“ და „საწარმოო სიმძლავრეები“ მზადაა მარკირები, რომლებიც გადაადგილდება პოზიციებში „საწარმოო გეგმა“ ($n=1, p=$ “Contr_1”) და „G“ ($w=1$). ეს უკანასკნელი არის სიგნალი იმის შესახებ, რომ 1-ელი კონტრაქტით გათვალისწინებული პროდუქციის საწარმოებლად საჭიროა „წარმოების ტექნოლოგიური მომზადება“, რასაც ასრულებს შესაბამისი გადასასვლელი. ამგვარად, თუ ნედლეულიც შემოსულია (პოზიციაში „ნედლეული“ არის 1 მარკერი), მაშინ გადასასვლელი „წარმოება“ ამუშავდება. პარალელურად ქსელში გაიშვება „პროდუქციის პარტიის გაცემის“ გადასასვლელი და მარკერი გადავა „გაგზავნის“ -პოზიციაში ($n=1, p=$ “Prod_1”). ტრანსპორტირების გარკვეული დროის შემდეგ (სტოქასტიკური დრო: @+DEL()) პროდუქცია მიაღწევს დამკვეთამდე და ა.შ.

CPN-ის სიმულაციის ინსტრუმენტით შესაძლებელია მარკირებათა მდგომარეობებისა და სტატისტიკური ანალიზის ჩატარება, შესაბამისი დიაგრამების აგებით.

ფერადი პეტრის ქსელებში კარგადაა შერწყმული პეტრის ქსელებისა და დაპროგრამების თეორია (იერარქიულობა, მოდულურობა – დიდი სისტემების მოდელირებისთვის), რაც მის დიდ პრაქტიკულ ღირებულებასაც განაპირობებს თანამედროვე ინფორმაციულ

ტექნოლოგიათა გამოყენების მრავალ სფეროში, განსაკუთრებით ბიზნესისა და მარკეტინგის მენეჯმენტის ამოცანების გადასაწყვეტად.



ნახ.1.15. იმიტაციური მოდელირების შუალედური ეტაპი

1.8. დაპროგრამების UML/2 ტექნოლოგია

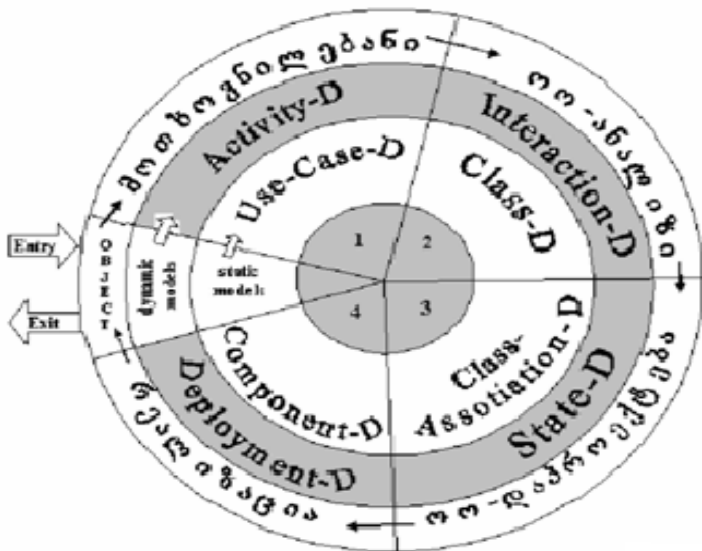
მაიკროსოფტის უახლესი პროგრამული .NET- პლატფორმა სულ უფრო ფართოდ იკიდებს ფეხს მსოფლიოს მოწინავე ქვეყნების საუნივერსიტეტო-სამეცნიერო და საწარმოო ფირმების ბიზნესის სფეროებში. იგი გამოიყენება Windows, Unix, და Linux ოპერაციული სისტემებისათვის. .NET-პლატფორმა შეიქმნა სპეციალურად განაწილებული გამოყენებითი 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) „გენეტიკური“ 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, ვინაიდან მასში შემუშავებულია მოდელირების კონსტრუქციების განსაზღვრული სიმრავლე აღწერის ერთიანი სისტემითა და სემანტიკით. 1.16 ნახაზზე. მოცემულია პროგრამული აპლიკაციების შექმნის 4 ეტაპი.

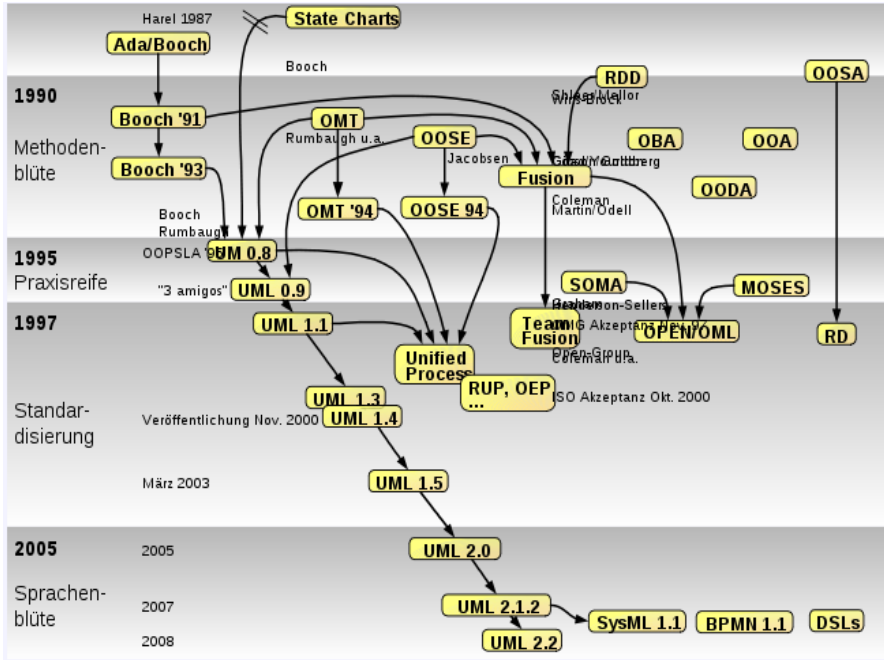
საკვლევი ობიექტის ავტომატიზაციის მოთხოვნის დადგენა, მისი ობიექტ-ორიენტირებული (ოთ) ანალიზი, ოთ-დაპროექტება (დეტალური დონე) და რეალიზაცია (პროგრამული კოდი). ეს ეტაპები სტატიკური და დინამიკური დიაგრამებით (D) ხორციელდება [20].



ნახ.1.16. UML–ტექნოლოგიის ეტაპები

ამგვარად, პროგრამული პაკეტების აგების ტექნოლოგიის სრულყოფის და სტანდარტიზაციის მოთხოვნამ ხელი შეუწყო უნიფიცირებული მოდელირების ენის შექმნას. როგორც ცნობილია, გრადი ბუჩის და მისი კოლეგების მიერ ფორმირან Rational Rose შემოთავაზებული UML-მეთოდოლოგია (1995/7წწ.), რომელსაც საფუძვლად ობიექტ-ორიენტირებულ მიდგომა დაედო, ასევე ოთხეტაპიანია. UML/1 გრაფო-ანალიზური და ვიზუალური კომპონენტების გამოყენებით, სტატიკური და დინამიკური მოდელების 8 სახის დიაგრამის აგების საშუალებას იძლევა კონკრეტული ობიექტისთვის. გამოჩნდა ამ ტექნოლოგიის ინსტრუმენტული საშუალებანიც: Rational_Rose, Paradigm_Plus, Ms_Visio და სხვ., რომელთა გამოყენებითაც დაპროექტების და დაპროგრამების ნაწილობრივი ავტომატიზაციაა შესაძლებელი [95,96].

2005 წლიდან დაფიქსირდა უნიფიცირებული მოდელირების ენის მე-2 თაობის პირველი ვერსია, რომელიც UML/2 სახელით გახდა ცნობილი, ხოლო 2008 წ. (ნახ.1.17). OMG-ჯგუფმა ოფიციალურად გამოაქვეყნა ამ პროდუქტის ბეტა-ვერსია [97].



ნახ.1.17. UML2–ის განვითარების ამსახველი დიაგრამა

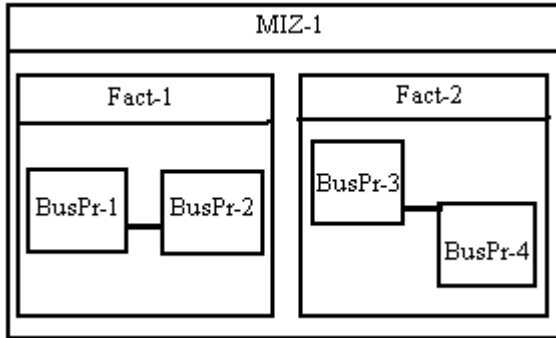
იგი იყენებს 13 სახის დიაგრამას, მათ შორის: ექვსი სტრუქტურული და შვიდი ყოფაქცევის დიაგრამა:

№	სტრუქტურული-დ	№	ყოფაქცევის-დ
1	კლასთა-დიაგრამა	1	აქტიურობის-დიაგრამა
2	კომპოზიციურ სტრუქტურული-დ.*)	2	გამოყენებით-შემთხვევათა-დ.
3	კომპონენტების-დ.	3	ინტერაქციათა-მიმოხილვის-დ.*)
4	განაწილების-დ.	4	კომუნიკაციის-დ.*)
5	ობიექტების-დ.*)	5	მიმდევრობითობის-დ.
6	პაკეტური-დ.*)	6	დროით-ობიექტური-დ.*)
		7	მდგომარეობათა-დ.

შენიშვნა: *) –ით აღნიშნულია UML2-ში დამატებული დიაგრამები.

კომპოზიციურ-სტრუქტურული დიაგრამა (Composite structure diagram) სტატიკური დიაგრამაა (ტიპით „მთელი-ნაწილი“), რომელიც

ასახავს კლასების შინაგან სტრუქტურას და კავშირებს მათ ელემენტებს შორის (მაგალითი ნახ.1.18).



ნახ.1.18: მიზანი, ფაქტორები, ბიზნეს-პროგრამები

ობიექტების დიაგრამა (Object diagram) ასახავს კლასის ეგზემპლარებს (ობიექტებს), ატრიბუტების მნიშვნელობებითა და მათ შორის კავშირებით. მომდევნო თავში ჩვენ დეტალურად განვიხილავთ ამ საკითხს ობიექტ-როლური მოდელების საფუძველზე.

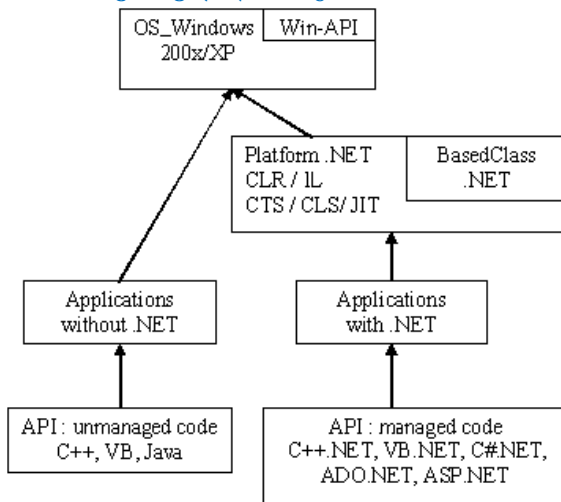
პაკეტური დიაგრამა

გარდა ახალი დიაგრამებისა, რომლებიც ობიექტის ყოფაქცევის მოდელში უფრო მდიდარი სემანტიკის ასახვის საშუალებას იძლევა, პირველად შეიქმნა ე.წ. „გაცვლითი ფორმატი“, რომლის დანიშნულებაა სხვადასხვა ფორმის UML-ინსტრუმენტებს შორის დიაგრამების გაცვლის შესაძლებლობა. ეს კი უფრო მაღალი საფეხურია პროგრამული პაკეტების სტანდარტიზაციის გზაზე.

1.9. .NET პლატფორმა და პროგრამული ინსტრუმენტები

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].

1.17 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.1.17. Windows და .NET-პლატფორმათა ურთიერთმოქმედება

მართვაში იგულისხმება ის, რომ ეს კოდები ამუშავდება უშუალოდ .NET-ის მიერ, იმართება მათი პროცესებისა და მონაცემთა

ნაკადები, მიეწოდება შესასრულებლად საჭირო დამხმარე რესურსები და ა.შ. პრინციპში, NET-პლატფორმა ასრულებს „ოპერაციული სისტემის“ გარკვეულ ფუნქციებს და მოქნილად ფუნქციონირებს Windows-თან.

.NET-ის კვებლოკი Based Class.NET არის ამ პლატფორმის საბაზო კლასების ბიბლიოთეკა (უმრავლესობა დაწერილია C#-ენაზე). იგი სრულად ობიექტ-ორიენტირებულია, შედგება ობიექტთა ერთობლიობისგან, რომელთაგანაც თითოეულში 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.

როგორც აღნიშნეთ, NET-პლატფორმისა და IL-შუალედური ენის არსებობის კონცეფციის საფუძველია ენის ობიექტ-ორიენტულობა და საერთო ტიპების სისტემის არსებობა, რომელთაც კომპილატორები ფლობს.

C# („სი შარფ“) ენა ობიექტ-ორიენტირებული ენების ერთ-ერთი ახალი და მძლავრი წარმომადგენელია, რომელიც შეიქმნა სპეციალურად NET-პლატფორმისათვის და თავსებადია Windows-ის თანამედროვე ვერსიებთან და ინტერნეტთან. ამ ენაზეა რეალიზებული NET-პლატფორმის უმრავლესი საბაზო კლასები.

როგორც ცნობილია, C++ ენა კომპილირდება ასემბლერულ კოდში, C# ენა კი - შუალედურ IL-ენაში.

IL-ენის დანიშნულებაა პლატფორმული და ენობრივი დამოუკიდებლობის განხორციელება ობიექტ-ორიენტირებულ გარემოში. Java-ენაც უზრუნველყოფს პლატფორმულ (Windows, Unix, Linux) დამოუკიდებლობას, მაგრამ მისი ბაიტ-კოდის შესრულების ეტაპზე იგი ინტერპრეტირდება (IL -კი კომპილირდება).

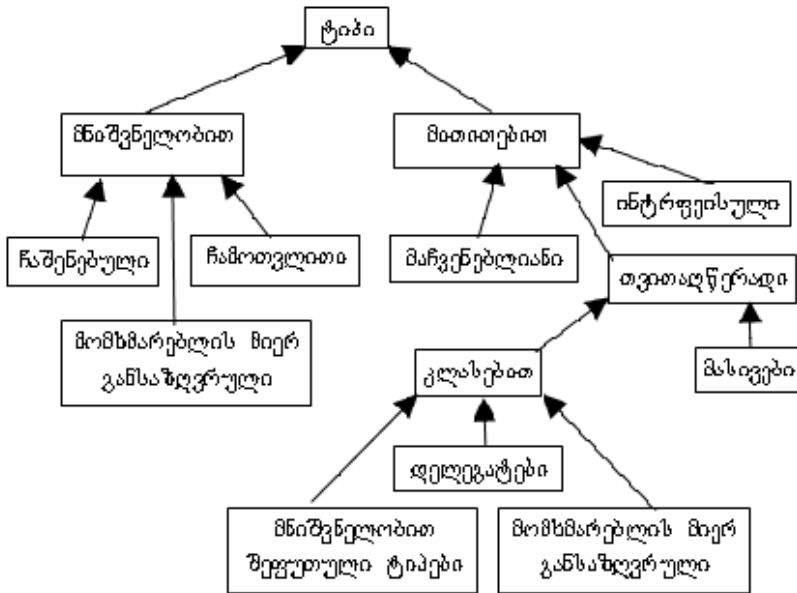
NET-პლატფორმისათვის ენობრივი თავსებადობა ხორციელდება IL ენაში არსებული ტიპების დიდი რაოდენობით, რომლებიც ორგანიზებულია ტიპთა იერარქიის ობიექტ-ორიენტირებული პრინციპებით. 1.18 ნახაზზე ილუსტრირებულია ტიპთა ასეთი იერარქია მემკვიდრეობითობის კავშირის გამოყენებით.

მოვიტანოთ ზოგიერთი კომენტარი, რომელიც ახსნის ნახაზს:

- **ტიპი** არის საბაზო კლასი, რომელიც ზოგადად ასახავს ნებისმიერ ტიპს;

- **ტიპი მნიშვნელობით** არის საბაზო კლასი, რომელიც ზოგადად ასახავს ნებისმიერ ტიპს მნიშვნელობით;

- **ჩაშენებული ტიპები მნიშვნელობით** არის სტანდარტული საბაზო ტიპები, რომლებიც აღწერს რიცხვებს, სიმბოლოებსა და ლოგიკურ მნიშვნელობებს;



ნახ.1.18. ტიპების ზოგადი სისტემა

- ჩამოთვლილ ტიპი არის ჩამონათვალთა ერთობლიობა, რომელშიც თითოეულ მნიშვნელობას შეესაბამება რიცხვითი მნიშვნელობა (0,1,... და ა. შ.) მისი მდებარეობის მიხედვით;

- მომხმარებლის მიერ განსაზღვრული ტიპი არის საწყის კოდში (მომხმარებლის პროგრამაში) აღწერილი ტიპები, რომლებიც ინახება მნიშვნელობებით (ესაა მაგალითად, სტრუქტურები).

- ტიპი მითითებით არის მონაცემთა ნებისმიერი ტიპები, რომელთანაც მიმართვა ზორციელდება მიმითებლებით და ინახება ნაკადში;

- თვითაღწერადი ტიპები არის ტიპები, რომლებიც ასახავს ინფორმაციას თავიანთ შესახებ;

- თვითაღწერადი ტიპები არის ტიპები, რომლებიც ასახავს ინფორმაციას თავიანთ შესახებ;

- მასივები არის ნებისმიერი ტიპი, რომელიც შეიცავს ობიექტების მასივს;

- ტიპები კლასებით არის თვითაღწერადი ტიპები, რომლებიც არაა მასივები;

- დელეგატები - ტიპებია, რომლებიც დამუშავდა მიმთითებელთა შესაძლებლობად კლასის მეთოდებისათვის;

- მნიშვნელობით შეფუთული ტიპები არის ტიპები მნიშვნელობით, რომლებიც დროებით დაიყვანება ტიპებამდე მიმთითებლით, რათა შენახულ იქნას ნაკადში;

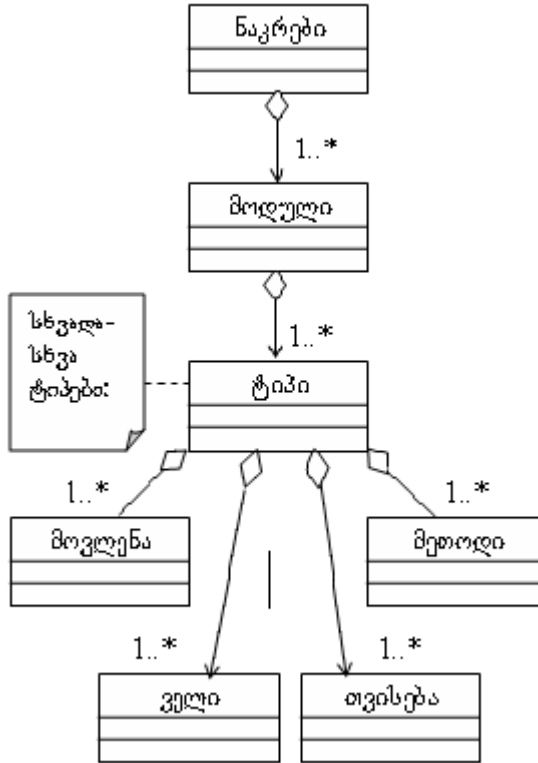
- მომზადების მიერ განსაზღვრული ტიპები მითითებით არის ტიპები, განსაზღვრული საწყის კოდში, როგორც ტიპები მითითებით. პროგრამაში ესაა მაგალითად, ნებისმიერი კლასი.

NET პლატფორმის საბაზო კლასების დიდი ნაწილი დაწერილია C# ენის გამოყენებით, ამიტომაც საილუსტრაციო მაგალითებს ამ ენაზე ვაგუგუთებთ კომენტარს..

.NET პლატფორმის კომპონენტებიდან ერთ-ერთი მთავარი ბლოკია Assembly (ანაწყობი, ნაკრები), რომელიც ლოგიკურად აერთიანებს კოდს, რესურსებს და მეტამონაცემებს. იგი ლოგიკური და არა ფიზიკური ერთეულია, რადგან შეუძლია მოთავსდეს რამდენიმე ფაილში. ასეთ შემთხვევაში არსებობს ერთი მთავარი ფაილი, რომელშიც ინახება ინფორმაცია დანარჩენებზე.

1.19 ნახაზზე ნაჩვენებია პროგრამული დანართის (Application) შესაბამისი ნაკრების ზოგადი იერარქიული სტრუქტურა აგრეგაციის კავშირის გამოყენებით.

ნახაზიდან ჩანს, რომ ნაკრები 1 ან რამდენიმე (1..*) მოდულისგან (module) შედგება. სწორედ მოდულში ინახება დანართის ან ბიბლიოთეკის კოდი, მისი მეტამონაცემებით. მოდულები შეიცავს ტიპებს. ესაა კოდის შაბლონები (კლასები), რომლებშიც ინკაფსულირებულია გარკვეული მონაცემები და მეთოდები. როგორც წინა პარაგრაფში ვახსენეთ, ტიპები ორი სახისაა: მიმთითებლებით (ანუ კლასები) და მნიშვნელობებით (ანუ სტრუქტურები).



ნახ.1.19. Assembly (ნაკრები) ზოგადი სტრუქტურა

ტიპებს აქვთ ველები, თვისებები და მეთოდები. ველი გამოყოფს მენსიერების ადგილს შესაბამის მონაცემთა ტიპისათვის. თვისებები ველების მსგავსია, ოღონდაც მათი დანიშნულებაა შესაბამის მონაცემთა საწყისი მნიშვნელობების განსაზღვრა და კონტროლი.

მეთოდები განსაზღვრავს მონაცემთა დასამუშავებლად კლასის ქცევას, ანუ რეაქციას გარედან შემოსულ შეტყობინებაზე (მოთხოვნაზე). შეტყობინება ინაფორმაციაა, რომელიც ამა თუ იმ მოვლენის შედეგად ფორმირდება.

პროგრამული ნაკრები შეიძლება იყოს ორი ტიპის: კერძო და საერთო გამოყენების. პირველ შემთხვევაში ნაკრები ინსტალირდება კერძო მომხმარებელს კატალოგში და მასთან სხვა მიმართვები გამორიცხულია.

საერთო გამოყენების ნაკრები შეიცავს პროგრამულ ბიბლიოთეკებს, რომელთაც იყენებს სხვადასხვა დანართი. აქ საჭიროა სპეციალური დაცვის მექანიზმების გამოყენება (სახელების კოლიზიისა და ნაკრებთა ვერსიების კონტროლის თვალსაზრისით).

კლასებს შორის სახელთა კოლიზიის აღმოფხვრის მიზნით .NET პლატფორმა იყენებს „სახელთა სივრცეს“.

სახელსივრცე (namespace): ესაა მონაცემთა ტიპების უბრალო დაჯგუფება. ყველა მონაცემთა ტიპის სახელს მოცემულ სახელთა სივრცეში ავტომატურად ემატება პრეფიქსი, რომელიც შედგენილია სახელთა სივრცის დასახელებისგან. ასევე შესაძლებელია ჩადგმული სახელთა სივრცეების შექმნა.

მაგალითად, საბაზო კლასების უმრავლესობისათვის, რომლებიც ზოგადი გამოყენებისთვისაა დანიშნული, მოთავსებულია სახელთა სივრცეში System, ვებ-გვერდებისათვის - System.Web და ა.შ.

C#-ის პროგრამის ტექსტის მაგალითზე შეიძლება შემდეგი კომენტარის გაკეთება:

```
namespace Magazia.Web // აქ მითითებულია სახელი Magazia.Web  
{  
    public class Checkout : PageBase  
    {  
        // და ა.შ.
```


დანართთა არეები (application area) არის .NET პლატფორმის მნიშვნელოვანი ელემენტი. მათი დანიშნულებაა ერთდროულად და ერთმანეთთან მომუშავე დანართების იზოლაცია, რათა არ მოხდეს მონაცემთა არასასურველი დამუშავება.

პროგრამული დანართების იზოლაციისათვის Windows გამოიყენებს „**პროცესის**“ ცნებას, რომელიც მისამართების სივრცეს ეხება. ყოველ პროცესს გამოეყოფა 4 გიგაბაიტი ვირტუალური მეხსიერება. ისინი დისკზე სხვადასხვა ფიზიკური მისამართებითაა და არ გადაიკვეთება.

პროცესებს აქვს მინიჭებული განსაზღვრული პრივილეგიები და ოპერაციული სისტემა აკონტროლებს მათ, თუ რომელ ოპერაციას რომელი პროცესის გამოყენება შეუძლია.

დანართთა არეების გამოყენების იდეა მდგომარეობს იმაში, რომ პროცესებს შორის მოხერხდეს მონაცემთა გაცვლა. ამიტომაც პროცესი იყოფა რამდენიმე დანართის არედ. თითოეულ დანართის არეში თავსდება ერთი დანართის კოდი.

.NET პლატფორმის მნიშვნელოვანი საშუალებაა JIT (Just-In-Time) კომპილატორი. იგი ახორციელებს პროგრამული კოდის ცალკეული ნაწილის დროულად კომპილირებას (საჭიროების შემთხვევაში).

Visual Studio.NET არის პროგრამული სისტემების დამუშავების ინტეგრირებული გარემო, რომელშიც შესაძლებელია კოდების დაწერა, კომპილირება და გამართვა VB.NET, C++.NET, C#.NET, ASP.NET, ADO.NET-ის და სხვა ტექნოლოგიებით.

ახლა მოვიტანოთ C++ და C# მარტივი კოდები, რათა დავინახოთ მსგავსება-განსხვავება ამ ენების სინტაქსებს შორის:

a) C++ -ის კოდის ფრაგმენტი:

```
#include <iostream.h> // C++ -----  
#include <Windows.h>  
int main(int argc, char *argv)  
{  
    cout << "Hello, my friend !";  
    MessageBox(NULL, "By-By !", "", MB_OK);  
    return 0;  
}
```

b) C#-ის კოდის ფრაგმენტი:

```
using System; // C# -----  
using System.Windows.Forms;  
namespace Console1;  
{  
    class Class1  
    {  
        static int Main(string[] args)  
        {  
            Console.WriteLine("Hello, my friend !");  
            MessageBox.Show("By-By !");  
            return 0;  
        }  
    }  
}
```

1.10. 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 ექპერტულ შეფასებათა გადაწყვეტილების მიღების მხარდამჭერი სისტემების უპირატესობა მონაცემთა საცავისა და მისი ოპერატიული ონლაინური ანალიზის პროცედურების ბაზაზე;

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

4. შემოთავაზებულია ასეთი პროცესების მოდელირებისათვის პეტრის ფერადი ქსელების CPN-ინსტრუმენტის, UML/2 ტექნოლოგიის და მისი რეალიზაციისთვის .NET პლატფორმის გამოყენება.

II თავი

ბიზნეს-პროგრამების მართვის ხელშეწყობა კომპიუტერული სისტემის აგების მეთოდებისა და მოდელის შემუშავება

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

როგორც 1-ელ თავში აღვნიშნეთ, 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 ძვირადღირებული 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 ცდილობენ 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 გამოიმუშავენ კოლექტიურ გადაწყვეტილებებს. მოხდება ამ გადაწყვეტილებათა ცხოვრებაში გატარების ღონისძიებების შემუშავება და მათი დანახარჯების ანალიზი.

ამგვარად, ფირმის ხელმძღვანელობა და მთავარი სპეციალისტები თვითონ არიან ექსპერტები, თვითონ ეხმარებიან კონსულტანტს საბოლოო გადაწყვეტილებების მოძებნაში. კონსულტანტს აქვს მეთოდიკა, თუ როგორ წარმართავს მუშაობას ექსპერტებთან. ეს მეთოდიკა გულისხმობს ობიექტის კვლევისათვის სისტემური მოდგომისა 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, „გონებრივი შეტევის“ მეთოდი, რომელიც ემყარება ახალი იდეის ჩამოყალიბებას (რალაც 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 ურთიერთზეავლენის 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 განზოგადებული 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 -ურ ინტერვალში. ხშირ შემთხვევაში, ექსპერტთა შეფასებებში ადგილი აქვს „ძალიან დიდ“ ან „ძალიან მცირე“ მნიშვნელობებს, რომლებიც გავლენას ახდენს საშუალო არითმეტიკულზე და, ეს უკანასკნელი, უკვე აღარ 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 ლუწია $2*n$, მაშინ მედიანად მიიღება n და $n+1$ წევრების მნიშვნელობათა საშუალო არითმეტიკული.

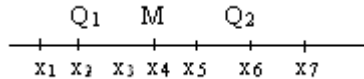
ექსპერტულ შეფასებათა ანალიზის დროს იყენებენ საშუალო შეფასებების ირგვლივ შეფასებათა მნიშვნელობების ვარიაციას. რაც

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 ღერძი

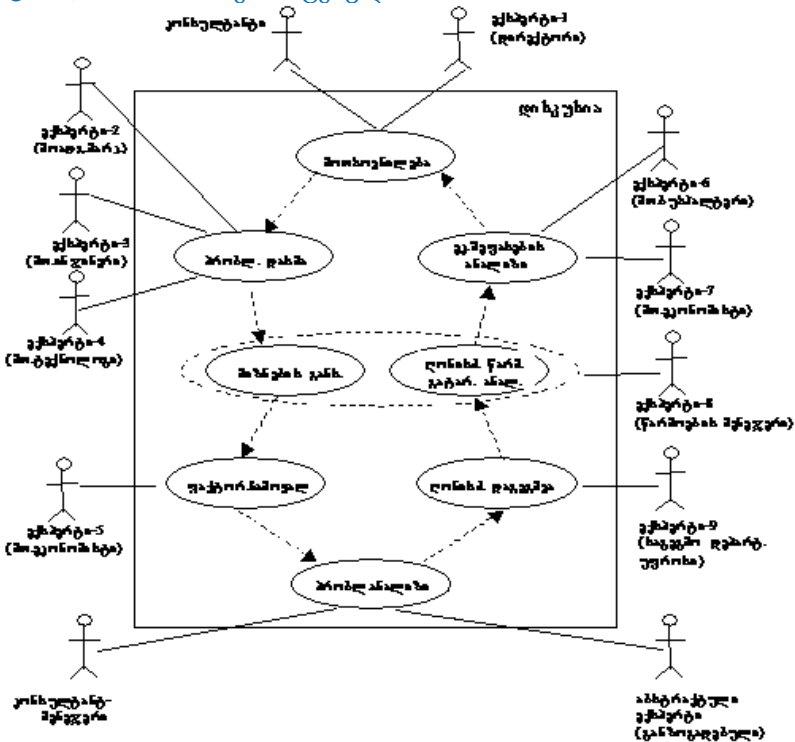
მაგალითად, თუ გვაქვს 7 ექსპერტული შეფასება, ისინი უნდა დალაგდეს მონოტონურად. შემდეგ საჭიროა ვიპოვოთ მედიანა (M). ჩვენს შემთხვევაში მედიანა ემთხვევა x_4 შეფასებას. ახლა განვსაზღვროთ ქვედა და ზედა კვარტილები (Q_1, Q_2). მედიანა და კვარტილები ქმნის ოთხ ინტერვალს: $< Q_1, Q_1M, MQ_2$ და $Q_2 >$ ამათგან Q_1M 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 გაუმხელად) გააცნობენ დანარჩენ ექსპერტებს.

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

გარდა ზემოგანხილულისა, ცნობილია აგრეთვე სხვა ექსპერტული მეთოდებიც, როგორცაა მაგალითად, დელფისა და ქსელური გრაფიკის განზოგადოებისა და პერსპექტიული დაგეგმვის სელექციური მეთოდი, რომელსაც საფუძვლად უძევს მიზნობრივი მიდგომა. აღნიშნული მეთოდის მსგავსია აგრეთვე პროგნოზირების ე.წ. „შეწონილ შეფასებათა“ მეთოდი, რომელიც იყენებს ასევე მიზნის ხეს და მატრიცებს. ცნობილია აგრეთვე პატერნ-მეთოდი, რომელიც

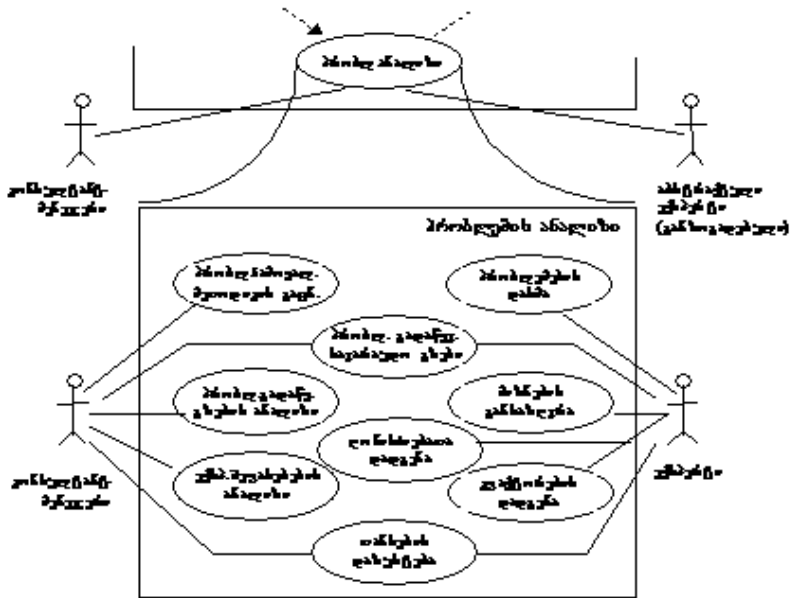
ანალოგიურია მიზნობრივი სტრუქტურის გრაფის გამოყენებისა ა.შ. [21].

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 თვითოეული პრეცედენტი (ოვალი) 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“ ლეტალიზაციის ღონე

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 საილენტიფიკაციო 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. განმორებითი 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 ღონისძიებათა დომინირებადობას მიზანზე;

16. კონსულტანტს შეაქვს თითოეულ ღონისძიების შესრულების ფასები;

17. სისტემა ამ ფასების გათვალისწინებით მოაწესრიგებს ღონისძიებებს;

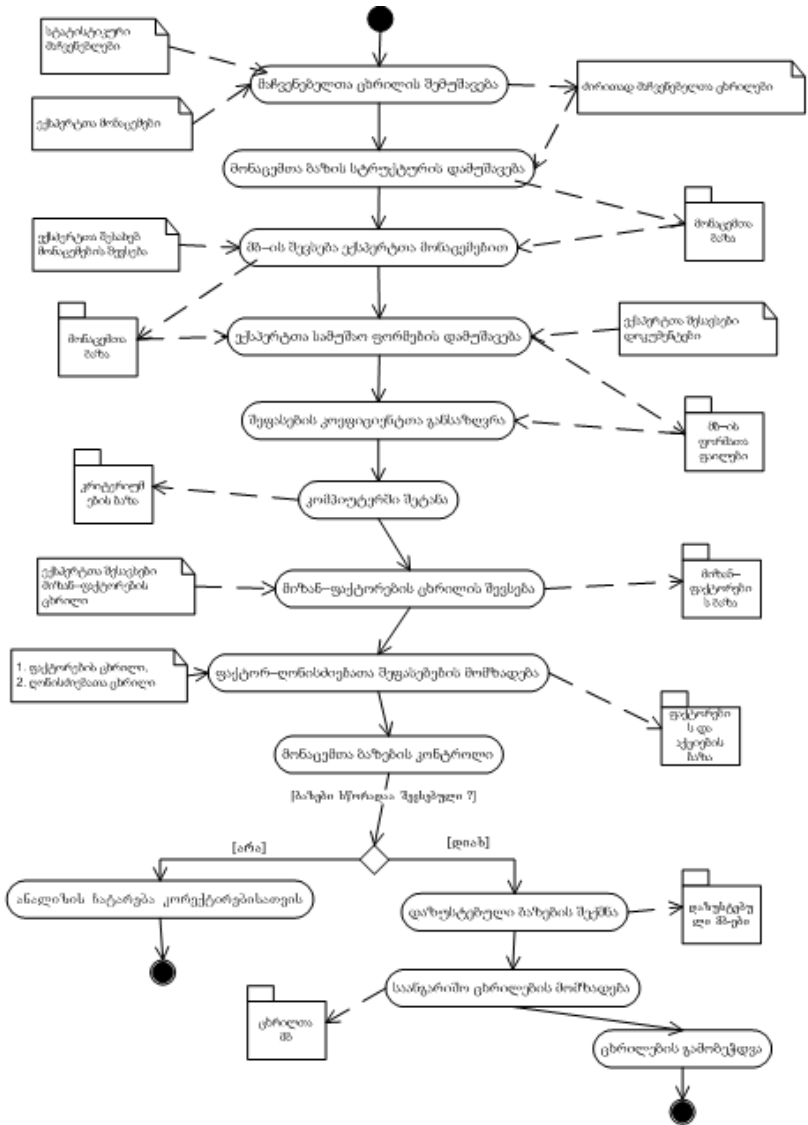
18. სისტემა ბეჭდავს საბოლოო შედეგებს;

19. კონსულტანტი ამზადებს დასკვნით ანგარიშს;

20. კონსულტანტი და ექსპერტები დასკვნით შეხვედრაზე განიხილავენ საბოლოო შედეგებს;

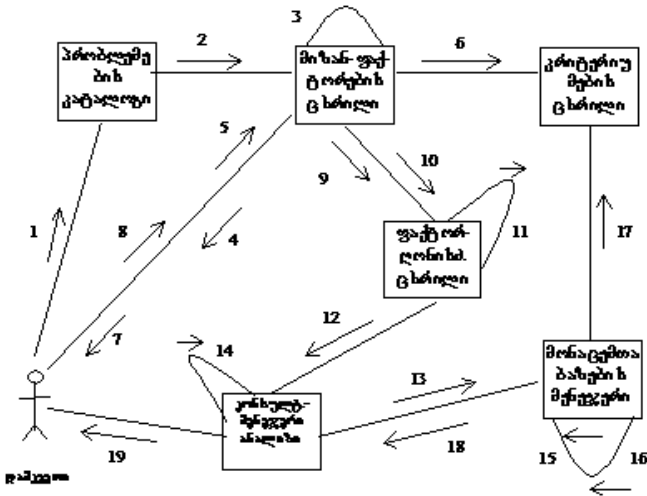
21. დასკვნით ლოკუმენტს მოეწერება ხელი ექსპერტების მიერ.

გამოყენებით შემთხვევათა დიაგრამების აგების შემდეგ განიხილება ბიზნეს-პროგრამების ექსპერტულ შეფასებათა მხარდამჭერი 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 მხარდამჭერი სისტემის აგების ტექნოლოგიური პროცესის აქტიურობის დიაგრამა

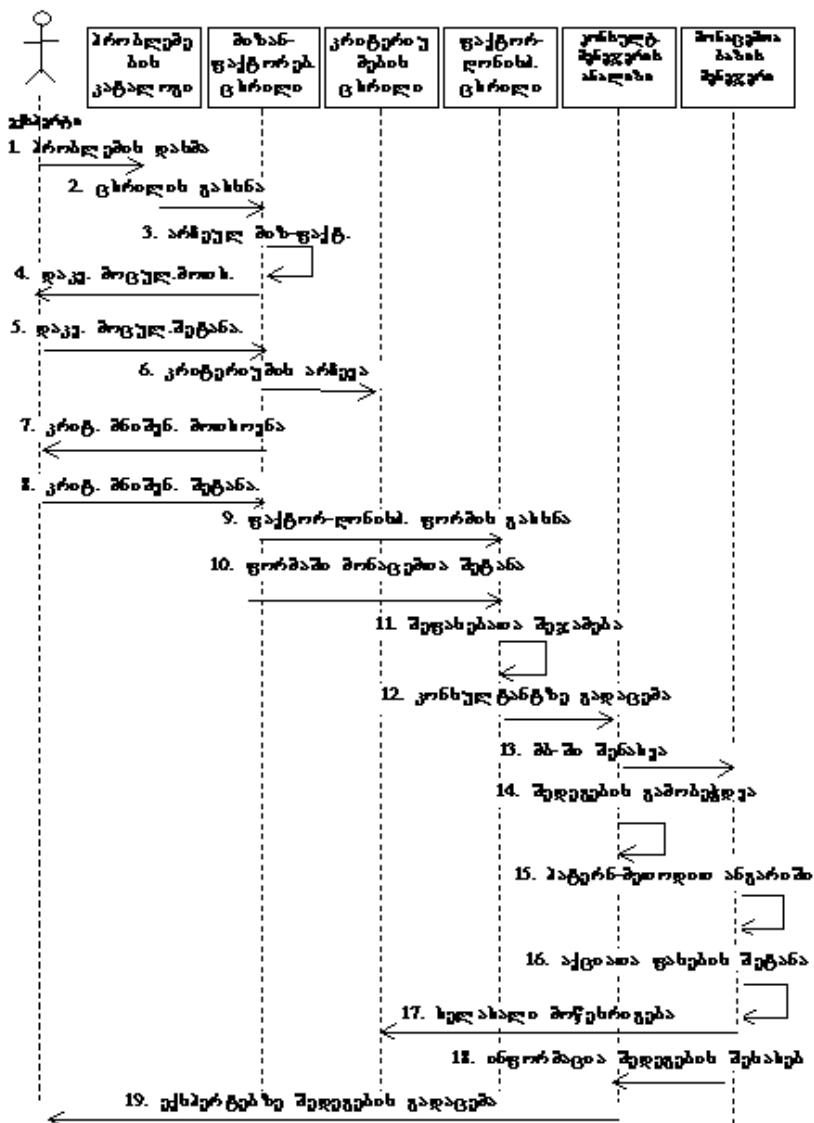
სისტემის კლასთა ობიექტებს შორის ურთიერთმოქმედების (Interaction) დიაგრამა. ამ მიზნით აიგება ორი სახის დიაგრამები: მიმდევრობითობის (Sequence) და თანამოქმედების (Collaboration). მათ საფუძველზე ხორციელდება ობიექტებს შორის შეტყობინებების გაცვლა დროის მიხედვით და შესაბამისი მეთოდების ამუშავება ინფორმაციული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



ნახ.2.5. თანამოქმედების დიაგრამის ფრაგმენტი

□ კლასი ; → შეტყობინება

ჩვენი ამოცანის გადასაწყვეტად, ანუ UML-დიაგრამების ასაგებად გამოვიყენეთ მაკროსოფტის 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].



ნან.2.6. მიმღვერობითობის ღიაგრამის ფრაგმენტი

2.2. მონაცემთა ბაზაში ექსპერტთა ცოდნის ასახვის ობიექტ-როლური მოდელების შემუშავება კატეგორიული 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 არაფორმალური მეთოდებით (ბუნებრივი სალაპარაკო ენის საშუალებით);

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

- კონცეპტუალური მოდელის აგება კატეგორიული 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 ამასთანავე, ორიენტირებულია გამოყენებითი სფეროს მომხმარებელზე.

ცოდნა, რომელიც აქვს საპრობლემო სფეროს მომხმარებელს, სპეციალური ინტერფეისების საშუალებით, რომელთა საფუძველია ფორმალურ გრამატიკათა აპარატის კატეგორიები და ლოგიკურ-ალგებრული მეთოდები, გადაეცემა 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 კატეგორიული მიდგომის გამოყენებით მონაცემთა ბაზებში [25,26].

კატეგორიები რომლებსაც აქ ვიხილავთ, არის ლოგიკური გრამატიკის შემადგენელი ნაწილი, რომელიც, როგორც ზოგადი ენის საძირკველი, ბევრად მდიდარია, ვიდრე ტრადიციული პრედიკატების ლოგიკა და რომელიც ცხადია დამოუკიდებელი უნდა იყოს ბუნებრივი ენის ემპირიული გრამატიკისგან. პრედიკატების ლოგიკა არის ლოგიკური გრამატიკა დანართების შინაარსის ასაგებად, მაგრამ ძალზე ღარიბია 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 \wedge 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 მანასიათებელი (აღნიშვნა). მარჯვნივ ზის პრედიკატორი (არსის დასახელება – განსხვავებული გამოსახულება).

საკუთარი სახელები და მანასიათებლები არის ნომინატორები.

ამგვარად, კატეგორია შეიძლება ჩაეწეროს სამეულთ:

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 ე (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“ ვლბულობთ:

„პოპული ე სწრაფი ^ პოპული ე გამყიდველი“.

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

„პოპული π გაყიდვა“ ან

„პოპული π (სწრაფი)(გამყიდველი)“

მოქმედების კოპულით π (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 წინადადების ლოგიკური ნაწილაკით შეერთებით.

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

„ვაჭარაძე იღებს პროდუქტს საწყობიდან“.

პროდუქტი - პირდაპირი ობიექტია, რომელზეც მიუთითებს მოქმედების პრედიკატორი „იღებს“. 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)	(პროლექტი)	(i საწყობი)
[ნომინატორი]		[მოქმედების პრედიკატორი]	[ობიექტის პრედიკატორი]	[მახასიათებელი]

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“.

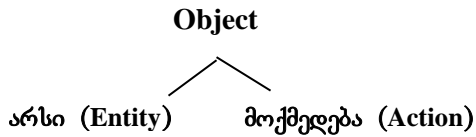
სქემების აგებისას დრო (ან დროის პუნქტი) არის ჩვეულებრივი პრედიკატორი, რომელიც „მანამ“, „შემდეგ“ და ა.შ. მიეთითება ორადგილიან პრედიკატორში „**x, y ε before**“ ან „**x, y ε after**“.

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

სქემის დროსთვის იყენებენ ტერმინს „ინტენსიონალი“ (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 ჩაწეროთ როგორც

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

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

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

AC - აბსტრაქტცია-კონკრეტიზაცია;

CD - კომპოზიცია – დეკომპოზიცია;

TP - მთელი - ნაწილი

LML - ენა – მეტაენა;

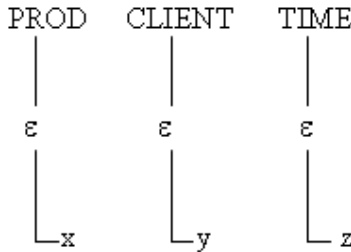
TIE - ინტენსიონალი- ექსტენსიონალი;

ERS - არსთა – დამოკიდებულებათა სქემა.

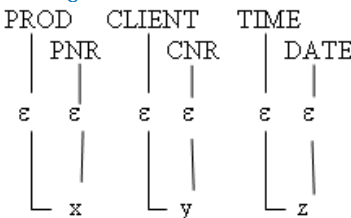
არსთა-დამოკიდებულების სქემის აგება საპრობლემო სფეროსთვის, ამ პროცესის კვლევა და მისი სრულყოფა ეფექტური ალგორითმული სქემებით და პროგრამული გრაფო-ანალიზური ინსტრუმენტებით – არის ჩვენი ნაშრომის ერთ-ერთი მთავარი საკითხი.

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

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



კატეგორიული თვალსაზრისით, ჩვენ ვიმყოფებით სქემის-დროში (schma-time) და არა მიმართვის-დროში (referent-time). მაგალითად, „დოლიძე ε კლიენტი“ - ეს არაა განზოგადება, პირიქით „ y ε კლიენტი“ არის y-ნომინატორული ცვლადით ყველა შემთხვევისთვის განზოგადებული. დოლიძე მხოლოდ კონკრეტულ მომენტში 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 პროლექციის წესი „თუ, მაშინ“), რომელიც ნიშნავს, რომ თუ მოცემულია მარცხენა მხარე (გამონათქვამი), მაშინ მარჯვენაც მართებულია:

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

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

მეორე მაგალითი: „S ურეკავს კლიენტს“, ჩაიწერება:

$$s \pi (\text{დარეკვა})(\text{CLIENT}).$$

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

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

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

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 დისკაპლინის ისეთ სფეროში, რომელიც დაკავშირებულია ენასა და შემეცნებასთან. მაგალითად, აქ შეიძლება გავიხსენიოთ 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 კონცეპტუალურ 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-დაკავშირებულ 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.

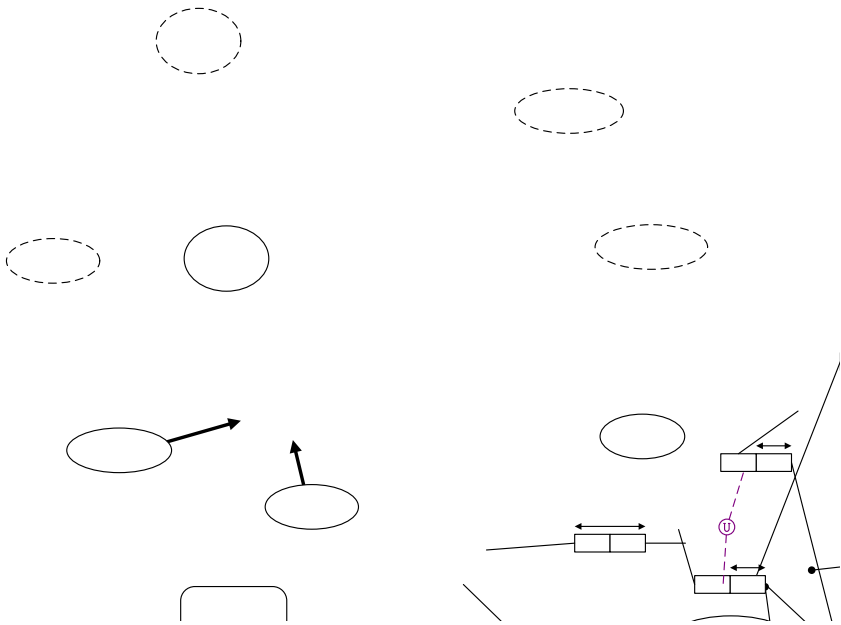
დიაგრამაზე გამოსახულია ისეთი შეზღუდვებიც, რომლებიც დამახასიათებელია ობიექტ-როლური მოდელირებისთვის. ობიექტები გამოისახება ელიფსებით. წვეტილ-კონტურიანი ელიფსი ასახავს ობიექტის მნიშვნელობას (value). პრედიკატები გამოსახულია მართკუთხედებით და ისინი აღწერს ობიექტთაშორის კავშირებს.

⊙ - 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 ქვეტიპზე (მაგალითად, შემკვეთი და მიმწოდებელი არის ზოგადად ფირმის ქვეტიპები).



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

„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 მხარდამჭერი სისტემის ობიექტ-როლური და კონცეპტუალური მოდელების ასაგებად.

2.3. ობიექტ-როლური მოდელების დაპროექტება და აგება ავტომატიზებულ რეჟიმში

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 [1,73].

ბიზნეს-პროგრამების ექსპერტულ შეფასებათა მხარდამჭერ 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 ინტუიციის 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 ღონისძიებათა შესახებ, ბიზნეს-პროგრამების შერჩევის ნაკლები რისკით. 2.2 ნახაზზე მოცემული იყო დასაპროექტებელი სისტემის როლურ-ფუნქციური UseCase დიაგრამა UML ენაზე, რომლის განხორციელებაც განიხილება ქვემოთ.

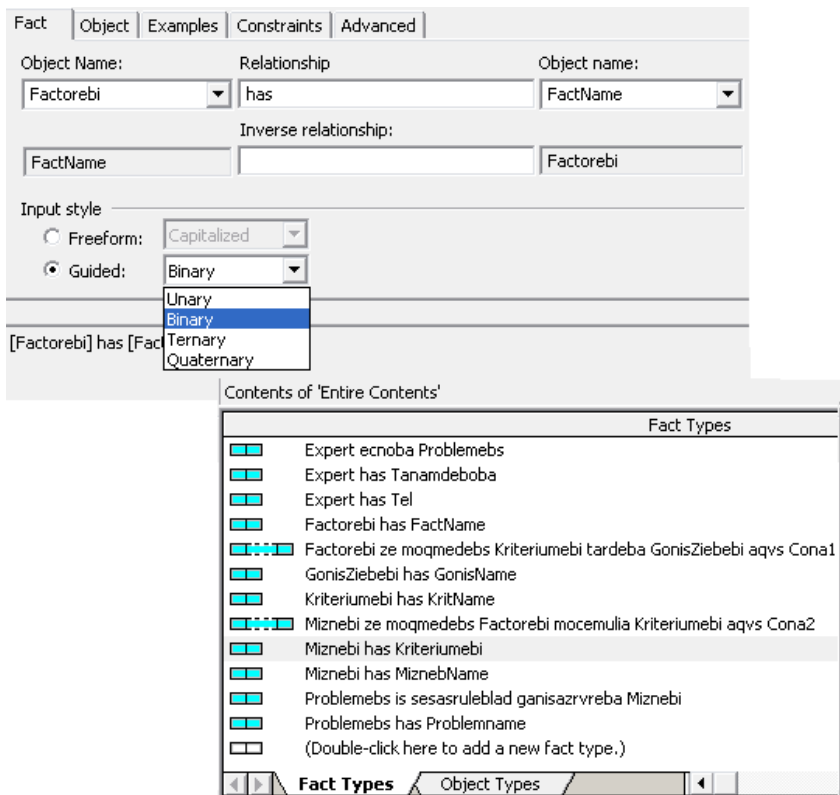
ექსპერტთა შეფასებების მხარდამჭერი სისტემის დაპროექტება იწყება საკვლევი სფეროს ობიექტ-ორიენტირებული ანალიზით, რომელიც, როგორც ცნობილია, მოითხოვს სტატიკური და დინამიკური დიაგრამების: პრეცედენტების, აქტიურობის, მიმდევრობითობის, კლასების და ა.შ. მომზადებას [2,3].

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

ცხრ.2.1

ფ-№	ფაქტის ჩაწერის ფორმა
f ₁	- ექსპერტს აქვს გვარი;
f ₂	- ექსპერტს აქვს თანამდებობა;
f ₃	- ექსპერტს აქვს ტელეფონი;
f ₄	- ექსპერტი ეცნობა პრობლემას;
f ₅	- პრობლემას აქვ დასახელება;
f ₆	- პრობლემის შესასრულებლად განისაზღვრება მიზნები;
f ₇	- მიზანს აქვს დასახელება;
f ₈	- მიზანს აქვს კრიტერიუმი;
f ₉	- მიზანზე მოქმედებს ფაქტორები;
f ₁₀	- ფაქტორს აქვს დასახელება;
f ₁₁	- ფაქტორს აქვს კრიტერიუმი;
f ₁₂	- ფაქტორზე მოქმედებს ღონისძიებები;
f ₁₃	- ღონისძიებას აქვს დასახელება;
f ₁₄	- ღონისძიებას აქვს კრიტერიუმი;
და	- კრიტერიუმს აქვს წონა;
ა.შ.	

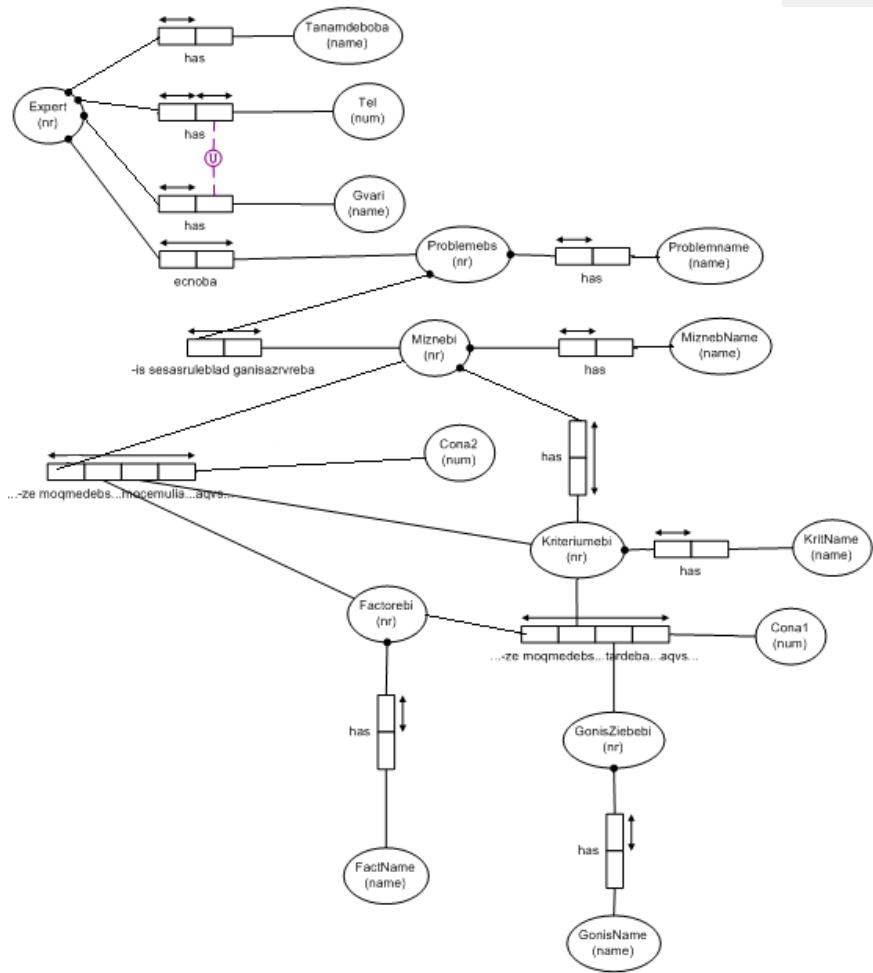
Visual Studio .NET გარემოში Ms Visio ინსტრუმენტის Database კატეგორიაში ORM Source Model- შაბლონით (Template) ვახორციელებთ ზემოაღწერილი ფაქტების ჩაწერას უნარული, ბინარული ან უფრო მაღალი რივის რეალაციური დამოკიდებულებით (ნახ.2.8).



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 მართკუთხედებით, მნიშვნელობის ტიპი გამოისახება წყვეტილი ელიფსით. იმ შემთხვევაში თუ განისაზღვრება ობიექტის მხოლოდ ერთი თვისება, საჭმე გვაქვს ერთ ადგილიან პრედიკატთან (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 გამოიყენება სხვადასხვა შეზღუდვები. ჩვენს შემთხვევაში გამოიყენებულია შემდეგი სახის შეზღუდვები.

- - იძულების შეზღუდვა გვიჩვენებს, რომ ექსპერტი ვალდებულია ჰქონდეს გვარი

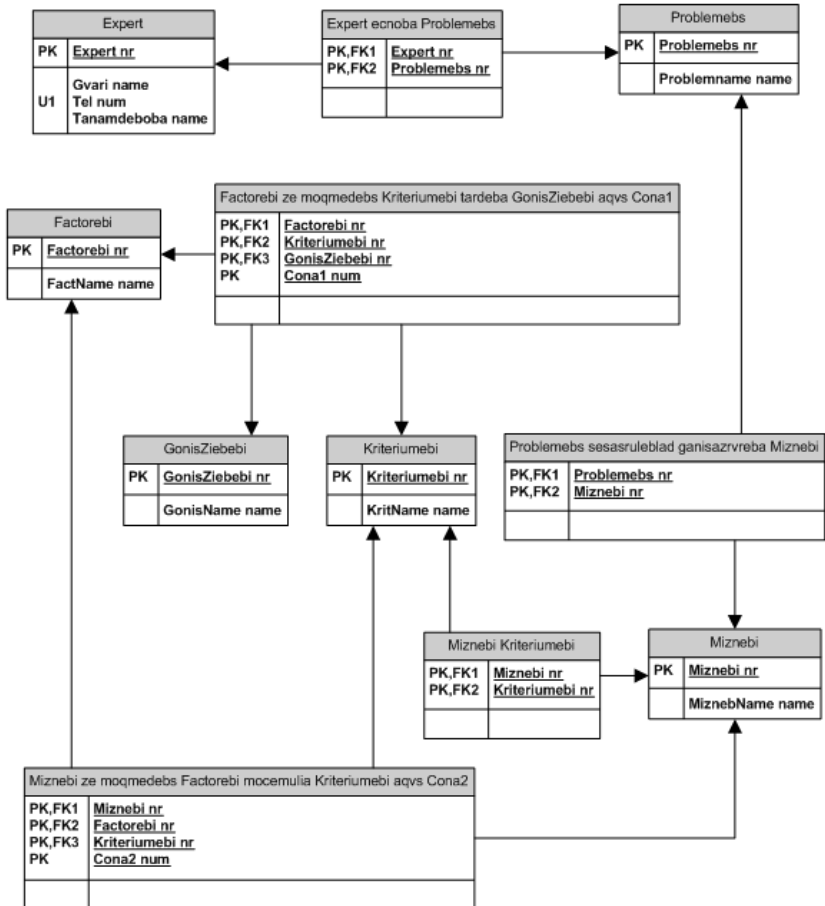
↔ - შიგა უნიკალურობის შეზღუდვა გვიჩვენებს, რომ რომ ექსპერტს აქვს მხოლოდ ერთი თანამდებობა.

Ⓢ - გვიჩვენებს, რომ მონაცემთა ბაზაში ექსპერტი უნიკალურედ განისაზღვრება გვარითა და ტელეფონის ნომრით.

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

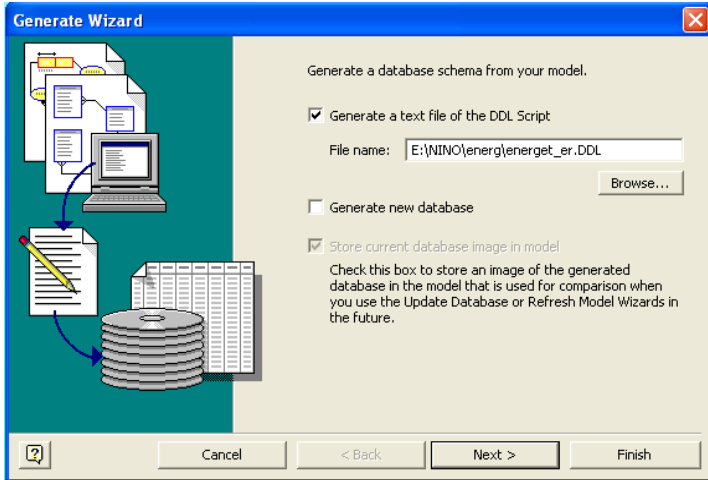
2.10 ნახაზზე მოცემულია ERM-დიაგრამა, რომელიც ავტომატიზებულ რეჟიმში აიგო ORM-დიაგრამის საფუძველზე. ამისათვის გამოყენებულ იქნა Ms Visio ინსტრუმენტის Database კატეგორიაში ERM მოდელის გენერაციის პროგრამა.

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",
"Miznebi 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 კავშირები, 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 დოკუმენტბრუნვის პრობლემათა გათვალისწინებით, ახალი საკანონმდებლო აქტების და ინტერნეტ-ინტრანეტის პირობებში.

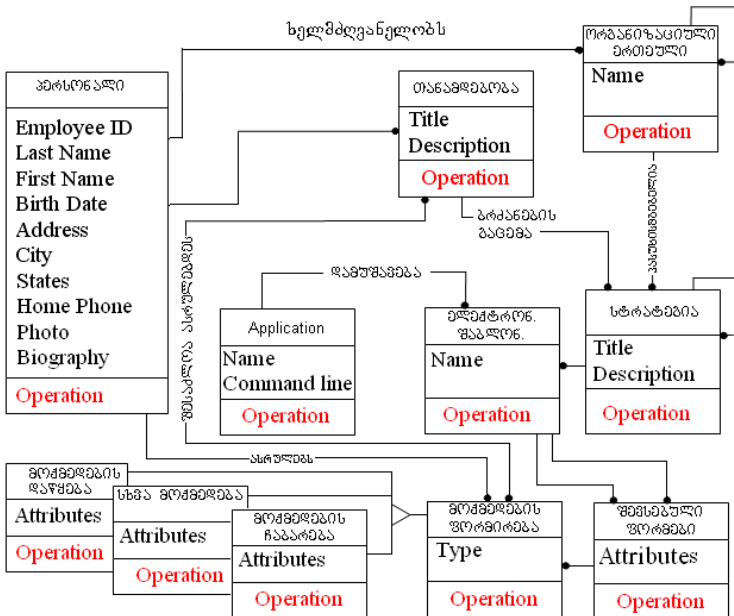
ბიზნეს-ობიექტი, ზოგადად წარმოადგენს ფიზიკურად განაწილებულ ბიზნეს-პროცესებს, მათი ლოგიკურად ერთიანი მონაცემთა საცავის (ბაზების ერთობლიობის) და ადმინისტრაციული მართვის კლასიკური მოდელის ინტეგრირებულ სტრუქტურას, უახლესი ინფორმაციული, მოდელურ-პროგრამული, ტექნიკურ-ტექნოლოგიური, ლინგვისტურ-ინტერფეისული, იურიდიული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

ყოველ ორგანიზაციაში მიმდინარეობს მისთვის დამახასიათებელი ტექნიკური, ტექნოლოგიური და ინფორმაციული პროცესები. წინამდებარე ნაშრომში ჩვენ გვანტერესებს ის საერთო, ზოგადი, რაც ახასიათებს ამ ოფისებს. მაგალითად, მართვის ამოცანები: დაგეგმვა,

აღრიცხვა, ანალიზი, გადაწყვეტილების მიღება, და რაც მთავარია, მათი ინფორმაციული უზრუნველყოფის თვალსაზრისით.

განაწილებულ ბიზნეს-სისტემებში მიმდინარე პროცესების ავტომატიზაციის ინფრასტრუქტურის 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 არაფორმალური (ტექსტური) ინფორმაცია, რომელშიც ასახული იქნება მოქმედებათა ზოგადი წესები.



ნახ.2.12. საქმიანი პროცესების ზოგადი ანალიზური მოდელი

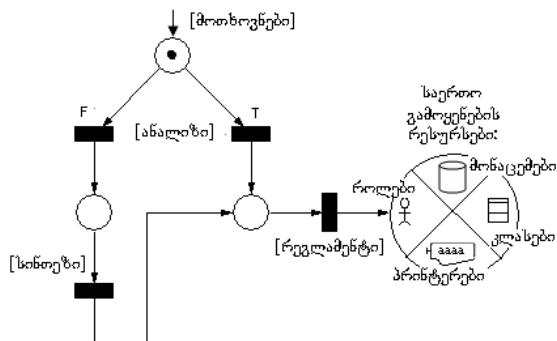
სტრატეგიის (Policy) ობიექტი წარმოადგენს მოდელის ძირითად რგოლს, მის ატრიბუტებს მიეკუთვნება დასახელება, რეალიზაციის თარიღი, მიზანი, პასუხისმგებელი, აღწერა და ტიპი. სტრატეგიის ობიექტები შეადგენს ორგანიზაციულ ერთეულს.

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 იყოს ერთი ან რამდენიმე ელექტრონული ფორმის შაბლონით. ელექტრონული ფორმები გამოიყენება ერთი ან რამდენიმე ელექტრონული ფორმის ეგზემპლარის მისაღებად. სისტემა შეიძლება შეიცავდეს ისეთ რთულ ელექტრონულ ფორმებს, რომლებიც განკუთვნილია პერსონალის სწავლებისათვის. ელექტრონული ფორმა შეიძლება შეიცავდეს ერთ ან რამდენიმე მოქმედებას, რომელიც უნდა შეასრულოს კონკრეტულმა თანამშრომელმა. ერთნაირი მოქმედებები შეიძლება ვრცელდებოდეს სხვადასხვა თანამდებობებზე ორგანიზაციის შიგნით (მაგალითად, ხარჯების უწყისის დამტკიცება) 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

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

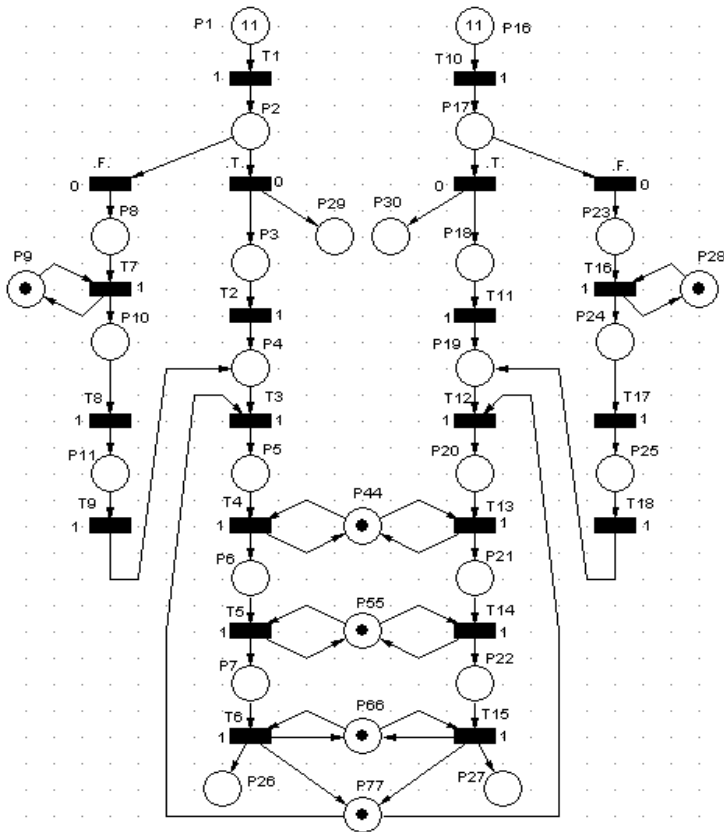
გადასასვლელი რეგლამენტი ასახავს კონკრეტული დოკუმენტის ფორმირების წესებს, სისტემის საერთო რესურსების მიმდევრობით-პარალელური ოპერაციების შესრულების ბაზაზე, მათი როლების, მონაცემების, პროგრამული მოდულებისა და სხვა ორგტექნიკის კოლექტიური გამოყენების თვალსაზრისით.

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 II-მომხმარებლისთვის.

2.15 ნახაზზე ილუსტრირებულია აღნიშნული პეტრის ქსელის იმიტაციური პროცესის ეტაპები. აქ მარკერების (მოთხოვნების) გადაადგილება ხორციელდება 1 და 16 პოზიციებიდან 26 და 27 პოზიციებამდე. მაგალითად, 1-ელ საწყის ეტაპზე 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 ღილაკებით) 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.2.14. დეტალიზებული პეტრის ქსელის ფრაგმენტი

ცხრ.2.3

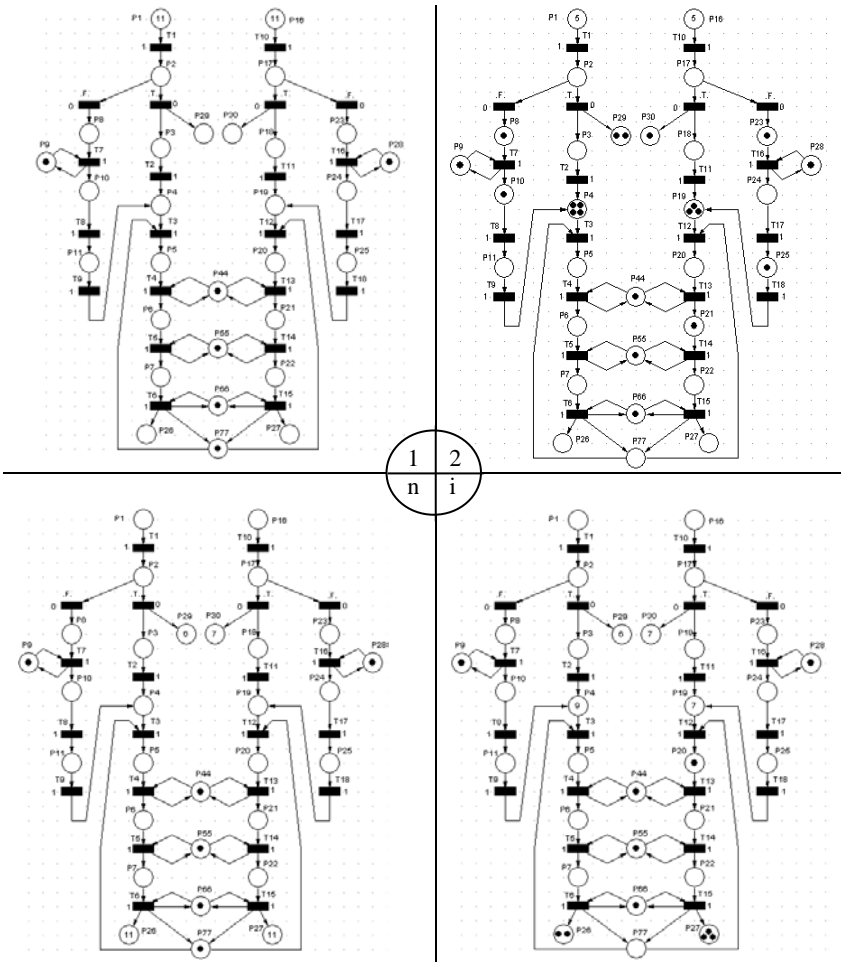
ცხრ.2.4

პოზიციების ცხრილი

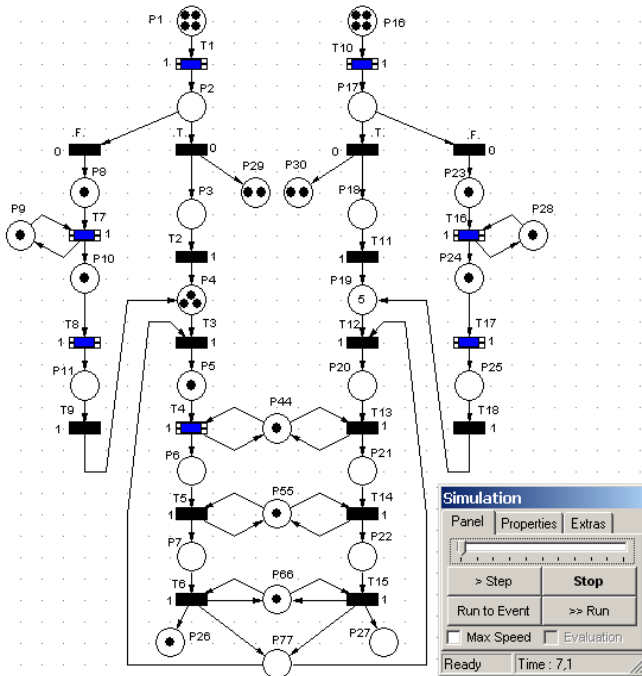
P _j	პოზიცია (მარკერით)
P _{1, 16}	მოთხოვნა არის (თუ მარკერია)
P _{2, 17}	დოკუმენტის ფორმა (კოდი)
P _{3, 18}	ცნობილი დოკუმენტის ფორმა
P _{4, 19}	დასამუშავებელი დოკუმენტები
P _{5, 20}	დოკუმენტი როლთან
P _{6, 21}	დოკუმენტი მონაცემთა რესურსით
P _{7, 22}	დოკუმენტი კლასთა რესურსით
P _{26,27}	დაბეჭდილი დოკუმენტი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

გადასასვლელების ცხრილი

T _i	გადასასვლელის ფუნქცია
T _{1, 10}	მოთხოვნის ანალიზი
T _{1, 10}	.T./ .F.
T _{2, 11}	დოკუმენტის ფორმის არჩევა
T _{3, 12}	ადამიანური რესურსით ფორმის შევსების წესის განსაზღვრა
T _{4, 13}	მონაცემთა რესურსის მიღება
T _{5, 14}	კლასთა რესურსის მიღება
T _{6, 15}	საბეჭდი მოწყობილობის რესურსის მიღება
T _{7, 16}	ატრიბუტთა სიმრავლის დადგენა
T _{8, 17}	რელაციათა სქემის სინთეზი
T _{9, 18}	რელაციათა სქემით დოკუმენტის ფორმის განსაზღვრა



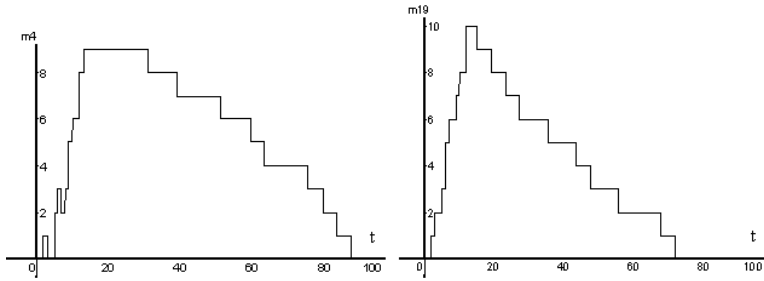
ნახ.2.15. პეტრის ქსელის სიმულატორით მიღებული ეტაპობრივი შედეგები



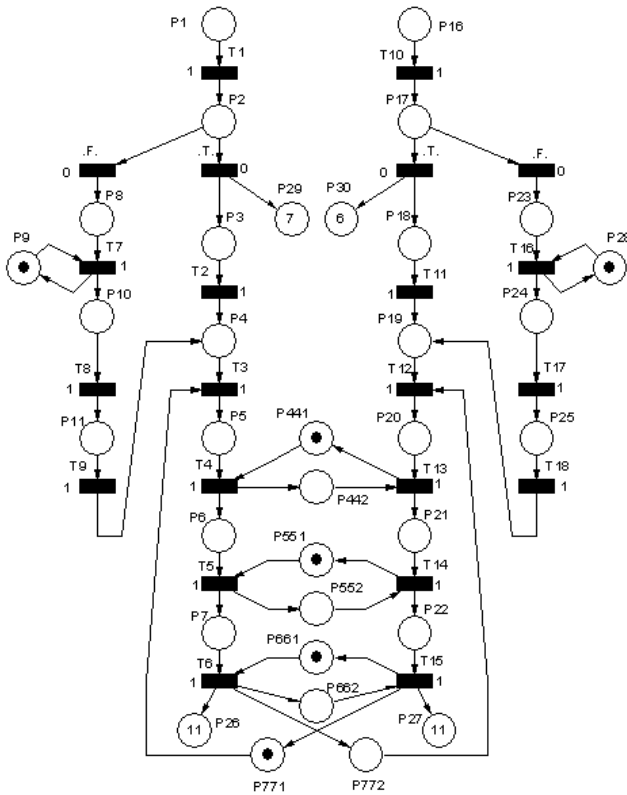
ნახ.2.16. იმიტაციის პროცესის ფრაგმენტი პეტრის ქსელით

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.

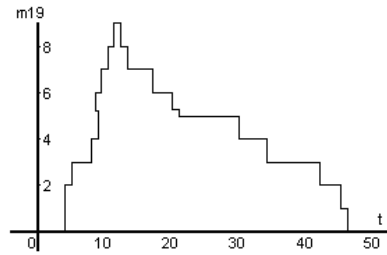
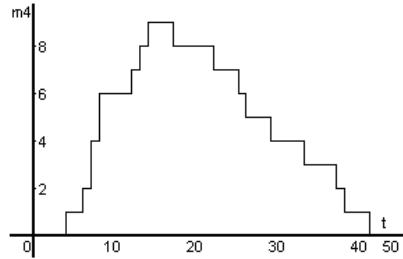
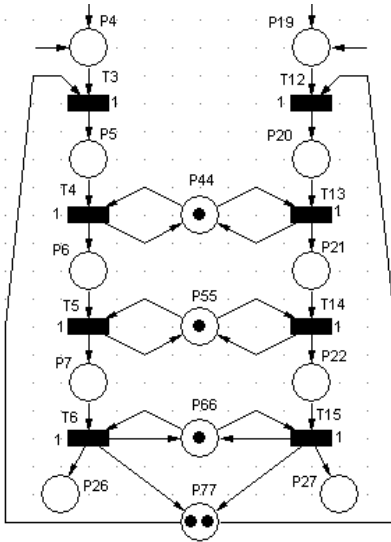
P_4 და P_{19} ის პოზიციებია, სადაც კონკრეტულ როლთან (ადამიანის რესურსი) თავს იყრის სხვადასხვა მოთხოვნები და საიდანაც უნდა მოხდეს მათი უზრუნველყოფა მონაცემთა ბაზებით, პროგრამებით და საბეჭდი მოწყობილობით. სქემის აღნიშნულ ფრაგმენტს განსაკუთრებული მნიშვნელობა აქვს მრავალმომხმარებლურ რეჟიმში მუშაობისას, როდესაც 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. P₄ და P₁₉ პოზიციების მდგომარეობათა ცვლილებები დროში



ნახ.2.18. რესურსების ტრივერული მართვის ფრაგმენტი



ნახ.2.21. P_4 და P_{19} პოზიციების მდგომარეობათა ცვლილებები დროში ადამიანური რესურსის გაორკეცვული სწრაფმედებით

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 ექსპერტული 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. 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 ჰეტრის ქსელებით საერთო რესურსების პირობებში.

III თავი ინფორმაციული უზრუნველყოფის დამუშავება MaSQL Server პაკეტით

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 გამოქვეყნებაში მდგომარეობს. აქტუალურია ასეთი პრობლემების გადასაწყვეტად კომპიუტერული მხარდაჭერი სისტემების აგება, რომელთა გული მონაცემთა საცავია, მრავალგანზომილებიანი, რელაციური მოდელის საფუძველზე დაპროექტებული, ხოლო გონება – მათი ოპერატიული ანალიზის სპეციალური მეთოდები (პროგრამები), რეალიზებული ობიექტ-ორიენტირებული დაპროგრამების UML–ტექნოლოგიით [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. აღნიშნული 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 მოსაწერებელი, რომელშიც თავმოყრილია არა მხოლოდ მიმდინარე, ოპერატიული ინფორმაცია, არამედ მასში ინახება აგრეთვე ისტორიული მონაცემებიც (შესაძლებელია მისი დაარსების დღიდან არსებული ინფორმაციული არქივების სახით). ამასთანავე მონაცემთა საცავის პროგრამული უზრუნველყოფა მოიცავს ოპერატიული 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 არამოდულიციურებას 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 – ესაა 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 განმასხვავებელი 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 თავის ოპერაციებზე და ღებულობს საშედეგო დოკუმენტებს და მოთხოვნებზე პასუხებს. ფილიალები უგზავნი 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

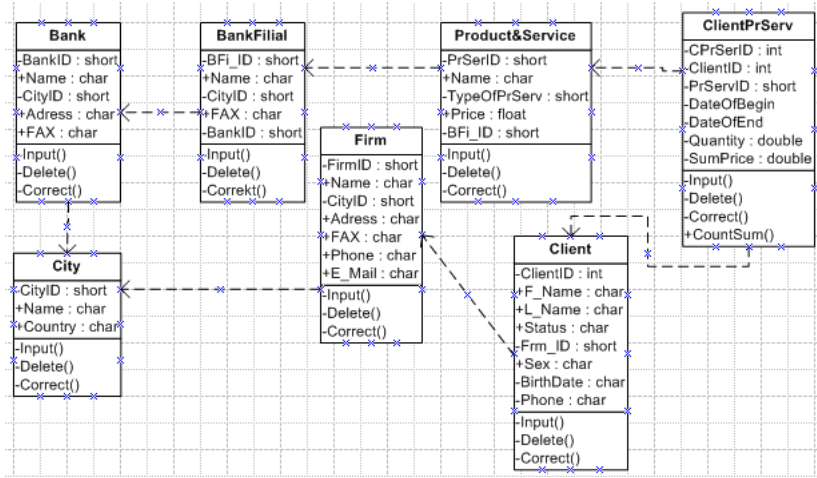
ტექნოლოგიების საშუალებებში, ბანკთაშორისი კონკურენციის გაზრდამ და გლობალური კომპიუტერული ტელეკომუნიკაციური ქსელების (პირველ რიგში Internet-ის) სწრაფმა განვითარებამ ხელი შეუწყო კლიენტების ელექტრონული მომსახურების სპექტრის გაფართოებას.

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 დასავგებმად, 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 „ვარსკვლავის“ ტოპოლოგიით (ნახ.3.2).

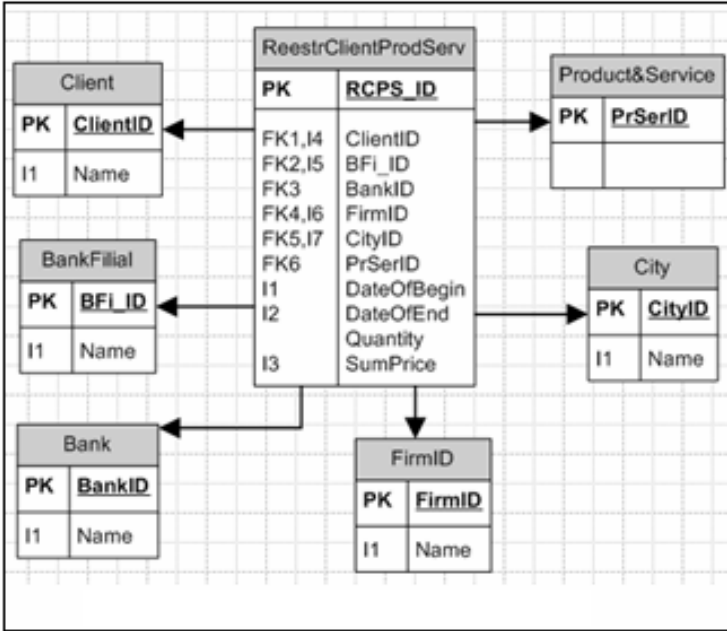


ნახ.3.2. ტოპოლოგია: ფიფქი და ვარსკვლავი

3.3 ნახაზზე 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 i -ური კლიენტის შემოსავლების ანგარიშის მეთოდით [30]. ჩვენი მოდელისთვის მას ექნება ასეთი სახე:

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

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 - ზედღებული ხარჯები კლიენტის მომსახურებიდან.

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

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

R- საერთო რისკია i-ური კლიენტის ყველა ოპერაციისათვის.

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

3.2. მონაცემთა საცავის დაპროექტება გადაწყვეტილებათა მიღების მხარდამჭერი 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 რევერსიულ (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-მოდელი (Entity-Relations Model), რომლის საფუძველზეც დაპროექტდება შემდგომში მონაცემთა ბაზის ლოგიკური სტრუქტურა [31].

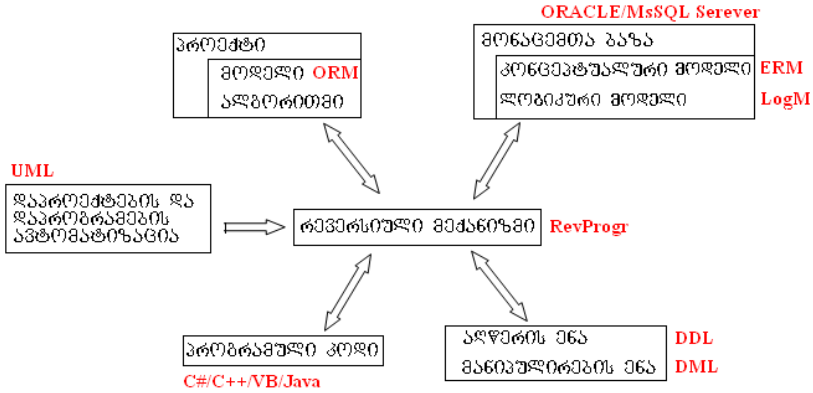
დაპროგრამების ენებთან შედარებით მბმს იყო პირველი ცდა პროგრამულ კოდში მონაცემთა სტრუქტურების აგების ავტომატიზაციისათვის. ობიექტ-ორიენტირებულ დაპროგრამების ენებში და UML-ტექნოლოგიაშიც ეს კონცეპტუალური ER-მოდელი, „კლასების დიაგრამების“ სახით, მნიშვნელოვანი ვიზუალური კომპონენტია. მომხმარებელი Use Case (გამოყენებითი 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 ამერიკისა და ევროპის თითქმის ყველა უნივერსიტეტსა და ბიზნესის მართვის სფეროში [54-:-60].

განსაკუთრებით საყურადღებოა ამ თვალსაზრისით უახლესი საინფორმაციო ტექნოლოგია, რომელიც .NET პლატფორმითაა ცნობილი. იგი აღჭურვილია ისეთი სპეციალური მედიატორული თვისებებით, რომლებიც უზრუნველყოფს შემოჩამოთვლილ დაპროგრამების ენებს შორის სრულ თავსებადობას [74-77,79,82,83].

წინამდებარე პარაგრაფის მიზანი სწორედ ამ ტექნოლოგიაზე მონაცემთა რელაციური ბაზების აგების (ORM/ERM – მოდელირების) ავტომატიზაციის საკითხები არის წარმოდგენილი (ნახ.3.4).

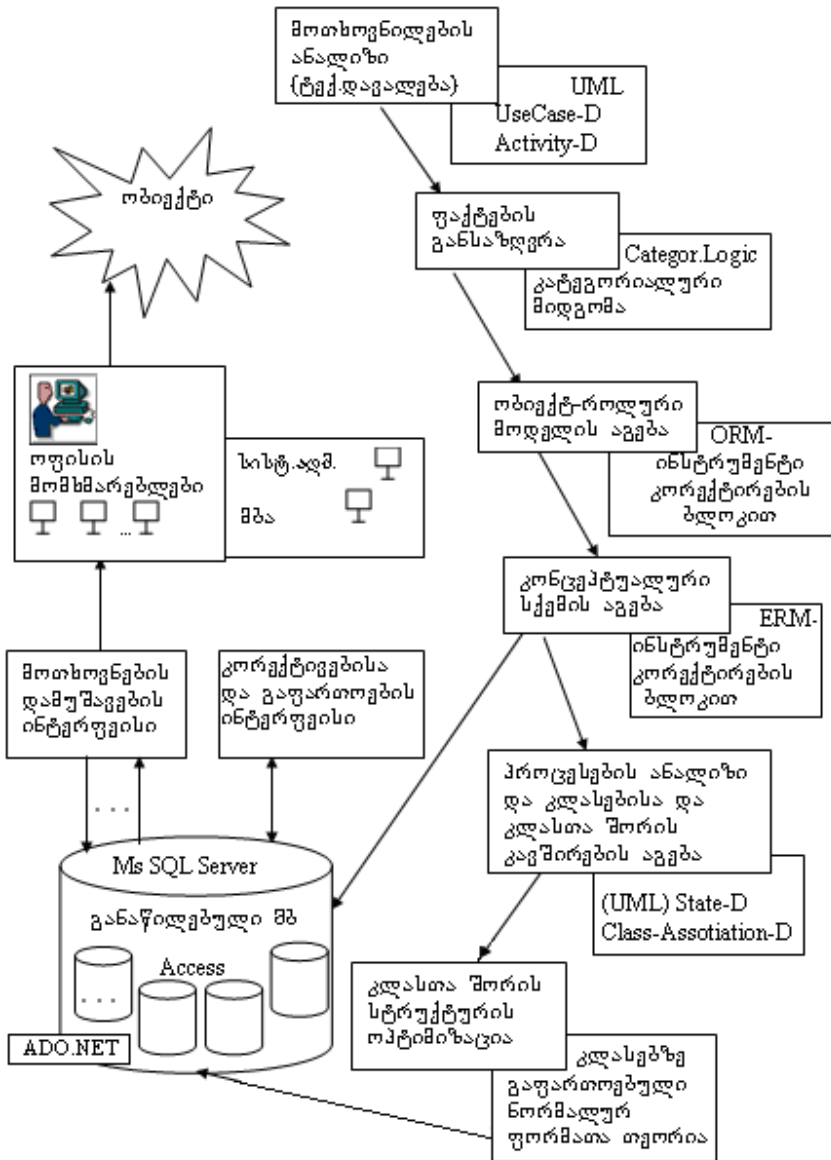


ნახ.3.4. მბ-ის დაპროექტების პროცესის რევერსიული მექანიზმის კომპონენტები

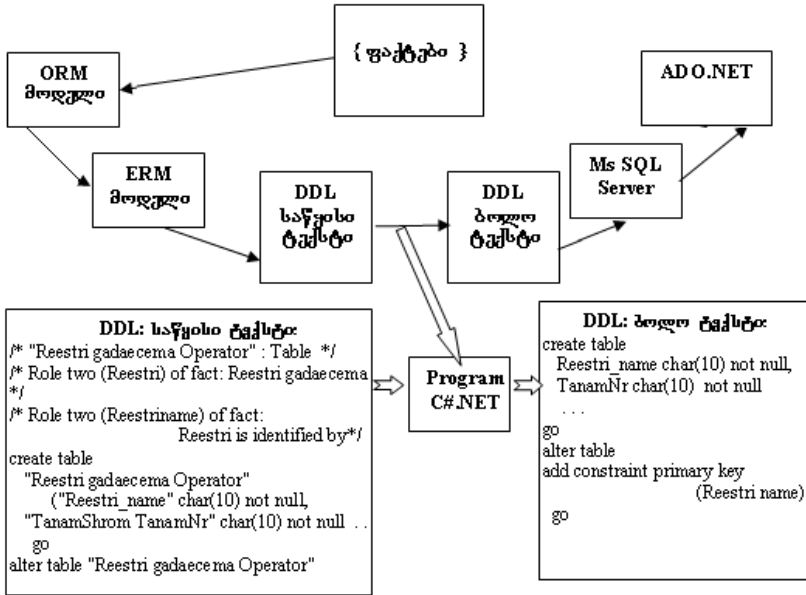
განიხილება განაწილებული ოფის-ობიექტების მონაცემთა ბაზების ობიექტ-ორიენტირებული მოდელირებისა და ობიექტ-ორიენტირებული დაპროექტების ავტომატიზებული პროცესების დამუშავების ამოცანა, რაც საგრძნობლად ამცირებს სისტემების ინფორმაციული და პროგრამული პაკეტების აგების დროს. 3.5 ნახაზზე მოცემულია საპრობლემო სფეროს მოდელირებისა და მონაცემთა ბაზაში ავტომატიზებული ასახვის ამოცანის ძირითადი 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-აღგებრული მეთოდები შეადგენს, გადაეცემა ობიექტ-როლური მოდელირების კომპიუტერულ პროგრამას [25].

3.6 ნახაზზე მოცემულია ERM-მოდელის შესაბამისი DDL-ფაილის ავტომატიზებული კორექტირების პროცედურები. შემდგომში ფაილი მიუერთდება MsSQL Server-მონაცემთა ბაზას (იხ. შემდეგი პარაგრაფი).



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-კომპონენტური, ობიექტ-ორიენტირებული მეთოდებითა და შესაბამისი ინსტრუმენტული საშუალებებით საგრძნობლად ამცირებს სისტემების რეალიზაციის დროს და ამაღლებს მისი ფუნქციონირების ხარისხს და მოქნილობას.

3.3. Ms SQL Server სისტემის არქიტექტურა

Microsoft ფირმამ Windows-2000 Server პლატფორმაზე შექმნა მონაცემთა განაწილებული რელაციური ბაზების მართვის სისტემა SQL Server. SQL Server მუშაობს აგრეთვე Unix, Linux, Macintosh და სხვა ოპერაციულ პლატფორმებთანაც, რაც მის უნივერსალურობასა და მოქნილობაზე მეტყველებს.

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

MsSQL Server შედგება ოთხი ძირითადი კომპონენტისგან:

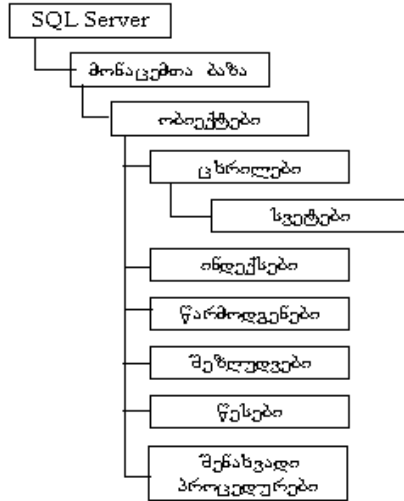
- Open Data Services SQL Server - უზრუნველყოფს ინტერფეისს ქსელურ ბიბლიოთეკებსა და თვით MSSQL Server-ის ბირთვის შორის;

- MSSQLServer - მართავს მონაცემთა ბაზის ყველა ფაილს, ამუშავებს მომხმარებელთა მოთხოვნებს, ანაწილებს სისტემურ რესურსებს, ამოწმებს მომხმარებელთა სააღრიცხვო ჩანაწერებს;

- SQLServer Agent - ახორციელებს დავალებათა დაგეგმვას და SQLServer მოვლენათა დამუშავების ავტომატიზაციას;

- MSDTC (Microsoft Distributed Transaction Coordinator) - როგორც განაწილებული ტრანზაქციების კოორდინატორი იგი მართავს მოთხოვნების შესრულებას მონაცემთა ბაზების რამდენიმე სერვერთან. MSDTC სერვისი შეიძლება ამუშავდეს როგორც SQL Server ბირთვიდან, ასევე კლიენტთა გამოყენებითი სისტემიდან.

SQL Server სისტემა მოთხოვნების დამუშავების SQL-ენის ნაცვლად იყენებს Transact-SQL დიალექტს. ეს არის მონაცემთა ბაზის ცხრილების, სვეტების, ჩანაწერების, ტრიგერებისა და შენახვადი პროცედურების შექმნის, მოდიფიკაციისა და წაშლის ენა. ამ ენის ინსტრუქციები დაყოფილია სამ ქვესიმრალედ: DDL - მონაცემთა ბაზების ცხრილებისა და წარმოდგენების შესაქმნელად, DML- მოთხოვნების შესაქმნელად და მონაცემთა დასამუშავებლად, DCL (Data Control Language) - მონაცემთა ბაზასთან მიმართვის პროცედურების სამართავად. 3.7 ნახაზზე მოცემულია SQL Server-ის არქიტექტურა.



ნახ.3.7

SQL Server სისტემის ინსტალირების დროს იქმნება ოთხი წვეილი სისტემური საბაზო ფაილი:

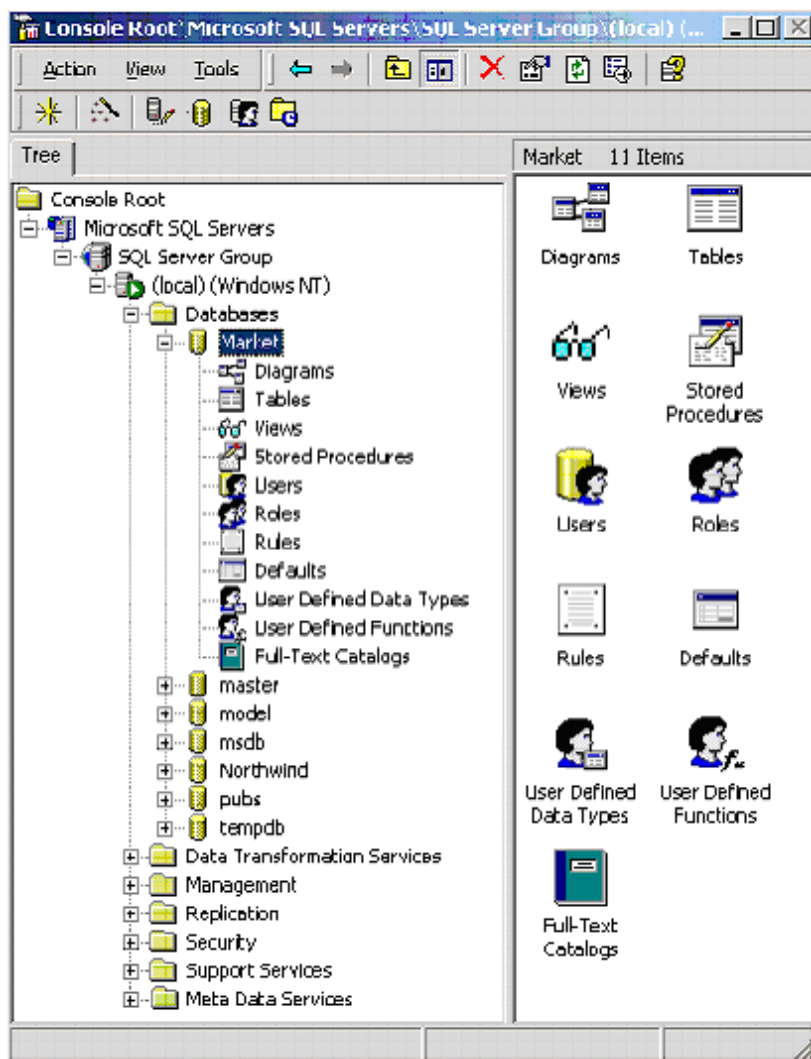
master.mdf - მონაცემთა ფაილი და mastlog.ldf - ტრანზაქციების ჟურნალის ფაილი. ამ ბაზებში ინახება SQL Server-ის კონფიგურაციისა და ფუნქციონირების შესახებ სრული ინფორმაცია. აქვეა მონაცემები სერვერის პარამეტრების, რეგისტრირებულ მომხმარებელთა და სისტემაში არსებულ სხვა ბაზების შესახებ.

model.mdf და modellog.ldf - ეტალონური (შაბლონური) მონაცემთა ბაზა, რომელიც გამოიყენება მომხმარებელთა ახალი ბაზების შესაქმნელად. იგი ავტომატურად გადასცემს ახალ ბაზას თავის პარამეტრებს (ცვლილებები დასაშვებია).

msdb.mdf და msdblog.ldf - ეს ბაზა შეიცავს ინფორმაციას დავალებების (jobs), მოვლენებისა (alerts) და ოპერატორების (operators) შესრულებათა მომდევრობების დასაგეგმად.

tempdb.mdf და tempdblog.ldf - ბაზაში ინახება დროებითი ცხრილები. იგი SQL Server-ის გლობალური რესურსია. მომხმარებლის მიერთებისას SQL Server-თან ყოველთვის იხსნება ეს ბაზა, მუშაობის დამთავრებისას კი იგი ავტომატურად წაიშლება.

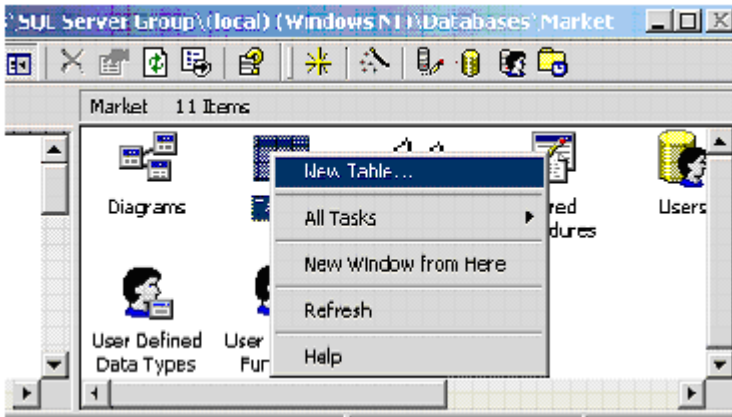
ახლა განვიხილოთ SQL Server-ის ძირითადი ობიექტები (ნახ.3.8). მათ საილუსტრაციოდ გამოვიყენებთ SQL Server Enterprise Manager უტილიტას (სერვისული პროგრამა).



ნახ.3.8

3.3.1. ცხრილები

Tables - ცხრილები მონაცემთა ბაზის ჩანაწერების (სტრიქონების) შესანახი კომპონენტია. იგი შედგება სახელმინიჭებელი ატრიბუტებისაგან (სვეტებისაგან), რომელთაც აქვს განსაზღვრული ტიპი და სივრძე. SQL Server-ში არსებობს სისტემური და მომხმარებლის ცხრილები. სისტემურის სახელები იწყება sys-ით. 3.9 ნახაზზე ნაჩვენებია ახალი ცხრილის შექმნის ინტერფეისი, ხოლო 3.10-ზე უკვე კონკრეტული მონაცემებით შევსებული „მწარმოებლის“ ცხრილის ფრაგმენტი.



ნახ.3.9. create New Table

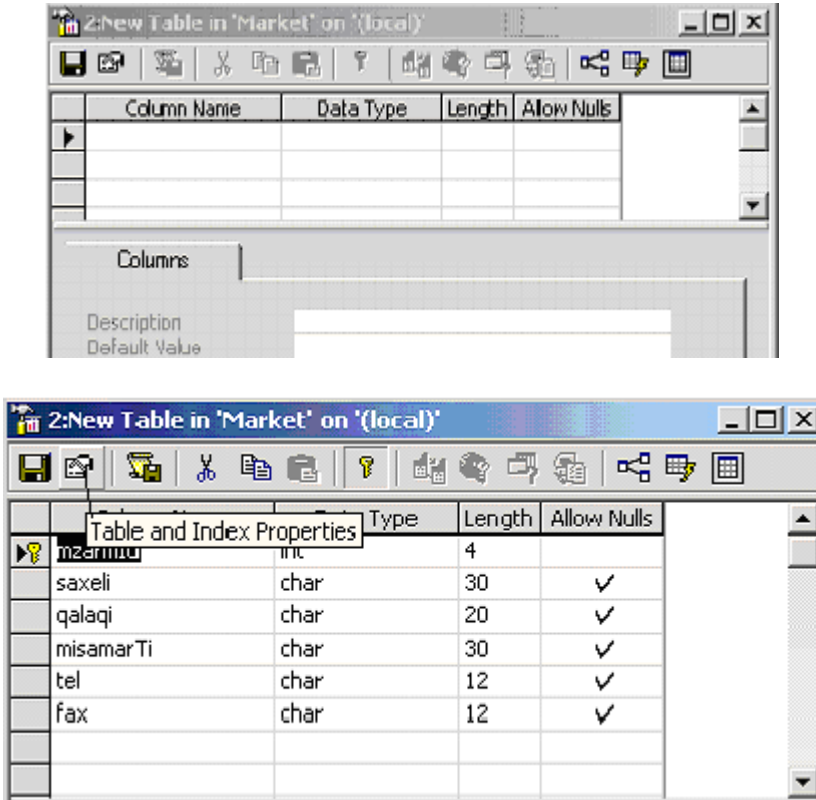
მწარმოებელი

	mzarmid	saxeli	qalagi	misamarTi	tel	fax
1		mzarmoebeli-1	Tbilisi	didi diRomi	34-20-20	34-20-21
2		mzarmoebeli-2	Gori	stalinis 79	2-22-22	<NULL>
3		mzarmoebeli-3	Tbilisi	muxiani	60-60-60	60-60-61
4		mzarmoebeli-4	baTumi	abaSiZis 100	6-66-66	6-66-99
*						

ნახ.3.10. Table

3.3.2. ატრიბუტები (სვეტები)

Columns - სვეტები, ანუ ცხრილის ველები (Column name), რომელთა მნიშვნელობები შეირჩევა მათი ტიპებისა (Data type) და სიგრძეების (Length) მიხედვით (ნახ.3.11).



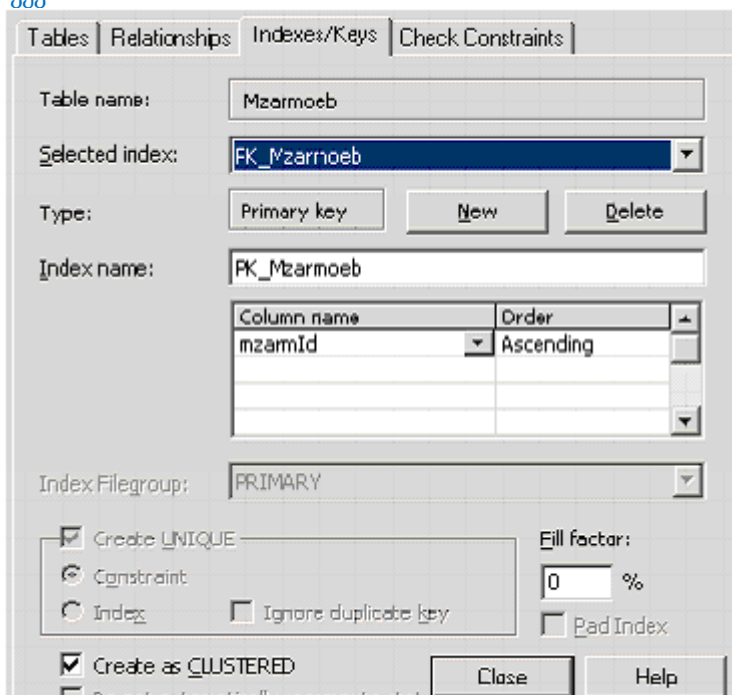
ნახ.3.11

SQL Server-ში მონაცემთა 30-მდე ტიპია. მაგალითად, int - მთელირიცხვა (4 ბაიტი), smallint - მთელირიცხვა (2 ბაიტი), real - ნამდვილირიცხვა (4 ბაიტი), float - ნამდვილირიცხვა (8 ბაიტი), money - ფულადი ტიპი (8 ბაიტი), char[n] - სიმბოლური

ფიქსირებული სიგრძით (n ბაიტით), varchar[n] - ცვლადი სიგრძის სიმბოლოები (n ბაიტით) და ა.შ. მონაცემთათვის კონკრეტული ტიპის შერჩევა შესაძლებელია Data type სვეტში "თავს" მარჯვენა ღილაკით მენიუს გამოძახებით.

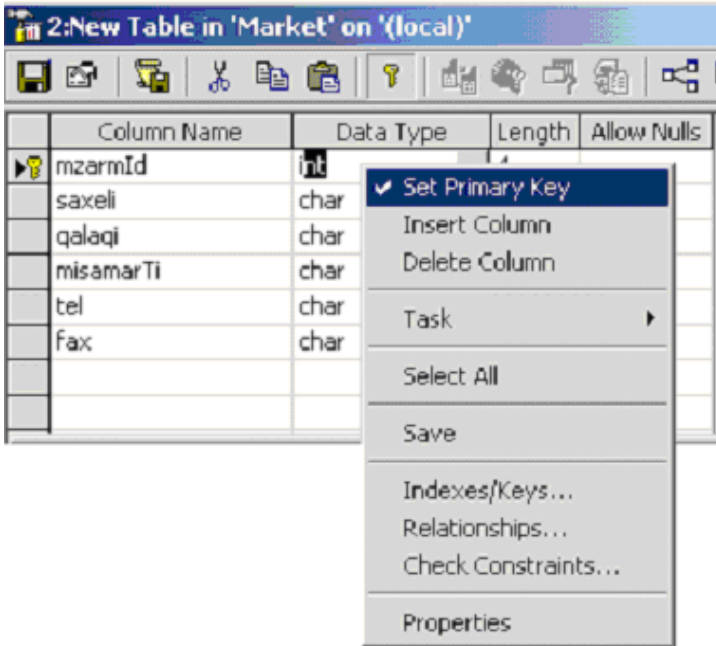
3.3.3. ინდექსები

Indexes - ინდექსები გამოიყენება დიდ ცხრილებში მონაცემთა მოწესრიგებული შენახვისათვის და შემდგომ ძებნის ოპერაციების დასაჩქარებლად. ერთ ცხრილს შეიძლება რამდენიმე სხვადასხვა ინდექსი ჰქონდეს. ფიზიკურად ესაა ცალკე ფაილი საკუთარი სახელით, რომელსაც კავშირი აქვს ძირითად ცხრილთან, სადაც მონაცემები ფიზიკურად ინახება. 3.12 ნახაზზე ნაჩვენებია ეს შემთხვევა.



ნახ.3.12

ინდექსი, რომელიც უნიკალურია ცხრილისათვის (ანუ მასში არ ხდება გასაღებური ველის მნიშვნელობის გამეორება), შეიძლება პირველად გასაღებად იქნეს არჩეული. ასეთია ჩვენს შემთხვევაში „მწარმოებლის იდენტიფიკატორი“ (mzarmid). 3.13 ნახაზზე მოცემულია გასაღებური ატრიბუტი.



ნახ.3.13

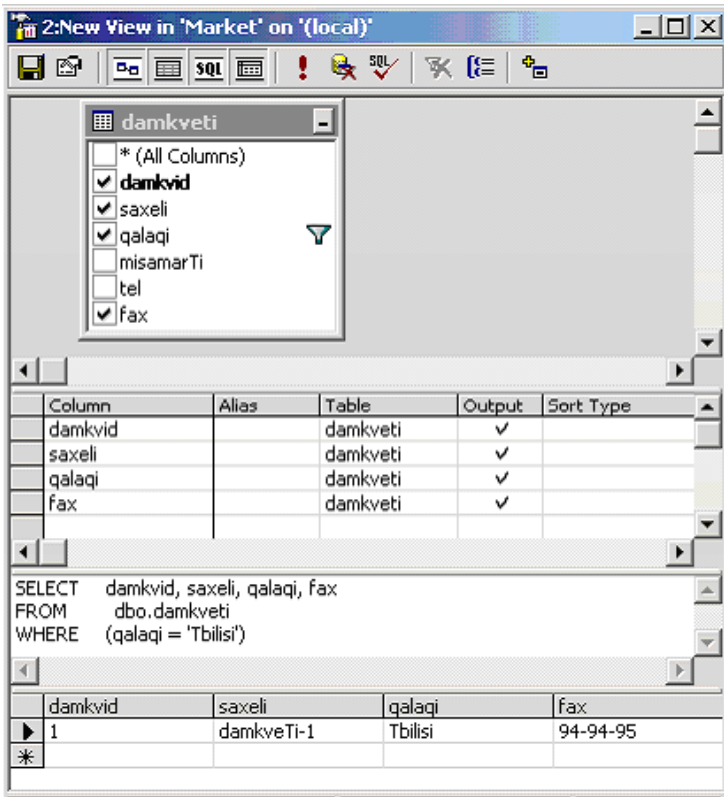
SQL Server-ში გამოიყენება ორი ტიპის ინდექსი: კლასტერული და არაკლასტერული. პირველის შემთხვევაში ცხრილი ფიზიკურად მოწესრიგდება ამ ინდექსით (ასეთია მაგ., გასაღებური ველი). ერთ ცხრილში დასაშვებია მხოლოდ ერთი კლასტერული ინდექსი.

არაკლასტერულია დანარჩენი ინდექსები (მაქსიმუმ 249), რომლებიც ამ ინდექსირებული ველის მოწესრიგებითაა დალაგებული, მაგრამ ძირითად ცხრილში ჩანაწერების მიმდევრობა არ იცვლება.

ინდექსური ველი შეიძლება შედგენილი იყოს ცხრილის რამდენიმე ველის სახელისგან. მაშინ ჩანაწერების მოწესრიგება ხდება ზრდადობით ან კლებადობით ამ ველების შესაბამისად. ჯერ ლაგდება პირველი ველით, შემდეგ მის შიგნით მეორით და ა.შ.

3.3.4. წარმოდგენები

Views - წარმოდგენები იქმნება შესაბამისი ცხრილ[ებ]იდან მომხმარებელთა მოთხოვნების საფუძველზე (მაგ., Select-ინსტრუქციით). მასში არაა ჩაწერილი რეალური მონაცემები, იგი ვირტუალური ცხრილია, სადაც შეიძლება გარკვეული მანიპულაციების ჩატარება მონაცემთა მიზნობრივი დამუშავებისთვის. 3.14 ნახაზზე ნაჩვენებია წარმოდგენის ფრაგმენტი ცხრილისათვის - „დამკვეთი“.



ნახ.3.14

შედეგში გამოსატანი ველები უნდა მონიშნოს ცხრილის მართკუთხედებში. ავტომატურად შეივსება ქვედა ცხრილი (Column, Alias, . . .), პარალელურად კი ფორმირდება SQL-მოთხოვნაც. აქ

შესაძლებელია მოთხოვნის კორექტირება ხელითაც, მაგ., ჩავამატოთ სტრიქონი WHERE qalaqi='Tbilisi'.

ამუშავება (Run) ხდება მენიუდან " ! " - პიქტოგრამით. საშედეგო წარმოდგენა მოცემულია ნახაზის ქვედა ცხრილში. ჩვენს შემთხვევაში ერთი სტრიქონია „დამკვეთი თბილისიდან“.

3.3.5. მთლიანობის შეზღუდვები

Constraints - შეზღუდვები მთლიანობაზე უზრუნველყოფს მონაცემთა მთლიანობას ცხრილების ან სვეტების დონეებზე. SQL Server-ში გამოიყენება ხუთი ტიპის შეზღუდვა:

- **PRIMARY KEY** : შეზღუდვა პირველად გასაღებზე. ცხრილის საზღვრებში პირველადი გასაღების მნიშვნელობა უნიკალურია (არ არსებობს მისი ორი ერთნაირი, ან Null-ის ტოლი მნიშვნელობა). ეს შეზღუდვა უზრუნველყოფს მონაცემთა ლოგიკურ მთლიანობას.

- **FOREIGN KEY** : შეზღუდვა მეორეულ გასაღებზე უზრუნველყოფს ცხრილთაშორისი კავშირების მთლიანობას. ეს კავშირები აიგება პირველადი და მეორეული გასაღებების ბაზაზე.

- **UNIQUE** : შეზღუდვა უნიკალურობაზე უზრუნველყოფს სვეტისათვის მნიშვნელობათა განუმეორებლობას. უნიკალურობა პირველადი გასაღების ფუნქციაა. შეიძლება ასევე უნიკალურობით ინდექსის შექმნა სხვა სვეტისათვისაც. მისთვის დასაშვებია Null მნიშვნელობაც.

- **CHECK** : შეზღუდვა მნიშვნელობაზე უზრუნველყოფს შესატანი მონაცემების კონტროლს. საკონტროლო მნიშვნელობათა დიაპაზონები წინასწარ განისაზღვრება სვეტებისათვის.

- **NOT NULL** : შეზღუდვა განუსაზღვრელ მნიშვნელობაზე უზრუნველყოფს სვეტისათვის არანულოვანი (განუსაზღვრელი) მნიშვნელობის თავიდან აცილებას.

Rules - წესები გამოიყენება ცხრილის სვეტებზე მნიშვნელობათა შეზღუდვების (CHECK) მსგავსად. ერთ (ან რამდენიმე) სვეტზე მხოლოდ ერთი წესია მიმავრებული. ყველა სვეტს შეიძლება საკუთარი წესი ჰქონდეს. CHECK-ით კი შეიძლება რამდენიმე შეზღუდვის გამოყენება ერთ სვეტზე. ამიტომაც რეკომენდებულია სისტემებში მისი ხმარება.

Defaults - მნიშვნელობები გამოუცხადებლად (ავტომატურად) მიენიჭება სვეტებს ცხრილის შექმნის დროს.

Trigger - ტრიგერი შენახვადი პროცედურაა, რომელიც სრულდება ავტომატურად SQL Server-ის ცხრილის განახლების დროს UPDATE, INSERT ან DELETE ინსტრუქციებით. ტრიგერების ინსტრუქციების ჩაწერა ხორციელდება Transact-SQL ენის ოპერატორთა ერთობლიობით.

ისინი გამოიყენება როგორც FOREIGN KEY-შეზღუდვები ცხრილთაშორისი კავშირების მთლიანობის უზრუნველსაყოფად, ოღონდ შედარებით რთული კავშირების აღსაწერად.

ტრიგერების ამუშავება დამოკიდებულია მონაცემთა მნიშვნელობებზე. მაგალითად, თუ ცხრილში მოხდება რაიმე მონაცემთა ცვლილება, ტრიგერს შეუძლია მისი დაფიქსირება ტრანზაქციების ჟურნალში.

3.3.6. შენახვადი პროცედურები

Stored procedure - შენახვადი პროცედურა არის Transact-SQL ენის ინსტრუქციების ერთობლიობა, რომელიც შექმნის დროს კომპილირდება სპეციალურ ფორმატში (შესრულების გეგმა).

ესაა მონაცემთა ბაზის ადმინისტრირების ძალზედ მოქნილი და ეფექტური საშუალება. მასზე შეიძლება ჩაიწეროს მონაცემთა დამუშავების რთული და მრავალფეროვანი ლოგიკური პროცესები.

შენახვადი პროცედურის შექმნის შემდეგ მისი შესაბამისი შესრულების გეგმა სისტემის მიერ განიცდის ოპტიმიზაციას გამოყენების ეფექტურობის ამაღლების თვალსაზრისით.

Extended stored procedures - გაფართოებადი შენახვადი პროცედურები იქმნება dll ფაილების სახით დაპროგრამების ენების საფუძველზე, მაგალითად Visual Basic, C++, C#, Java და სხვ.

მათი შექმნის შემდეგ ყოველი ფუნქცია უნდა დარეგისტრირდეს SQL Server-ში sp_addextendedproc შენახვადი პროცედურით.

3.4. მოთხოვნების დამუშავების ინსტრუმენტი Query Analyzer

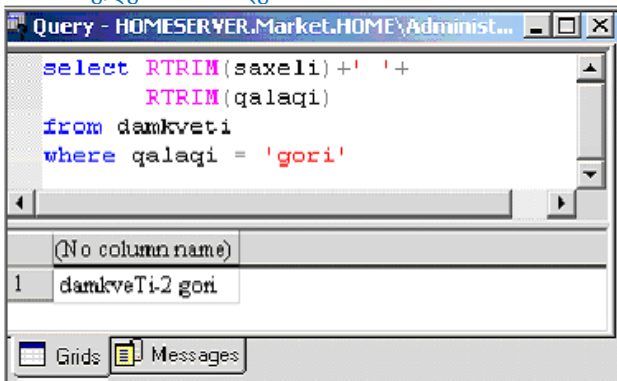
სერვისული პროგრამა Query Analyzer (მოთხოვნების ანალიზატორი) გამოიყენება მონაცემთა ბაზასთან ინტერაქტიულ რეჟიმში სამუშაოდ, კერძოდ, SQL-ინსტრუქციების ჩასაწერად, შენახვადი პროცედურების შესაქმნელად, მოთხოვნების შესრულების გასაანალიზებლად და მათი სტატისტიკის გასაცნობად.

იგი გამოიძახება შემდეგი სქემით:

Start | Programs | MS SQL Server | Query Analyzer.

ახლა განვიხილოთ კონკრეტული მოთხოვნები მონაცემების მისაღებად:

დავუშვათ, გვინტერესებს არის თუ არა ბაზაში პროდუქციის დამკვეთი რომელიმე კონკრეტული ქალაქიდან (მაგ., გორიდან). 3.15 ნახაზზე ილუსტრირებულია ამ SQL-მოთხოვნის ჩაწერის ფრაგმენტი. ვინაიდან შედეგში გამოიტანება ორი სტრიქონული ატრიბუტის (ველის) მნიშვნელობა სიმბოლოების რაოდენობის სხვადასხვა სიგრძით, სასურველია მათი ზედმეტი ცარიელი სიმბოლოების უგულებელყოფა. ამისათვისაა შემოტანილი RTRIM-ფუნქცია (მარჯვენა მხრიდან პრობლების წაშლა). პასუხში "დამკვეთი-2 გორი" ერთმანეთთან ახლოს განლაგდება. მოთხოვნის ტექსტში "+" "+" ნიშნავს ორი სტრიქონის კონკატენაციას (გადაბმას) მათ შორის სასურველი პრობლების რაოდენობით.



ნახ.3.15

მოთხოვნა: „დაკვეთილი პროდუქციის მიხედვით რამდენი კონტრაქტია გაფორმებული, რა ჯამური რაოდენობით და რა ჯამური თანხებით. შედეგები დაიბეჭდოს ჯამური თანხების მნიშვნელობების კლებადობით“.

ასეთი მოთხოვნების შესაბამისი SQL-ინსტრუქციები გამოიყენებს აგრეგატულ (ანუ სტატისტიკურ) ფუნქციებს (AVG - სვეტის მნიშვნელობათა საშუალო, COUNT - სტრიქონების რაოდენობა, MAX, MIN, SUM -ჯამი). გასაღებური სიტყვა DISTINCT <სვეტი> გამოიყენება აგრეგატულ ფუნქციებში განმეორებადი სტრიქონების გამოსარიცხად.

3.16 ნახაზზე ილუსტრირებულია ჩვენი მოთხოვნის შესაბამისი SQL-ინსტრუქცია აგრეგატული ფუნქციების გამოყენებით. ცხრილში სვეტების ახალი დასახელებების შემოტანა განხორციელებულია AS - კონსტრუქციით, მაგალითად, sum(DamkvProd.Tanxa) AS jami.

```

select Producti.dasaxeleba,
       count(DamkvProd.Tanxa) AS kontraqtebi,
       sum(DamkvProd.raod) AS dakveTis_raod,
       sum(DamkvProd.Tanxa) AS jami

from DamkvProd, producti

where DamkvProd.productid=producti.productid

group by Producti.dasaxeleba
order by sum(DamkvProd.Tanxa) Desc
    
```

	დასახელება	კონტრაქტები	დაკვეთის-რაოდ	ჯამი
1	პროდუქტი-1	3	1100	149500.0000
2	პროდუქტი-4	4	1750	146900.0000
3	პროდუქტი-2	2	1487	121525.0000
4	პროდუქტი-6	1	70	1890.0000
5	პროდუქტი-3	1	20	800.0000

ნახ.3.16

ვინაიდან ჯამი (sum) გამოითვლება თითოეული პროდუქტის შესაბამისად, ამიტომ აუცილებელია **group by** - წინადადების გამოყენება, რომელშიც მიეთითება დაჯგუფების კრიტერიუმი, ჩვენს შემთხვევაში "Producti.dasaxeleba". და ბოლოს, შედეგების სტრიქონები გამოიტანება ჯამური თანხების მნიშვნელობების მოწესრიგებით (კლებადობით).

3.5. მესამე თავის დასკვნები

3.1.

3.2.

3.3.

IV ტაპი

პროგრამული უზრუნველყოფის დამუშავება

4.1. გადაწყვეტილების მიღების მხარდამჭერ სისტემაში მრავალფაქტორული ამოცანის გადაწყვეტის პროგრამული რეალიზაცია

წინამდებარე პარაგრაფში განხილულია ბიზნეს-პროცესების ოპერატიული ანალიზის OLAP-ისტრუმენტის გამოყენების საკითხი [8]. შემოთავაზებულია განაწილებული მონაცემთა ბაზებიდან შერჩეული მონაცემების ერთიან გადაწყვეტილების მიღების მხარდამჭერ სისტემაში სტრუქტურული 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 შემოთავაზებულ იყო ცნობილი ამერიკელი მეცნიერის, ედვარდ 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 მოსახერხებელია 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.

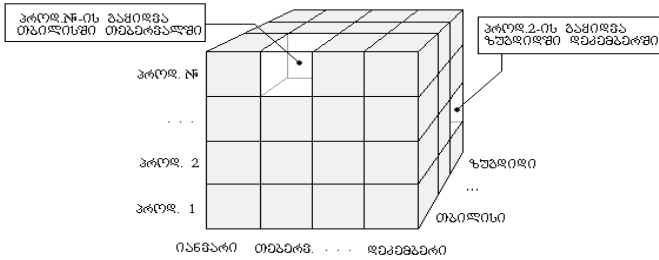
პირველ ეტაპზე ხდება ინფორმაციის თავმოყრა, რომელიც შეეხება საქონლის შემოსვლას, დატვირთვას, შეკვეთას, ანგარიშსწორებას და ა.შ. ამის შემდეგ უნდა შესრულდეს დოკუმენტბრუნვასთან დაკავშირებული ყველა პრობლემის გადაწყვეტა, ბოლოს კი დგება საკითხი, სისტემიდან საჭირო ინფორმაციის დროულად მისაღებად.

სისტემის მიმართ მოთხოვნა შესაძლოა გაუჩნდეს ორგანიზაციის ნებისმიერ თანამშრომელს, მაგალითად, თუ რა რაოდენობის პროდუქციაა დარჩენილი სარეალიზაციოდ, როგორ არის დაფუძვებული 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 წარმოგადგინოთ 4.1 ცხრილის სახით, რომელშიც არსებული მონაცემები განთავსდება სამგანზომილებიან კუბში:

ცხრ.4.1

ქალაქი	პროდ. დასახელება	იანვარი	თებერვალი	მარტი	აპრილი
თბილისი	უთო	10	22	15	47
	მტკვრსასრუტი	2	7	5	14
	ჩაიდან	17	34	20	71
ჯამი		29	63	40	132
გორი	მაცივარი	2	0	3	5
	ჩაიდან	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
	მტკვრსასრუტი	2	3	0	5
ჯამი		19	22	20	61

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



ნახ.4.1. მონაცემები სამგანზომილებიან კუბში

თუ მნიშვნელობათა განსაზღვრას ვაწარმოებთ ვერტიკალურად, მაშინ მივიღებთ ანგარიშთა 4.2 ცხრილს:

ცხრ.4.2

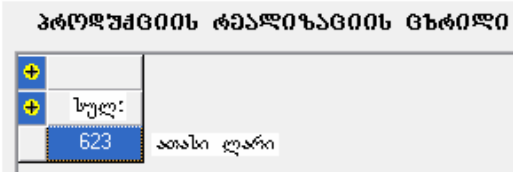
ქალაქი	იანვარი	თებერვალი	მარტი	აპრილი
თბილისი	29	63	40	132
გორი	19	28	13	60
ზუგდიდი	19	22	20	61
ჯამი	67	113	73	253

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.

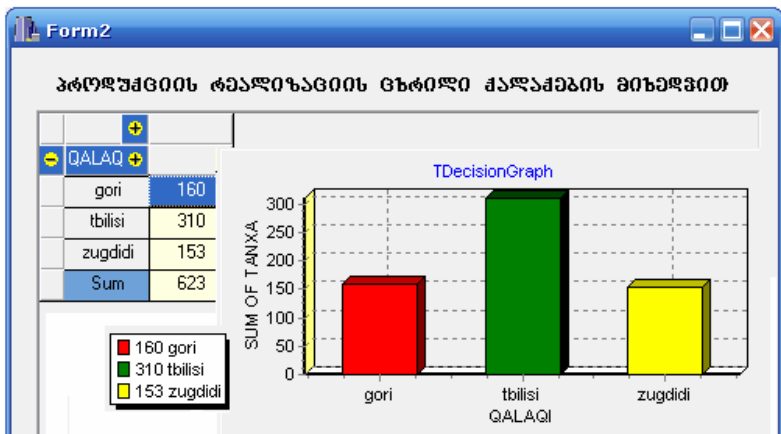
იმისათვის, რომ მივიღოთ კარგი შედეგი, აუცილებელია ეკრენზე გამოვიდეს არა მთლიანად კუბი არამედ მხოლოდ ის ნაწილი, სადაც წარმოდგენილია ჩვენთვის საჭირო ინფორმაცია, თუ არ გვინტერესებს კონკრეტული ინფორმაცია მაგ. რომელიმე ქალაქის შესახებ, სადაც აწარმოებენ პროდუქციის გაყიდვას თავიდანვე ამოვადგებთ განზომილებას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

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

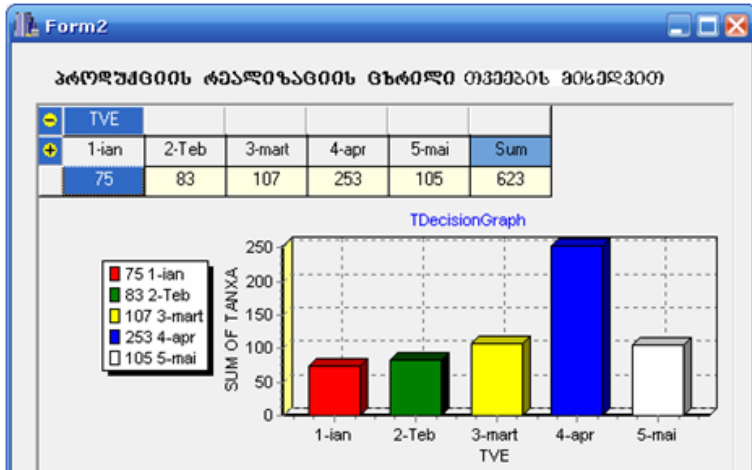
4.2 და 4.3 ნახაზებზე მოცემულია ჩვენი მაგალითისათვის რეალიზებული მომხმარებლის ინტერფეისის ფრაგმენტები პროდუქციის რეალიზაციის ჯამური თანხებისა და ამ თანხების ქალაქების მიხედვით გადანაწილების შესახებ. პროგრამირების საშუალო გარემოდ გამოყენებულია ვიზუალური, ობიექტ-ორიენტირებული დაპროგრამების პაკეტი Borland C++ Builder. „+“ სიმბოლო საშუალებას იძლევა ვმართოთ მრავალჯექტორიანი ანალიზის ცხრილები. მაგალითად, 4.4 ნახაზზე წარმოდგენილია თანხების განაწილება მხოლოდ თვეების მიხედვით, ხოლო 4.5 ნახაზზე ქალაქებისა და თვეების მიხედვით.



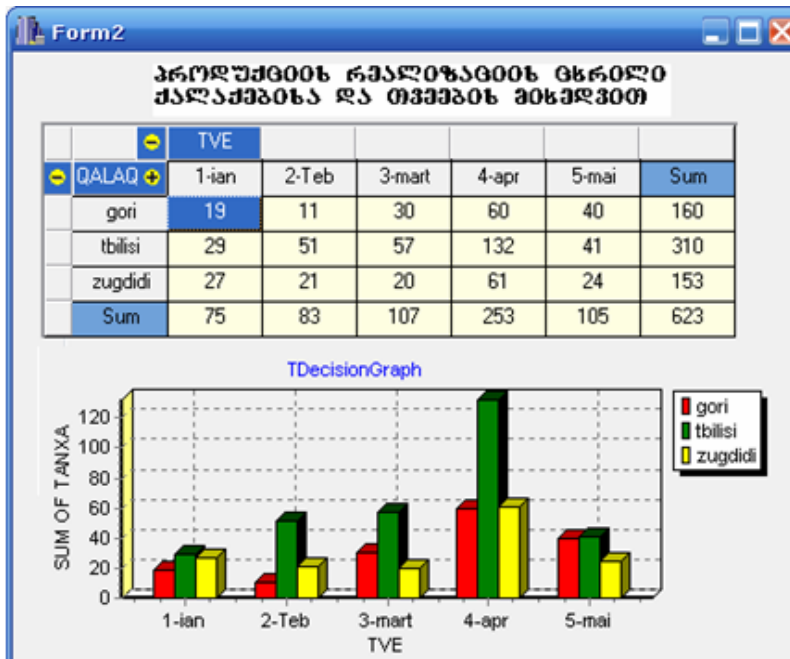
ნახ.4.2. ჯამური რეალიზაციის თანხები



ნახ.4.3.. თანხები ქალაქების მიხედვით

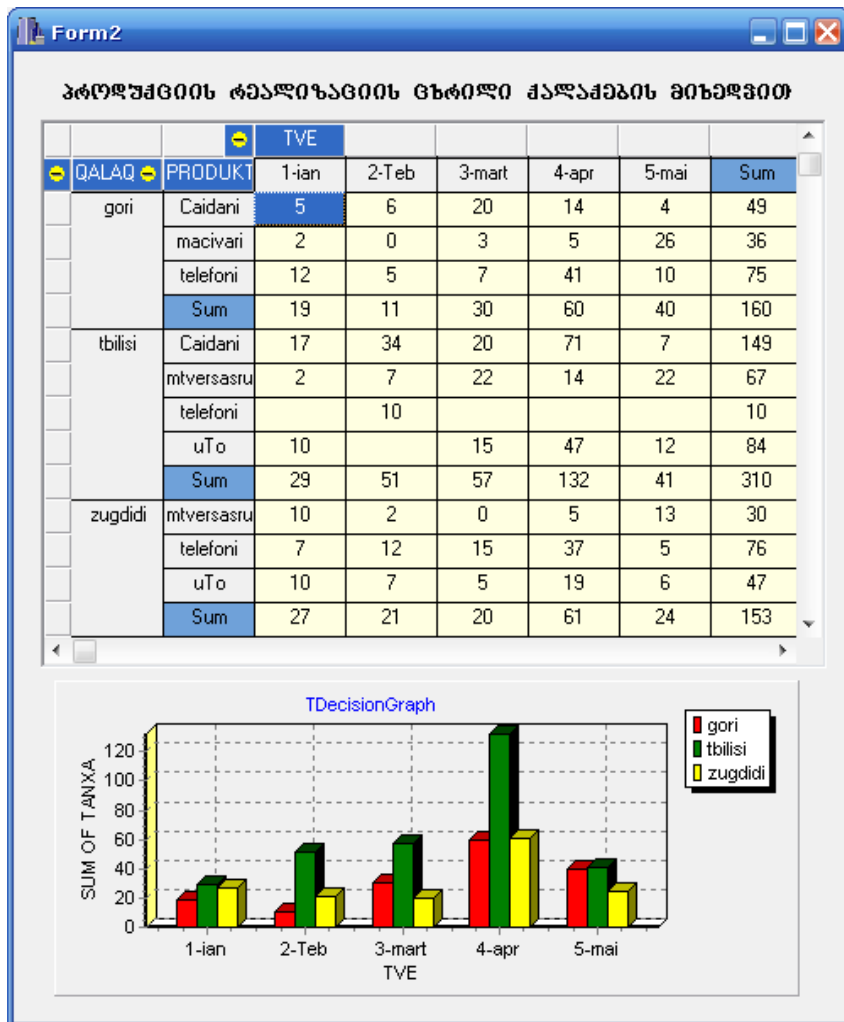


ნახ.4.4. პროექტის რეალიზაცია თვეების მიხედვით



ნახ.4.5. პროექტის რეალიზაცია ქალაქებისა და თვეების მიხედვით

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



ნახ.4.6. ცხრილი სამი ფაქტორის გათვალისწინებით

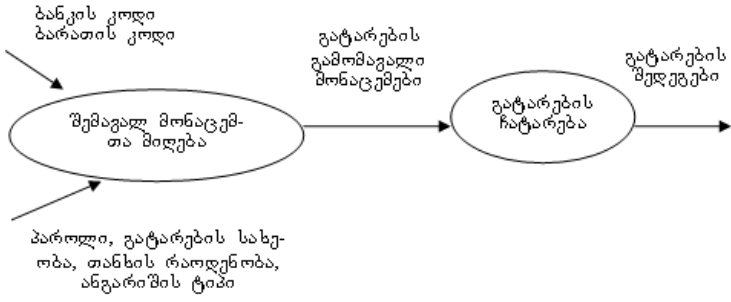
ამგვარად, OLAP-კონცეფცია არის ინფორმაციის მოპოვების ინსტრუმენტი, რომლის გამოყენება შესაძლებელია მრავალფაქტორიული ანალიზისათვის. ამ კონცეფციის რეალიზაცია ვიზუალური, ობიექტ-ორიენტირებული დაპროგრამების ინსტრუმენტების საშუალებით იძლევა მოქნილი და ეფექტური ინტერფეისი აგების საშუალებას მრავალფაქტორული ამოცანების გადასაწყვეტად, რომელთა ინფორმაცია განთავსებულია რელაციურ მონაცემთა ბაზებში.

4.2. ავტომატიზებული საბანკო სისტემის პროცესების ანალიზი და ინფორმაციული ნაკადების დიაგრამები

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

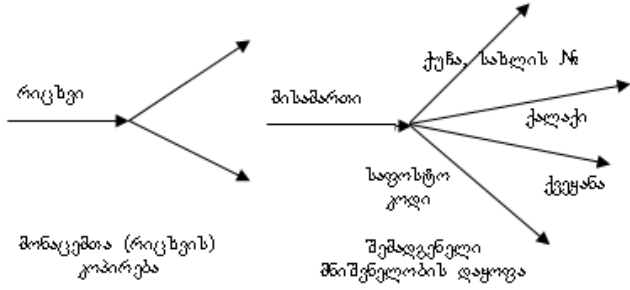
პროცესები: როგორც ზემოთ აღვნიშნეთ, პროცესები გარდაქმნის მონაცემთა მნიშვნელობებს. ყველაზე ქვედა დონის პროცესები წარმოადგენს ფუნქციებს გარე ეფექტების გარეშე. ასეთი ფუნქციების მაგალითად შეიძლება მოვიყვანოთ: „ორი რიცხვის ჯამის გამოთვლა“, „საბანკო ბარათით ჩატარებული გატარების საკომისიო ნაერთის გამოთვლა“. მონაცემთა ნაკადის მთლიანი გრაფი ასევე წარმოადგენს პროცესს, თუმცა უფრო მაღალი დონისას. პროცესს შეიძლება გააჩნდეს გარე ეფექტები, თუ იგი შეიცავს არაფუნქციურ კომპონენტებს, როგორცაა 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.4.7).



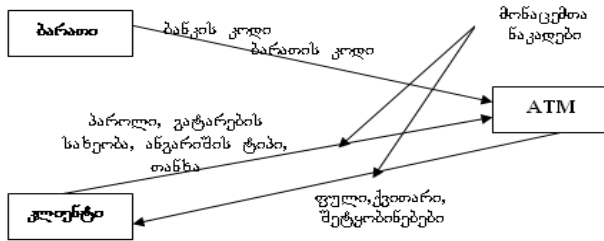
ნახ.4.7. პროცესთა ნაკადების დიაგრამის ფრაგმენტი

მონაცემთა ნაკადები: მონაცემთა ნაკადები აკავშირებს ერთ ობიექტს (ან პროცესს) მეორესთან. მათი მაგალითები მოცემულია 4.8 ნახ.აზზე. ისინი გამოისახება ისრების სახით და აკავშირებს ერთმანეთთან მონაცემების მწარმოებლებსა და მომხმარებლებს. პირველ მაგალითზე ნაჩვენებია მონაცემთა კოპირება, ორ ობიექტზე ერთიდაიმავე მნიშვნელობის გადაცემისას, მეორეზე – სტრუქტურის ველებად დაყოფა, სხვადასხვა ობიექტზე სხვადასხვა ველების გადაცემისას.



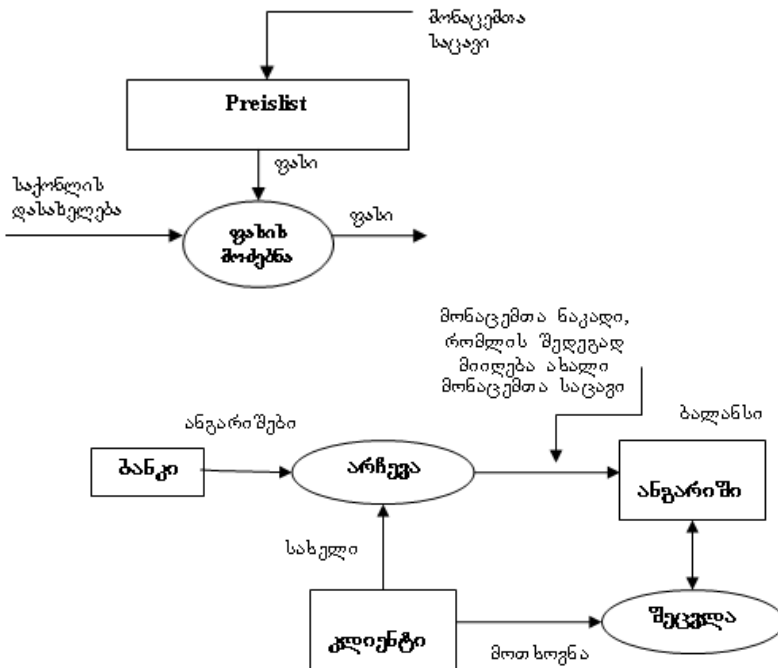
ნახ.4.8. მონაცემთა ნაკადების დიაგრამის ფრაგმენტი

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. 4.9 ნახაზზე ისინი გამოსახულია მართკუთხედებით.



ნახ.4.9. აქტიური ობიექტების დიაგრამის ფრაგმენტი

მონაცემთა საცავი: მონაცემთა საცავი არის პასიური ობიექტი მონაცემთა ნაკადების დიაგრამაზე, რომელშიც მონაცემები ინახება შემდგომ გამოყენებად. მისი მაგალითები მოცემულია 4.10 ნახაზზე. მონაცემთა აგრეგატული საცავები, როგორცაა მაგალითად სიები და ცხრილები, უზრუნველყოფს მონაცემთა შელწევას იგივე თანმიმდევრობით, როგორც ისინი იქნენ მიღებული.

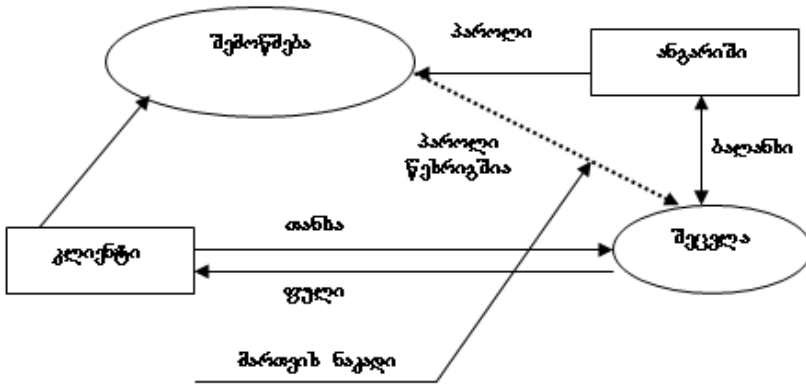


ნახ.4.10. მონაცემთა საცავის დიაგრამის ფრაგმენტები

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.

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



ნახ.4.11. 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 იქნას 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 4.12. ნახ.აზზე. ეფექტის აღწერისას გამოყენებულია ოპერაციები: „გატარების შეწყვეტა“, „მოთხოვნის გაცემა“, „ფულის გაცემა“, „ანგარიშის დებუტირება“, „ანგარიშის დაკრედიტება“.

ოპერაცია: „ანგარიშის შეცვლა“ (ანგარიში, თანხა, გატარების ტიპი)
შედგენი: ფული, ქვითარი

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

თუ თანხა იხსნება და იგი ნაკლებია ბალანსზე, მაშინ
“თანხის დებუტირება” 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.4.12. ოპერაციის აღწერის მაგალითი

ინფორმაციული ტექნოლოგიის ყველა ოპერაცია შეიძლება დაიყოს სამ კატეგორიად: მოთხოვნები, მოქმედებები და აქტიურობები.

მოთხოვნად ჩაითვლება ოპერაცია ობიექტის გარეგანი მასსიათებლებისათვის უკურეაქციის გარეშე.

მოქმედებად იწოდება ოპერაცია, რომელსაც გააჩნია სისტემის ობიექტზე მოქმედი გვერდითი მოვლენები. ყოველი მოქმედება განისაზღვრება 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.

4.3. კორპორაციული მართვის სისტემის Web-აპლიკაციის დაშუშავება 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 დაშუშავებულია .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 ოპერატიულად დაშუშავებისა 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 დაფუძნებული ინფორმაციული სისტემების დაპროექტების და მისი შემდგომი რეალიზაციის საკითხები.

ისეთ საფინანსო ორგანიზაციის ფილიალებში, როგორცაა 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 დამონტაჟებით 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 ამცირებს ადმინისტრირების ხარჯებს.

ვინდოუსის-პროგრამების შემთხვევაში სისტემის „მწყობრიდან გამოსვლის“ სისშირე 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 დროის გაზრდას. რადგან როდესაც პროგრამული უზრუნველყოფის ახალი ვერსია გამოდის საჭიროა მისი ყოველ

კომპიუტერზე ხელახალი დაინსტალირება, რაც თავისთავად გამოიწვევს დროის ხარჯვას.

ვებ-პროგრამის შემთხვევაში ყველა ეს პრობლემა იხსნება. რადგან ვებ პროგრამა ჩაწერილია სერვერზე, და პროგრამისტა ერთი ჯგუფივც 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 Dedeveloper 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 პროგრამების 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, გამოვიყენეთ აუტენტიფიკაციის მექანიზმი, რისთვისაც პირველ რიგში მომხმარებელი შეიტანს თავისი იდენტიფიკატორს და პაროლს.

სწორი აუტენტიფიკაციის შემდეგ მომხმარებელი მიიღებს სისტემის მთავარ გვერდს, რომელზეც გამოსახულია ვალუტის გაცვლის კურსი, მენიუ და ლილაკები (ნახ.4.13).

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

ინფორმაცია კლიენტზე

ინფორმაცია ანგარიშზე

you are logged in
[Logout](#)

ოპერაციები ▾

სალაროს ოპერაციები ▾

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.4.13. ინტერფეისის მთავარი გვერდი

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

თუ ოპერატორს სჭირდება ინფორმაცია კლიენტის ანგარიშების შესახებ, მას შეუძლია გადავიდეს გვერდზე „ინფორმაციის ანგარიში”, შესაბამისი სახელწოდების ლილაკით. 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.4.14).

ინფორმაცია ანგარიშზე

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

სახელი გვარი

ნახ.4.14. 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.4.15). გამოტანილი იქნება ინფორმაცია მითითებული ანგარიშის შესახებ. თანხის შეტანისას ველში “თანხა” ოპერატორმა უნდა მიუთითოს თანხის მნიშვნელობა და დანიშნულება.

Address <http://localhost:1891/Bank/Tanxis%20Setana.aspx>

ინფორმაცია კლიენტზე

ინფორმაცია ანგარიშზე

ანგარიშის ნომერი

you are logged in ოანხა

[Logout](#) დანიშნულება

ოპერაციები ▶

ხალაროს ოპერაციები ▶ თანხის შეტანა

თანხის გატანა

კონვერტიციები ▶

ნახ.4.15. 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.4.16).

ინფორმაცია ანგარიშზე

ანგარიშის ნომერი ანგარიშის ნახვა

თანხა

დანიშნულება

	AccountNumber	TypeofAccount	Balance	Overdraft	Loan	Currency	Name	Surname
Edit	333010525	Current	1700.0000	0	0	GEL	Irma	Berdzenishv

ნახ.4.16. თანხის შეტანის საბოლოო შედეგი

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

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

4.4. ავტომატიზებული ანალიზის სისტემის დაპროექტება სატენდერო კომისიის ექსპერტებისთვის

მოცემულ პარაგრაფში აღწერილია სატენდერო პროცესის ხელშემწყობი კომპიუტერული სისტემა, რომლის მიზანია სატენდერო კომისიის ექსპერტებისათვის აუცილებელი საინფორმაციო ბაზისა და მისი ოპერატიული 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.

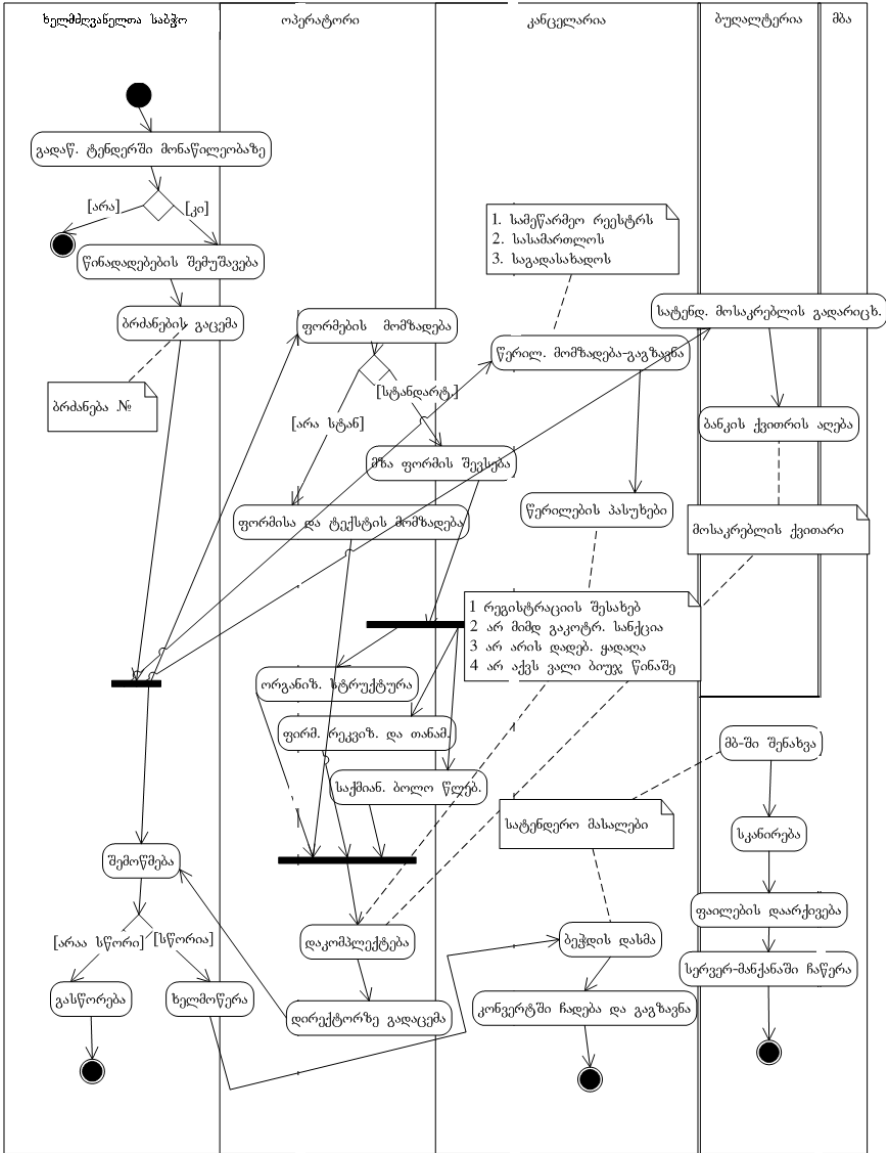
უმეტესად ასეთი საქმიანობით სხვადასხვა ტიპის სპეციალიზებული ფირმებია დაინტერესებული. ტენდერში მონაწილეობის მისაღებად აუცილებელია საჭირო იურიდიული დოკუმენტაციის შეკრება სხვადასხვა დაწესებულებებიდან, ასევე აუცილებელია სატენდერო მოსაკრებლის გადახდა ტენდერის მოთხოვნებში მითითებულ ბანკის ანგარიშზე. ტენდერში მონაწილეობის მიღების მიზნით შესასრულებელი სამუშაო დეტალურადაა წარმოდგენილი UML Activity-დიაგრამაზე (ნახ.4.17).

კონცეპტუალური მოდელის დასაპროექტებლად გამოყენებულია ობიექტ-როლური მოდელირება (ORM), რომელიც კონცეპტუალური მოდელირების განვითარებულ ტექნიკას წარმოადგენს.

ობიექტ-როლური მოდელირება მიახლოებულია ბუნებრივ სალაპარაკო ენასთან. ესაა მოდელირება ფაქტების საფუძველზე, სადაც საპრობლემო არე 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.



ნახ.4.17. სატენდერო პროცესის მომზადების აქტიურობის დიაგრამა

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.

სატენდერო ამოცანების გადაწყვეტის ხელშემწყობი კომპიუტერული სისტემის აგების მიზნით შეიძლება შემდეგი ზოგადი ფაქტების ჩამოყალიბება, რაც შეესაბამება ზემოთ ნახაზზე მოცემულ აქტიურობათა დიაგრამას:

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

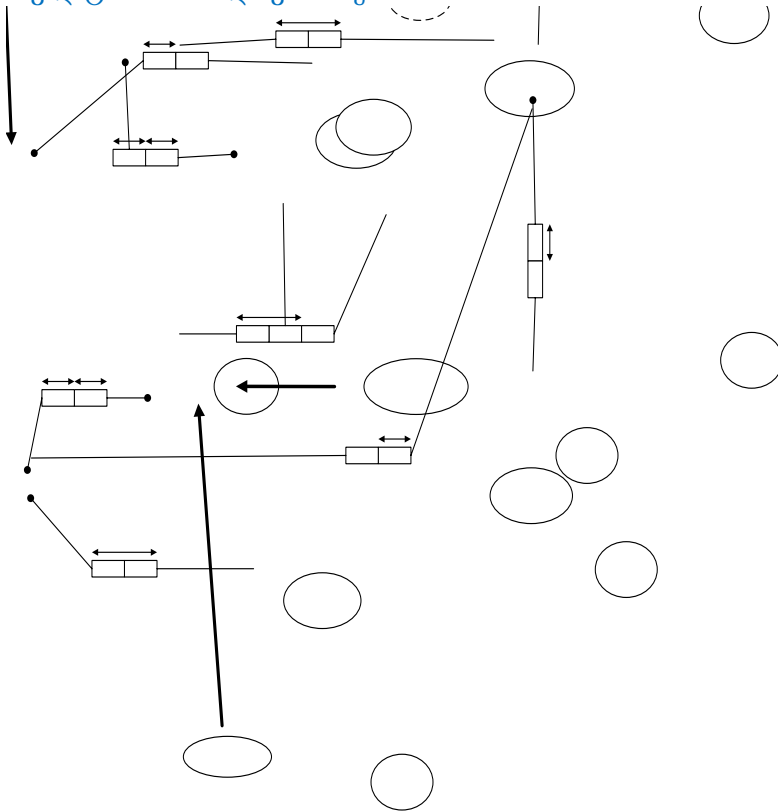
ტენდერის ჩატარების დროს ყველა მონაწილე ორგანიზაციის წარმომადგენელი, ვალდებულია გამოცხადდეს მითითებულ მისამართზე. სატენდერო კომისიის წევრები საჯაროდ გახსნიან დალუქულ კონვერტებს, შეამოწმებენ იურიდიულად საჭირო დოკუმენტაციის არსებობას და შეადგენენ ოქმს ტენდერის დაწყებისა და მასში მონაწილე ორგანიზაციების 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 არაფორმალური ადწერიდან ფორმალურზე გადასასვლელად არის საჭირო სწორედ ფაქტების გამოკვეთა,

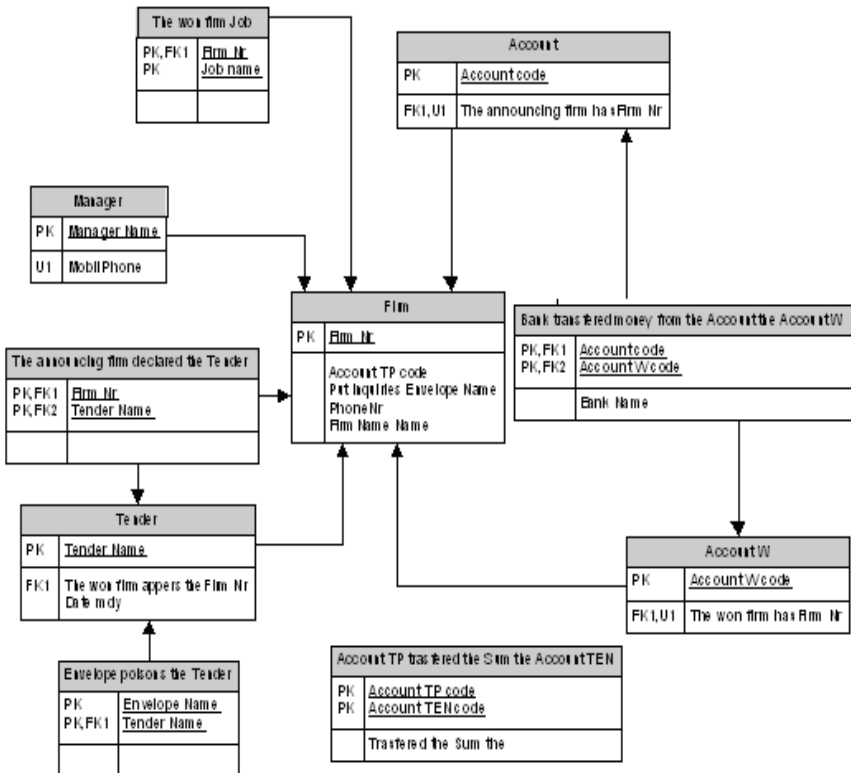
რომლებიც შემდგომ ობიექტ-როლური დიაგრამისთვის გამოგვადგება. 4.18 ნახაზზე მოცემულია „ტენდერის ჩატარების“ პროცესის ფაქტების გადატანა ORM-დიაგრამაზე.



ნახ.4.18. სატენდერო პროცესის ORM-დიაგრამა

Ms Studio.Net პროგრამული პაკეტი, კერძოდ ს Visio საშუალებას გვაძლევს დ-დიაგრამიდან ავტომატურად ავაგოთ საპრობლემო სფეროს კონცეპტუალური მოდელი, დ-დიაგრამა, რომელიც ჩვენი შემთხვევისთვის მოცემულია 4.19 ნახაზზე. ERM (Entity Relation Modeling)-ის საფუძველზეც აივება რელაციურ

მონაცემთა ბაზების ლოგიკური სტრუქტურა. მონაცემთა ბაზის დაპროექტების მიზნით ვიყენებთ SQL-სერვერს [31,38,83].



ნახ.4.19. „ტენდერის“ საპრობლემო სფეროს ER-მოდელი

4.20 ნახაზზე მოცემულია ტენდერის ჩატარების ხელშემწყობი 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-E0AD098E2F'

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

Design Table 'Prize list' in 'tender' on 'USER-E0AD098E2F'

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

ნახ.4.20. 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 ხარისხს.

4.5. მეოთხე თავის დასკვნები

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

2. გაანალიზებულია ავტომატიზებული საბანკო სისტემის პროცესები და ავებულია შესაბამისი ინფორმაციული ნაკადების დიაგრამები;

3. რეალიზებულია საბანკო კორპორაციული სისტემის Web-აპლიკაცია Internet-Intranet-გარემოში NET-პლატფორმაზე, რაც უზრუნველყოფს მოქნილი კლიენტ-სერვერული სისტემის ფუნქციონირებას;

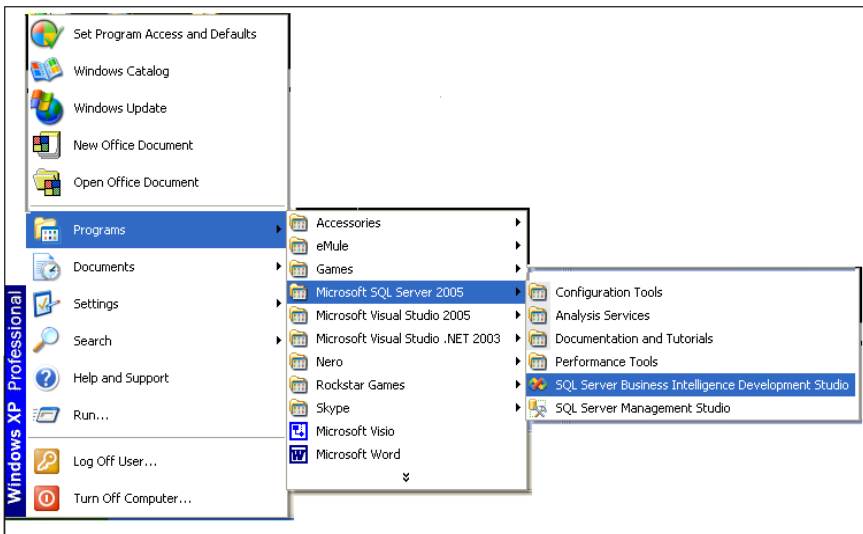
4. დაპროექტებული და რეალიზებულია ავტომატიზებული ანალიზის სისტემა სატენდერო კომისიის ექსპერტებისთვის

V თავი

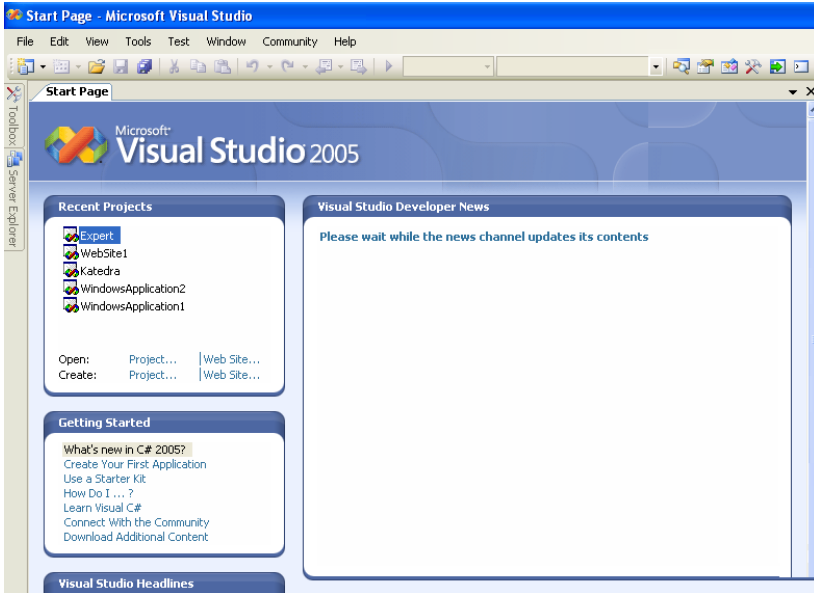
ექსპერტულ შეფასებათა ავტომატიზებული დამუშავების სისტემის რეალიზაცია

5.1. საბუთო გარემო: Visual Studio.NET პლატფორმა

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 (ნახ.5.1-5.), ბაზები და პროგრამული კოდის ლისტინგები:



ნახ.5.1. SQL Server 2005 -> SQL Server Business Intelligence Development Studio



ნახ.5.2. 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.Selected
            edValue),
            Convert.ToInt32(CBMizani.SelectedValue),
            Convert.ToInt32(CBKriteriomi.SelectedValue),
            Convert.ToInt32(CbFaqtori.SelectedValue),
            Convert.ToInt32(textBox1.Text));
                this.Close();
            }
            catch (Exception ex)
            {

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

        }

        private void bindingexpert()
        {

        }

        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.Fac
torebi);
            // 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.Mizne
bi);
            // 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
);

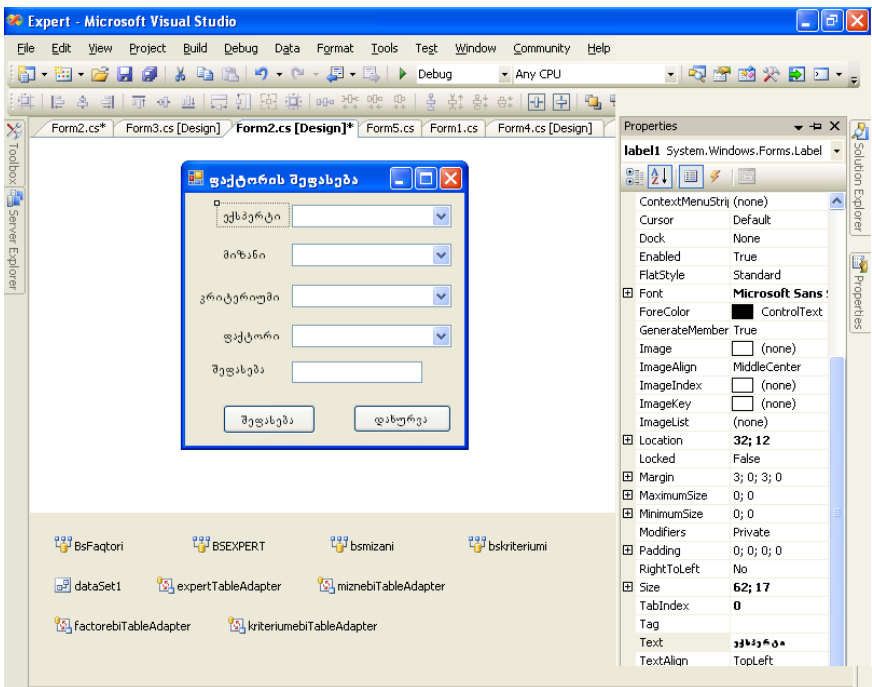
```

```

    }

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

```



ნახ.5.3-1. Form2.C# - პროგრამის შედეგი

საილუსტრაციოდ ვიხილავთ განათლების საუნივერსიტეტო სისტემის სასწავლო პროცესისი სრულყოფის პრობლემას და მისი გადაწყვეტის ერთ ვარიანტს ექსპერტულ შეფასებათა საფუძველზე (იხ. დანართი).

ფაქტორის შეფასება

ექსპერტი: სურგულაძე

მიზანი: მაღალკვალიფიციური ინჟი

კრიტერიუმები: ეკონომიკური

ფაქტორი: მატერიალურ-ტექნიკური შა

შეფასება:

შეფასება დახურვა

ფაქტორის შეფასება

ექსპერტი: სურგულაძე

მიზანი: მაღალკვალიფიციური ინჟი

კრიტერიუმები: ეკონომიკური

ფაქტორი: ტერიალურ-ტექნიკური შაზ

შეფასება:

შეფასება

- მატერიალურ-ტექნიკური შაზა
- კადრები
- ამიტური ინტენი
- სასწავლო პროცესის ორგანიზაც
- ლექტორთა პირობები
- მეცადინეობების ხარისხი

ნახ. 5.3-2. Form2.C# - პროგრამის შედეგი. ამორჩევის შესაძლებლობები

ფაქტორის შეფასება

ექსპერტი: სურველაძე

მიზანი: შონია

კრიტერიუმები:

- გაშედავა
- თოფურია
- სუხიაშვილი
- ვლემიანიშვილი**
- ფრანგიშვილი

ფაქტორი: მატერიალურ-ტექნიკური შა

შეფასება:

შეფასება დახურვა

ფაქტორის შეფასება

ექსპერტი: სურველაძე

მიზანი: კური ინჟინრემის გამოშვება

კრიტერიუმები:

- შაღალვეალიფიციური ინჟინრემ
- სწავლემის ხარისხის აშაღლება
- ლექტორთა სოციალური საკითხი**

ფაქტორი: მატერიალურ-ტექნიკური შა

შეფასება:

შეფასება დახურვა

ნახ. 5.3-2. Form2.C# პროგრამის შედეგი - გაგრძელება.

ლისტინგი-დ2

```
// Form1.cs --- ექსპერტის ინტერფეისი - ფაქტორის
// მიხედვით ღონიადიებათა 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.SpGetRelationTableAd
            apter();

            spGetDataTableBindingSource.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.SpGetGonisdziebaRelationTab
leAdapter gon = new
Expert.DataSet1TableAdapters.SpGetGonisdziebaRela
tionTableAdapter();

spGetGonisdziebaRelationDataTableBindingSource.Da
taSource = gon.GetData(HedId);
    }

    private void
spGetRelationDataTableBindingSource_CurrentItemCh
anged(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)spGetGonisdziebaRelationDataTabeBi
            ndingSource.Current).Row as
            DataSet1.SpGetGonisdziebaRelationRow;
            Form3 fr = new Form3(row.RowId,
            true, row1.GonisdziebaId, row1.shefaseba,
            row1.RowId);

            fr.ShowDialog();
            this.bindingdetail(row.RowId);
        }

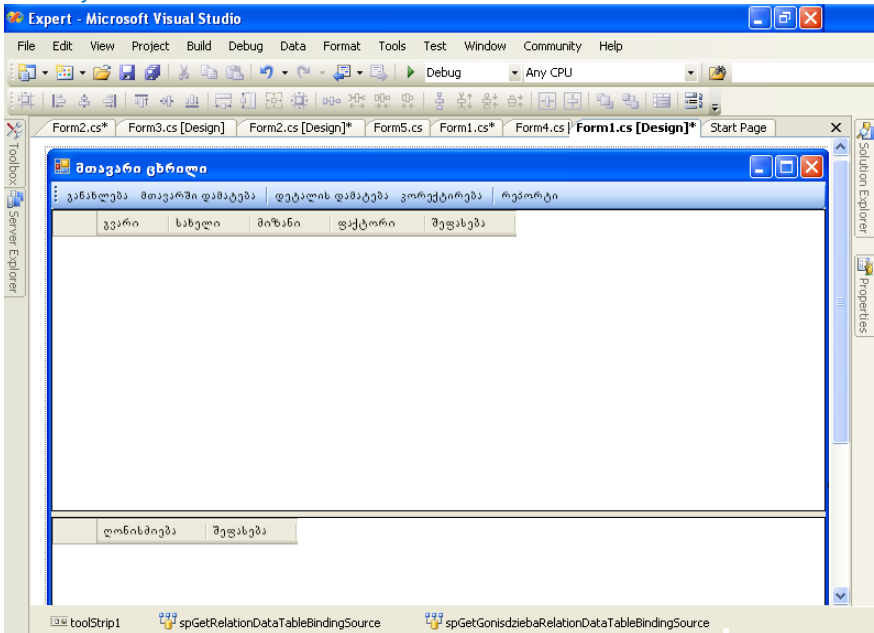
        private void toolStripButton3_Click(object
        sender, EventArgs e)
        {
            Form4 fr = new Form4();
            fr.ShowDialog();
        }
    }

```

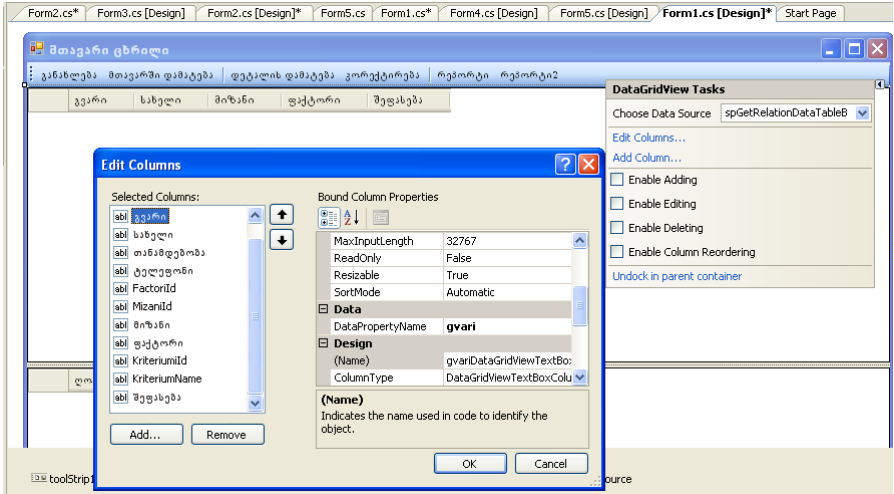
```

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

```



ნახ.5.4. Form1.C# - პროგრამის შედეგი



ნახ.5.5-1. ცხრილის სვეტებში ველების რედაქტორი

გვარი	სახელი	მიზანი	დეტალი	შეფასება
სურგულაძე	გია	მაღალკვალიფიციური ინჟინრების გამოშვება	მატერიალურ-ტექნიკური ბაზა	20
სურგულაძე	გია	მაღალკვალიფიციური ინჟინრების გამოშვება	აბიტურიენტები	10
სურგულაძე	გია	მაღალკვალიფიციური ინჟინრების გამოშვება	კადრები	20
სურგულაძე	გია	მაღალკვალიფიციური ინჟინრების გამოშვება	სასწავლო პროცესის ორგანიზაცია	20
სურგულაძე	გია	მაღალკვალიფიციური ინჟინრების გამოშვება	ლექტორთა პირობები	10
სურგულაძე	გია	მაღალკვალიფიციური ინჟინრების გამოშვება	შეკადინებების სარისხი	20
ლონისძიება				შეფასება
▶ ახალი ტექნიკის შემენა				40
ახალი ტექნოლოგიის დანერგვა				40
აუდიტორიების აღჭურვა მულტიმედიალური ტექნიკით				20

ა)

ნახ.5.5-2. ა-ვ) ფაქტორით ღონისძიებების არჩევისა და შეფასებების შეტანის ინტერფეისები

მთავარი ცხრილი					
გვარი	სახელი	მიზანი	ფაქტორი	შეფასება	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	მატერიალურ-ტექნიკური ბაზა	20	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	ამიტურიენტები	10	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	ვადრები	20	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	სასწავლო პროცესის ორგანიზაცია	20	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	ლექტორთა პირობები	10	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	შეცადინეობების ხარისხი	20	
ლონისძიება				შეფასება	
მაღალქულიანი ამიტურიენტების მოზიდვა				70	
ამიტურიენტების საწყისი ტესტირების ჩატარება				30	

ბ)

მთავარი ცხრილი					
გვარი	სახელი	მიზანი	ფაქტორი	შეფასება	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	მატერიალურ-ტექნიკური ბაზა	20	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	ამიტურიენტები	10	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	კადრები	20	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	სასწავლო პროცესის ორგანიზაცია	20	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	ლექტორთა პირობები	10	
სურგულაძე	გია	მალაღვკალიფიციური ინჟინრების გამოშვება	შეცადინეობების ხარისხი	20	
ლონისძიება				შეფასება	
ლექტორთა კვალიფიციაციის ამაღლება საზღვარგარეთ				60	
ლექტორთა კვალიფიციაციის ამაღლების ფაქულტეტის შექმნა				40	

გ)

ნახ.5.5-2. ინტერფეისები (გაგრძელება)

მთავარი ცხრილი					
განახლება	მთავარში დამატება	დეტალის დამატება	კორექტირება	რეპორტი	რეპორტი2
გვარი	სახელი	მიზანი	ფაქტორი	შეფასება	
სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	მატერიალურ-ტექნიკური ზაზა	20	
სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	აბიტურიენტები	10	
სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	კადრები	20	
▶ სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	სასწავლო პროცესის ორგანიზაცია	20	
სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	ლექტორთა პირობები	10	
სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	მეცადინეობების სარისხი	20	
ლონისძიება				შეფასება	
▶ ევროპული დონის სასწავლო პროგრამების შედგენა				20	
კორუფციასთან ბრძოლა				15	
ლექტორთა კვალიფიკაციის ამაღლების ფაკულტეტის შექმნა				10	
წარმოებიდან მალაღვალიფიციურ ლექტორთა მოწვევა				25	
სრული მონიტორინგის განხორციელება სასწავლო პროცესზე				10	
სტუდენტთა შეფასების ტესტირების ფორმების დანერგვა				20	

ღ)

მთავარი ცხრილი					
განახლება	მთავარში დამატება	დეტალის დამატება	კორექტირება	რეპორტი	რეპორტი2
გვარი	სახელი	მიზანი	ფაქტორი	შეფასება	
▶ განახლება	სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	მატერიალურ-ტექნიკური ზაზა	20
	სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	აბიტურიენტები	10
	სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	კადრები	20
	სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	სასწავლო პროცესის ორგანიზაცია	20
▶	სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	ლექტორთა პირობები	10
	სურველამე	გია	მალაღვალიფიციური ინჟინრების გამოშვება	მეცადინეობების სარისხი	20
ლონისძიება				შეფასება	
▶ ლექტორთა ხელფასების მომატება				60	
საათობრივი დატვირთვის ანაზღაურების ამაღლება				40	

ე)

ნახ.5.5-2. ინტერფეისების (გაგრძელება)

გვარი	სახელი	მიზანი	ფაქტორი	შეფასება
სურგულაძე	გია	მაღალეკალიფიციური ინჟინრების გამოშვება	მატრიალურ-ტექნიკური მუშა	20
სურგულაძე	გია	მაღალეკალიფიციური ინჟინრების გამოშვება	აბიტურიენტები	10
სურგულაძე	გია	მაღალეკალიფიციური ინჟინრების გამოშვება	კადრები	20
სურგულაძე	გია	მაღალეკალიფიციური ინჟინრების გამოშვება	სასწავლო პროცესის ორგანიზაცია	20
სურგულაძე	გია	მაღალეკალიფიციური ინჟინრების გამოშვება	ლექტორთა პირობები	10
სურგულაძე	გია	მაღალეკალიფიციური ინჟინრების გამოშვება	მეცადინეოების ხარისხი	20

ლონისძიება	შეფასება
მეცადინეოების სასწავლო-მეთოდური დონის ამაღლება	45
ლექტორთა მეცადინეოებზე ურთიერთდასწრება	15
სტუდენტთა მიერ ლექტორთა ანონიმური შეფასების დანერგვა	15
სტუდენტთა შეფასების ტესტირების ფორმების დანერგვა	25

3)

ნახ. 5.5-2. 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 ROWid)
        {
            InitializeComponent();
            relid = relationid;
            editstatus = status;
            _gon = gonisdieba;
            shef = shefaseba;
            _rowid = ROWid;
        }

```

```

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();
        }
    }

```

```

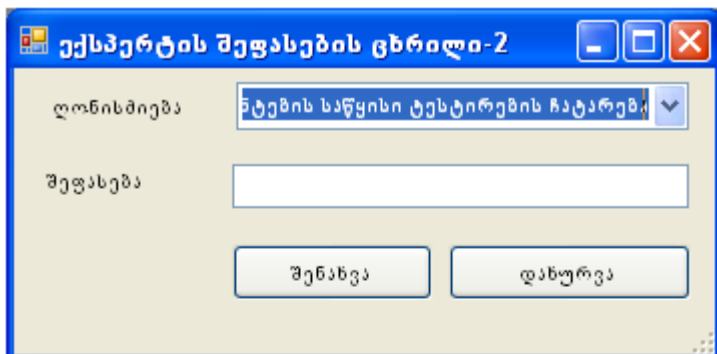
        private void button1_Click(object
sender, EventArgs e)
        {
            try
            {

```

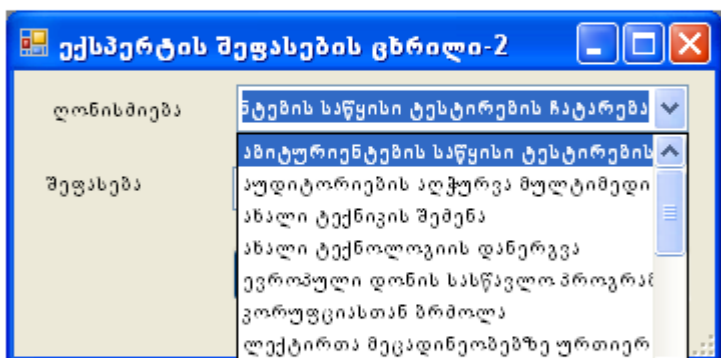
```

DataSet1TableAdapters.QueriesTableAdapter q = new
Expert.DataSet1TableAdapters.QueriesTableAdapter(
);

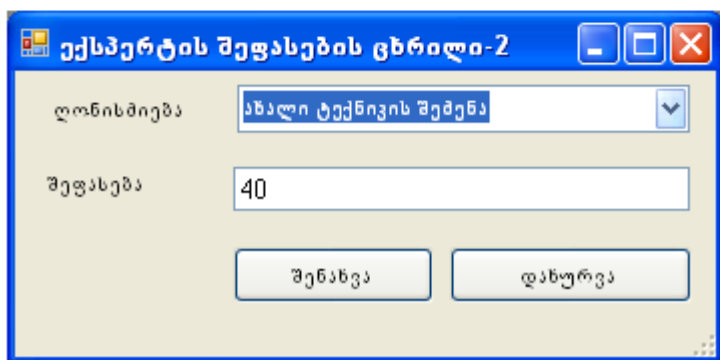
```

ა)



ბ)

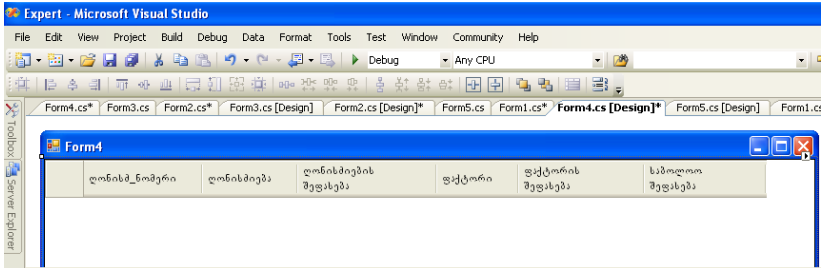


გ)

ნახ.5.6-2. 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.5.7-1. შედეგების გამოტანის ინტერფეისი

ღონისძემორი	ღონისძიება	ღონისძიების შეფასება	ფაქტორი	ფაქტორის შეფასება
1	ასალი ტექნიკის შექმნა	40	მატრიალურ-ტექნიკური შპს	20
2	ასალი ტექნოლოგიის დანერგვა	40	მატრიალურ-ტექნიკური შპს	20
3	უფლოტორიების აღჭურვა მულტიმედიადური ტექნიკით	20	მატრიალურ-ტექნიკური შპს	20
6	მკვლევადიანი აპოტორენტების მოხდენა	70	აპოტორენტები	10
7	აპოტორენტების სწვისი ტესტირების დანერგვა	20	აპოტორენტები	10
4	ლექტორთა კვალიფიკაციის ამაღლება სწავლავრეკრუცი	60	კადრები	20
5	ლექტორთა კვალიფიკაციის ამაღლების ფაქტორის შექმნა	40	კადრები	20
8	ვერობული დონის სასწავლო გროგრაქმის შეფასება	20	სასწავლო პროცესის ორგანიზაცია	20
13	კორფიციციის მოხდენა	15	სასწავლო პროცესის ორგანიზაცია	20
5	ლექტორთა კვალიფიკაციის ამაღლების ფაქტორის შექმნა	10	სასწავლო პროცესის ორგანიზაცია	20
11	წარმოებიდან მკვლევადიფიკაციურ ლექტორთა მოწვევა	25	სასწავლო პროცესის ორგანიზაცია	20
12	წრული მოტივაციის განხორციელება სასწავლო პროცესზე	10	სასწავლო პროცესის ორგანიზაცია	20
15	სტუდენტთა შეფასების ტესტირების დანერგვა	70	სასწავლო პროცესის ორგანიზაცია	20
9	ლექტორთა ხელფასების მომატება	60	ლექტორთა პირობები	10
10	სააღმრეო ფაქტორის ამაღლების ამაღლება	40	ლექტორთა პირობები	10
14	შეადინერების სასწავლო-მედიფიკაციური დონის ამაღლება	45	შეადინერების ხარისხი	20
16	ლექტორთა შეადინერებზე ურთიფიკაციურება	15	შეადინერების ხარისხი	20
17	სტუდენტთა მერ ლექტორთა ამაღლების შეფასების დანერგვა	15	შეადინერების ხარისხი	20
15	სტუდენტთა შეფასების ტესტირების დანერგვა	25	შეადინერების ხარისხი	20

ნახ.5.7-2. შედეგების გამოტანის ინტერფეისი

ღოსტინგი-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;

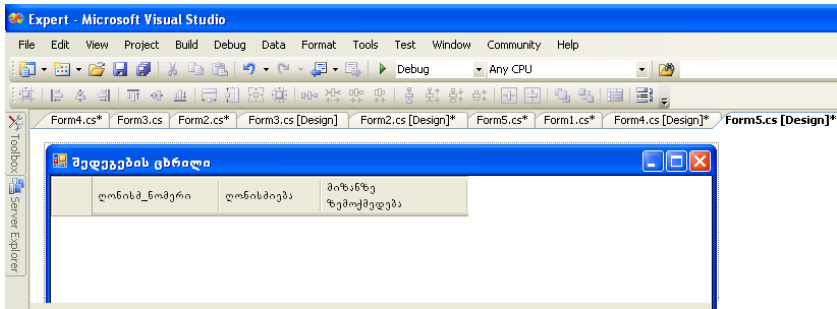
namespace Expert
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();

            DataSet1TableAdapters.SpGetGonisdziebafaqtoriDist
inctTableAdapter ta=new
Expert.DataSet1TableAdapters.SpGetGonisdziebafaqto
riDistinctTableAdapter();

            spGetGonisdziebafaqtoriDistinctDataTableBindingSo
urce.DataSource = ta.GetData();
        }

        private void Form5_Load(object
sender, EventArgs e)
        {
        }
    }
}

```

ნახ.5.8-1. საბოლოო შედეგების გამოტანის ცხრილი

	ლონის_ნომერი	ლონის_ბიუჯეტი	მიზანშედეგის_ზომები
▶	4	ლექტორთა ვვალიდაციის ამალგა საზღვარგარეთ	1200
	5	ლექტორთა ვვალიდაციის ამალგის ფაქტობრივი შექმნა	1000
	14	შეცდომების სასწავლო-მეთოდური დონის ამალგა	900
	15	სტუდენტთა შეფასების ტესტირების ფორმების დანერგვა	900
	1	ახალი ტექნიკის შექმნა	800
	2	ახალი ტექნოლოგიის დანერგვა	800
	6	მაღალკლასიანი აპიტურენტების მოზიდვა	700
	9	ლექტორთა ხელფასების მომატება	600
	11	წარმოებიდან მაღალკვალიფიციურ ლექტორთა მოწვევა	500
	10	საათობრივი დატვირთვის ანაზღაურების ამალგა	400
	3	კუდიტორების აღჭურვა მულტიმედიალური ტექნიკით	400
	8	ვერობილი დონის სასწავლო პროგრამების შედგენა	400
	13	გორუფციასთან ბრძოლა	300
	16	ლექტორთა შეცდომებზე ურთიერთდასწრება	300
	7	აპიტურენტების საწყისი ტესტირების ჩატარება	300
	17	სტუდენტთა მიერ ლექტორთა ანონიმური შეფასების დანერგვა	300
	12	სრული მონიტორინგის განხორციელება სასწავლო პროცესზე	200

ნახ.5.8-2. საბოლოო შედეგების ცხრილის ფრაგმენტი

5.2. მონაცემთა ბაზის ცხრილები

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.5.9-1. სტრუქტურა ექსპერტები

Expert - Microsoft Visual Studio

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

Change Type SQL

Expert: Query(gia.Expert) Start Page Gonisdziebebi: Query(gia.Expert) dbo.Gonisdzieb...le(gia.Ex

ExpertId	gvari	Saxeli	Tanamd	Telefoni	staJi
1	სურგულაძე	ვია	მიმართულების ხელმძღვანელი	37-37-37	35
2	შონია	ოძარი	სრ. პროფესორი	55-55-55	30
3	ვახუტაძე	ოძარი	ასოც. პროფესორი	25-25-25	25
4	თაფურია	ნინო	ასოც. პროფესორი	39-39-39	15
5	სუხიაშვილი	თეიმურაზი	ასოც. პროფესორი	99-99-99	30
6	ვოინაშვილი	ვიორგი	დებარტამენტის უფროსი	32-32-32	37
7	ფრანგიშვილი	არჩილი	რექტორის მოადგილე	95-95-95	25
*	NULL	NULL	NULL	NULL	NULL

ნახ.5.9-2. ექსპერტები

dbo.Miznebi: Table(gia.Expert)		dbo.Expert: Table(gia.Exper	
	Column Name	Data Type	Allow Nulls
▶	MizaniId	int	<input type="checkbox"/>
	dasaxeleba	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

ნახ.5.10-1. სტრუქტურა მიზნები

Miznebi: Query(gia.Expert)		dbo.Ronisdieba...ble(gia.Expert)	
	MizaniId	dasaxeleba	
▶	1	მალაღვაძლიფიციური ინჟინრების გამოშვება	
	2	სწავლების ხარისხის ამაღლება ვერაქულ დონემდე	
	3	ლექტორთა სოციალური სავითხების გაუმჯობესება	
*	NULL	NULL	

1 of 3 | Cell is Read Only.

ნახ.5.10-2. მიზნები

dbo.Factorebi: Table(gia.Expert)		dbo.Miznebi: Table(gia.Ex	
	Column Name	Data Type	Allow Nulls
▶	FactorId	int	<input type="checkbox"/>
	FactorName	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

ნახ.5.11-1. სტრუქტურა ფაქტორები

Factorebi: Query(gia.Expert)		dbo.Factorebi: Table(gia.Expert)
	FactorId	FactorName
▶	1	მატერიალურ-ტექნიკური მანა
	2	კადრები
	3	ამიტორინტები
	4	სასწავლო პროცესის ორგანიზაცია
	5	ლექტორთა პირობები
	6	მეცადინეობების ხარისხი
*	NULL	NULL

1 of 6 | Cell is Read Only.

ნახ.5.11-2. ფაქტორები

dbo.Kriteriume...ble(gia.Expert)		dbo.Gonisdzieb...le(gia.Exp)	
	Column Name	Data Type	Allow Nulls
▶	KriteriumId	int	<input type="checkbox"/>
	KriteriumName	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

ნახ.5.12-1. სტრუქტურა კრიტერიუმები

Kriteriumebi: Query(gia.Expert)		Gonisd...
	KriteriumId	KriteriumName
▶	1	ეკონომიკური
	2	ტექნიკური
	3	სოციალური
	4	ეროვნული
*	NULL	NULL

ნახ.5.12-2. კრიტერიუმები

dbo.Gonisdziebi...le(gia.Expert)		dbo.Factorebi: Table(gia.Ex	
	Column Name	Data Type	Allow Nulls
▶	GonisdziebaId	int	<input type="checkbox"/>
	GonisdziebaName	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

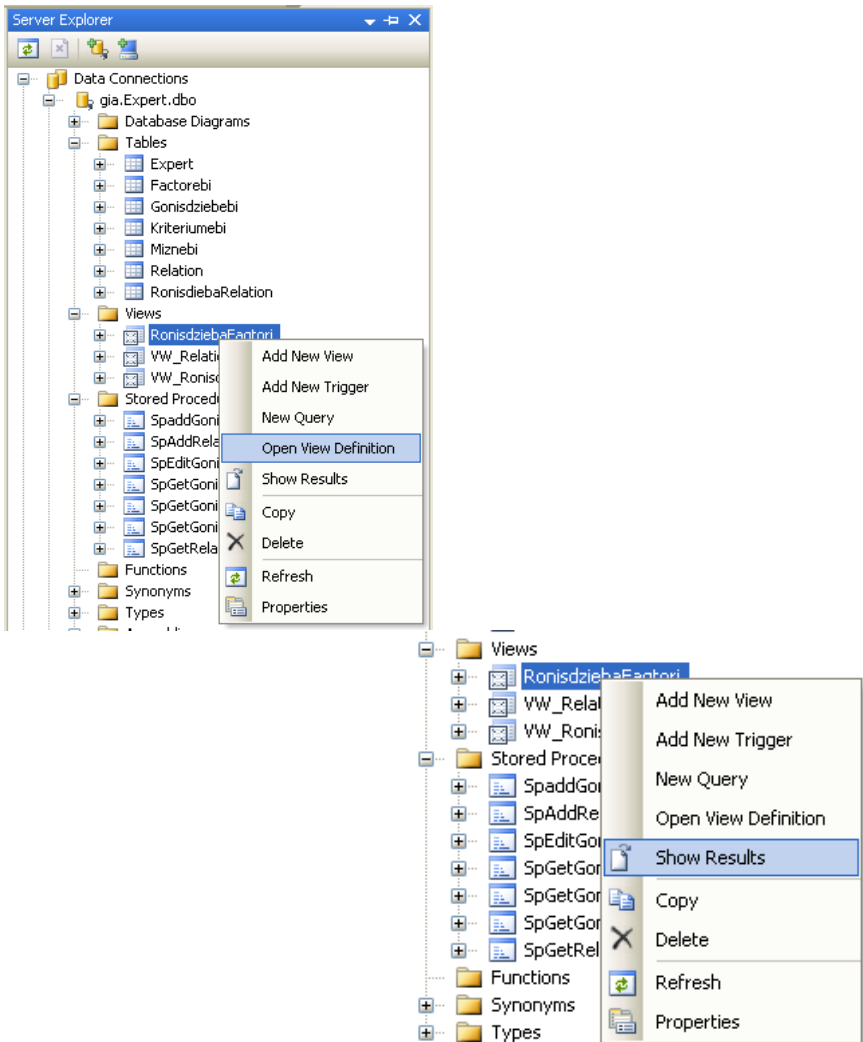
ნახ.5.13-1. სტრუქტურა ღონისძიებები

Gonisdziebi: Query(gia.Expert)		Factorebi: Query(gia.Expert)	dbo.Factorebi:
	GonisdziebaId	GonisdziebaName	
▶	1	ახალი ტექნიკის შექმნა	
	2	ახალი ტექნოლოგიის დანერგვა	
	3	აუდიტორიუმის აღჭურვა მულტიმედიალური ტექნიკით	
	4	ლექტორთა კვალიფიკაციის ამაღლება საზღვარგარეთ	
	5	ლექტორთა კვალიფიკაციის ამაღლების ფაკულტეტის შექმნა	
	6	მადამლედიანი ამიტურინტევის მოზიდვა	
	7	ამიტურინტევის საწყისი ტესტირების ჩატარება	
	8	ვერობელი ღონის სასწავლო პროგრამების შედგენა	
	9	ლექტორთა ხელფასების მომატება	
	10	საათორიგი დატვირთვის ანაზღაურების ამაღლება	
	11	წარმოემიდან მადამლედილიფიკირ ლექტორთა მოწვევა	
	12	სრული მონიტორინგის განხორციელება სასწავლო პროცესზე	
	13	კორუფციისთან ბრძოლა	
	14	მეცადინეოების სასწავლო-მეთოდური ღონის ამაღლება	
	15	სტუდენტთა შეფასების ტესტირების ფორმების დანერგვა	
	16	ლექტორთა მეცადინეოებზე ურთიერთდასწრება	
	17	სტუდენტთა მიერ ლექტორთა ანონიმური შეფასების დანერგვა	
*	NULL	NULL	

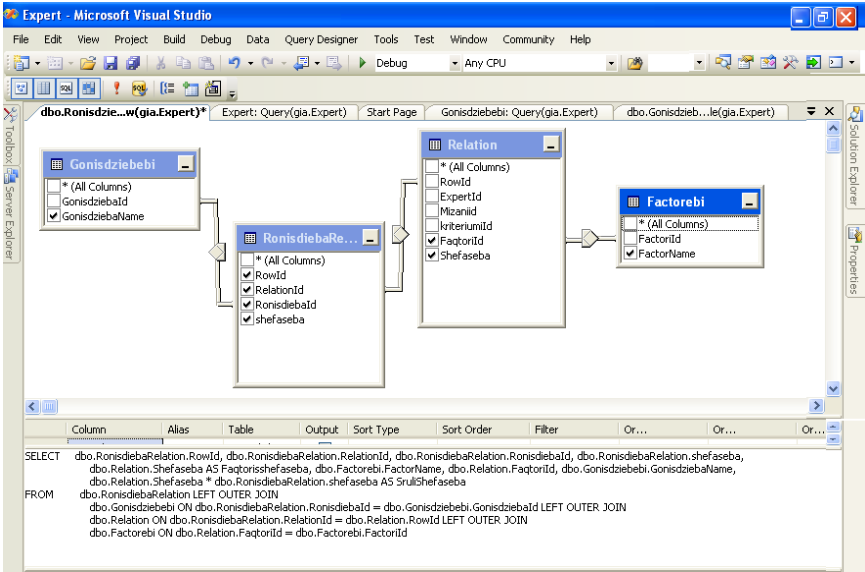
1 of 17 | Cell is Read Only.

ნახ.5.13-2. ღონისძიებები

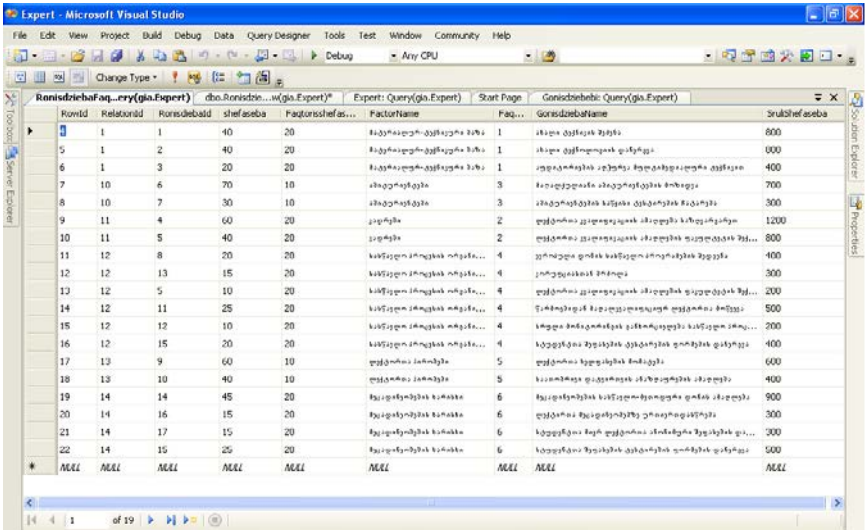
5.3. Expert.dbo მონაცემთა ბაზის Views წარმოდგენები SQL ტექსტებით



ნახ.5.14-1. View ცხრილთაკავშირების და შესაბამისი SQL-ის გამოტანის საშუალებები



ნახ.5.15-1. View ღონისძიება-ფაქტორი SQL-ით



ნახ.5.15-2. შედეგი - View ღონისძიება-ფაქტორი SQL-ით

Expert - Microsoft Visual Studio

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

Debug Any CPU

dbo.VW_Relatio...w(gia.Expert)* RoniszdiebaFaq...ery(gia.Expert) dbo.Roniszdie...w(gia.Expert)* Expert: Query(gia

```

SELECT  dbo.Expert.ExpertId, dbo.Expert.gvari, dbo.Expert.Saxeli, dbo.Expert.Tanamd, dbo.Expert.Telefoni, dbo.Factorebi.FactorId,
        dbo.Factorebi.FactorName, dbo.Miznebi.MizaniId, dbo.Miznebi.dasaxeleba, dbo.Kriteriumebi.KriteriumId,
        dbo.Kriteriumebi.KriteriumName, dbo.Relation.Shefaseba, dbo.Relation.RowId
FROM    dbo.Expert INNER JOIN
        dbo.Relation ON dbo.Expert.ExpertId = dbo.Relation.ExpertId INNER JOIN
        dbo.Miznebi ON dbo.Relation.MizaniId = dbo.Miznebi.MizaniId INNER JOIN
        dbo.Kriteriumebi ON dbo.Relation.kriteriumId = dbo.Kriteriumebi.KriteriumId INNER JOIN
        dbo.Factorebi ON dbo.Relation.FaqtorId = dbo.Factorebi.FactorId

```

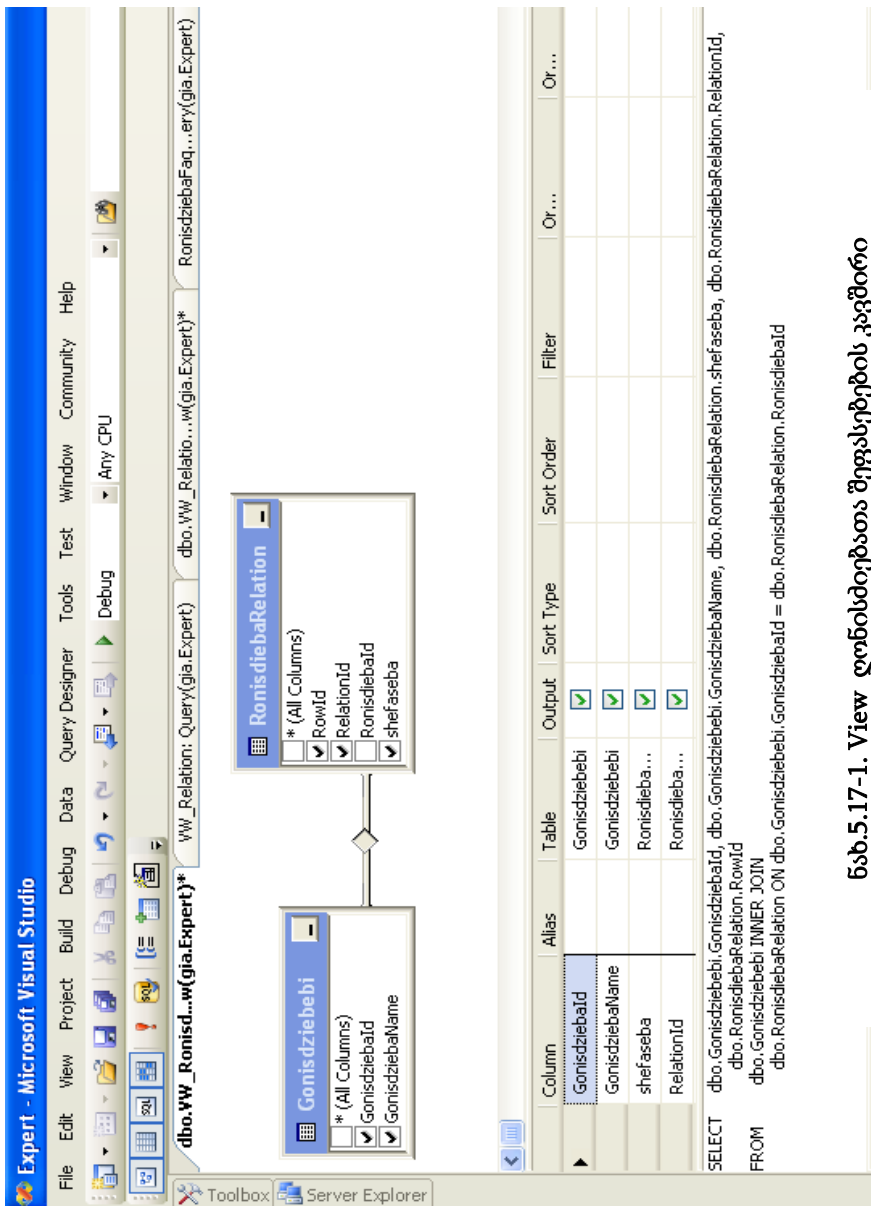
ნახ.5.16-1. View ექსპერტი-ღონისძიება-ფაქტორი კრიტერიუმი SQL-ით

vw_Relation: Query(gis.Expert)							dbo.Vw_Relatio...w(gis.Expert)*		RonisdziebFaq...ery(gis.Expert)		dbo.Ronisdzie...w(gis.Expert)	
ExpertId	gvari	Saxeli	Tanamd	Telefoni	FactorId	FactorName	McanId					
1	უბრუნებ	კაა	მასობულებს ხელმძღვანელი	37-37-37	1	მასობულებს ხელმძღვანელი	1					
1	უბრუნებ	კაა	მასობულებს ხელმძღვანელი	37-37-37	3	ახიერებებს	1					
1	უბრუნებ	კაა	მასობულებს ხელმძღვანელი	37-37-37	2	კატეგორიები	1					
1	უბრუნებ	კაა	მასობულებს ხელმძღვანელი	37-37-37	4	საინჟინერო პროექტების ინჟინერები	1					
1	უბრუნებ	კაა	მასობულებს ხელმძღვანელი	37-37-37	5	დოკუმენტაციის კატეგორიები	1					
1	უბრუნებ	კაა	მასობულებს ხელმძღვანელი	37-37-37	6	მასობულებს ხელმძღვანელი	1					
	NULL	NULL	NULL	NULL	NULL	NULL	NULL					NULL

ს)*

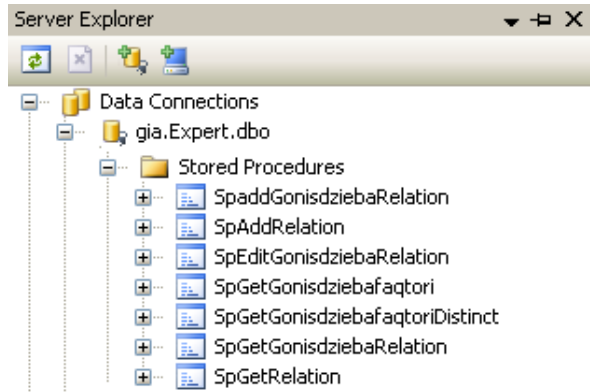
RonisdziebFaq...ery(gis.Expert)		dbo.Ronisdzie...w(gis.Expert)*		Expert: Query(gis.Expert)	
dasaxeleba	KriteriumId	KriteriumName	Shelfaseba	RowId	
მასობულებს ხელმძღვანელი	1	კომპიუტერი	20	1	
მასობულებს ხელმძღვანელი	1	კომპიუტერი	10	10	
მასობულებს ხელმძღვანელი	1	კომპიუტერი	20	11	
მასობულებს ხელმძღვანელი	1	კომპიუტერი	20	12	
მასობულებს ხელმძღვანელი	1	კომპიუტერი	10	13	
მასობულებს ხელმძღვანელი	1	კომპიუტერი	20	14	
NULL	NULL	NULL	NULL	NULL	

ნახ.5.16-2. შედეგი View ვესბერტი-ლონისიება-ფაქტორი

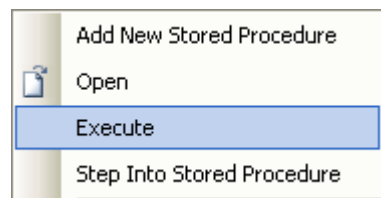
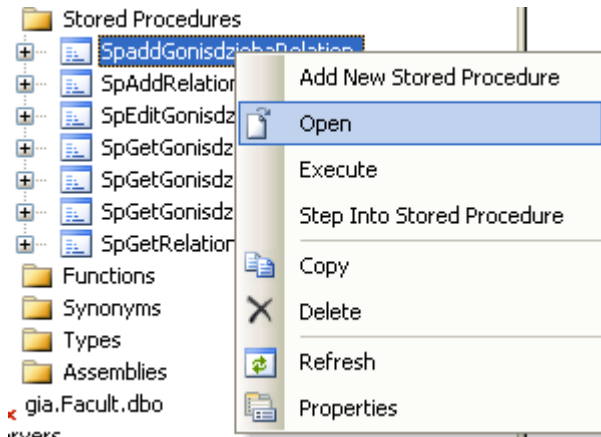


6sb.5.17-1. View დონისებობათა შეფასების კავშირი

5.4. Expert.dbo მონაცემთა ბაზის Stored Procedures



ნახ.5.18-1. შენახვადი პროცედურების სია



ნახ.5.18-2. ა) Open

ბ) Execute:

5.5. შენახვადი პროცედურების ლისტინგები:

ლისტინგი 1:

```
-- SpaddGonisdziebaRelation
ALTER PROCEDURE [dbo].[SpaddGonisdziebaRelation]
@relationID int,
@gonisdziebaId int,
@shefaseba int
AS
BEGIN
    --ღონისძიების შეფასების შეტანა.

    INSERT INTO [Expert].[dbo].[RonisdiebaRelation]
        ([RelationId]
        ,[RonisdiebaId]
        ,[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

```

-- დონისძიების 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 RonisdiebaId,GonisdziebaName,
sum(SruliShefaseba) as SruliShefaseba from
dbo.RonisdziebaFaqtori
    group by RonisdiebaId,GonisdziebaName order by
sum(SruliShefaseba) desc
END

```

ლისტინგი 6:

```

-- SpGetGonisdziebaRelation
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
--ჰედერის(მთავარის) წამოღება
    select * from dbo.VW_Relation
END

```

5.6. მეზუთე თავის დასკვნები

- 5.1.
- 5.2.

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-ინსტრუმენტის გამოყენების საკითხი. შემოთავაზებულია განაწილებული მონაცემთა ბაზებიდან შერჩეული მონაცემების ერთიან გადაწყვეტილების მიღების მხარდამჭერ სისტემაში სტრუქტურული 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.

ლიტერატურა

1. Абрамсон Р., Халсет У. Повышение эффективности работы предприятия с помощью планирования. Пер. с англ., Тбилиси, 1987.
2. გ. სურგულაძე, ი. ვაჭარაძე, ნ. 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> - გადამოწ. 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 – გადამოწ. 20.05.08
8. სურგულაძე გ., პეტრიაშვილი ლ. მონაცემთა საცავის აგების ტექნოლოგია ინტერნეტული ბიზნესის სისტემებისათვის. სტუ, თბ., 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., Rozenberg 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. თბ., 2006.
16. სურგულაძე გ. ობიექტ-ორიენტირებული დაპროგრამების მეთოდი. სტუ. თბ., 2007.

17. ჩოგვაძე ვ., გოგინაშვილი ვ., სურგულაძე ვ., შეროზია თ., შონია ო. მართვის ავტომატიზებული სისტემების დაპროექტება და აგება (თეორიულ-პრაქტიკული ინფორმატიკა). სტუ, თბილისი, 2001.

18. რეისიგი ვ., სურგულაძე ვ., გულუა დ. ვიზუალური, ობიექტ-ორიენტირებული დაპროგრამების მეთოდები. სტუ. თბილისი, 2002.

19. Архангельский А. Программирование в BorlandC++ Builder. Москва, 2001.

20. სურგულაძე ვ. დაპროგრამების ვიზუალური მეთოდები და ინსტრუმენტები: UML, Ms Visio, Borland C++Builder . სტუ. თბილისი, 2000.

21. ბუკია ვ., სურგულაძე ვ., დოლიძე თ., შარაშიძე ბ., შონია ო. ექსპერტთა შეფასებების დამუშავება PC-ზე. სმში, თბილისი, 1990.

22. სურგულაძე ვ., ვაჭარაძე ი., ფოლადაშვილი ნ., ტყემელაშვილი 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).

24. Овчинников В.В., Повышение управляемости больших концептуальных моделей. Информационные технологии, №10, 2004.

25. სურგულაძე ვ., ვედეკინდი ჰ., 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, Mannheim/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.

28. სურგულაძე ვ., 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. Ашкинадзе А. Практика финансового управления: расчет доходности клиентов. М., Банки и Технологии, №2, 2005.

31. სურგულაძე ვ., შონია ო., ყვავაძე ლ. მონაცემთა ბაზების მართვის სისტემები: Ms Access, SQL Server, InterBase, Oracle, Corba. სტუ, თბ., 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. ბოტკე კ., სურგულაძე გ., დოლიძე თანამედროვე პროგრამული პლატფორმები და ენები. სტუ. თბილისი, 2003.

35. <http://msdn.microsoft.com/vstudio/express/vwd/> გადამოწ. 10.04.08

36. <http://msdn.microsoft.com/vstudio/express/sql/> გადამოწ. 10.04.08

37. Halpin T.A., Microsoft's new database modeling tool. *Journal of Conceptual Modeling* www.orm.net, 2002 . გადამოწ. 10.04.08

38. გ. სურგულაძე, ნ. თოფურია, ვ. ქაჩიბაია, ი. ილიძე. კონცეპტუალური მოდელის დაპროექტება -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*, No 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. Чоговадзе Г., Качибая В., Сургуладзе Г. Теория реляционных зависимостей и проектирование логических структур баз данных. "Гос.Университет", Тб., 1988.

49. ჩოგვაძე გ., სურგულაძე გ. რელაციური აღვებრის ოპერაციების შესრულების ეფექტური პროცედურის აგების ერთი ინსტრუმენტის შესახებ მონაცემთა ბაზებში. საქ.მეცნ.აკად. „მოამბე“, 148-№3, 1993.

50. სურგულაძე გ., დოლიძე თ. საწარმოო ფორმებში მარკეტინგული პროცესების მართვის ინფორმაციული სისტემის დაპროექტება და რეალიზაცია UML-ტექნოლოგიით. სტუ-ს შრ.კრებ., №4(437) თბ., 2001.

51. რეისივი ვ., სურგულაძე გ., გულუა დ. ვიზუალური 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. მცნე, თბ., 1993.

54. ბოტჰე კ., სურგულაძე გ., დოლიძე თ., შონია ო. თანამედროვე პროგრამული პლატფორმები და ენები. სტუ-ს შრ.კრ., თბ., 2003.

55. Калверт Ч., Рейсдорф К. Borland C++ Builder: Энциклопедия программиста. Пер.с англ., "DiaSoft", К., 2001.

56. Оутей М., Конте П. SQL – Server2000. СПбетербург, 2002.

57. Харитоновна И., Михеева В. MS ACCESS 2000: Разработка приложений. С – Петербург, 2002.

58. Landy M., Siddiqui S., Swisher J, et al. . Borland JBuilder. Developer's Guide. Williams Publ., 2004.

59. Грин Дж. и др. Oracle8/8i Server. Энциклопедия пользователя. Пер.с англ. Киев, DiaSoft, 2000.

60. Хотка Д. Oracle 9/9i Server. Энциклопедия пользователя. Пер.с англ. Киев, Diasoft, 2004.

61. სურგულაძე გ., პეტრიაშვილი ლ. კუბში მონაცემთა აგრეგაცია გრაფების თეორიის გამოყენებით. ჟურნალი „ინტელექტი“ №3(20),თბ., 2004

62. Albrecht, J., Hummer,W., Lehnen 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 theGTU, 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. Managment 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 DWH: გადამოწ. 10.05.08
69. http://mf.grsu.by/other/lib/olap/bd_wh/doc03.htm OLAP: გადამოწ. 10.05.08
70. http://mf.grsu.by/other/lib/olap/bd_wh/doc09.htm OLAP: გადამოწ. 10.05.08
71. http://mf.grsu.by/other/lib/olap/bd_wh/doc10.htm RelationalDBS to Multi-Dimensional: გადამოწ. 10.05.08
72. http://mf.grsu.by/other/lib/olap/bd_wh/doc18.htm DWH: გადამოწ. 10.05.08
73. სურგულაძე გ., თურქია ე. ბიზნეს-გეგმის ავტომატიზებული დამუშავების პროცესის საშუალო 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, М., 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.Specific. <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/> გადამოწ.15.03.08
82. Майо Дж. С#: Искусство программирования. Энциклопедия программиста. Пер.с англ., "DiaSoft", СПб., 2002.
83. Robinson S., Cornes O., Glynn J., Harvey B., McQueen C., Moemeka J., Nagel C., Skinner M., Watson K. Professional C#. Birmingham, WP, 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 - გადამოწ.18.03.08.

85. ჯავახაძე გ. ბიზნეს-პროგრამების მართვისას გადაწყვეტილებათა მიღების მხარდამჭერი მოდელები და მეთოდები : ავტორეფ. ტექნ. მეცნ. დოქტ. 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 შრ.კრებ. მას-№1, 2006.

88. Козлов Л.А. Когнитивный подход к исследованию информационных процессов на ранних стадиях проектной деятельности. «Системы автоматизированного проектирования», №5. 1986.

89. CPN Tools. www.daimi.au.dk/CPNTools/. გადამოწ. 1.10.08

90. Jensen K., Kristensen M.L., Wells L. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. University of Aarhus. Denmark. 2007. 24]

91. <http://www.gnuplot.info> გადამოწმებულია 10.10.08

92. სურგულაძე გ., ოხანაშვილი მ., სურგულაძე ვრ. მარკეტინგის ბიზნეს-პროცესების უნიფიცირებული და იმიტაციური მოდელირება. სტუ, თბ., სტუ, 2009.

93. ოხანაშვილი მ., შარაშიძე თ. პროდუქციის წარმოების ინფორმაციულ-ტექნოლოგიური პროცესების უნიფიცირებული და იმიტაციური მოდელირება. სტუ შრომები №1(4) გვ.108–113 2008.

94. სურგულაძე გ., თურქია ე., ოხანაშვილი მ., სურგულაძე გ. მარკეტინგული პროცესების მართვის ერთი მოდელის შესახებ ფერადი პეტრის ქსელებით- №2(5), 2008. გვ. 9–16.

95. ბოტჭე კ., სურგულაძე გ., კაშიბაძე მ. მემკვიდრეობითობა მართვის ინფორმაციული სისტემების დაპროგრამებაში: მონაცემთა ბაზებიდან UML-ტექნოლოგიამდე. სტუ შრ.კრ. №4(437), თბ., 2001.

96. სურგულაძე გ., კაშიბაძე მ. ორგანიზაციულ სისტემებში ინფორმაციული რესურსების მართვა. სტუ, თბილისი. 2009.

97. Oesterreich B. <http://swt.informatik.uni-freiburg.de/teaching/winter-term-2008/software-design-modelling-and-analysis-in-uml>

ავტომატიზებული ანალიზის სისტემის მოდელი უნივერსიტეტის სასწავლო პროცესის სრულყოფისთვის

უნივერსიტეტის ექსპერტების (პროფესორ-მასწავლებელთა) გამოკითხვისა და მათი შეხედულებების ანალიზის საფუძველზე MsExcel პაკეტის გარემოში (ნახ.დ-1) აგებულია სისტემის პირველი მოდელი, რომლის მიზანია „მაღალკვალიფიციური ინჟინრების გამოშვება“-ამოცანის ანალიზი. ნახაზი დ-2 შეესაბამება ამ ცხრილის მიზნის, ფაქტორებისა და ბიზნეს-პროგრამების (გასატარებელი ღონისძიებების) იერარქიულ კავშირების ასახვას. მე-5 თავში ეს ამოცანა რეალიზებულია MsSQL Server მონაცემთა ბაზის და ობიექტ-ორიენტირებული დაპროგრამების ინტეგრირებული VisualStudio პაკეტის გამოყენებით.

L3 * = 100·(B3+D3+F3+H3+J3)

A	B	C	D	E	F	G	H	I	J	K	L	M
1					მიზანი: მაღალკვალიფიციური ინჟინრების მოშვება							1-ღრმე
2												
3	10		20		20		20		20			10
4	აბიტურიენტი	მატ-ტექნ. ბაზა			ადრები		სასწ პროგ. ორგანიზება		მეცადინეობის სარისკო		ლექტორთა პრობები	2-ღრმე
5												
6							20		45		80	
7							ვებო-დონის სასწ პროგრ. შედგენა		სასწ-მეთოდ. დონის ამაღლება		შეფასების მომატება	
8		70		40		60		15		15		55
9	მაღალკვალი ნების მოზიდვა		ასალი ტექნოლოგიის შეტენა		გვალუფაცია საზღვარგარეთ		10	კორუფციასთან ბრძოლა		ლექტორთა მეცადინეობებზე ურობორიზაცია		სააოდატვირთ. ანაზღ. ვაზრდა
10						40			20	25		15
11	საწესის ტესტირება		ასალი ტექნოლოგიის დანერგვა				გვალუფაცია ცლის ფ-ალუტ. შექმნა		სტუდენტთა შეფასების ტესტირების ფორმების დანერგვა			ლექტორთა შეფასება სტუდენტების მიერ
12												
13							25			10		
							წარმოებდან მაღალკვალიფ. ლექტორთა მოწვევა		სასწ პროცესის სრული მონიტორინგი			3-ღრმე

ნახ. დ-1



ნახ. დ-2

იბეჭდება ავტორთა მიერ
წარმოდგენილი სახით

გადაეცა წარმოებას 4.05.2009 წ. ხელმოწერილია დასაბეჭდად
14.05.2009 წ. ოფსეტური ქაღალდის ზომა 60X84 1/16. პირობითი
ნაბეჭდი თაბახი 9. ტირაჟი 100 ეგზ.

საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“
თბილისი, მ. კოსტავას 77



Verba volant,
scripta manent

ი.მ. „გონა დალაქიშვილი“,
ქ. თბილისი, ვარკეთილი 3, კორპ.333, ბ.38