

საქართველოს ტექნიკური უნივერსიტეტი

ბია სურგულაძე, მარინა კაშიბაძე

**ორბანიზაციულ სისტემებში
ინფორმაციული რესურსების მართვა**



რეგისტრირებულია
სტუ-ს სარედაქციო-
საგამომცემლო
საბჭოს მიერ

თბილისი – 2009

უაკ 681.3.06

გადმოცემულია კორპორაციული მართვის სისტემებში ბიზნეს-პროცესების უნიფიცირებული მოდელირების და ანალიზის, მონაცემთა ბაზების ობიექტ-ორიენტირებული დაპროექტების, მომხმარებელთა ინტერფეისების ვიზუალური დაპროგრამების თეორიული საფუძვლები და მათი რეალიზაციის ინსტრუმენტული საშუალებანი. გამახვილებულია ყურადღება ორგანიზაციული მართვის საინფორმაციო სისტემებში კოლექტიური მოხმარების რესურსების გამოყენების ეფექტიანობის ამაღლების ამოცანის გადაწყვეტაზე. UML-ტექნოლოგიისა და პეტრის ქსელების ინსტრუმენტული საშუალებით განხორციელებულია ექსპერიმენტული შედეგების კვლევა.

დამხმარე სახელმძღვანელო განკუთვნილია მართვის საინფორმაციო სისტემების (Management Information Systems) სპეციალობის მაგისტრანტებისათვის, აგრეთვე თეორიული და პრაქტიკული ინფორმატიკის სპეციალისტების, დოქტორანტების და სტუდენტებისთვის.

რეცენზენტი: ტ.მ.დ., პროფ. ვ. სესაძე

პროფ. გ. სურგულაძის რედაქციით

© საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, 2009

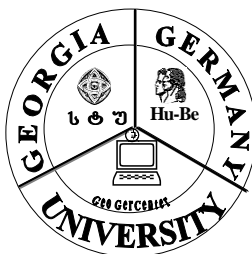
ISBN 978-9941-14-447-6

Georgian Technical University

GIA SURGULADZE, MARINA KASHIBADZE

**MANAGEMENT OF
INFORMATION RESOURCES IN
ORGANIZATIONAL SYSTEMS**

Supported by DAAD
(Germany)



© Publication House "Technical University", Tbilisi, 2009
ISBN 978-9941-14-447-6

Tbilisi 2009

Abstract

The work “Management of information resources in organizational systems”. Questions of the decision of problems of designing and implementation of the distributed databases for Management Information systems are considered on the basis of the object-oriented approach, and also questions of management by processes of collective use of information and program resources.

Corporate Management Information System which functions on the basis of a local network of computers, operative memory, processors, printers, scanners, etc. has the certain computing resources as which it is possible to consider the automated workplaces, channels of a network, hard disks, including shared devices. It is necessary to note, that active resources of a computer network are programs and files of databases.

At designing multiprocessor systems the set of parameters, a choice which optimum value are provided is significant and difficult. Optimization such value not only raises productivity of a network, but also reduces charges for creation of such network. Considerably to take into account optimum distribution of such parameters what capacities are, the common resources, etc.

From resources of a computer network especially it is necessary to allocate an information resource, in particular bases melon. As the majority of inquiries demand from information bases allocation of the certain fragments, therefore in the dissertation the big attention on intensional and extensional resources of databases will be given.

In the dissertation the following problems are offered:

Modelling and the analysis of dynamic processes of the distributed systems of organizational management are considered on the basis of graphical-analytical tools, modelling business of processes on the basis of diagrams of activity and a state with the purpose of a choice of the best model from alternative results designing is offered to model of representation of business - processes and operations with the help of Petri networks.

Disputed situations of development and transfer of information streams in the distributed systems, in particular opportunities of existence

of deadlock processes and their exceptions are considered. The model of management is developed by such processes on the basis of the tool of Petri networks. Simulation of these processes is lead and corresponding time characteristics are constructed.

Problems of the automated construction of conceptual ER-models on basis UML of technology of programming and the tool of Objective Role Modelling (ORM) of automated workplaces are considered. On the basis of use of categorial approach and logic and algebraic methods of modelling structures of the data and knowledge is realized.

Problems of converting of the information in big corporative systems, as necessary and rather complex process of processing of data warehouses of the data are analysed. In work the structure of process of converting of data warehouse of the data is offered and its communication with time characteristics and process of acceptance the decision in organizational systems is investigated.

The concept and problems of perspective development of virtual marketing, as an example of business and electronic business of commerce is considered. The problem of construction of automated workplace of the virtual distributor, with an actual database, algorithms of modelling and the analysis of processes of an estimation of efficiency of their activity is offered.

წინასიტყვაობა

წიგნში „ორგანიზაციულ სისტემებში ინფორმაციული რესურსების მართვა“ განიხილება კორპორაციული მართვის ობიექტებზე მონაცემთა განაწილებული ბაზების დაპროექტების და რეალიზაციის ამოცანების გადაწყვეტა ობიექტ-ორიენტირებული მიდგომით. ასეთ სისტემებში ინფორმაციული და პროგრამული რესურსების კოლექტიური გამოყენების პროცესების მართვა.

კორპორაციული მართვის ავტომატიზებული სისტემა, რომელიც კომპიუტერების ლოკალურ ქსელზე ფუნქციონირებს, ფლობს გარკვეულ გამოთვლით რესურსებს, როგორებიცაა სამუშაო ავტომატიზებული ადგილები ფუნქციურ განყოფილებებში, ქსელის არხებს (კაბელები და ქსელის პლატები), მაგნიტური დისკოები (ვინჩესტერები), მათ შორის კოლექტიური მოხმარებისათვის, ოპერაციული მეხსიერებები და პროცესორები, პრინტერები და სკანერები და ა.შ. აქვე უნდა აღნიშნოს, რომ კომპიუტერული ქსელის აქტიური რესურსებია პროგრამები და მონაცემთა ბაზების ფაილები.

მულტიპროცესორული სისტემის დაპროექტების დროს გაითვალისწინება მრავალი მახასიათებელი, რომელთა ოპტიმალური მნიშვნელობების შერჩევა ძალზე მნიშვნელოვანია და ამავე დროს რთულიც. ამ სიდიდეთა ოპტიმიზაცია არა მარტო გაზრდის კომპიუტერული ქსელის მწარმოებლურობას, არამედ შეამცირებს მის შესაქმნელად საჭირო ხარჯებსაც. მნიშვნელოვანია ისეთი მომენტების გათვალისწინება, როგორიცაა სიმძლავრეების, საერთო რესურსების და ა.შ. ოპტიმალური განაწილება.

კომპიუტერული ქსელის რესურსებიდან ჩვენ განსაკუთრებით გვინდა გამოვყოთ ინფორმაციული რესურსი – კერძოდ მონაცემთა ბაზები. ვინაიდან მოთხოვნათა უმეტესი რაოდენობა ითხოვს ინფორმაციულ ბაზებიდან გარკვეული ფრაგმენტის ამოღებას, ამიტომაც ნაშრომში მონაცემთა ბაზების ინტენსიონალურ და ექსტენსიონალურ რესურსზე გვექნება გამახვილებული ყურადღება.

წინამდებარე ნაშრომის მიზანია კორპორაციული ობიექტებისათვის მონაცემთა განაწილებული, რელაციური ბაზების დაპროექტებისა და

აგების ეფექტური მეთოდებისა და ინსტრუმენტების შემუშავება. ამ მიზნის მისაღწევად ნაშრომში წარმოდგენილი არის შემდეგი ამოცანების გადაწყვეტის საკითხები:

განიხილება ორგანიზაციული მართვის განაწილებულ სისტემებში დინამიკური პროცესების მოდელირებისა და ანალიზის გრაფო-ანალიზური სისტემების გამოყენება. სამუშაო პროცესების მოდელირება აქტიურობისა და მდგომარეობათა დიაგრამებით. დაპროექტების ალტერნატიული შედეგებიდან საუკეთესო მოდელის შესარჩევად შემოთავაზებულია ბიზნეს-პროცესებისა და ოპერაციების ასახვის მოდელები პეტრის ქსელის გამოყენებით.

განიხილება განაწილებულ სისტემებში ინფორმაციული ნაკადების დამუშავების და გადაცემის კონფლიქტური სიტუაციები, კერძოდ ჩიხური პროცესების არსებობისა და მათი გამორიცხვის შესაძლებლობანი. შემუშავებულია ასეთი პროცესების მართვის მოდელი პეტრის ქსელის ინსტრუმენტის საფუძველზე. ჩატარებულია ამ მოდელის იმიტაციური გამოკვლევა და აგებულია შესაბამისი დროითი მახასიათებლები. ამგვარად, კომპიუტერულ ქსელებში მიმდინარე მოვლენების (დინამიკური პროცესების) მოდელირებისათვის მოსახერხებელია პეტრის ქსელების აპარატის გამოყენება, რაოდენობრივი მახასიათებლების ანალიზისათვის კი – მასობრივი მომსახურეობის სისტემების თეორია

განიხილება დაპროგრამების UML ტექნოლოგიის და ობიექტ-როლური მოდელირების ORM ინსტრუმენტის გამოყენებით არსთა-დამოკიდებულების ERM მოდელის ავტომატიზებულ რეჟიმში აგების ამოცანა. კატეგორიალური მიდგომისა და ლოგიკურ-ალგებრული მეთოდების გამოყენებით განხორციელებულია საპრობლემო სფეროს მონაცემთა სტრუქტურებისა და ცოდნის ობიექტ-ორიენტირებული მოდელირება.

გაანალიზებულია ინფორმაციის კონვერტაციის პრობლემები დიდ კორპორაციულ სისტემებში, როგორც მონაცემთა საცავების გადაამუშავების აუცილებელი და საკმაოდ რთული პროცესი. ნაშრომში წარმოდგენილია მონაცემთა საცავის კონვერტაციის პროცესის სტრუქტურა და გამოკვლეულია მისი კავშირი ორგანიზაციულ სისტემებში გადაწყვეტილების მიღების პროცესის დროით მახასიათებლებთან.

განხილულია ელექტრონული ბიზნესის და ელექტრონული კომერციის სისტემების დამკვიდრებისა და ვირტუალური მარკეტინგის პერსპექტიული განვითარების კონცეფცია და ამოცანები. შემოთავაზებულია ვირტუალური მარკეტინგის სისტემაში პროდუქციის ვირტუალური დისტრიბუტორ-რეალიზატორის ავტომატიზებული სამუშაო ადგილის აგების ამოცანა, მონაცემთა აქტუალური ბაზით, საქმიანობის ეფექტურობის შეფასების პროცესის მოდელირებისა და ანალიზის ალგორითმით.

დამხმარე სახელმძღვანელოში წარმოდგენილია აგრეთვე ავტორთა უშუალო მონაწილეობით შესრულებული საპროექტო სამუშაოები, დაკავშირებული მონაცემთა განაწილებული ბაზებისა და მართვის ავტომატიზებული სისტემების დაპროექტების და რეალიზაციის ეტაპებთან. კერძოდ, აგებულია სერვის-ფირმის ტიპის სააგენტოს საინფორმაციო სისტემა უძრავი ქონების აღრიცხვისა და გაყიდვა-გაქირავების კონტროლის (მონიტორინგის) მიზნით. დამუშავებულია მომხმარებელთა ინტერფეისები თანამედროვე ობიექტ-ორიენტირებული დაპროგრამების ენებისა და კლიენტ-სერვერული არქიტექტურის სისტემისთვის .NET პლატფორმაზე, C++, C#, Ms SQL Server, ADO.NET, Ms Access და სხვ პაკეტების ბაზაზე.

პროფ. გ. სურგულაძე

შინაარსი

წინასიტყვაობა	6
შესავალი	11
1 თავი. ორგანიზაციული მართვის სისტემების რესურსების კლასიფიკაცია და ანალიზი	15
1.1. განაწილებული სისტემების ინფორმაციული რესურსების კლასიფიკაცია, მათი მართვის პრობლემები და ამოცანები	15
1.2. მონაცემთა რელაციური ბაზების სემანტიკური პრობლემები და მათი გაფართოებული მოდელები	20
1.3. მონაცემთა განაწილებული, ობიექტ-ორიენტირებული ბაზების მართვის სისტემები და დაპროექტების ტექნოლოგია	25
1.4. უნიფიცირებული მოდელირების ენა (UML) და მონაცემთა განაწილებული ბაზების დაპროექტების ამოცანა	36
1.5. განაწილებული სისტემის ბიზნეს-პროცესების მოდელირების და ანალიზის ინსტრუმენტი პეტრის ქსელების ბაზაზე	38
2 თავი. განაწილებული სისტემების ინფორმაციული რესურსების მართვის პროცესების მოდელირება UML/PN ტექნოლოგიით	41
2.1. ინფორმაციული რესურსების მართვის პროცესების ტექნოლოგია, სტატიკური და დინამიკური მოდელები	41
2.2. უნიფიცირებული მოდელირების ენა და ობიექტური პეტრის ქსელები. UML/PN ინსტრუმენტული საშუალების ბირთვი	49
2.3. დინამიკური პროცესების ტრანსფორმაციის ძირითადი პრინციპები და UML/PN ტექნოლოგიის დამუშავება მონაცემთა ბაზებისთვის	55
2.4. ობიექტების სასიცოცხლო ციკლის მოდელირება და ანალიზი პეტრის ქსელებით	60
2.5. განაწილებული სისტემების რესურსების ადმინისტრირების ამოცანები: ჩიხური სიტუაციების მართვა	64
2.6. მონაცემთა განაწილებული ბაზების დაპროექტება კატეგორიული მიდგომით და ობიექტ-როლური მოდელის ინსტრუმენტით	68
2.7. განაწილებული სისტემის კონცეპტუალური მოდელის დაპროექტების და მონაცემთა ბაზის აგების პროცედურების ავტომატიზებული დაპროგრამება	79

2.8. ტრანსფორმაციისა და კონვერტაციის ალგორითმების დაძუშაება	89
3 თავი. შიდეშების პრაქტიკული რეალიზაცია	102
3.1. ვირტუალური მარკეტინგული პროცესების მოდელირება და დისტრიბუტორ-რეალიზატორის ავტომატიზებული სამუშაო ადგილი	102
3.2. საწარმოო გაერთიანების მარკეტინგული უზრუნველყოფის დეპარტამენტის განაწილებული მონაცემთა ბაზის მთლიანობის დაცვა	108
3.3. ინფორმაციის უსაფრთხოებისა და საიდუმლოების დაცვის საკითხები	115
3.4. მონაცემთა ბაზა და სერვისული პროგრამები უძრავი ქონების სააგენტოს მაგალითზე	117
ლიტერატურა	123

შესავალი

განაწილებული ორგანიზაციული მართვის სისტემების დაპროექტება და მათი პროგრამული რეალიზაცია თანამედროვე ობიექტ-ორიენტირებული ინფორმაციული ტექნოლოგიების გამოყენებით მეტად მნიშვნელოვანი და აქტუალური მიმართულებაა. ამაზე მეტყველებს დიდი სიმრავლე იმ სამეცნიერო-ტექნიკური შრომებისა და რეალიზებული ბიზნეს-სისტემებისა, რომლებიც ბოლო 15-20 წლის მანძილზე ვითარდებოდა ამერიკის, ევროპისა და სხვა ქვეყნების უნივერსიტეტებსა თუ საპროექტო-კვლევით ორგანიზაციებში [1,2,3].

განსაკუთრებით მნიშვნელოვანია კორპორაციული მართვის სისტემებში მონაცემთა საცავების, განაწილებული რელაციური ბაზების, ობიექტ-ორიენტირებული მოდელირებისა და დაპროგრამების ენების ახალი კონცეფციების დანერგვა, რასაც მივყავართ ზოგადად კომპიუტერული აპლიკაციების სტანდარტიზაციის მეთოდოლოგიის ჩამოყალიბებამდე. უნიფიცირებული მოდელირების (UML, ARIS და სხვ.) ენები და სისტემები, ანუ დაპროექტებისა და დაპროგრამების პროცესების ავტომატიზაციის იდეა პრიორიტეტული გახდა ინტეგრირებული მართვის საინფორმაციო გამოყენებითი პაკეტების შექმნის თერიაში. ამ, ერთგვარად რთული და ინტერდისციპლინარული სამეცნიერო მიმართულების სფეროში მნიშვნელოვანი წვლილი შეიტანეს ე. კოდმა, ბ. სტრაუსტრუპმა, გ. ბუჩმა, ი. ჯაკობსონმა, ჯ. რამბომ, ტ. ჰალპინმა, ჰ. ვედეკინდმა, ა. შეერმა, ვ. გორბატოვმა, ლ. კუზინმა, დ. პოსპელოვმა, მ. ცალენკომ და სხვ., საქართველოში მონაცემთა რელაციური და ინტელექტუალური ბაზების მართვის სისტემების გამოყენებისა და განვითარების სფეროში გარკვეული კვალი დატოვეს სტუ-ს პროფესორებმა გ. ჩოგოვაძემ, გ. გოგიჩაიშვილმა, გ. სურგულაძემ, ვ. ქაჩიბაიამ, გ. ღარიბაშვილმა და სხვ. [4,7,8,9,20,22,35].

წინამდებარე სახელმძღვანელოს მიზანია სტუდენტებსა და მაგისტრანტებს მიაწოდოს კორპორაციული ობიექტებისათვის მონაცემთა განაწილებული, რელაციური ბაზების დაპროექტებისა და აგების ეფექტური მეთოდებისა და ინსტრუმენტების შემუშავების მეთოდოლოგია. ამ მიზნის მისაღწევად ნაშრომში განიხილება შემდეგი ამოცანები:

- ორგანიზაციული მართვის ობიექტების ობიექტ-ორიენტირებული ანალიზისა და ობიექტ-ორიენტირებული დაპროექტების მეთოდების

პროცესების კვლევა და მათი ეფექტური გამოყენების ალგორითმული სქემების შემუშავება;

- საპრობლემო სფეროს კონცეპტუალური მოდელირების მეთოდების სრულყოფა უახლესი ინფორმაციული ტექნოლოგიების კომპლექსურად გამოყენების საფუძველზე;

- მონაცემთა ბაზების სტრუქტურების დაპროექტების პროცესების ავტომატიზაცია კატეგორიული მიდგომისა და ლოგიკურ-ალგებრული მეთოდების საფუძველზე, სპეციალური ინსტრუმენტული საშუალებების შექმნა და სრულყოფა ტექნოლოგიური პროცესების ავტომატიზაციის მიზნით;

- განაწილებული კორპორაციულ სისტემების ინფორმაციული და გამოთვლითი რესურსების კოლექტიური გამოყენების პროცესების მოდელირება და კვლევა პეტრის ქსელების საფუძველზე;

- მომხმარებელთა ინტერფეისების შემუშავება კლიენტ-სერვერული არქიტექტურის სისტემისთვის .NET პლატფორმაზე, C++, C#, MsSQL Server, ADO.NET, MsAccess და სხვ პაკეტების ბაზაზე.

დისერტაციის **პირველ თავში** განიხილება დასმული პრობლემის ირგვლივ არსებული დღევანდელი მდგომარეობა, ლიტერატურული წყაროების ანალიზი. გამოვლენილია ის ძირითადი ამოცანები, რომელთა გადაწყვეტა სრულყოფს განაწილებული სისტემების ინფორმაციული უზრუნველყოფის დაპროექტებისა და შექმნის პროცესებს. როგორც ცნობილია, მულტიპროცესორული სისტემის დაპროექტებისას გაითვალისწინება მრავალი მახასიათებელი, რომელთა ოპტიმალური მნიშვნელობების შერჩევა ძალზე მნიშვნელოვანია და ამავე დროს რთულიც. ამ სიდიდეთა ოპტიმიზაცია არა მარტო გაზრდის კომპიუტერული ქსელის მწარმოებლურობას, არამედ შეამცირებს მის შესაქმნელად საჭირო ხარჯებსაც. მნიშვნელოვანია ისეთი მომენტების გათვალისწინება, როგორცაა სიმძლავრეების, საერთო რესურსების და ა.შ. ოპტიმალური განაწილება.

მეორე თავში ყურადღებას ვამახვილებთ მართვის ავტომატიზებული სისტემების ინფორმაციული უზრუნველყოფის დაპროექტებისა და რეალიზაციის ამოცანებზე, ინფორმაციული რესურსების მართვის პროცესების ტექნოლოგიის ასახვაზე უნიფიცირებული მოდელირების UML ენის სტატიკური და დინამიკური მოდელების გამოყენებით.

საყურადღებოდ მიგვაჩნია ასეთი დინამიკური პროცესების მოდელირება და კვლევა პეტრის ქსელების ინსტრუმენტით, ანუ ასეთი პროცესების ეკვივალენტური (ალტერნატიული) აქტიურობათა, ინტერაქტიურობათა და მდგომარეობათა დიაგრამების შესაბამისი პეტრის ქსელის მოდელების აგებითა და ანალიზით შესაძლებელი ხდება უკეთესის არჩევა, გარკვეული კრიტერიუმების ბაზაზე.

მოდელირებისა და ანალიზის ინსტრუმენტად ვიყენებთ პეტრის ელემენტარულ და სისტემურ ქსელებს, მდგომარეობებისა და გადასასვლელების გრაფული ასახვის საშუალებით, მიზნულ-შედეგობრივი დამოკიდებულებებისა და პროცესების ასახვისათვის. ასეთ სისტემებში შეზღუდვები გამოხატავს მონაცემთა ბაზის მდგომარეობის (რელაციები) ზოგად, აბსტრაქტულ თვისებებს ანუ მონაცემთა ბაზის სემანტიკას. შეზღუდვების გამოხატვის ყველაზე ბუნებრივი ხერხია გამონათქვამები I რიგის პრედიკატების ენაზე, რომელიც განსაზღვრავს მონაცემთა ბაზის დასაშვებ მდგომარეობათა სიმრავლეს. მონაცემთა ბაზა, გადადის რა ერთი მდგომარეობიდან მეორეში (რელაციური ცვალებადობა), აღწერს მონაცემთა მოდელში გარკვეულ ტრაექტორიას. ყოველი მომდევნო მდგომარეობა შეიძლება დამოკიდებული იყოს მის წინა მდგომარეობაზე. ამ მდგომარეობებს შორის კავშირები აღიწერება მთლიანობის დინამიკური შეზღუდვებით (პრედიკატებით), რომლებიც ქმნის მონაცემთა ბაზის ე.წ. დინამიკურ სქემას.

ამგვარად, უნიფიცირებული მოდელირების ენისა და პეტრის ქსელების საფუძველზე შემუშავებულია UML/PN – მეთოდი და ინსტრუმენტი, რომლის საფუძველზეც შესაძლებელია ბიზნეს-პროცესების ასახვა UML–დიაგრამებით, მათი მოდელირება პეტრის ქსელებით და ამ მოდელების ანალიზი, რათა დადგინდეს დინამიკური ბიზნეს-პროცესების ეფექტურობა, მათი შესრულების დროის შემცირების მიზნით.

საპრობლემო სფეროს კონცეპტუალური მოდელებისა და მონაცემთა რელაციური ბაზების სტრუქტურების ავტომატიზებულ რეჟიმში დასაპროექტებლად და შემდგომ MsSQL Server ბაზაში გამოსაყენებლად, დამუშავდა ოთხეტაპიანი კომპლექსური ალგორითმიული სქემა < Fact → ORM → ERM → DDL → MsSQLServer >. პირველ ეტაპზე ხდება საპრობლემო სფეროს ექსპერტის მიერ შედგენილი ფაქტების (ობიექტზე არსებული კანონზომიერებანი, ჩაწერილი ბუნებრივი

სალაპარაკო ენაზე) სიმრავლის ასახვა კატეგორიალური მიდგომით და ობიექტ-როლური მოდელირების ინსტრუმენტით. მეორე ეტაპზე ORM-მოდელიდან ვაპროექტებთ არსთა-დამოკიდებულების ER-მოდელს (ავტომატიზებულ რეჟიმში). მესამე ეტაპზე მონაცემთა აღწერის ენაზე (DDL), კოდების გენერატორით იქმნება შესაბამისი ფაილი, რომელიც ბოლო ეტაპზე მიეძღვნება MsSQL Server-პროგრამულ პაკეტს. ნაშრომში განხილულია აგრეთვე მოაცემთა საცავეებში ინორმაციის კონვერტაციის პრობლემები და შემოთავაზებულია სპეციალური პროგრამული Import/Export ინსტრუმენტი მონაცემთა სტრუქტურების თავსებადობის განსახორციელებლად სხვადასხვა პროგრამულ პაკეტებს შორის.

მესამე თავი ეხება მიღებული თეორიული შედეგების პრაქტიკულ რეალიზაციას. ნაშრომში წარმოდგენილია ექსპერიმენტებისა და რეალური პროექტების ფორმატით შერულებული სამუშაოთა შედეგები. კერძოდ, ვირტუალური მარკეტინგული პროცესების მოდელირება და ვირტუალური დისტრიბუტორ-რეაილიზატორის ავტომატიზებული სამუშაო ადგილის შექმნა. განიხილება პროდუქციის მწარმოებელი ფირმის მარკეტინგული უზრუნველყოფის სამსახურის მონაცემთა განაწილებული ბაზების მთლიანობის დაცვისა და ინფორმაციის უსაფრთხოების საკითხები. დამუშავებულია ბიზნეს-ფირმის „უძრავი ქონების სააგენტო“ სისტემის მონაცემთა ბაზები და სერვისული პროგრამები.

I ტაზი

ორგანიზაციული მართვის სისტემების რესურსების კლასიფიკაცია და ანალიზი

1.1. განაწილებული სისტემების ინფორმაციული რესურსების კლასიფიკაცია, მათი მართვის პრობლემები და ამოცანები

ორგანიზაციული მართვის სისტემების დაპროექტებისა და რეალიზაციის საკითხები მნიშვნელოვანადაა დამოკიდებული მათი ინფორმაციული უზრუნველყოფის, კერძოდ მონაცემთა განაწილებული ბაზების აგების ეფექტური მეთოდებისა და ინსტრუმენტების გამოყენებაზე.

70-იანი წლებიდან ჩვენს ქვეყანაში დაიწყო პირველი მართვის ავტომატიზებული სისტემების მიმართულების განვითარება, 80-იანი წლებიდან კი – მონაცემთა ბაზების მართვის სისტემებისა [7,8,41]. თითქმის 40 წლის განმავლობაში ვითარდებოდა თეორიულიად და პრაქტიკულად ეს ორი ძირითადი მიმართულება, საგრძნობლად შეიცვალა ასევე კომპიუტერული ტექნიკა და ინფორმაციული ტექნოლოგიები, გაჩნდა ინტერნეტიც. მართვის ამოცანები, მართვის სისტემები და მართვის ავტომატიზებული სისტემებიც უახლეს ინფორმაციულ ტექნოლოგიებს მოითხოვს.

ასეთ ვითარებაში კორპორაციები, ორგანიზაციული თუ ადმინისტრაციული მართვის სისტემები, დიდი თუ მცირე ბიზნესის ობიექტები – ყველა მოითხოვს საკუთარი საინფორმაციო სისტემების შექმნას, რათა მსოფლიო ბაზარზე გაჩაღებულ კონკურენციაში არ დამარცხდეს. ამისათვის კი საჭიროა არა მხოლოდ ახალი ტექნიკა და ტექნოლოგიები, არამედ უტყუარი ინფორმაციაც ! ინფორმაცია გახდა ერთ-ერთი მნიშვნელოვანი კატეგორია მართვის პროცესის სრულყოფისათვის.

თანამედროვე მართვის ობიექტები, კორპორაციები და ა.შ., რთულ, განაწილებულ სისტემებს წარმოადგენს, რომელთა ტოპოლოგიური სქემა მოიცავს მრავალ განყოფილებას, დეპარტამენტს და ფილიალს, გეგრაფიული ტერიტორიის სხვადასხვა წერტილებში. ამიტომაც განაწილებული მონაცემთა ბაზების დაპროექტებისა და აგების ამოცანა

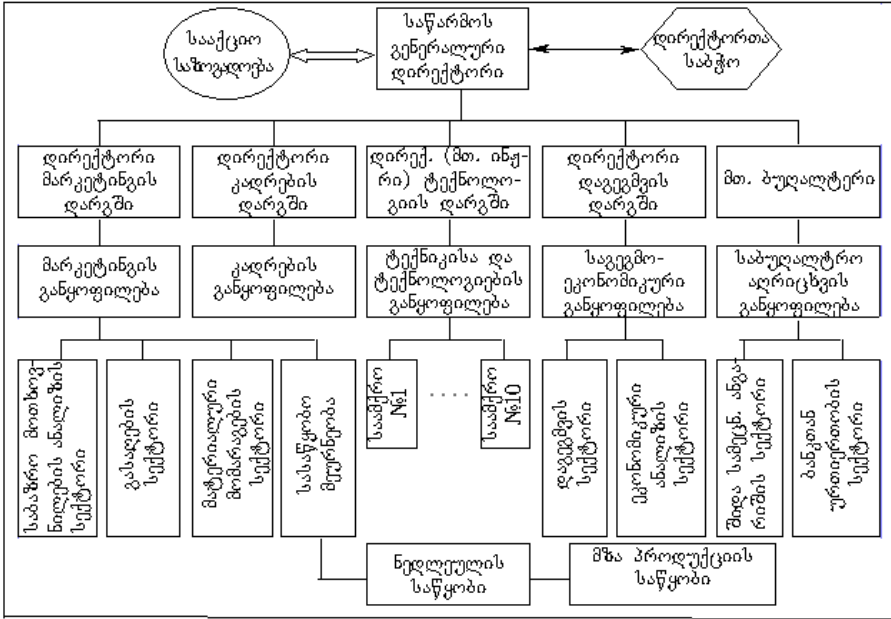
მეტად მნიშვნელოვანი და აქტუალურია კლიენტ-სერვერული არქიტექტურისა და Windows- და Web-სისტემების გარემოში.

წინამდებარე პარაგრაფში განვიხილავთ საწარმოო ფირმის ზოგად მაგალითს და მისი განაწილებული მონაცემთა ბაზების აგების კონცეფციას ქსელურ გარემოში.

ორგანიზაციული მართვის ობიექტი შეიძლება იყოს ნებისმიერი საწარმოო-სამეურნეო ორგანიზაცია (დიდი ან საშუალო საწარმოო ფირმა), რომლის ქვედანაყოფები აღჭურვილია სათანადო ორგტექნიკით, მათ შორის პერსონალური კომპიუტერებითაც. ეს მანქანები შეერთებულია ერთმანეთთან კაბელური არხით და იმართება ერთი ოპერაციული სისტემით. დავუშვათ, ეს ოპერაციული სისტემაა Windows-NT/XP, ქსელში ჩართული N კომპიუტერით, M საერთო გამოყენების პრინტერით, სკანერებით და სხვ. ქსელში რამდენიმე, მაგალითად, S კომპიუტერი ასრულებს სერვერის ფუნქციას, ანუ კოლექტიური გამოყენებისთვისაა გამიზნული და აქვთ მძლავრი კონფიგურაცია.

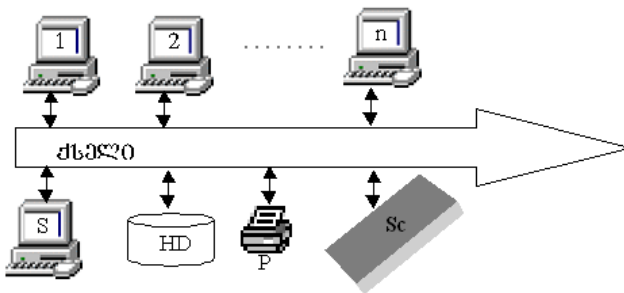
კომპიუტერები მოთავსებულია ორგანიზაციის ხელმძღვანელობის, ფუნქციური განყოფილებების, საამქროებისა და საწყობების ტერიტორიაზე. ამგვარად ქსელის სტრუქტურა, რომელიც „საერთო სალტის“ (ან „ვარსკვლავური“) ტოპოლოგიით შეიძლება იყოს ჩართული, მსგავსი იქნება საწარმოო-ორგანიზაციული სტრუქტურისა, რომელიც მოცემულია 1.1. ნახაზზე.

მართვის ავტომატიზებული სისტემა, რომელიც ამ ლოკალურ ქსელზე იფუნქციონირებს, ფლობს გარკვეულ გამოთვლით რესურსებით, როგორებიცაა სამუშაო ავტომატიზებული ადგილები ფუნქციურ განყოფილებებში, ქსელის არხებს (კაბელები და ქსელის პლატები), მაგნიტური დისკოები (ვინჩესტერები), მათ შორის კოლექტიური მოხმარებისათვის, ოპერაციული მეხსიერებები და პროცესორები, პრინტერები და სკანერები და ა.შ. აქვე უნდა აღნიშნოს, რომ კომპიუტერული ქსელის აქტიური რესურსებია პროგრამები და მონაცემთა ბაზების ფაილები.

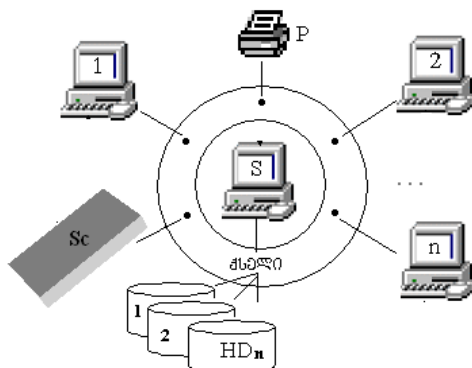


ნახ.1.1. კორპორაციული ობიექტის ორგანიზაციული სტრუქტურა

1.2 ნახაზზე წარმოდგენილი გვაქვს ლოკალური ქსელის კონკრეტული მაგალითით მანქანების „საერთო სალტის“ (ა) და „ვარსკვლავური“ (Switch-ის ბაზაზე) (ბ) ჩართვის პრინციპით.



ნახ.1.2-ა. ქსელი „საერთო სალტე“



ნახ.1.2-ბ. ქსელი „ვარსკვლავი“

ასეთ პირობებში ავტომატიზებული სისტემა უნდა ფუნქციონირებდეს შემდეგი პრინციპით: მონაცემთა ბაზები და პროგრამები გადანაწილებულია ქსელის სამუშაო კვანძებში. ყველა განყოფილებას ან სპეციალისტს აქვს საკუთარი მონაცემთა ბაზა, რომელთანაც მუშაობის უფლება მხოლოდ თვითონ გააჩნია. მაგრამ არის ინფორმაციის ისეთი ბაზებიც, რომლებიც საერთო მოხმარებისაა, მათში ინფორმაციის ჩაწერის უფლება აქვთ მხოლოდ სამომსახურეო რეგლამენტით განსაზღვრულ თანამშრომლებს (სპეციალური სტატუსის მქონე თანამშრომლები). დანარჩენს ექნებათ მხოლოდ წაკითხვის უფლება, ჩასწორების და ჩამატების გარეშე.

დიდი კორპორაციების ლოკალურ კომპიუტერულ ქსელში ძალზედ მნიშვნელოვანია საერთო გამოყენების რესურსების ეფექტური მართვის ამოცანის გადაწყვეტა.

როგორც ცნობილია, მულტიპროცესორული სისტემის დაპროექტებისას გათვალისწინება მრავალი მახასიათებელი, რომელთა ოპტიმალური მნიშვნელობების შერჩევა ძალზე მნიშვნელოვანია და ამავე დროს რთულიც. ამ სიდიდეთა ოპტიმიზაცია არა მარტო გაზრდის კომპიუტერული ქსელის მწარმოებლურობას, არამედ შეამცირებს მის შესაქმნელად საჭირო ხარჯებსაც. მნიშვნელოვანია ისეთი მომენტების გათვალისწინება, როგორცაა სიმძლავრეების, საერთო რესურსების და ა.შ. ოპტიმალური განაწილება.

კომპიუტერულ ქსელებში მიმდინარე მოვლენების (დინამიკური პროცესების) მოდელირებისათვის მოსახერხებელია პეტრის ქსელების

აპარატის გამოყენება, რაოდენობრივი მახასიათებლების ანალიზისათვის კი – მასობრივი მომსახურების სისტემების თეორია [11,13,41].

წინამდებარე ნაშრომში ყურადღებას გავამახვილებთ მართვის ავტომატიზებული სისტემების ინფორმაციული უზრუნველყოფის დაპროექტებისა და რეალიზაციის ამოცანებზე თანამედროვე უნიფიცირებული მოდელირების ტექნოლოგიების გამოყენებით, როგორცაა UML-ტექნოლოგია, მათი დინამიური პროცესების კვლევის მექანიზმების შემუშავებაზე პეტრის ქსელებით.

განვიხილოთ კომპიუტერული ქსელი, სადაც რამდენიმე მომხმარებელი და რამდენიმე სერვერია. დავუშვათ, რომ სერვერთაგან ერთი ასრულებს განმანაწილებლის ფუნქციას, ე.ი. მომხმარებელთაგან იღებს მოთხოვნებს და მას მომხმარებისათვის უზავენის იმ სერვერს, რომელიც თავისუფალია. თუ ყველა სერვერი დაკავებულია, მოთხოვნა დგება და ელოდება ერთ-ერთი მათგანის განთავისუფლების. სერვერი მიიღებს რა მოთხოვნას განაწილებული სერვერიდან, მოემსახურება მას და დაუბრუნებს შედეგს ისევ განაწილებულ სერვერს, რომელიც თავის მხრივ პასუხს აბრუნებს მომხმარებელს. უნდა ვიგულისხმოთ, რომ მოთხოვნები მომხმარებელთაგან გარკვეული სიხშირით მოდის უწყვეტად. თითოეული სერვერი ერთული მოთხოვნის მომსახურებას რაღაც დროს ანდომებს. ამ შემთხვევაში, როდესაც მოთხოვნათა ფორმირების სიხშირე დიდია, განაწილებულ სერვერთან წარმოიქმნება რივი.

კომპიუტერული ქსელის რესურსებიდან ჩვენ განსაკუთრებით გვინდა გამოვყოთ ინფორმაციული რესურსი – კერძოდ მონაცემთა ბაზები. ვინაიდან მოთხოვნათა უმეტესი რაოდენობა ითხოვს ინფორმაციულ ბაზებიდან გარკვეული ფრაგმენტის ამოღებას, ამიტომაც ნაშრომში მონაცემთა ბაზების ინტენსიონალურ და ექსტენსიონალურ რესურსზე გვექნება გამახვილებული ყურადღება.

ინტენსიონალი - კონკრეტულ მონაცემთა ერთობლიობაა, ხოლო ექსტენსიონალი კი ცოდნა (მოდელი, ფორმულა) მათ შესახებ.

1.2. მონაცემთა რელაციური ბაზების სემანტიკური პრობლემები და მათი გაფართოებული მოდელები (კატეგორიული მიდგომა)

მონაცემთა ბაზების თეორია იცნობს იერარქიულ, ქსელურ, რელაციურ და ჰიბრიდულ მონაცემთა სტრუქტურებს, რომლებიც ხანგრძლივი დროის განმავლობაში ვითარდებოდა და კვლავაც ვითარდება [41]. მრავალი მონაცემთა ბაზების მართვის სისტემა ისტორიას ჩაბარდა, ბევრი ახალიც გამოჩნდა, რომეთა შორის ლიდერებია, კერძოდ Oracle, MsSQL Server, MySQL, Intabase, ინსტეცესს, dBase, Visual_FoxPro, Paradox და სხვ. ყველა ეს „გადარჩენილი“ თუ „ახალდაბადებული“ ბაზების მართვის სისტემები რელაციურ მოდელზეა აგებული (ან სუბრელაციურია) [7,33,38,39,42].

მონაცემთა ბაზა დისკრეტული დინამიკური სისტემაა, რომლის მდგომარეობები (რელაციები) მიეკუთვნება ფიქსირებულ მონაცემთა მოდელს. მონაცემთა მოდელი შეიძლება განვმარტოთ, როგორც ალგებრული სისტემა, რომელიც შედგება ლექსიკონური სიმრავლეების (საგნობრივი სფეროს ამსახველი სიტყვების ერთობლიობა) და მათ შორის დამოკიდებულებებისაგან, რომლებზეც შესაძლებელია განხორციელდეს სხვადასხვა სახის ოპერაცია.

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

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

ამ გამონათქვამების ერთობლიობა მოდელიდან ამოსარჩევი ელემენტის სახელეთან ერთად (ატრიბუტების დასახელება) ქმნის მონაცემთა ბაზის ე.წ. სტატიკურ სქემას.

მონაცემთა ბაზა, გადადის რა ერთი მდგომარეობიდან მეორეში (რელაციური ცვალებადობა), აღწერს მონაცემთა მოდელში გარკვეულ ტრაექტორიას. ყოველი მომდევნო მდგომარეობა შეიძლება დამოკიდებული იყოს მის წინა მდგომარეობაზე. ამ მდგომარეობებს შორის კავშირები აღიწერება მთლიანობის დინამიკური შეზღუდვებით (პრედიკატებით), რომლებიც ქმნის მონაცემთა ბაზის ე.წ. დინამიკურ სქემას.

მთლიანობის პრედიკატების ცოდნის საფუძველზე, რომელიც ერთ-ერთ მნიშვნელოვან ნაწილს ფუნქციონალური დამოკიდებულებები შეადგენს, ხდება სინამდვილის (საგნორივი სფეროს) მოდელირება ნორმალურ ფორმათა თეორიის საშუალებით [41]. ფორმირდება მონაცემთა ბაზის სქემა, რომელიც ამ დამოკიდებულებათა სასრულ ერთობლიობას შეადგენს. მაგრამ ფუნქციონალური დამოკიდებულებები ასახავს სინამდვილის შინაარსობრივი მხარის მხოლოდ ნაწილს. მონაცემთა ბაზის მოდიფიკაცია კი სემანტიკური ბუნების მატარებელია. აქედან გამომდინარე, არაა ცხადად წარმოდგენილი, თუ როგორ შეიძლება დამოკიდებულებათა ნორმალიზაციის სინტაქსური პროცედურებით გადაწყდეს სემანტიკური პრობლემები ?

ფორმისა და შინაარსის მიმართების საკითხი განვიხილოთ რამდენიმე კონკრეტული მაგალითით. როგორც აღნიშნული იყო, სინამდვილე მონაცემთა მოდელში აისახება დამოკიდებულებათა სიმრავლით (R). დამოკიდებულება გამოისახება ქვესქემით. ეს უკანასკნელი კი აიგება დამოკიდებულების შესაბამისი მდგომარეობის ანუ რელაციის ანალიზის საფუძველზე. რელაციის ძირითადი ელემენტებია: კორტეჟი ანუ ჩანაწერი, რომელიც წარმოადგენს მონაცემთა დომენების ამონარჩევიდან ფორმირებულ სტრიქონს; ატრიბუტების დასახელებები, რომლებიც განსაზღვრულია დომენებზე და ატრიბუტის მნიშვნელობები, რომლებიც განხილული დომენების ქვესიმრავლეებია. ე.ი. დამოკიდებულების გამოსახვის ძირითადი კონსტრუქციული (ფორმალური) ელემენტები მონაცემთა ბაზაში, არის „ჩანაწერი“. ჩანაწერის იდენტიფიკატორია გასაღებური ატრიბუტი (ან ატრიბუტები).

ვთქვათ, შეიცვალა გასაღებური ატრიბუტის მნიშვნელობა, ე.ი. მოხდა შესაბამის ჩანაწერში მისი სინტაქსური შემადგენლობის ანუ ფორმის ცვლილება ახლით. რას ნიშნავს ეს სემანტიკურად, როგორ შეიცვალა შინაარსი მოცემული ჩანაწერისა ? აი, ამ შეკითხვაზე

მონაცემთა რელაციური ბაზა ვერ გვიპასუხებს. მაგალითად, შეიცვალა თანამშრომლის ნომერი მის ჩანაწერში. რას ნიშნავს ეს – ახალი თანამშრომელი მივიღეთ ძველის მაგიერ, თუ ეს უკანასკნელი გადავიდა სხვა განყოფილებაში? სისტემას არ გააჩნია დამატებით არავითარი ინფორმაცია, რათა პასუხი გავცევს ამ შეკითხვაზე.

ახლა განვიხილოთ ორი დამოკიდებულება $R_1(M,N)$ და $R_2(M,N)$. როგორც ჩანს, $R_1 \subseteq M^*N$ და $R_2 \subseteq M^*N$ ორივე ბინარული დამოკიდებულება განსაზღვრულია ერთსა და იმავე დომენებზე, ე.ი. ფორმალურად ისინი მსგავსი კონსტრუქციისაა. როგორია თითოეულის სემანტიკა? შეიძლება თუ არა მხოლოდ დამოკიდებულებათა საფუძველზე, მათემატიკური თვალსაზრისით გავიგოთ მათი შინაარსების განსხვავება? პასუხი აქ უარყოფითია და ამ შემთხვევაშიც რელაციური მოდელი ვერ ხსნის დასმულ პრობლემას. მაგალითად, $R_1 =$ თანამშრომლის ასაკი და $R_2 =$ თანამშრომლის მუშაობის სტაჟი; $M =$ {თანამშრომლის გვარი}, $N =$ {ნატურალურ რიცხვთა მწკრივი}. ორივე დამოკიდებულების ჩანაწერები ფორმირდება თანამშრომლის გვარისა და ნატურალური რიცხვის მნიშვნელობებისაგან. მაგალითად, <„აბესაძე, 23“>. რას ნიშნავს კონკრეტულად ეს უკანასკნელი, რომ აბესაძე 23 წლისაა, თუ რომ აბესაძეს აქვს 23 წლის სამუშაო სტაჟი, თუმცა ერთი მეორეს არ გამოორიცხავს. ზოგადი ფორმალური ჩაწერით ($R \subseteq$ თანამშრომლის გვარი \times ნატურალური რიცხვი) ვერ ვგებულობთ R დამოკიდებულების შინაარსს.

ჩაწერის ფორმალური კონსტრუქციის გამოყენება ერთგვარად ზღუდავს მონაცემთა ბაზასთან მიმართვის ენის შესაძლებლობებს მაგ., თუ გვაქვს თანამშრომელთა ბაზა და გვინდა ორი შეკითხვის დაბეჭდვა: „რა თანამდებობაზე მუშაობს თანამშრომელი აბესაძე და “მუშაობს თუ არა აბესაძე ორგანიზაციაში“. პირველ შეკითხვაზე რელაციური ბაზა ადვილად უპასუხებს, ვინაიდან იგი ჩანაწერებში მოძებნის „აბესაძეს“ (ატრიბუტი გვარი) და ამოიღებს მის შესაბამის თანამდებობას „უფროსი კონსტრუქტორი“ (ატრიბუტი-თანამდებობა).

მეორე შემთხვევას, რომელიც ერთი შეხედვით მარტივია, რელაციური ბაზა ვერ უპასუხებს, თუ მას არ ექნება დამატებითი პროცედურული საშუალებანი ინფორმაციის ლოგიკური დაბეჭდვისა. ჩვენს შემთხვევაში პასუხი მეორე კითხვაზე უნდა იყოს „კი“ ან „არა“.

ე.ი. „მუშაობს“ ან „არ მუშაობს“. აღნიშნულმა პროცედურამ უნდა გადასინჯოს თანამშრომელთა გვარები ჩანაწერების მიხედვით და თუ იპოვა გვარი „აბესაძე“, მაშინ დააფორმიროს დადებითი პასუხი, თუ არა, უარყოფითი.

სავნობრივი არის სტრუქტურისა და კანონზომიერებების მათემატიკური მოდელირებისას რელაციურ მოდელში გამოიყენება სიმრავლეთა თეორიის ელემენტები. ამიტომ ისმის კითხვა, რამდენად შეესაბამება მონაცემთა ბაზაში ფორმალურად ასახული ცნებები სიმრავლეთა თეორიის ცნებებს (ტერმინოლოგიის მათემატიკური სიზუსტის თვალსაზრისით). მაგალითად, თანამშრომლის-ასაკი (a, b), ე.ი. a თანამშრომელს აქვს b ასაკი. ამავე დროს გვაქვს თანამშრომელთა ერთობლიობა მონაცემთა ბაზაში. მაგრამ „თანამშრომელთა ერთობლიობა“ განისაზღვრება კონკრეტული დროის მომენტით, ხოლო სიმრავლეთა თეორიაში ასეთი ცნება არ არსებობს: a მიეკუთვნება ან არ მიეკუთვნება სიმრავლეს ყოველგვარი დროის ფაქტორზე დამოუკიდებლად.

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

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

რელაციური მოდელების სემანტიკური ანალიზი ფუნდამენტურად ჩაატარა ედგარ კოდმა და თავის ნაშრომში (1970 წ.) გადმოსცა ის ძირითადი მოთხოვნები, რომლებიც დამატებული უნდა იყოს რელაციურ ბაზაში [43]. მისი აზრით სისტემა უნდა გაფართოვდეს ახალი ცოდნის დელექციური გამოყვანის საშუალებებით არსებული ფაქტების საფუძველზე. იგი მნიშვნელოვნად თვლის აგრეთვე მონაცემთა ბაზის დამოკიდებულებებისა და მათი ძირითადი სტრუქტურების ეკრანზე გამოტანის გრაფიკული საშუალებების არსებობას, რაც მომხმარებლისათვის ერთგვარ ფსიქოლოგიურ კომფორტს ქმნის. დასასრულ, მან აღნიშნა, რომ რელაციური მოდელების შემდგომი განვითარება უნდა მოიცავდეს დროის წარმოდგენის საკითხსაც.

ე. კოდის ეს მოსაზრება დრომ და პრაქტიკამ დაამოწმა. კომპიუტერული და ინფორმაციული ტექნოლოგიების განვითარებამ აქტუალური გახადა მონაცემთა საცავების (სუპერ-დიდი და მრავალტიპური მონაცემთა ბაზების ერთობლიობა) ტექნოლოგიების დამუშავება [20]. მისი ერთ-ერთი მნიშვნელოვანი კომპონენტია OLAP-მონაცემთა ოპერატიული ანალიზის პაკეტი. სწორედ ე. კოდმა ისევ, ჩამოაყალიბა (1993 წელს) ძირითადი პრინციპები და მოთხოვნები, რომლებიც რელაციურ მონაცემთა ბაზებთან მჭიდრო კავშირშია [44].

ახლა განვიხილოთ ზოგიერთი ე.წ. „სემანტიკურად გაფართოებული“ რელაციური მოდელი, რომელიც რეალიზებულია ზემოთ აღნიშნული მოსაზრების საფუძველზე.

მაგალითად, მონაცემთა ბაზის დამოკიდებულების სქემის აღწერაში გათვალისწინებულია დამატებითი სემანტიკური ინფორმაციის ჩამატება ბუნებრივი სალაპარაკო ენის კონსტრუქციის მიხედვით. ამ შემთხვევაში დამოკიდებულების სტრიქონი (რელაციის კორტეჟი) ეთანადება ბუნებრივ ენაზე გამონათქვამის შაბლონს. მაგალითად, დამოკიდებულება:

თანამშრომელი ატრიბუტებით გვარი, თანამდებობა, განყოფილება

შეესაბამება შემდეგ ფრაზას:

თანამშრომელი . . . მუშაობს . . . კათედრაზე . . . თანამდებობაზე

თუ წერტილებს შევცვლით სათანადო ატრიბუტების კონკრეტული მნიშვნელობებით, მივიღებთ კორტეჟს:

„თანამშრომელი შურაია მუშაობს რობოტტექნიკის კათედრაზე პროფესორის თანამდებობაზე“

შესაბამის რელაციურ მოდელს სემანტიკური დამატების გარეშე ექნებოდა შემდეგი სახე (ცხრ.1.1):

ცხრ.1.1

თანამშრომელი		
გვარი	კათედრა	თანამდებობა
...
შურაია	ინფორმატიკის	პროფესორი
...

განსხვავება ადვილი შესამჩნევია. რელაციური მოდელის სემანტიკური გაფართოების საშუალებით კორტეჟი ხდება შედარებით ინფორმატიული, ვინაიდან იგი ჩვეულებრივ სალაპარაკო ენის სტრუქტურას მოიცავს. ე.ი. ამ შემთხვევაში წინადადების თითოეულ სიტყვას (არსებით სახელს, ატრიბუტს და მის მნიშვნელობას) აქვს თავისი როლი: სუბიექტი, ობიექტი, დამატება და ა. შ., გამოკვეთილია აგრეთვე მოქმედება (შემასმენელი), რომელიც ერთგვარად ხსნის დამოკიდებულების ატრიბუტთა ურთიერთმიმართებას [41].

არსებობს მონაცემთა და ცოდნის ბაზებისა აგების მრავალი მიდგომა, რომელსაც ჩვენ აქ არ შეგვხვებით, ვინაიდან ეს ცალკე დიდი მიმართულებაა [7,9].

ამ თვალსაზრისით ჩვენს ნაშრომში განხილული გვექნება მონაცემთა ბაზების სქემების, ანუ კონცეპტუალური სტრუქტურების ავტომატიზებული აგების მეთოდები და მოდელები, ძირითადად კატეგორიალური მოდგომის საფუძველზე [23,24]. ვინაიდან სწორედ ასეთი მიდგომით შესაძლებელია სალაპარაკო ენის გრამატიკული კონსტრუქციებისა და ლოგიკურ-ალგებრული მეთოდების ერთობლივი გამოყენებით მივაღწიოთ რელაციური ბაზების მართვის სისტემებში სემანტიკური პრობლემების სრულყოფას.

1.3. მონაცემთა განაწილებული ობიექტ-ორიენტირებული ბაზების მართვის სისტემები და აპროექტების ტექნოლოგია

დღეისათვის აქტუალური და ფართოდ გამოყენებადი პროდუქტები, მონაცემთა ბაზების მართვის სისტემების თვალსაზრისით, არის რელაციური განაწილებული არქიტექტურის მქონე პაკეტები,

როგორებიცაა, მაგალითად Oracle, Ms_Access, Ms_SQL_Server, InterBase, JavaDBC, MySQL და სხვ.

ყველა ეს პროგრამული პაკეტი ობიექტ-ორიენტირებული მონაცემთა ბაზის კონცეფციითაა აგებული, ფლობს მდიდარ კლასთა ბიბლიოთეკებს და ვიზუალურ-კომპონენტურ საშუალებებს [7,32].

კონკრეტული მონაცემთა ბაზის მართვის სისტემის არჩევა უნდა მოხდეს საპრობლემო სფეროს დაპროექტების შემდეგ, როდესაც აგებულ იქნება მისი კონცეპტუალური სქემა და ლოგიკური მოდელი [41]. ამიტომ ჯერ ყოველთვის წყდება თეორიული საკითხები, რომლებიც ზოგადაა, არაა დამოკიდებული კონკრეტულ რეალიზაციაზე.

ჩვენ აქ შევხებით მონაცემთა განაწილებული ბაზების აგების მეთოდებს და მის ეტაპებს, რომლებიც კლასიკურ თეორიაში გამოიყენება [8].

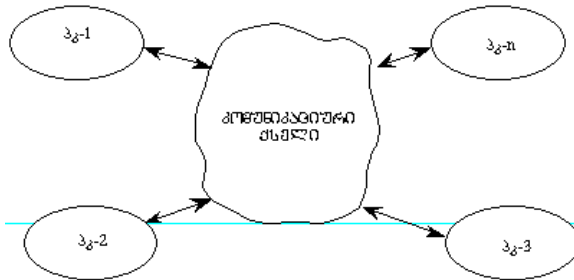
გამომთვლელ მანქანებში ინფორმაცია გადამუშავდება მისი ძირითადი კომპონენტის – პროცესორის საშუალებით. პროცესორების ერთმანეთთან მიერთებით მიიღება მრავალპროცესორიანი ანუ მულტიპროცესორული კომპლექსი. კომპლექსის დანიშნულების მიხედვით პროცესორები შეიძლება სხვადასხვა ხერხით გაერთიანდეს (მიმდევრობით, პარალელურად) ერთ მანქანაში ან მანქანათა ქსელში. ეს საკითხი გვანტერესებს მონაცემთა განაწილებული დამუშავების თვალსაზრისით. ამიტომ, ზოგადად სიტყვაში – „განაწილებული“ ვიგულისხმებთ გეოგრაფიულ (ტერიტორიალურად) სხვადასხვა ადგილზე მდებარე დაკავშირებულ პროცესორებს (ანუ კომპიუტერებს).

შეიძლება გამოვყოთ განაწილებული კომპლექსების რამდენიმე ტიპი: სალტით (შინით) შეერთებული ორი ან რამდენიმე პროცესორი ქმნის ერთიან გამოთვლით კომპლექსს; კაბელით შეერთებული სისტემები გამოთვლითი ცენტრის ერთ სამანქანო დარბაზში; კაბელით შეერთებული სისტემები ერთი შენობის ფარგლებში; მუდმივი ტელეკომუნიკაციური საშუალებებით (სატელეფონო არხები) რამდენიმე კილომეტრის რადიუსში; კავშირგაბმულობის ტელეკომუნიკაციური საშუალებანი ქმნის კოლექტიური მოხმარების ქსელს რეგიონული მასშტაბით (ქალაქი, რესპუბლიკა, დარგი, ქვეყანა და ა.შ.) სატელიტური კავშირის საშუალებების გამოყენებით აგებული ქსელები.

დღეისათვის ერთ-ერთი აქტუალური და მსოფლიოში ფართოდ გავრცელებული ქსელური სისტემის შესანიშნავი მაგალითია Internet, რომლის გამოყენება ჩვენს რესპუბლიკაშიც დღითიდღე იზრდება.

შეიძლება აღვნიშნოთ, რომ ტელეკომუნიკაციური არხები, რომლებიც გამოყენებულია დღეს მონაცემების გადასაცემად, მუშაობს საგრძნობლად ნელა, დაახლოებით 1000-ჯერ ნელა, ვიდრე სამანქანო არხები გამოთვლით ცენტრში. აქედან გამომდინარე, შეიძლება გასაგები იყოს, რომ ერთ სამანქანო დარბაზში თავმოყრილი სისტემებისა და გეოგრაფიულად განაწილებული კომპლექსების ფუნქციები განსხვავებული იქნება.

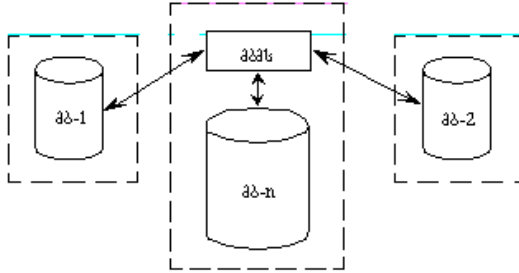
1.3 ნახაზზე ნაჩვენებია გამოთვლითი ქსელის ზოგადი სტრუქტურა მონაცემთა განაწილებული დამუშავებისათვის.



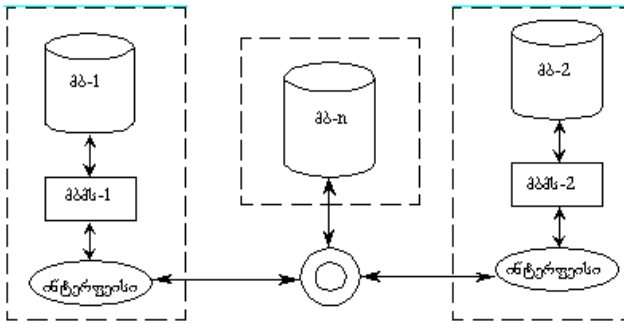
ნახ.1.3 გამოთვლითი ქსელის ზოგადი სტრუქტურა: პკ-პერსონალური კომპიუტერი

როგორც პირველ თავში იყო აღნიშნული, მონაცემთა ბანკი უნდა წარმოვიდგინოთ მონაცემთა ბაზების და მისი მართვის სისტემის ერთობლიობით. ამიტომ, მონაცემთა განაწილებული ბაზებისათვის საჭიროა წინასწარ განისაზღვროს, თუ რა თვალსაზრისით უნდა განაწილდეს იგი – მონაცემების ბაზების დონეზე, ბაზების მართვის სისტემების დონეზე, თუ ორივე ერთად (ნახ. 1.4 და 1.5).

ცენტრალიზებულ ბაზებთან შედარებით ასეთი სახის განაწილებულ სისტემებს აქვს გარკვეული უპირატესობაც და ნაკლოვანებაც. განვიხილოთ ისინი. მონაცემთა განაწილებული დამუშავების სისტემები საშუალებას იძლევა შევათავსოთ მართვის დეცენტრალიზებული და ცენტრალიზებული სისტემების დადებითი მომენტები. შევქმნათ საშუალებანი, რომლებიც უზრუნველყოფს მონაცემთა მთელი ერთობლიობის მთლიანობის დაცვას.



ნახ.1.4. სისტემა განაწილებული მონაცემთა ბაზებით (მბ)



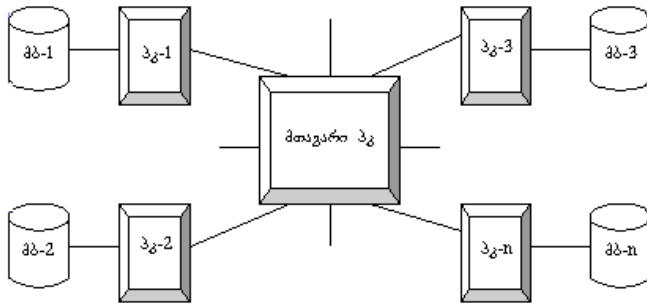
ნახ.1.5. სტრუქტურა მონაცემთა ბაზების მართვის სისტემის (მპმს) განაწილებით

ქსელის რომელიმე კომპონენტის დაზიანების შემთხვევაში სხვა კომპონენტები მუშაობს ნორმალურ რეჟიმში (ცენტრალიზებულის დროს მთლიანად შეჩერდებოდა მისი ფუნქციონირება დაზიანების აღმოფხვრამდე).

შინაარსობრივად ინფორმაციის განაწილებულ ბანკებში მონაცემთა ბაზები „გაფანტულია“ მხოლოდ ფიზიკურად სხვადასხვა კომპიუტერში, ლოგიკურად იგი ერთი ინტეგრირებული ბაზაა. ლოგიკური ინტეგრაცია ნიშნავს, რომ მთლიანი ბაზა პოტენციურად ხელმისაწვდომია მანქანათა ქსელის ყოველი წერტილიდან. ამგვარად, მონაცემთა განაწილებული ბაზების პირობებში შემოდის ახალი ტერმინები: სუპერსქემა, რომელიც ცალკეულ მონაცემთა ბაზების სქემების ლოგიკურად მთლიან ინფორმაციულ ერთობლიობას ეთანადება და ინტეგრირებული ბაზის

ადმინისტრატორი, რომელიც ფუნქციურად უკავშირდება მონაცემთა ლოგიკურ ბაზათა ადმინისტრატორებს.

1.3 ნახაზზე კომუნიკაციური ქსელის (ლოკალური ქსელი) როლს ასრულებს თვით კომპიუტერი (ნახ.1.6). როგორც ცნობილია, არსებობს სხვადასხვა ვარიანტი აღნიშნული ქსელების ორგანიზაციისა (იერარქიული, ვარსკლავისებრი და ა.შ) ერთი მთავარი მანქანით (ასრულებს მონაცემთა განაწილებული ბაზების მართვის ფუნქციას).



ნახ.1.6. მთავარი პკ ასრულებს კომუნიკაციურ ფუნქციებს ქსელის კვანძებს შორის

ქსელური კომპონენტის საშუალებით (სერვერ-მანქანით) ხორციელდება ისეთი ფუნქციები, როგორცაა ამა თუ იმ მონაცემთა ერთობლიობის მოძებნა ქსელში, მის განსაზღვრულ ნაწილში შემოსული მოთხოვნის გადაგზავნა დასამუშავებლად, მოთხოვნების დამუშავების კოორდინაცია, მონაცემთა სხვადასხვა ტიპის სტრუქტურების გარდაქმნა, ინფორმაციის დაცვის საკითხები და ა.შ.

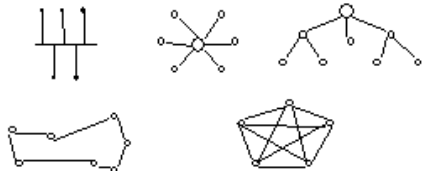
ინფორმაციის ბანკებს თავისი ხარვეზებიც აქვს. ეს პირველ რიგში ეხება რეალიზაციისა და ექსპლუატაციის სირთულეებს. ცხადია, რომ მონაცემთა განაწილებული ბაზების მართვის სისტემას აქვს საგრძნობლად მეტი ფუნქცია, ვიდრე ჩვეულებრივ (ცენტრალიზებულ) მონაცემთა ბაზების მართვის სისტემას. გართულებულია მოთხოვნების (შეკითხვების) დამუშავების სინქრონიზაციისა და მათი კორექტირების პროცესები. ყველაზე მეტად საყურადღებოა განაწილებულ ბანკებში (ქსელებში) არასანქციური მიმართვებისაგან ინფორმაციის დაცვის

საკითხი. ეს კი მოითხოვს სათანადო დაცვის პროგრამების (პაროლური სისტემები) შექმნას და რეალიზაციას.

აქვე საჭიროა აღინიშნოს ფაქტი, რომ დღეისათვის ინფორმაციის გადაცემა არხების საშუალებით (ქსელებში) საკმაოდ ძვირია, საიმედოობა კი დაბალია, რაც გარკვეულად ზღუდავს მისი გამოყენების ფარგლებს. ამ საკითხთან უშუალო კავშირშია ინფორმაციის დუბლირების ასპექტი. ინფორმაციის სიჭარბის მოცულობის თვალსაზრისით ასხვავებენ მთლიანად ჭარბ განაწილებულ ბაზებს (ქსელის ყოველ კვანძში არსებობს მთლიანი ინფორმაციის დუბლი) და განაწილებულ ბაზებს ინფორმაციის ერთჯერადი შენახვით (სიჭარბე 0%).

შუალედურია ნაწილობრივ სიჭარბიანი სისტემები, რომლებშიც წინასწარ განისაზღვრება, რომელი მონაცემები, სიჭარბის რა %-ით, ქსელის რომელ კომპონენტებში უნდა იქნეს განაწილებული.

ინფორმაციის განაწილებული ბანკები ქსელის ტოპოლოგიის შესაბამისად შეიძლება იყოს სალტური, იერარქიული, ქსელური და ვარსკლავისებრი (ნახ.1.7). განაწილებული ბანკის ტოპოლოგია სავალდებულო არაა ზუსტად ემთხვეოდეს თვით ქსელის ტოპოლოგიას. ამიტომ ასხვავებენ მონაცემთა ბანკების ფიზიკურ და ლოგიკურ ტოპოლოგიებს. ფიზიკური ტოპოლოგია განსაზღვრავს ქსელში მოთხოვნის დაბუშვების რეალურ გზას, ხოლო ლოგიკური ტოპოლოგია – კავშირებს მონაცემთა ბაზებსა და მომხმარებლებს შორის (ფიზიკური რეალიზაციის გარეშე).



ნახ.1.7. მანქანათა ქსელის ტოპოლოგიის ტიპები

ამგვარად, ინფორმაციის განაწილებული ბანკების შექმნა და გამოყენება სამეურნეო ობიექტების მართვის პროცესში აუცილებელი და სასურველი მომენტია. ჩვენს ქვეყანაში აღნიშნული საკითხი (პრაქტიკული რეალიზაციის თვალსაზრისით) ჯერჯერობით არაა გადაჭრილი როგორც სათანადო კომპიუტერული ტექნიკის, ასევე კავშირგაბმულობის აუცილებელი ელემენტების (გამოთვლითი კომპლექსების ქსელის კომპონენტების) დეფიციტის გამო. ამიტომ

შეიძლება ითქვას, რომ ინფორმაციის განაწილებული ბანკების აგება და მისი ეფექტური გამოყენება ჩვენი ხვალისდელი დღის საქმეა.

მონაცემთა განაწილებული ბაზების დაპროექტება და რეალიზაცია ექვს ეტაპს მოიცავს (ნახ. 1.8). როგორც ნახაზიდან ჩანს, საგნობრივი არის არაფორმალისზეული აღწერა ხორციელდება საკვლევე ობიექტის სისტემური ანალიზისა და მომხმარებელ-ექსპერტთა ინფორმაციული მოთხოვნების ანალიზის საფუძველზე.

1-ელ ეტაპზე ფორმირდება საპრობლემო სფეროს ძირითად ცნებათა ლექსიკონი.

მე-2 ეტაპზე დაპროექტდება საგნობრივი არის კონცეპტუალური (სემანტიკური) მოდელი. შედეგად ფორმირდება მონაცემებისა და პროცედურების გლობალური სქემები (ცოდნის ბაზა).

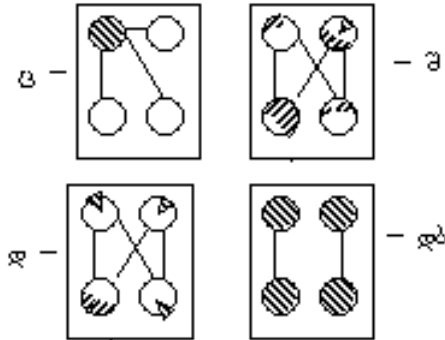
მე-3 ეტაპზე წყდება გლობალური ლოგიკური მოდელის დაპროექტების საკითხი. ამ მოდელის რეალიზება შეიძლება მონაცემთა ერთ-ერთი კლასიკური მოდელის (რელაციური, ქსელური ან იერარქიული) გამოყენებით.

ამ ეტაპზე ფორმირდება გამარტივებული გლობალური სქემა და ლოგიკური მიმართვის ცხრილები, რომლებიც ეთანადება შესაბამისად მონაცემთა აბსტრაქტულ მოდელს (სემანტიკას) და მათი წვდომის გზებს (ურთიერთკავშირებს).

მე-4 ეტაპი დამახასიათებელია მხოლოდ განაწილებული ბაზებისათვის, ამიტომ მას შევეხებით დეტალურად სამი ძირითად ქვეეტაპის სახით. განაწილების დაპროექტების ეტაპზე გაითვალისწინება როგორც აპარატურის, ოპერაციული სისტემის, მონაცემთა განაწილებული ბაზების მართვის სისტემის (მგბმს) და ქსელის ტოპოლოგიის მახასიათებლები, ასევე ბაზის განაწილების მოთხოვნების ანალიზის შედეგები. ჩვენი მიზანია გლობალური ლოგიკური სქემის „დახლენა“ ცალკეულ ლოგიკურ ქვესქემებად (ფრაგმენტაცია), რათა ისინი ლოკალური ქსელის კვანძებში განაწილდნენ.

1-ელ ქვეეტაპზე ფორმირდება განაწილებული ბაზის ცალკეული ნაწილის გამოყენების სინშირეთა ცხრილი, წინააღმდეგობებისა და დაყოფის ცხრილები. წინააღმდეგობრიობის ცხრილებში აისახება მოთხოვნების დამუშავების ლოკალიზაციაზე ბაზების განაწილების გავლენის რაოდენობრივი შეფასება, დაყოფის ცხრილებში კი – გლობალური სქემის თითოეული ობიექტის განაწილების კრიტერიუმები.

მე-2 ქვეტაბაზე უნდა გადაწყდეს ბაზის ლოგიკურ ფრაგმენტებად დანაწევრების ამოცანა. ამ დროს მნიშვნელოვანია ბაზების განაწილების სწორი სტრატეგიის განსაზღვრა. როგორც აღვნიშნეთ, ამის 4 ალტერნატივა არსებობს (ნახ. 1.9): (ც) – ცენტრალიზაცია (გვაქვს ბაზის ერთი პირი მხოლოდ ერთ მანქანაში, სხვები სარგებლობენ მასთან მიმართებით და მონაცემების დროებითი გადაწერით თავიანთ მეხსიერებაში), (დ) – დანაწევრება (გვაქვს ბაზის ერთი პირი, რომელიც დანაწევრებულია ცალკეულ კვანძში. ამასთან, დუბლირება 0%-ია), (დბ) – დუბლირება (ქსელის ყველა მანქანაში გვაქვს ბაზის სრული პირი) და (შ) – შერეული (ყოველ კომპიუტერში შეიძლება მოთავსდეს ბაზის ნებისმიერი ფრაგმენტი).



ნახ.1.9. ალტერნატიული სტრატეგიები

არსებობს ჰორიზონტალური და ვერტიკალური ფრაგმენტაცია. ჰორიზონტალურის დროს ბაზა (ფაილები) „იხლიჩება“ სტრიქონებად გარკვეული პრედიკატის მიხედვით. ფრაგმენტი შეიძლება იყოს მინიმუმ ფაილის ერთი სტრიქონი (კორტეჟი) ან მაქსიმუმ მთლიანი ფაილი (რელაცია). ვერტიკალური ფრაგმენტაციის დროს ბაზა იყოფა ლოგიკური სქემის ატრიბუტების მიხედვით. ამ დროს შესაძლებელია ნორმალურ ფორმათა თეორიის გამოყენება, რომელიც ადგენს ბაზის ფაილების ოპტიმალურ ლოგიკურ სტრუქტურებს. ოპტიმალურობის კრიტერიუმია მოთხოვნების სწრაფი დამუშავება, რაც შესაძლებელია ატრიბუტების ერთობლიობის (დამოკიდებულების) სწორად დაკომპლექტების შემთხვევაში.

ბაზის განაწილების სტრატეგიის შესაფასებლად განიხილება შემდეგი ძირითადი ფაქტორები: ექსპლუატაციის სიმარტივე (ან სირთულე), საიმედოობა, მონაცემთა ამორჩევის ეფექტურობა, გადაგზავნის ლოკალიზაციის ხარისხი, განაზღვრების სირთულე, ბაზის მთლიანობის დაცვა, მოქნილობა, პარალელური დამუშავების შესაძლებლობა, მონაცემთა სიჭარბე, ქსელისა და მანქანების მენსიერების მოცულობა, კავშირების რეალიზაციის ღირებულება, ქსელის თანაბარი დატვირთვა გადაცემების დროს და ა.შ.

ბაზის ცენტრალიზაციის სტრატეგიის სიკეთეა ქსელში მისი ექსპლუატაციის სიმარტივე, მაგრამ იგი შეზღუდულია ერთი კომპიუტერის მენსიერების მოცულობით და ნაკლებად საიმედოა (თუ მანქანა გამოვიდა მწყობრიდან, მაშინ ქსელის ვერც ერთი სხვა კვანძი ვერ იღებს მონაცემებს მოთხოვნების დასამუშავებლად).

დუბლირების სტრატეგია ხასიათდება მაქსიმალური საიმედოობით, მონაცემთა ამორჩევის ეფექტურობით და ბაზის ამორჩევის სიმარტივით. მისი ნაკლია მონაცემთა დიდი სიჭარბე (N-დუბლის არსებობა), რაც განაპირობებს დანახარჯების ზრდას მეორეულ მენსიერებასა და ბაზის განაზღვრების პროცესების სინქრონიზაციაზე (ბაზის ყველა პირის შეთანხმება).

დანაწევრების სტრატეგიის დროს მენსიერების მოცულობა მონაცემთა ბაზისათვის შემოიფარგლება არა ერთი ეგმ-ით, არამედ ქსელის ყველა კვანძის მენსიერებათა ჯამით. ცენტრალიზებულთან შედარებით გაზრდილია საიმედოობა და მონაცემებთან მიმართვის ლოკალიზაციის ხარისხი. მაგალითად, თუ ერთი მანქანა გამოვიდა მწყობრიდან, დანარჩენს მაინც შეუძლია ისეთი მოთხოვნების დამუშავება, რომლებიც არ მოითხოვს საჭირო მონაცემებს დაზიანებული კომპიუტერიდან. დანაწევრების სტრატეგია მოითხოვს დამატებით ხარჯებს ბაზების დანაწილების ცნობარების შესანახად.

შერეული სტრატეგია აერთიანებს დანაწევრებული და დუბლირებული სტრატეგიების დადებით მხარეებს. მაღლდება სისტემის მოქნილობა (ვინაიდან მოძებნილია გარკვეული კომპრომისი ქსელის მთლიან მენსიერებასა და ცალკეულ მანქანებს შორის), იზრდება მიმართვის ლოკალიზაციის ხარისხი (დუბლირების გამო), შესაძლებელია მონაცემთა პარალელური დამუშავების რეალიზაცია და საიმედოობის რეგულირება (ფრაგმენტის პირის რაოდენობის ცვლილებით). ამ

სტრატეგიის უარყოფითი მხარეა მაღალი დანახარჯები განაწილებული ბაზების ფრაგმენტთა პირების საცნობარო ცხრილების არსებობასა და ამ ფრაგმენტების სინქრონიზაციაზე (შეთანხმებაზე). მოთხოვნების დამუშავება და მათი ოპტიმიზაცია საკმაოდ პრობლემატურია, ამიტომაც შერეული სტრატეგია ბაზების განაწილების დროს გამოიყენება მხოლოდ მაშინ, როცა დანარჩენი სტრატეგიები არაეფექტურია.

მე-2 ქვეეტაპზე დაპროექტდება აგრეთვე განაწილებული ბაზის ქსელური ცნობარი (ქც). ამ ცნობარში შეინახება ინფორმაცია ბაზის ფრაგმენტების პირების ქსელის მანქანებში განაწილების შესახებ. ყველა მოთხოვნა იყენებს ქსელურ ცნობარს, რათა „მიაგნოს“ საჭირო მონაცემების ადგილმდებარეობას. ხშირ შემთხვევაში ასეთი ცნობარები საკმაოდ მოცულობისაა და ქსელის კნაქებში მათი მოთავსება მოითხოვს განსაკუთრებულ ყურადღებას. აღნიშნული საკითხის გადაჭრა შესაძლებელია ზემოთ განხილული განაწილების სტრატეგიების გამოყენებით. ამგვარად, ქსელური ცნობარი შეიძლება იყოს ცენტრალიზებული ერთ მანქანაში, დუბლირებული ყველა ეგმ-ში ან განაწილებული კვანძებში მოთხოვნათა ლოკალიზაციის წინასწარი ანალიზის საფუძველზე. განაწილებული და შერეული სტრატეგიების დროს საჭიროა ქსელური ცნობარის დანაწილების ცნობარის (ცნობარის ცნობარი) აგება, რომელიც მოთავსდება ქსელის ყველა კვანძში (მოძებნის პროცედურების ეფექტურობის ასამაღლებლად).

მოცემულ ქვეეტაპზე ერთ-ერთი მნიშვნელოვანი საკითხია განაწილებული ბაზების განახლების პროცესების დაპროექტება. ეს ამოცანა დაკავშირებულია ბაზის მთლიანობისა და ექსპლუატაციის სიმარტივის ფაქტორებთან და წყდება ე.წ. „დიფერენციალური ფაილების“ (დფ) გამოყენებით. დიფერენციალური ფაილი არის ცალკე ფაილი (თავიდან ცარიელი), რომელშიც გროვდება ბაზის ფაილების შეცვლილი მონაცემების მნიშვნელობები. პერიოდულად (დფ-ფაილის გარკვეული მოცულობის შევსებისას) მოხდება ამ ფაილის შერწყმა ძირითად ბაზასთან და იგი კვლავ გასუფთავდება. დიფერენციალური ფაილის გამოყენების იდეა განაწილებული ბაზის განახლების პროცედურების ჩატარების ხარჯების შემცირებაშია. განახლების ხარჯები მაღალია, როცა ცვლილებები ბაზებში ხორციელდება სისტემატურად, უშუალოდ განახლების ტრანზაქციის შესრულების დროს (დაგროვების გარეშე). მეორე მხრივ, დიფერენციალური ფაილის

არსებობა ზრდის მიმართვის დროს, რადგან ნებისმიერი მოთხოვნა ჯერ მიმართავს ამ ფაილს და შემდეგ ძირითად ბაზას. საპრობლემო სფეროს მიზნებიდან გამომდინარე, უნდა განისაზღვროს განაწილებული ბაზების განახლების რეგლამენტი დიფერენციალური ფაილების გამოყენებით.

მე-3 ქვეტაპი ეხება ბაზის დანაწილებული ლოგიკური ფრაგმენტების ქსელის კვანძებში განლაგების რაციონალური (ან ოპტიმალური) ვარიანტის შერჩევას. ცენტრალიზებული ან დუბლირებული სტრატეგიების გამოყენების დროს (მე-2 ქვეტაპზე) შედარებით ტრივიალურია განლაგების ამოცანის გადაწყვეტა. საჭიროა მხოლოდ განისაზღვროს, თუ ქსელის რომელ მანქანაში მოთავსდეს ცენტრალიზებული ბაზა ან რომელ კვანძებში განლაგდეს ბაზის პირები. დანაწევრებული ან შერეული სტრატეგიების გამოყენებისას რთულდება განლაგების ამოცანის გადაჭრა. აქ საჭიროა დადგინდეს, თუ რომელი ლოგიკური ფრაგმენტი რომელ კვანძებში განლაგდეს, რამდენი იყოს მათი პირების რაოდენობა და ა.შ. ამას ემატება ქსელის ცნობარების განლაგების საკითხიც, რომლის ეფექტურ გადაწყვეტაზეც დამოკიდებულია სისტემის მწარმოებლობა. შესაძლოა მრავალი ვარიანტის განხილვა (ვერტიკალური მიდგომით) ან წრფივი დაპროგრამების აპარატის გამოყენება. შედეგების შესაფასებლად საჭიროა დამუშავდეს სპეციალური პროგრამები.

მე-5 ეტაპზე ბაზის ცალკეული ლოგიკური ფრაგმენტებისა და მათთან მიმართვის ცხრილების საფუძველზე ხდება ლოგიკური დაპროექტება. გათვალისწინებულია ქსელის თითოეული კვანძის აპარატული და კონკრეტული მბმს-ის მახასიათებლები.

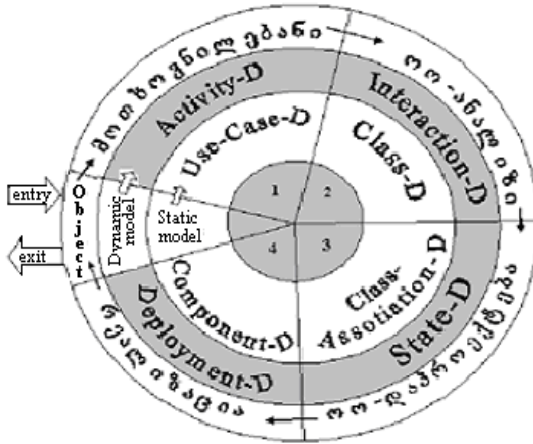
მე-6 ეტაპზე ხდება ლოკალური ლოგიკური სქემების ფიზიკური დაპროექტება, რაც გულისხმობს მონაცემთა ფაილების ორგანიზაციის აღწერას მანქანების პირდაპირი მიმართვის მეხსიერებაში და მათთვის მიმართვის მეთოდების დადგენას.

1.4. უნიფიცირებული მოდელირების ენა (UML) და მონაცემთა ბაზების დაპროექტების ამოცანა

ახლა განვიხილოთ UML-ტექნოლოგიის განზოგადებული კონცეფცია და მისი, როგორც კომპიუტერული სისტემების

ავტომატიზებული დაპროგრამების მეთოდოლოგიური საფუძლის კონკრეტული რეალიზაციის ეტაპები.

პროგრამული პაკეტების შექმნა UML-ტექნოლოგიის მიხედვით ოთხ ეტაპად სორციელდება (ნახ.1.10).



ნახ.1.10. UML-ეტაპები

საკვლევი ობიექტის ავტომატიზაციის მოთხოვნილებების დადგენა, მისი ობიექტ-ორიენტირებული (ოო) ანალიზი, ოო-დაპროექტება (დეტალური დონე) და რეალიზაცია (პროგრამული კოდი). ეს ეტაპები სტატიკური და დინამიკური დიაგრამებით (D) სორციელდება.

UseCase-D დიაგრამა უჩვენებს როლებს (Actor) და მათ ფუნქციებს (Action), აგრეთვე მათ კავშირებს (ნახ.1.11);

ყოველ UseCase-ფუნქციას (ოვალს) შეესაბამება ერთი დინამიკური მოდელი, რომელიც **Activity-D** დიაგრამის სახით ფორმირდება (ნახ.1.12). აქტიურობათა დიაგრამას ერთი დასაწყისი და რამდენიმე დასასრული შეიძლება ჰქონდეს. მასში მონაწილეობს რამდენიმე "როლის" შემსრულებელი (მაგ., ლექტორი, სტუდენტი, დეკანი და ა.შ.).

მრავალკუთხედებში მოთავსებულია მათ მიერ შესასრულებელ პროცედურათა დასახელებები. პროცედურები შეიძლება შესრულდეს მიმდევრობით ან პარალელურად.

1.5. განაწილებული სისტემის ბიზნეს-პროცესების მოდელირების და ანალიზის ინსტრუმენტი პეტრის ქსელების ბაზაზე

აღნიშნული საკითხების ასახსნელად და უკეთ აღსაქმელად განვიხილოთ გაფართოებული პეტრის ქსელების ერთ-ერთი კონცეფცია და მისი ინსტრუმენტი, რომლის გამოყენებითაც შემდგომში შევძლებთ პროცესების მოდელირებას.

განვიხილოთ მაღალი დონის პეტრის ქსელთა ტიპები. ყოველ პეტრის ქსელს, ჩვეულებისამებრ, გააჩნია შემდეგი საერთო ელემენტები - პოზიციები, - გადასასვლელები, - რკალები (წიბოები), რომლებითაც პეტრის ქსელის გრაფი იქმნება.

პეტრის ქსელის ახალი ტიპის განსაზღვრისას საფუძვლად სწორედ პეტრის ქსელის გრაფია აღებული და იგი შემდგომი ასახვებითა და ფუნქციებით პეტრის ქსელის კონკრეტულ ტიპამდე ფართოვდება.

პეტრის ქსელის სხვადასხვა ტიპები ერთმანეთისგან შეიძლება განსხვავდებოდეს მარკერთა ტიპებით და მათგან გამომდინარე ერთიანი მარკირების სისტემით, ქსელის ელემენტების აღწერით (ჭდეები) ან/და გადასასვლელთა გაშვების წესებით.

ჭდეები პეტრის ქსელის ელემენტებზე, ძირითადად მხოლოდ წარწერებია და შეიცავს ელემენტის სინტაქსს, მაგრამ არა სემანტიკას. შესაბამისად, ისინი ქსელის შესრულების პროცესში ვერაფერს ცვლის. ჭდეების დანიშნულება პეტრის ქსელის სინტაქსური კონტროლია.

ამის მიუხედავად, ახალი ტიპის ჭდის განსაზღვრა უკვე საკმარისია იმისთვის, რომ ახალი პეტრის ქსელის ტიპი იქნეს განსაზღვრული.

პოზიციებზე, გადასასვლელებზე ან/და რკალებზე დროითი დაყოვნების განსაზღვრას დროითი პეტრის ქსელის ტიპი შემოაქვს, დაყოვნების დროთა აღბათურ განაწილებას – სტოქასტური პეტრის ქსელის ტიპი და ასე შემდეგ.

პეტრის ქსელების ტიპების განსაზღვრის უფრო ფართო გარემო მაღალი დონის პეტრის ქსელების ქვეკლასებია (მაგალითად, სისტემური პეტრის ქსელები). ქვემოთ მოკლედ აღვწერთ ყველაზე კარგად დამუშავებულ და გავრცელებულ პეტრის ქსელის ტიპებს.

ფერადი პეტრის ქსელები (Coloured Petri Nets) მაღალი დონის პეტრის ქსელების კლასში შედის და სხვადასხვა ტიპის ანუ ფერის მარკერებს შეიცავს [13]. ტერმინი „ფერადი“ მხოლოდ ტრადიციისთვის გამოიყენება და ქსელში განსხვავებული მარკერების არსებობაზე მიანიშნებს - ამგვარი ქსელების დაბალი დონის პეტრის ქსელებისგან გამოსარჩევად, რომლებიც ერთგვაროვან, „შავ“ მარკერებს შეიცავს. ფერადი პეტრის ქსელის სტრუქტურა საკმაოდ რთულია და იგი მრავალი სახეობის ჯღეებს შეიცავს.

დროითი პეტრის ქსელები, ფაქტობრივად, ყოველი ტიპის პეტრის ქსელისთვის დროითი გაფართოების დამატებით მიიღება. დროითი გაფართოება აუცილებელია რეალური საპრობლემო სფეროს მოდელირებისთვის, მის გარეშე პეტრის ქსელი მხოლოდ სისტემის რაოდენობრივი ანალიზისთვის გამოდგება.

დროითი პეტრის ქსელის 4 ტიპი არსებობს: პოზიციურ-დროითი (Timed Places Petri Nets - TPPNs), ტრანზიციულ-დროითი (Timed Transition Petri Nets - TTPNs), რკალურ-დროითი და მარკერულ-დროითი [11].

პოზიციურ-დროითი ტიპისთვის განისაზღვრება დაყოვნების ერთი და იგივე დრო პოზიციაში მოთავსებული ყველა მარკერისთვის და დროის ათვლა იწყება შესაბამისი გადასასვლელის გააქტიურებისთანავე (როცა მისი გახსნა ნებადართული ხდება). ყველა შემავალი პოზიციის დაყოვნების დროის გასვლის შემდეგ გადასასვლელი გაიხსნება.

ტრანზიციულ-დროით პეტრის ქსელებში დაყოვნების დრო გადასასვლელისთვის (ტრანზიციისთვის) განისაზღვრება. პეტრის ქსელების ეს ტიპი 2 ქვეტიპს შეიცავს: წინასწარი არჩევანისა და შეჯიბრების მოდელებს.

წინასწარი არჩევანის შემთხვევაში გადასასვლელი გააქტიურებისთანავე იღებს მონოპოლურ უფლებას ყველა შემავალ პოზიციაში მოთავსებულ მარკერების იმ ოდენობაზე, რაც მისი გახსნისთვის აუცილებელია (სხვა პოზიციებთან კონფლიქტში იმარჯვებს). ამის შემდგომ იწყება დაყოვნების დროის ათვლა. მისი გასვლისთანავე გადასასვლელი გაიშვება პეტრის ქსელის წესების მიხედვით, ანუ გადასასვლელის გააქტიურებას აუცილებლად მისი გახსნა მოჰყვება.

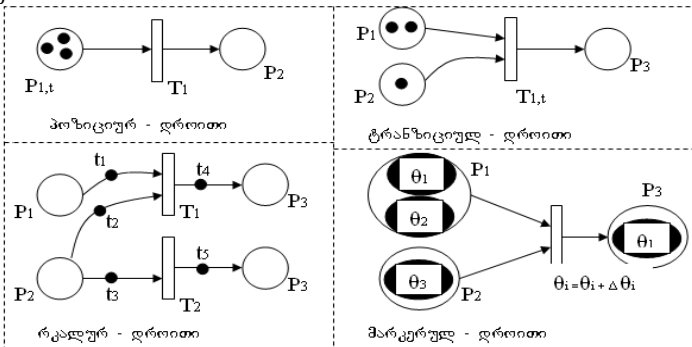
შეჯიბრის მოდელში მთავარი დროითი ფაქტორია, მარკერები ყველა აქტიურ გადასასვლელს ეკუთვნის და გაივლის მას, რომლის დაყოვნების დროც უფრო მაღე გავა.

რკალურ-დროით პეტრის ქსელში დროითი დაყოვნების სიდიდეები რკალებს ენიჭება, განისაზღვრება მარკერის რკალში „მოგზაურობის“ დრო და გადასასვლელის გახსნა შესაძლებელია მხოლოდ მაშინ, როცა ყველა შემავალ რკალში მოძრავი მარკერი უწყევს გადასასვლელს.

მარკერთა „გადაადგილება“ გადასასვლელისკენ იწყება მხოლოდ მაშინ, როცა გადასასვლელის გახსნა ნებადართული ხდება. გახსნის შემდგომაც ყოველ რკალს ენიშნება მასში მარკერის „გადაადგილების“ დრო, სანამ იგი გამოძვალ პოზიციას მიაღწევს.

მარკერულ-დროითი პეტრის ქსელი ყოველი მარკერისათვის ცალკე დაყოვნების დროის განსაზღვრას მოითხოვს. ამგვარი ტიპი მოხერხებულია დროითი პრიორიტეტების მოდელირებისთვის.

დროითი პეტრის ქსელის სხვადასხვა ტიპები 1.11 ნახაზზეა მოცემული. ტრანზიციულ-დროით პეტრის ქსელებს, სადაც გადასასვლელის დაყოვნების დრო შემთხვევით განაწილებულ ექსპონენციალურ ფუნქციას წარმოადგენს, ალბათური ანუ სტოქასტური პეტრის ქსელები (Stochastic Petri Nets) ეწოდება, ხოლო სტოქასტური პეტრის ქსელი, რომელიც დროითთან ერთად არადროით (მყისიერ) გადასასვლელებსაც შეიცავს, განზოგადებულ სტოქასტურ პეტრის ქსელს (Generalized Stochastic Petri Nets) წარმოადგენს [13]. ამგვარი ქსელის ქცევა ალბათური (მაგალითად, მარკოვის) პროცესებით აღიწერება.



ნახ.1.11. დროითი პეტრის ქსელის ტიპები

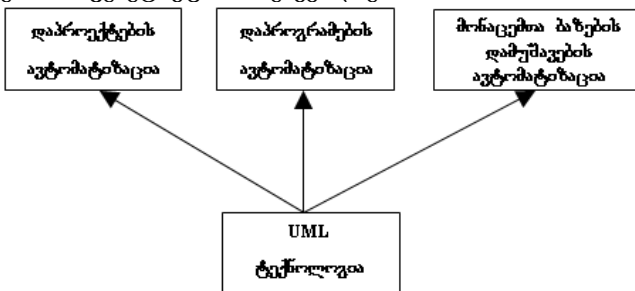
II ტაზი

ბანაწილებული სისტემების ინფორმაციული რესურსების მართვის პროცესების მოდელირება UML/PN ტექნოლოგიით

2.1. ინფორმაციული რესურსების მართვის პროცესების ტექნოლოგია, მათი ასახვა სტატიკური და დინამიკური მოდელებით

ახალი საუკუნის დასაწყისი კომპიუტერული ინდუსტრიისა და ინფორმაციული ტექნოლოგიების განვითარების მაღალი დონით ხასიათდება. მსოფლიო ბაზარზე გამოჩნდა არა ერთი ახალი აპარატურული (Hardware) და პროგრამული (Software, Groupware) სისტემები. მათ შორის საყურადღებოა ამერიკული კორპორაციის Rational Software მეცნიერ-კონსტრუქტორების გრადი ბუჩის, ჯეიმს რამბოს და ივარ ჯაკობსონის ახალი, UML-ტექნოლოგია კომპიუტერული სისტემების პროგრამული უზრუნველყოფის ასაგებად [1,2]. უნიფიცირებული მოდელირების ენა (Unified Modeling Language), როგორც უახლესი სტანდარტი, საფუძვლად დაედო თანამედროვე ინფორმაციული ტექნოლოგიების სისტემებს და მას აქტიურად იყენებს IBM, Hewlet Packard, Oracle და სხვა ფირმები [3].

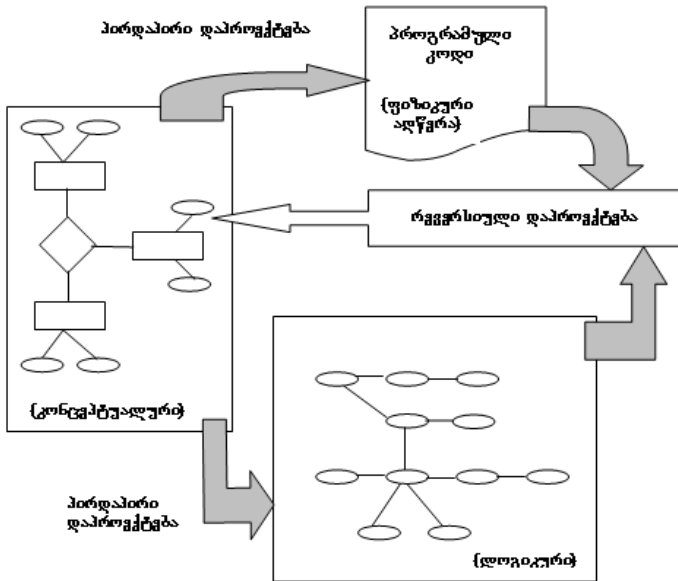
UML-ტექნოლოგია ობიექტ-ორიენტირებული მოდელირებისა და სტრუქტურული დაპროგრამების იდეოლოგიის მატარებელია, რომელიც თეორიული და პრაქტიკული ინფორმატიკის სამი ძირითადი მიმართულებისა „გენეტიკური“ მემკვიდრეა (ნახ.2.1):



ნახ.2.1. UML ტექნოლოგიის კონცეფცია

მართვის უზრუნველყოფის	კომპიუტერული აგების პროცესების	სისტემების ასეთი	პროგრამული სრულფასოვანი
--------------------------	-----------------------------------	---------------------	----------------------------

ავტომატიზაცია ვიზუალური მოდელირების სახელწოდებით დამკვიდრდა და იგი მოდელის გრაფიკულ წარმოდგენას ეკრძნობა. ასეთი ინსტრუმენტები ფლობს როგორც პირდაპირ (გრაფიკიდან პროგრამული კოდისაკენ), ასევე რევერსიულ (კოდიდან გრაფიკისაკენ) ტექნოლოგიას (ნახ.2.2):



ნახ.2.2. რევერსიული დაპროგრამების კონცეფციის მოდელი მონაცემთა ბაზებისთვის

შეიძლება დავასკვნათ, რომ UML–ტექნოლოგია (ენა) დაპროგრამების ენებისა და მონაცემთა ბაზების მართვის სისტემების ერთგვარ სიმბიოზს, ინტეგრირებულ ინსტრუმენტს წარმოადგენს [4,8].

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 [5].

რამ განაპირობა ასეთი ტიპის ინტეგრირებული მძლავრი კომპიუტერული ინსტრუმენტის შექმნა ? განვიხილოთ მოკლედ კომპიუტერული სისტემების აგების ტექნოლოგიების განვითარების

გზაზე ინსტრუმენტების ძირითადი თვისებები და კონცეფციები, რომლებმაც თავი მოიყარა („ემპკვიდრული“ თვისებების სახით) UML-ტექნოლოგიაში.

50-70-იან წლებში, როგორც ცნობილია, გავრცელებული იყო ოპერატორული და პროცედურულ-ორიენტირებული სტილის დაპროგრამების ენები. „ტანდემურ“ პრინციპზე მომუშავე ეს ტრადიციული ინსტრუმენტები პროგრამულ კოდებს და მათთვის საჭირო ინფორმაციულ მონაცემებს ერთად, ერთ ფაილში ათავსებდა. ეს პრინციპი დღეს ობიექტ-ორიენტირებული სტილის ენებშიც შენარჩუნებულია: კლასი არის მონაცემებისა და ამ მონაცემების დამუშავების ფუნქციების ინკაფსულირებით მიღებული კომპონენტი.

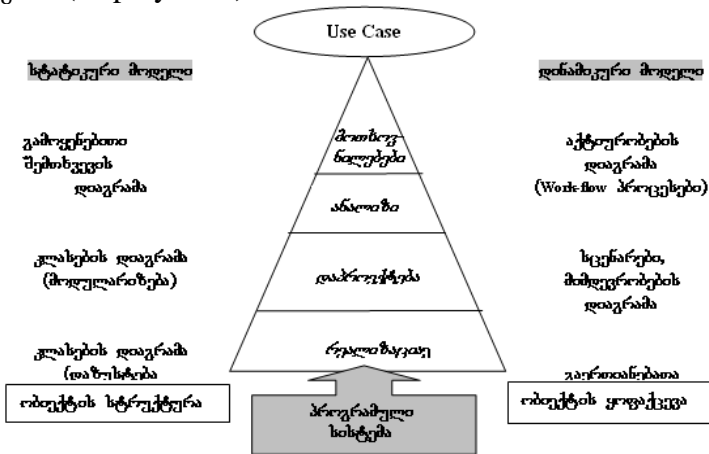
ასეთი მიდგომა აღნიშნულ ტრადიციულ ენებში, ერთის მხრივ სისტემის საიმედოობას ზრდიდა, მაგრამ, მეორეს მხრივ ასევე იზრდებოდა პროგრამული კოდისა და მონაცემთა მოვლის, მოდიფიკაციის, შედეგების მიღების დრო, მანქანური მეხსიერების რესურსების ხარჯი ინფორმაციის დუბლირების გამო და ა.შ.

70-90-იან წლებში განვითარდა მონაცემთა ბაზების მართვის სისტემების (მბმს) პროგრამული პაკეტები. აქ ძირითადი კონცეფცია, განსაკუთრებით რელაციურ ბაზებში, პროგრამული კოდისა და მონაცემების ცალკ-ცალკე შენახვის იდეაში მდგომარეობდა. რელაციურ მონაცემთა ბაზების მართვის სისტემები დღესაც აქტუალურ მიმართულებად ითვლება (მაგ, Oracle, Access, Visual FoxPro, SQL 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.

დაპროგრამების ენებთან შედარებით მბმს იყო პირველი ცდა პროგრამულ კოდში მონაცემთა სტრუქტურების აგების ავტომატიზაციისა. 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 [2,5].

90-ანი წლებიდან მნიშვნელოვნად განვითარდა სტრუქტურული დაპროგრამებისა და ობიექტ-ორიენტირებული მოდელირების კონცეფციები და ისინი ინფორმაციული ტექნოლოგიებისა და კომპიუტერული ენების თითქმის ყველა პაკეტშია რეალიზებული. ამგვარად, UML-ტექნოლოგიამ შეითვისა როგორც ტრადიციული დაპროგრამების ენების, ასევე მონაცემთა ბაზების მართვის სისტემების საუკეთესო ჩვევები და გაამდიდრა ისინი სისტემების დაპროექტების ვიზუალური კომპონენტებითაც. პროგრამული კოდების კონსტრუირება აქ ავტომატიზებულია. მომხმარებელმა მხოლოდ უნდა აავოს ზუსტად კლასების დიაგრამები და მიუთითოს სისტემას, თუ რომელი ენა (C++, Visual Basic, Java, Oracle, Power Builder თუ სხვ.) ესაჭიროება. 2.3 ნახაზზე მოცემულია UML-ტექნოლოგიის განხორციელების ეტაპობრივი სქემა [3].

UML არაა კონკრეტული მეთოდი. იგი არის ბაზა სხვადასხვა მეთოდებისათვის, ვინაიდან მასში შემუშავებულია მოდელირების კონსტრუქციების განსაზღვრული სიმრავლე აღწერის ერთიანი სისტემითა და სემანტიკით. 2.4 ნახაზზე მოცემულია UML-ტექნოლოგიის შედგენილობა და სტრუქტურა. იგი შედგება ოთხი წარმოდგენისაგან (View): გამოყენებითი შემთხვევის დიაგრამა (Use Case), ლოგიკური (Logical), კომპონენტებისა (Component) და ფიზიკური განლაგების (Deployment).



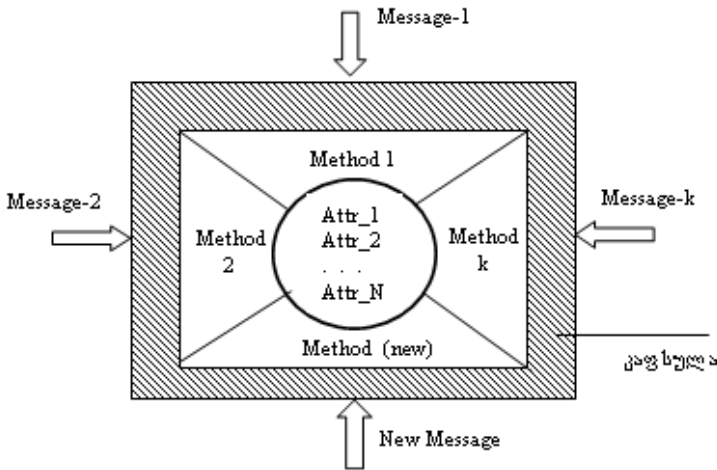
ნახ.2.3. სტატისკური და დინამიკური მოდელირება

თითოეულისათვის მოდელირების ელემენტები იყოფა სხვადასხვა ტიპის დიაგრამებად (UseCase Diagram, Interaction Diagram-ურთიერთმოქმედების დიაგრამა, Class Diagram, State Transition-მდგომარეობათა დიაგრამა (ყოფაქცევის მოდელი), Component Diagram-მაგ., cpp, h, exe ფაილების ერთობილობა, Deployment Diagram-კომპონენტების განლაგება ფიზიკურ მატარებლებზე (სერვერ-კლიენტის ქსელში).

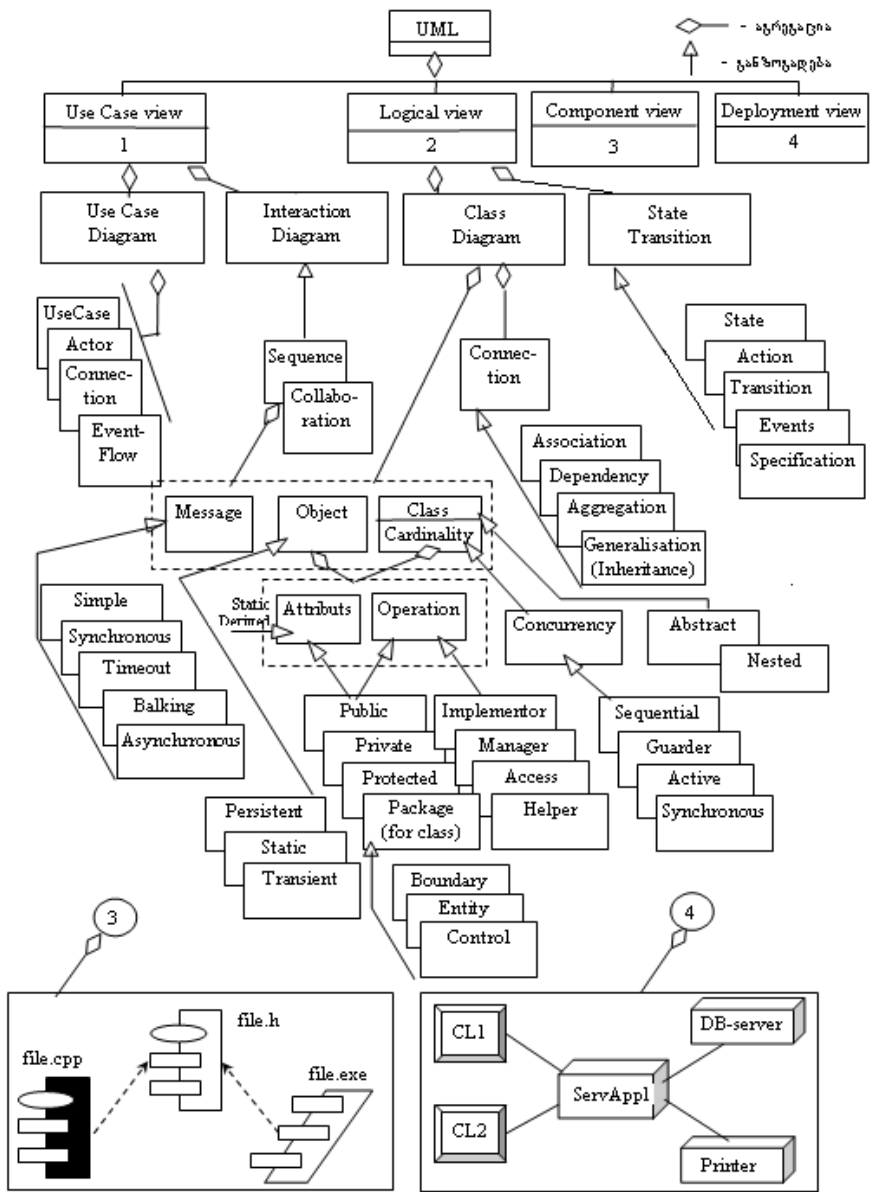
ასეთი იდეოლოგია მემკვიდრეობითაა მიღებული მონაცემთა ბაზების დაპროექტების თეორიიდან, სადაც შესაბამისად გვაქვს: მომხმარებელთა წარმოდგენის დონე, კონცეპტუალური მოდელის დონე, მონაცემთა ლოგიკური სტრუქტურის დონე და ფიზიკური სტრუქტურის დონე [7].

მბმს-ის კონცეპტუალური და ლოგიკური დონეების, ასევე UML-ენის UseCase და Logical დიაგრამების სიღრმისეული ანალიზი გვიჩვენებს, რომ მათი ძირითადი შემადგენელ ელემენტებია ატრიბუტები (Attributes) და მათი დამუშავების მეთოდები (Operations), რომლებიც შესაბამისი შეტყობინებებით (Messages) ინიციალიზირდება (ნახ.2.4).

ეს კი, როგორც ცნობილია, ობიექტ-ორიენტირებული მოდელირების ძირითადი ელემენტი – კლასია (Class). მისი ილუსტრაცია 2.5 ნახაზზეა მოცემული.



ნახ.2.5. კლასის მოდელი



ნახ.2.4. UML სტრუქტურა Rational Rose მაგალითზე

კლასის ატრიბუტებისა და ოპერაციებისათვის სპეციფიცირებულია მათი ხილვადობის არეალი („+“-Public, „-“-Private, „#“-Protected, Package), ატრიბუტებისათვის სტატიკურობა („\$“-Static, რაც ნიშნავს მისი მნიშვნელობის უცვლელობას ყველა ობიექტისათვის) და მწარმოებლურობა („/“-Derived, ანუ იგი მიიღება სხვა ატრიბუტებისაგან), ოპერაციებისთვის ტიპები (Implementor-შესრულების, Manager-მართვის, Access-მიმართვის, Helper-დახმარების).

კლასთან და ობიექტებთან ერთად საჭიროა განისაზღვროს შეტყობინებების მახასიათებლები. მაგ., მისი დასახელება და კავშირი შესაბამის ოპერაციასთან, მისი გადაცემის სინქრონიზაციის პარამეტრები, როგორცაა Simple (შეტყობინება სრულდება ერთი ნაკადში მიმდევრობით), Synchronous (სინქრონული, კლიენტი ელოდება პასუხს), Timeout (მოლოდინი ლიმიტირებული დროით), Balking (მოთხოვნა რიგში დაუდგომლად მოიხსნება), Asynchronous (ასინქრონული, კლიენტი პასუხს არ ელოდება, აგრძელებს მუშაობას).

კლასთა შორის კავშირები ოთხი ტიპისაა: ასოციაციური (Association), დამოკიდებულებითი (Dependency), აგრეგირებული (Aggregations) და განზოგადებული (Generalization), რომელიც არის „მემკვიდრეობითობა“ (inheritance) - ობიექტ-ორიენტირებული დაპროგრამების ერთ-ერთი საფუძველი.

UML-ენის კლასები შეიძლება გაერთიანდეს სპეციალურ პაკეტებში (Packages), მაგ., საზღვრების (Boundaries), არსთა (Entities) და მართვის (Control). ამ პაკეტებში ჯგუფდება მსგავსი სპეციფიკაციის მქონე კლასები.

კლასი შეიძლება იყოს აბსტრაქტული (Abstract), რომელშიც კონკრეტული ობიექტები არაა, მაგრამ არის მონაცემთა და ფუნქციათა ერთობლიობა, რომელიც მემკვიდრეობით გადაეცემა შვილ-კლასებს. გამოიყენება აგრეთვე ჩადგმული (Nested) კლასი. ამ დროს შვილ-კლასში მიეთითება მშობელი-კლასის სახელი. შეიძლება რამდენიმე ჩადგმული იერარქიის ორგანიზება.

პროცედურების შესრულების თვალსაზრისით, როდესაც არსებობს რამდენიმე მართვის ნაკადი კლასებისათვის განისაზღვრება სპეციფიკაცია პარალელიზმი (Concurrency). მათ შორის მიმდევრობითი (Sequential),

შემზღუდველი (Guarded), აქტიური (Active) და სინქრონული (Synchronous).

მართვის ინფორმაციული სისტემების დაპროექტების დროს მნიშვნელოვან საკითხად ითვლება კლასების შერჩევა და მათი დაკომპლექტება ატრიბუტებითა და ოპერაციებით. დღეისათვის არ არსობს რაიმე მეთოდი (თეორია), რომელიც კლასების დაპროექტების ოპტიმიზაციას ემსახურებოდეს. შეიძლება სისტემა აიგოს ერთი კლასით, რომელშიც მოთავსდება ყველა ატრიბუტი და ყველა ფუნქცია. ეს ვარიანტი მსგავსია მონაცემთა ბაზების თეორიაში პირველი ნორმალური ფორმისა (1ნფ), ოღონდაც, როგორც ცნობილია, ეს რელაციური ცხრილი ბაზის სტატიკური ელემენტია. მისი დამუშავების ფუნქციები თავმოყრილია პროგრამულ ნაწილში. ნორმალურ ფორმათა თეორია [7] მონაცემთა სტრუქტურების ოპტიმიზაციას ემსახურება და მისი ალგორითმები ფართოდ გამოიყენება მონაცემთა ბაზების სხვადასხვა ინსტრუმენტებში (მაგ., Oracle, Access).

ნაშრომში შემოთავაზებულია ნორმალიზაციის თეორიის ალგორითმების ადაპტაცია კლასების შედგენილობისა და სტრუქტურების ავტომატიზებული განსაზღვრისათვის. მისი იდეა მდგომარეობს შემდეგში: დავუშვათ, მოცემულია საპრობლემო სფეროს ყველა ატრიბუტის $\{A_i\}$ და ყველა ფუნქციის $\{F_j\}$ სიმრავლეები, აგრეთვე დამოკიდებულებათა მატრიცა M_{ij} , რომელიც აღწერს თუ რომელი ფუნქცია რომელ ატრიბუტთა ერთობლიობას იყენებს. ზოგადად ეს შეიძლება იყოს $M:N$ სახის დამოკიდებულება. მატრიცის ელემენტი ჩაიწერება ასე:

$$M(i, j) = \begin{cases} 1, & \text{if } A_i \text{ is argument of } F_j \\ 0, & \text{if otherwise.} \end{cases}$$

ატრიბუტების ერთობლიობა მოცემული ამოცანისათვის წარმოდგენილია ერთ ცხრილში (1ნფ-ში). საჭიროა მისი გადაყვანა შედარებით მაღალი რიგის ნორმალურ ფორმებში სპეციალური ალგორითმების დახმარებით [7]. შედეგად მიიღება რამდენიმე მცირე ზომის ცხრილი (მაგ., 3ნფ-ში). ამ ცხრილების ატრიბუტთა და მათი დამუშავების ოპერაციათა არგუმენტების ავტომატიზებული ანალიზის საფუძველზე განისაზღვრება მიზნობრივი კლასების ფუნქციათა სტრუქტურები, და

ამგვარად საბოლოოდ დაკომპლექტდება გამოყენებითი სფეროს კლასთა დიაგრამები.

შემდგომ ეტაპზე შესაძლებელია კლასების დიაგრამის დაზუსტება, კორექტირება და კლასთა შორის სიმრავლითი დამოკიდებულებების (Multiplicity) განსაზღვრა. კლასებში შეიძლება მიეთითოს აგრეთვე მათი სიმძლავრეები (Cardinality), რომელიც გვიჩვენებს მასში ობიექტების რიცხვს.

ობიექტები შეიძლება სპეციფირებულ იქნას ისეთი თვისებებით, როგორცაა მდგრადობა (Persistent), სტატიკურობა (Static) და დროებითობა (Transient). ეს მახასიათებელი ადგენს თითოეული ობიექტის არსებობის ხანგრძლიობას შესაბამისი პროგრამის აქტიურობის დროის ინტერვალში ან მის გარეთ.

ზოგიერთი კლასისათვის საჭიროა აიგოს მდგომარეობათა დიაგრამა (State Transition), რომელიც აღწერს ყველა იმ შესაძლო მდგომარეობას, რომელშიც შეიძლება იმყოფებოდეს ან გადადიოდეს კლასის ობიექტი. ეს მექანიზმი გამოიყენება ობიექტების ყოფაქცევის მოდელირებისათვის. მისი კომპონენტებია მდგომარეობა (State), გადასასვლელი (Transition), მოვლენა (Event), შეზღუდვის პირობები (Guard conditions) და მოქმედება (Action).

კომპონენტების წარმოდგენა (Component view) ემსახურება წინა ეტაპზე დაპროექტებული სისტემის ფიზიკურ ორგანიზებას პროგრამულ პროდუქტში. ანუ კლასების დიაგრამების ასახვა კომპიუტერულ ფაილებში, მაგ., DLL, H, CPP, EXE ან სხვ. ფაილების სახით (ნახ.2.4).

განლაგების წარმოდგენა (Deployment view) უზრუნველყოფს დაპროგრამებული საბოლოო პროდუქტის ფიზიკურ განლაგებას ქსელურ ტოპოლოგიაში. აქ განისაზღვრება ის, თუ სისტემის რომელი კომპონენტი (კოდი) სერვერ-კლიენტის ქსელის რომელ კვანძებში უნდა მოთავსდეს (ნახ.2.4).

2.2. უნიფიცირებული მოდელირების ენა და ობიექტური პეტრის ქსელები. UML/PN ინსტრუმენტული საშუალების ბირთვი

ორგანიზაციული მართვის ავტომატიზებული სისტემების დაპროექტებისა და რეალიზაციის დროს განსაკუთრებულ მნიშვნელობას იძენს დინამიკური პროცესების მოდელირებისა და ანალიზის გრაფი-

ანალიზური სისტემებისა და ინსტრუმენტების გამოყენება. ამ მხრივ მნიშვნელოვანია სამუშაო პროცესებისა და ოპერაციების პეტრის ქსელის გრაფებით ასახვა და მისი კვლევა.

აქ შევეხებით პრაქტიკული ღირებულების მქონე თეორიულ საკითხებს, რომლებიც ეყრდნობა, ერთის მხრივ, ობიექტ-ორიენტირებულ მოდელირებას და მეორეს მხრივ, თვით პეტრის ქსელების გამოყენებას.

ობიექტური პეტრის ქსელები (Object Petri Nets) (და მისი გაფართოებები: ობიექტ-ორიენტირებული და მაჩვენებლიანი პეტრის ქსელები) პეტრის ქსელების თეორიისა და ობიექტ-ორიენტირებული დაპროგრამების თეორიის შეჯერებით მიღებული პეტრის ქსელის ტიპია [13]. ობიექტური პეტრის ქსელი ერთი სისტემური და რამდენიმე ობიექტური ქსელისგან შედგება, სადაც ობიექტური ქსელები მარკერთა როლში გამოდის. ფაქტობრივად, მიიღება პეტრის ქსელების სიმრავლე ერთი პეტრის ქსელის პოზიციებში.

ობიექტური ქსელები ელემენტარულ სისტემურ ქსელებს წარმოადგენს, ხოლო სისტემური ქსელი მაღალი დონის პეტრის ქსელია, რომლის პოზიციებშიც დაშვებულია როგორც ობიექტური ქსელების, ასევე ჩვეულებრივი შავი მარკერების არსებობა, ოღონდ არა ერთსა და იმავე პოზიციაში. შესაბამისად, რკალის ანოტაცია შეიძლება იყოს ნატურალური რიცხვი შავი მარკერებისთვის ან ობიექტური ქსელების განსაზღვრულ იდენტიფიკატორთა სიმრავლე-სისტემური ქსელის ყოველ გადასასვლელს შეუძლია ობიექტური ქსელის გადატანა არაუმეტეს ერთი შემავალი პოზიციიდან არაუმეტეს ერთ გამომავალ პოზიციაში. ამასთანავე ერთ გაშვებაზე არაუმეტეს ერთი ობიექტური ქსელის გადატანაა ნებადართული.

ობიექტურ პეტრის ქსელებს სხვა ტიპის პეტრის ქსელებისგან გადასასვლელის როლის ზრდაც გამოარჩევს: სისტემური და ობიექტური ქსელების ზოგიერთ გადასასვლელს ემატება სპეციალური ფუნქცია, რომელსაც ინტერაქცია ეწოდება. ინტერაქცია 2 ტიპისაა: სისტემ-ობიექტური და ობიექტ-ობიექტური. პირველი სისტემური და ობიექტური ქსელების გადასასვლელთა სინქრონულ ურთიერთობას უზრუნველყოფს, მეორე – ობიექტური ქსელების ურთიერთ-სინქრონიზაციას.

სისტემ-ინტერაქციული გადასასვლელის გაშვების წესი შემდეგია: თუ სისტემური ქსელის გადასასვლელი ინტერაქციულია და მისი ინტერაქცია ქსელში მარკერის სახით მოძრავი ობიექტური ქსელის

ნებადართულ ინტერაქციას თანხვდება, მაშინ სისტემური ქსელის ინტერაქციული გადასასვლელის გაშვებისას ქსელში მოძრაობის პარალელურად გაიხსნება ობიექტური ქსელის ინტერაქციული გადასასვლელიც.

სხვა შემთხვევაში (თუ ინტერაქციები არ თანხვდება, ან სისტემური ან ობიექტური ქსელის გადასასვლელები ინტერაქციებს არ შეიცავს), ობიექტური ქსელი სისტემურში უცვლელი სახით გადაადგილდება. სწორედ ამგვარი მიდგომა განაპირობებს ობიექტური პეტრის ქსელების ობიექტ-ორიენტირებულ ხასიათს.

ფერადი პეტრის ქსელები სტრუქტურული დაპროგრამების თეორიასთან მჭიდრო კავშირში იმყოფება, შესამაბისად, შესაძლებელია ფერადი პეტრის ქსელებიდან მოქნილი გადასვლა ობიექტურ პეტრის ქსელებზე (როგორც სტრუქტურულიდან ობიექტორიენტირებული დაპროგრამების იდეოლოგიაზე) გარკვეული ახალი თვისებების შემოტანით.

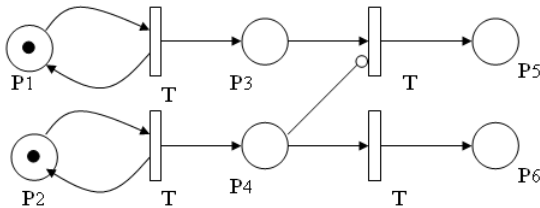
ობიექტ-ორიენტირებულ დაპროგრამებასა და ობიექტურ პეტრის ქსელებს შორის რამდენიმე მარტივი შესაბამისობა 2.1 ცხრილშია მოცემული.

ცხრ.2.1

ობიექტ- ორიენტირებული დაპროგრამება	ობიექტური პეტრის ქსელები
პროგრამული მოდული	სისტემური ქსელი
კლასი	ობიექტური ქსელის განსაზღვრა
ობიექტი	ობიექტური ქსელი კონკრეტული მარკირებით
ცვლადი კლასის წევრი-ცვლადი	სისტემური ქსელის მარკერი ობიექტური ქსელის მარკერი
გარე ფუნქცია კლასის წევრი-ფუნქცია	სისტემური ქსელის ინტერაქციული გადასასვლელი ობიექტური ქსელის გადასასვლელი

პეტრის ქსელების მთელი აღწერილი მრავალფეროვნების მიუხედავად მისი მამოძღვრებელი სიმბოლურე შეზღუდულია და რიგი ამოცანების მოდელირებისთვის არასაკმარისი. ამ პრობლემის გადასაჭრელად განსაზღვრულია პეტრის ქსელის **გაფართოებები**, რომლებიც სპეციფიკური საპრობლემო სფეროების მოდელირებისთვის გამოიყენება [11,13].

ყველაზე ფართოდ გავრცელებული გაფართოებაა **შემაკავებელი (ამკრძალავი) რკალი (Inhibitor Arc)**, რომელიც პოზიციის „ნულზე შემოწმების“ პროცედურას ასრულებს, ანუ გადასასვლელს ნებადართულს ხდის მხოლოდ მაშინ, როცა შესაბამის შემავალ პოზიციაში არცერთი მარკერი არაა წარმოდგენილი. შემაკავებელი რკალი წრიული ბოლოთი გამოისახება. 2.6 ნახაზზე მისი დახმარებით გადასასვლელთა პრიორიტეტული გაშვების ამოცანა მოდელირდება (გადასასვლელი **T₃** ნებადართულია მხოლოდ მაშინ, როცა **T₄** არ არის ნებადართული, ანუ **P₄** პოზიციაში მარკერი არ არის), რაც კლასიკური პეტრის ქსელით შეუძლებელი იყო.

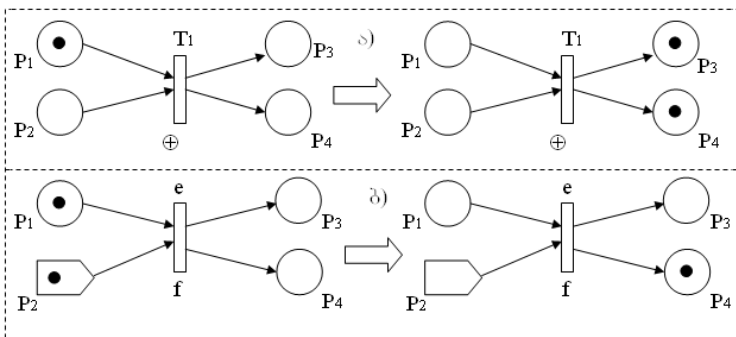


ნახ.2.6. პეტრის ქსელი შემაკავებელი რკალით “გადასასვლელთა პრიორიტეტული გახსნის” ამოცანისათვის

გარდა შემაკავებელი რკალებისა, პეტრის ქსელებში განისაზღვრება პოზიციების და გადასასვლელთა სხვადასხვა გაფართოებები.

შეზღუდვის არე შემაკავალი პოზიციების გარკვეული სიმრავლეა, რომელთათვისაც გადასასვლელის გახსნის წესი შექმნილია მოდიფიცირდება: გადასასვლელი შეიძლება გაიხსნას მხოლოდ მაშინ, როცა ერთი მაინც შემაკავალი პოზიცია ცარიელია.

გაფართოება **„გამომრიცხავი ან“** გულისხმობს გადასასვლელის გახსნას მაშინ, როცა შემაკავალი პოზიციებიდან მარკერი ერთსა და მხოლოდ ერთ პოზიციაშია (ნახ.2.7).



ნახ.2.7. პეტრის ქსელები ა) “გამომრიცხავი ან” და ბ) “გადამრთველი პოზიცია – გადასასვლელის” გაფართოებით. გადასასვლელთა გაშვებამდე და გაშვების შემდეგ.

გადამრთველი გადასასვლელი შეიცავს სპეციალურ შემავალ გადამრთველ პოზიციას და ზუსტად 2 გამომავალ პოზიციას (**e** და **f** პოზიციები).

გადამრთველი გადასასვლელისთვის გადამრთველ პოზიციაში მარკერის არსებობა-არარსებობას მნიშვნელობა არა აქვს, იგი გაიხსნება, თუ სხვა შემავალი პოზიციები შევსებულია მარკერებით. ამასთან, გამომავალი პოზიციებიდან მარკერს მიიღებს მხოლოდ: პოზიცია **e** - თუ გადამრთველ პოზიციაში მარკერი არ არის, პოზიცია **f** – თუ მარკერი მოცემულია.

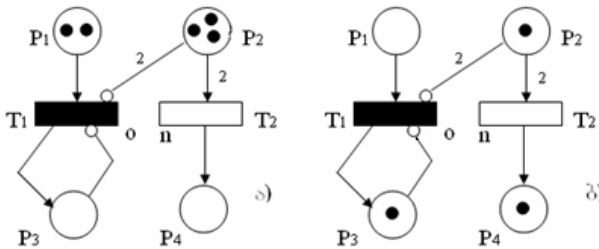
აღწერილი სამი გაფართოება, როგორც წესი, ჩანაცვლებადია შემაკავებელი რკალებით, ოღონდ ამ დროს პეტრის ქსელის სტრუქტურა რთულდება.

ზოგადად, გაფართოებები ზრდის პეტრის ქსელების მამოძღვრებელ სიმძლავრეებს, მაგრამ იმავდროულად ამცირებს ქსელის ანალიზის შესაძლებლობას. ამ პრობლემის შემდგომი გადაწყვეტა ცალკე საკითხია.

ამ მოკლე შესავლის შემდეგ პეტრის გაფართოებული ქსელების თეორიიდან, ჩავატაროთ კონკრეტული სამუშაო პროცესების და ოპერაციების მოდელირება და ვაჩვენოთ ის მიდგომა, რომელიც შესაძლებელს ხდის მოღვაწეობის (Activity) დიაგრამები წარმოვადგინოთ პეტრის ქსელის გრაფებით.

მოლვაწობის დიაგრამა ასახავს დინამიკურ პროცესს, ამიტომაც მის შესასრულებლად გარკვეული დროა საჭირო. შესაბამისი დროითი ელემენტი პეტრის ქსელში აისახება გადასასვლელში. პეტრის ქსელის გადასასვლელი ორი სახისაა: მყისიერად შესრულებადი (დროის დაყოვნების გარეშე) და დაყოვნებით (დროითი) [11,13].

2.8 ნახაზზე ნაჩვენებია მაგალითი ასეთი გადასასვლელის მოდელირებისათვის. აკრძალვის რეალზე (P2-დან T1-კენ) ზის 2-ანი, რაც იმას ნიშნავს, რომ T1 გაიხსნება მაშინ, როცა P2 -ში მარკერების რაოდენობა გახდება ნაკლები ან ტოლი 2-ს.



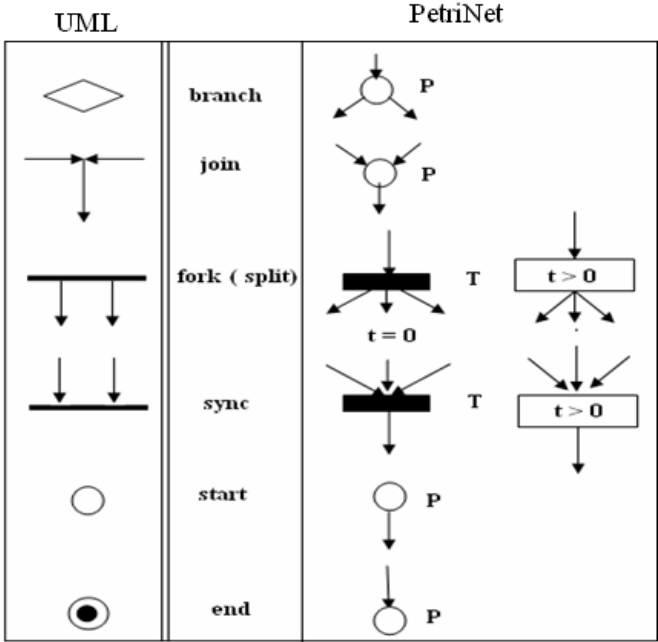
ნახ.2.8. პეტრის ქსელის საწყისი ა) და საბოლოო ბ) მდგომარეობები

ამგვარად, ა) სქემა იმუშავებს ასე: ჯერ გაიხსნება T2 გადასასვლელი და ორი მარკერი გამოაკლდება P2-ს, ერთი მარკერი გადავა P4-ში. ეს მდგომარეობა შეესაბამება ბ) ნახაზს.

როგორც აღნიშნული იყო აქტიურობის დიაგრამაზე გამოიყენება ელემენტები: საწყისი და საბოლოო კვანძები, მოქმედება, შედგენილი-მოქმედება (იერარქიულად ჩადგმული პროცესი), განშტოება და შეერთება.

ეს ელემენტები პეტრის ქსელებში მოდელირდება გრაფის (Pi) პოზიციებით. დაყოფისა (fork, split) და გაერთიანების (join, sync) ელემენტები კი მოდელირდება პეტრის ქსელის გადასასვლელებით (Tj).

2.9 ნახაზზე ნაჩვენებია ეს იზომორფული ელემენტები.



ნახ.2.9. UML და PetriNet იზომორფული ელემენტები

2.3. დინამიკური პროცესების ტრანსფორმაციის ძირითადი პრინციპები და UML/PN ტექნოლოგიის დამუშავება მონაცემთა ბაზებისთვის

მოცემულ პარაგრაფში განიხილება განაწილებულ მართვის ავტომატიზებულ სისტემებში დინამიკური პროცესების მოდელირებისა და ანალიზის გრაფო-ანალიზური სისტემების გამოყენების საკითხები [10].

ბიზნეს-პროცესების მოდელირება აქტიურობათა დიაგრამით კარგ მეთოდურ ინსტრუმენტს წარმოადგენს, მაგრამ მრავალვარიანტულობის გამო დაპროექტებისა და რეალიზაციის პროცესი თავიდან ბოლომდე დამოკიდებულია დამპროექტებელ-სპეციალისტზე. ამიტომ, საუკეთესო მოდელის განსაზღვრისათვის მნიშვნელოვანია სამუშაო პროცესებისა და ოპერაციების პეტრის ქსელის გრაფებით ასახვა და მისი კვლევა [1,11].

თანამედროვე ინფორმაციული ტექნოლოგიები ეფუძნება, ერთის მხრივ კომპიუტერებისა და სატელეკომუნიკაციო აპარატურის მაღალ შესაძლებლობებს, მეორეს მხრივ ახალ პროგრამულ სისტემებს და ინფორმაციის დამუშავების ახალ ტექნოლოგიებს. წინა პლანზეა წამოწეული ქსელური, განაწილებული ინფორმაციული სისტემები, რომლებიც ფუნქციონირებს გლობალურ (ინტერნეტი) ან ლოკალურ ქსელებში, ან ორივეში ერთად. ეს საშუალებას გვაძლევს მაქსიმალურად გაემიჯნოთ მომხმარებელთა ფუნქციები და უზრუნველყოთ ინფორმაციული ბაზის დაცვა არასანქცირებული მიმართვებისაგან.

თანამედროვე მდგომა მსგავსი სისტემების დაპროექტებისა ეფუძნება UML ტექნოლოგიას, რომელშიც სამუშაო ადგილების მოდელირებისათვის გამოიყენება აქტიურობათა დიაგრამა. ვიზუალურად აქტიურობის დიაგრამა გამოიხატება გრაფის სახით, რომელსაც გააჩნია მწვერვალები და წიბოები:

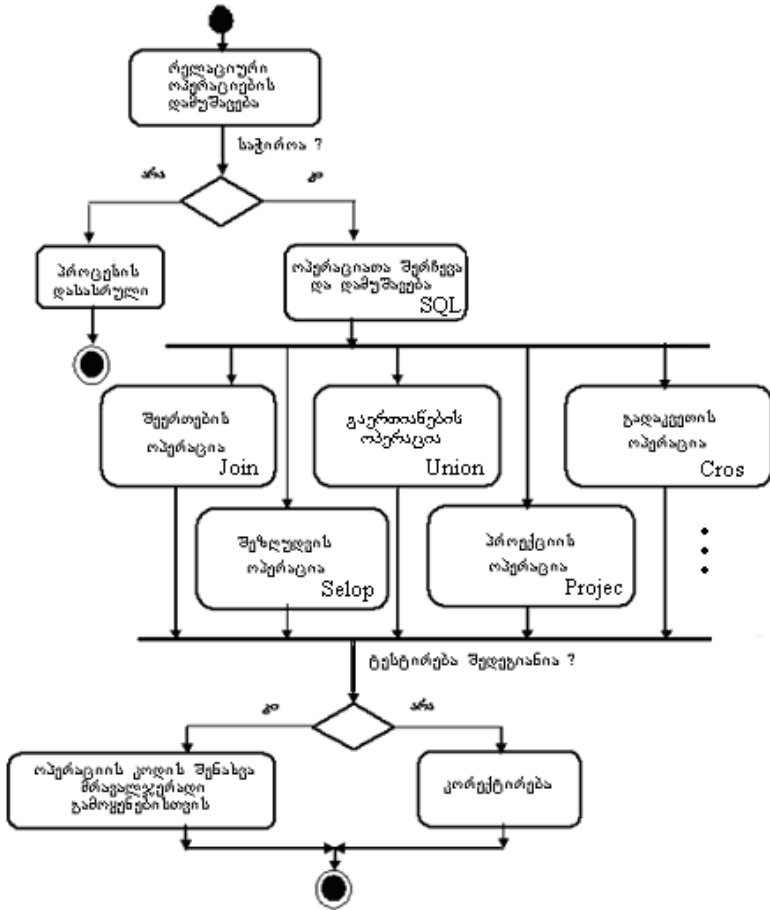
$$D = \langle Sa, Sm, R, O \rangle,$$

სადაც

Sa – აქტიურობის (პროცესის) და **Sm** – მოქმედებათა (პროცედურები) მდგომარეობებია, **R** – გადასვლები და **O** - ობიექტებია.

მოქმედების მდგომარეობა ისეთი მდგომარეობაა, რომლის შემდგომი დეკომპოზიცია შეუძლებელია. მისგან განსხვავებით, შეიძლება აქტიურობის მდგომარეობების შემდგომი დეკომპოზიცია, ამის შედეგად შესრულებადი აქტიურობა შეიძლება წარმოვადგინოთ სხვა აქტიურობის დიაგრამების სახით. სემანტიკურად აქტიურობის დიაგრამა ეკვივალენტურია აქტიურობის გრაფის ტრანზიტული გაფართოებისა, მანამდე სანამ არ დარჩება მხოლოდ მოქმედებები. შევდივართ რა ერთერთ ასეთ მდგომარეობაში, სრულდება შესაბამისი მოქმედება ან აქტიურობა, ხოლო გამოსვლისას მართვა გადაეცემა შემდეგ მოქმედებას ან აქტიურობას. განვიხილოთ 2.10 ნახაზზე მოცემული მაგალითი, რომელიც ასახავს მონაცემთა რელაციური ბაზის ოპერაციების დამუშავების პროცესის ფრაგმენტს – აქტიურობის დიაგრამით [9].

მართვის გადაცემის აღწერისათვის გამოიყენება გადასვლები (Transitions), რომლებიც მიუთითებს გზას ერთი მოქმედების მდგომარეობიდან მეორეში. გრაფიკულად იგი გამოიხატება ისრით.



ნახ.2.10. მბ-ის რელაციური ოპერაციების დაშუქების პროცესის აქტიურობის დიაგრამა

გადასვლებისათვის შესაძლებელია გამოვიყენოთ როგორც მარტივი არატრიგერული გადასვლა, ასევე განშტოება (Decision – პროგრამირებაში if...else ბლოკი) და გაყოფა-შერწყმა (Fork-Join), რომლებიც საშუალებას იძლევა მოვახდინოთ პარალელური ნაკადების მოდელირება.

აქტიურობის დიაგრამა, როგორც უნივერსალური საშუალება, კარგ მეთოდოლოგიურ ინსტრუმენტს წარმოადგენს, მაგრამ თავიდან ბოლომდე დაპროექტებისა და რეალიზაციის პროცესი დამოკიდებულია დამპროექტებელ-სპეციალისტზე.

აქ ძირითადად ევრისტიკულ მოდელირებასთან გვაქვს საქმე, რამეთუ UML-მეთოდოლოგიით ამოცანათა გადაწყვეტის მრავალ-ეტაპიანობა და კომპლექსურობა შესაძლებელს ხდის შედეგები მივიღოთ მრავალვარიანტული, გამომდინარე მათი კომბინატორული ხასიათისა. ევრისტიკული მიდგომით კი ვერ დავადგენთ, თუ რამდენად კარგია ჩვენს მიერ მიღებული შედეგები.

ამ თვალსაზრისით საინტერესოა გადაწყვედეს სინტაქსური და სემანტიკური ეკვივალენტობის (მოდელირების) მეთოდებით დასმული ამოცანა [12]. განისაზღვროს მისი გადაწყვეტის ოპტიმალური მეთოდი, შეფასების კრიტერიუმები.

ორგანიზაციული მართვის ავტომატიზებული სისტემების დაპროექტებისა და რეალიზაციის დროს განსაკუთრებულ მნიშვნელობას იძენს დინამიკური პროცესების მოდელირებისა და ანალიზის გრაფო-ანალიზური სისტემებისა და ინსტრუმენტების გამოყენება.

ამ მხრივ მნიშვნელოვანია სამუშაო პროცესებისა და ოპერაციების პეტრის ქსელის გრაფებით ასახვა და მისი კვლევა [10].

ნაშრომში შევეხებით პრაქტიკული ღირებულების მქონე თეორიულ საკითხებს, რომლებიც ეყრდნობა, ერთის მხრივ, ობიექტ-ორიენტირებულ მოდელირებას და მეორეს მხრივ, თვით პეტრის ქსელების გამოყენებას.

ამოცანის გადაწყვეტა სამი ეტაპისგან შედგება:

1. შეიქმნას სამუშაო პროცესებისა და ოპერაციების ტრანსფორმაციის (მოდელირების) თეორიული საფუძვლები პეტრის ქსელის გრაფებში;

2. აიგოს სისტემური პეტრის ქსელის გრაფებით წარმოდგენილი პროცესების დიაგრამები გრაფო-ანალიზური ინსტრუმენტების საშუალებით;

3. ჩატარდეს აგებული პეტრის ქსელების მოდელების მანქანური ანალიზი (შესაძლოა მასობრივი მომსახურების მეთოდების გამოყენება).

აღნიშნული საკითხების ასახსნელად და უკეთ აღსაქმელად განვიხილოთ გაფართოებული პეტრის ქსელების ერთ-ერთი კონცეფცია და მისი ინსტრუმენტი, რომლის გამოყენებითაც შემდგომში შევძლებთ პროცესების მოდელირებას.

განვიხილოთ მაღალი დონის პეტრის ქსელთა ტიპები. ყოველ პეტრის ქსელს, ჩვეულებისამებრ, გააჩნია შემდეგი საერთო ელემენტები: პოზიციები, გადასასვლელები და რკალები (წიბოები), რომლებითაც პეტრის ქსელის გრაფი იქმნება.

პეტრის ქსელის ახალი ტიპის განსაზღვრისას, საფუძვლად სწორედ პეტრის ქსელის გრაფია აღებული და იგი შემდგომი ასახვებითა და ფუნქციებით პეტრის ქსელის კონკრეტულ ტიპამდე ფართოვდება.

პეტრის ქსელის სხვადასხვა ტიპები ერთმანეთისგან შეიძლება განსხვავდებოდეს მარკერთა ტიპებით და მათგან გამომდინარე ერთიანი მარკირების სისტემით, ქსელის ელემენტების აღწერით (ჭდეები) ან/და გადასასვლელთა გაშვების წესებით [13].

ჭდეები პეტრის ქსელის ელემენტებზე, ძირითადად მხოლოდ წარწერებია და შეიცავს ელემენტის სინტაქსს, მაგრამ არა სემანტიკას. შესაბამისად, ისინი ქსელის შესრულების პროცესში ვერაფერს ცვლის. ჭდეების დანიშნულება პეტრის ქსელის სინტაქსური კონტროლია.

ამის მიუხედავად, ახალი ტიპის ჭდის განსაზღვრა უკვე საკმარისია იმისთვის, რომ ახალი პეტრის ქსელის ტიპი იქნეს განსაზღვრული. პოზიციებზე, გადასასვლელებზე ან/და რკალებზე დროითი დაყოვნების განსაზღვრას დროითი პეტრის ქსელის ტიპი შემოაქვს, დაყოვნების დროთა ალბათურ განაწილებას – სტოქასტური პეტრის ქსელის ტიპი და ასე შემდეგ.

ამ მოკლე შესავლის შემდეგ პეტრის გაფართოებული ქსელების თეორიიდან, ჩავატაროთ კონკრეტული UML-პროცესების მოდელირება და ვაჩვენოთ ის მიდგომა, რომელიც შესაძლებელს ხდის მოღვაწეობის დიაგრამები წარმოვადგინოთ პეტრის ქსელის გრაფებით.

მოღვაწეობის დიაგრამა ასახავს დინამიკურ პროცესს, ამიტომაც მის შესასრულებლად გარკვეული დროა საჭირო. შესაბამისი დროითი ელემენტი პეტრის ქსელში აისახება გადასასვლელში.

პეტრის ქსელის გადასასვლელი ორი სახისაა: მყისიერად შესრულებადი (დროის დაყოვნების გარეშე) და დაყოვნებით (დროითი) [13].

UML-ის აქტიურობის დიაგრამაზე გამოიყენება ელემენტები: საწყისი და საბოლოო კვანძები, მოქმედება, შედგენილი-მოქმედება (იერარქიულად ჩადგმული პროცესი), განშტოება და შეერთება. ეს ელემენტები პეტრის ქსელებში მოდელირდება გრაფის (Pi) პოზიციებით. დაყოფისა (fork, split) და გაერთიანების (join, sync) ელემენტები კი მოდელირდება პეტრის ქსელის გადასასვლელებით (Tj). 2.11 ნახაზზე ილუსტრირებულია 2.10 ნახაზის აქტიურობის დიაგრამის ტრანსფორმაციის მაგალითი შესაბამის პეტრის ქსელში.

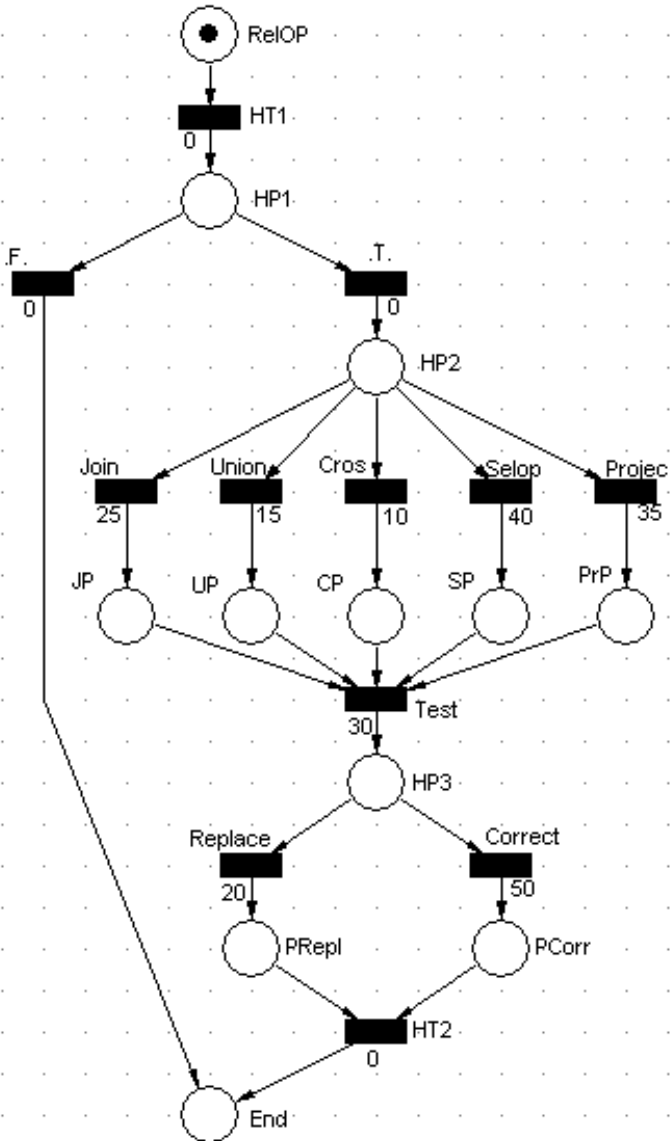
როგორც ნახაზიდან ჩანს, პეტრის ქსელის სქემაზე გაჩნდა დამატებითი ელემენტები: დამხმარე-პოზიცია (Help Position - HP) და დამხმარე-გადასასვლელი (Help Transition - HT). ისინი აუცილებელია სქემის შესაკრელად, როდესაც მოსაზღვრეა ორი პოზიცია ან ორი გადასასვლელი.

დასასრულ შეიძლება დავასკვნათ, რომ ამგვარი მეთოდით შესაძლებელია დინამიკური პროცესების მოდელირების განხორციელება პეტრის ქსელების ინსტრუმენტის გამოყენებით და შემდგომ მიღებული ქსელების რაოდენობრივი და თვისობრივი მახასიათებლების ანალიზის ჩატარება.

2.4. ობიექტების სასიცოცხლო ციკლის მოდელირება და ანალიზი პეტრის ქსელებით

წინა პარაგრაფში ჩვენ განვიხილეთ ბიზნეს-პროცესების მოდელირების ამოცანა, როგორც აქტიურობის დიაგრამების გამოყენებით, ასევე მათი მოდელირებისა და ანალიზის საშუალებები გრაფო-ანალიზური სისტემებისა და ინსტრუმენტების გამოყენებით, კერძოდ პეტრის ქსელებით.

საკვლევი ობიექტის დინამიკურ პროცესებს ასახავს მდგომარეობათა დიაგრამაც (State Diagram), რომელთა დეტალური ანალიზი სასარგებლოა, რათა მოვანერხოთ მისი კლასების (მონაცემთა ცხრილებისა და მეთოდების, ანუ შესასრულებელი ფუნქციების) ზუსტად განსაზღვრა. ამ მიზნისათვის გამოიყენება პეტრის ქსელების ინსტრუმენტი. აქ შევხებით ობიექტის სასიცოცხლო ციკლის ცნებას, რომელიც გამოისახება მდგომარეობათა დიაგრამებით, პეტრის ქსელის გრაფებით ასახვას და მის კვლევას.



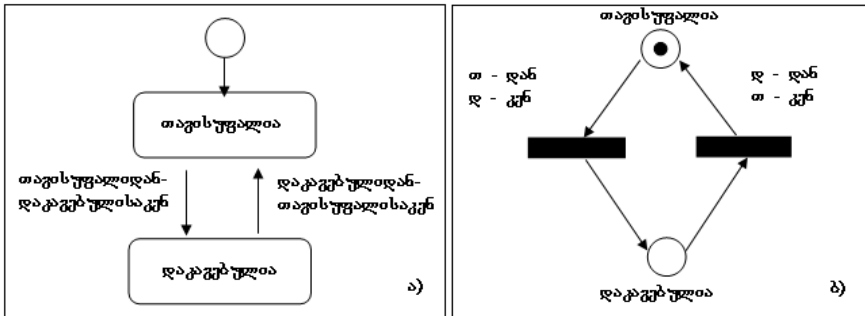
ნახ.2.11. პეტრის ქსელის გრაფის ფრაგმენტი

აღნიშნული საკითხების ასახსნელად და უკეთ აღსაქმელად წინა თავში უკვე განვიხილეთ გაფართოებული პეტრის ქსელების ერთ-ერთი კონცეფცია და მისი ინსტრუმენტი, რომლის გამოყენებითაც ავაგეთ საშუალო პროცესების ამსახველი ქსელი.

ვეყრდობით რა წინა თავში გამოთქმულ დებულებებს, მდგომარეობათა დიაგრამაზე კონკრეტულ მდგომარეობას შევუსაბამებთ პეტრის ქსელებში Pi-პოზიციას, ხოლო გადასასვლელს კი - Tj-გადასასვლელს. კავშირებისათვის ორი რკალი გამოიყენება (გადასასვლელში შემავალი და მისგან გამომავალი). მარკერი თავსდება საწყის პოზიციაში.

2.12 ნახაზზე ნაჩვენებია მარტივი მდგომარეობათა დიაგრამის შემთხვევა ორი მდგომარეობით (ა) და შესაბამისი პეტრის ქსელი (ბ).

შედგენილ (რთულ) მდგომარეობათა დიაგრამების შემთხვევაში გამოიყენება მდგომარეობების დაყოფა ზემდგომარეობათა (super-) და ქვემდგომარეობათა (sub-) დიაგრამებად. აქ შესაძლებელია იერარქიული, ჩადგმული და სხვა სახის ორგანიზაციასთან გვექონდეს საქმე.



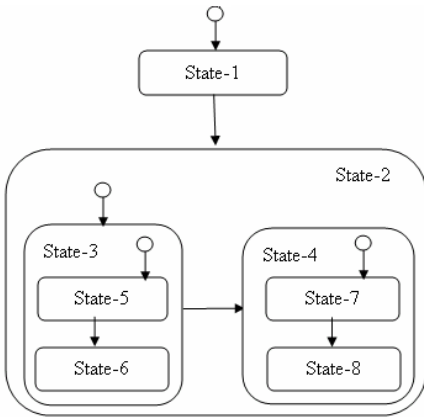
ნახ.2.12.. მარტივი მდგომარეობათა დიაგრამის (ა) პეტრის ქსელში ტრანსფორმაცია (ბ)

მათი ტრანსფორმაცია პეტრის ქსელებში (მარტივთან შედარებით) განსხვავებულია. განვიხილოთ მაგალითები შესაბამისი გრაფიკული ილუსტრაციებით (ნახ.2.13 და 2.14).

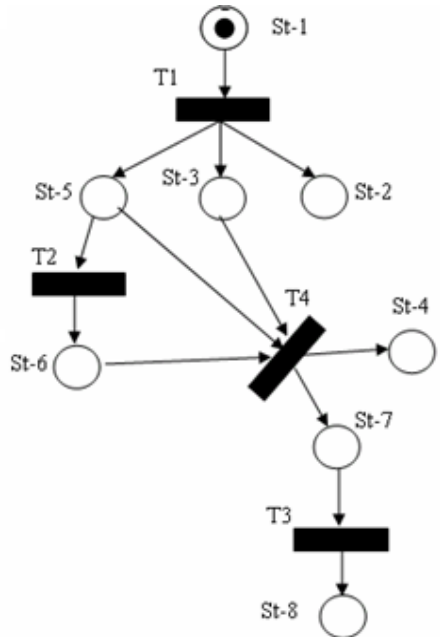
ნახაზზე გამოსახულია რვა მდგომარეობა. აქედან პირველი (State-1) მარტივია, მეორე (State-2) შედგება ორი იერარქიულად

ჩადგმული (State-3) და (State-4) ქვემდგომარეობისგან, რომლებიც კიდევ თავისთავად შედგება 5, 6, 7 და 8 - გან.

როგორც ტრანსფორმირების პრინციპებიდან გამომდინარეობს, UML-მდგომარეობები (State-D) ტრანსფორმირდება პეტრის ქსელებში პოზიციებით (P), კავშირები, ჩვეულებრივი ისრები - პეტრის ქსელის გადასასვლელებით (T).



ნახ.2.13. რთული, იერარქიული მდგომარეობათა შემთხვევა (UML)



ნახ.2.14. მდგომარეობათა დიაგრამის შესაბამისი პეტრის ქსელის გრაფი (PN)

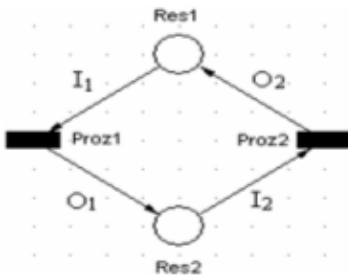
წრისთავიანი ისრები ასახავს „შესასვლელს“ მარკერის გადასაცემად და ისინი პეტრის ქსელის შემავალი და გამომავალი რკალებით აისახება ქსელზე. 2.14 ნახაზზე მოცემულია შესაბამისი პეტრის ქსელის გრაფი.

2.5. განაწილებული სისტემების რესურსების ადმინისტრირების ამოცანები: ჩიხური სიტუაციების მართვა

განიხილება ოპერაციულ სისტემების პლატფორმაზე მონაცემთა ბაზების ფაილების კოლექტიური გამოყენების დროს ჩიხური პროცესების არსებობისა და მათი გამორიცხვის შესაძლებლობანი. შემუშავებულია ასეთი პროცესების მართვის მოდელი პეტრის ქსელის ინსტრუმენტის საფუძველზე. ჩატარებულია ამ მოდელის იმიტაციური გამოკვლევა და აგებულია შესაბამისი დროითი მანასიათებლები [15].

თანამედროვე მულტიპროცესორულ სისტემებში, მათ შორის ლოკალურ კომპიუტერულ ქსელებში, რომლებიც გამოთვლითი რესურსების საერთო გამოყენების კონცეფციას ეყრდნობა, განსაკუთრებული მნიშვნელობა ენიჭება პროცესების ეფექტურად ორგანიზაციის საკითხს ჩიხური სიტუაციების აღმოსაფხვრელად [16,17].

პროცესი ჩიხურია (deadlock), თუ იგი ელოდება ისეთი ხდომილების შესრულებას, რომელიც არასოდეს მოხდება. ორი ან რამდენიმე პროცესი შეიძლება მოხვდეს ჩიხში, თუ თითოეული მათგანი აბლოკირებს რესურსებს (მაგალითად, მონაცემთა ბაზის ცხრილებს, ან მის ფრაგმენტებს), რომლებიც ესაჭიროება სხვა პროცესებს და თვითონ კი მოითხოვს ისეთ რესურსებს, რომლებიც ბლოკირებულია სხვა პროცესების მიერ. ოპერაციულ სისტემას ჩვენ განვიხილავთ როგორც გამოთვლითი რესურსების ადმინისტრატორს, ხოლო რესურსებად გვევლინება ცენტრალური პროცესორი, ოპერატიული მეხსიერება, დისკოები, ფაილური სისტემები, პროგრამები და მონაცემთა ბაზები, პრინტერები, ქსელური არხები და ა.შ.



ნახ.2.15. ჩიხური სიტუაციის მაგალითი

2.15 ნახაზზე ნაჩვენებია ელემენტარული ჩიხური ოპერაციის მაგალითი, ჩაწერილი პეტრის ქსელის გრაფით. აქ Proz1, Proz2 პროცესებია, ხოლო Res1, Res2 - რესურსები. პოზიცია-გადასასვლელთა შემაერთებელი რკალები შემდეგი დანიშნულებითაა: I_1 : Res1-რესურსი გამოეყო Proz1-პროცესს; O_1 : Proc1-პროცესი მოითხოვს Res2-რესურსს; I_2 : Res2-რესურსი გამოეყო Proz2-პროცესს;

O_2 : Proc2-პროცესი მოითხოვს Res1-რესურსს.

როგორც ნახაზიდან ჩანს, Proz1 პროცესს ბლოკირებული აქვს Res1 რესურსი და მუშაობის გასაგრძელებლად სჭირდება Res2 რესურსი. Proz2 პროცესს კი პირიქით, ბლოკირებული აქვს Res2 რესურსი და მუშაობის გასაგრძელებლად სჭირდება Res1 რესურსი. ამგვარად, ორივე პროცესი იმყოფება მუდმივად მოლოდინის რეჟიმში.

ჩიხური პროცესების არსებობისათვის ოთხი აუცილებელი პირობა იქნა განსაზღვრული [16]. ურთიერთგამორიცხვის (პროცესებს აქვს რესურსების მონოპოლური გამოყენების უფლება), დამატებითი რესურსების მოლოდინის (პროცესებს აქვს უკვე გამოყოფილი რესურსები, მაგრამ ელოდება დამატებითს), არაგადანაწილებადობის (პროცესებს არ შეიძლება ჩამოერთვას რესურსები მათ საბოლოო შესრულებამდე) და წრიული მოლოდინის (არსებობს პროცესების წრიული ჯაჭვი, რომელშიც ყოველი პროცესი აბლოკირებს ერთ ან რამდენიმე რესურსს, რომელიც ესაჭიროება ჯაჭვში მომდევნო პროცესს).

ჩიხური პროცესების მართვის პრობლემა ოპერაციულ სისტემებში განიხილება შემდეგი ამოცანების გადაწყვეტით:

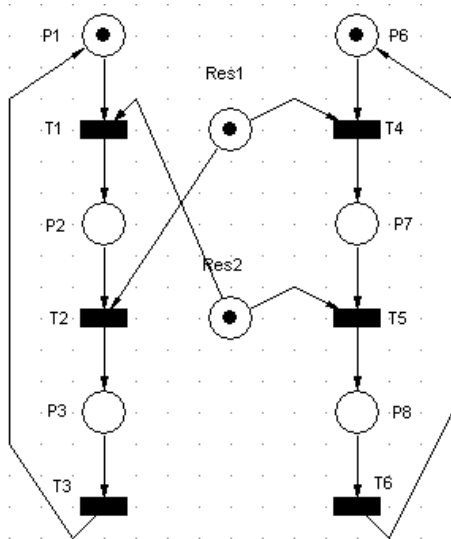
- ჩიხების თავიდან აცილება. თუ ჩიხების არსებობის აღწერილი პირობებიდან მოხერხდება ერთი ან რამდენიმე პირობის მოხსნა, მაშინ შესაძლებელია ჩიხების აღმოცენების თავიდან აცილება;

- ჩიხების გერდის ავლა. აქ პრინციპულად დასაშვებია ჩიხური სიტუაციის არსებობა, მაგრამ მისი მოახლოებისას მიიღება შესაბამისი გამაფრთხილებელი ზომები. ამ დროს შესაძლებელია რესურსების უფრო რაციონალური გამოყენება, ვიდრე წინა შემთხვევაში;

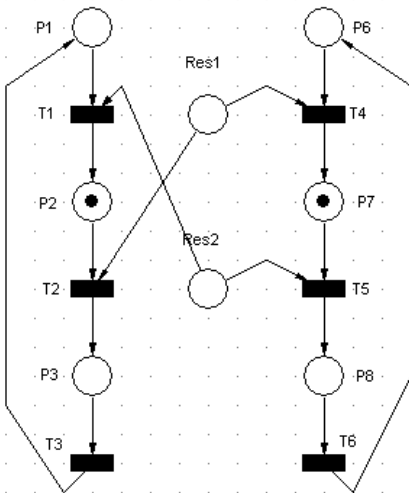
- ჩიხების აღმოჩენა. ამ დროს ჩიხური სიტუაციები ლოკალიზდება და ოპერატორს მიეწოდება სათანადო ინფორმაცია მათ შესახებ;

- ჩიხური სიტუაციის აღდგენა. ესაა ჩიხური სიტუაციიდან გამოსვლა მიმდინარე მუშაობის შედეგების გარკვეული დანაკარგებით.

ახლა განვიხილოთ კონკრეტული შემთხვევა ორი პროცესისთვის (Proz1, Proz2), რომლებიც ორი საერთო რესურსის (Res1, Res2) გამოყენებით ასრულებენ გარკვეულ პროცედურათა მიმდევრობას. 2.16 ნახაზზე წარმოდგენილია შესაბამისი პეტრის ქსელის გრაფი საწყის და შუალედურ (ჩიხურ) მდგომარეობაში.



ა)

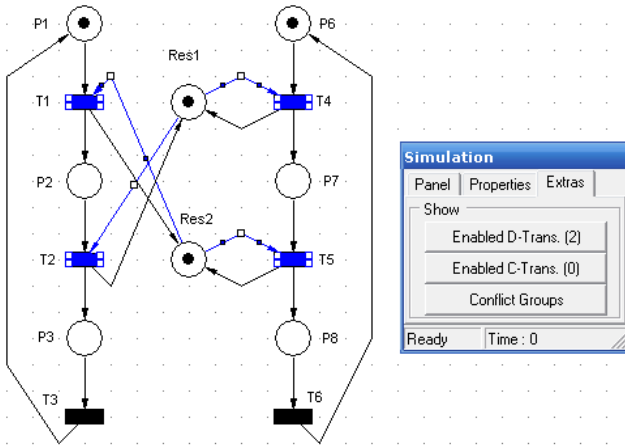


ბ)

ნახ.2.16. საწყისი მდგომარეობა (ა),
ჩიხური სიტუაცია ბლოკირებული რესურსებით (ბ)

ორივე რესურსი ბლოკირებულია შუალედურ პროცედურაში და ელოდება მეორე რესურსს. ამ შემთხვევაში პეტრის ქსელი უძლურია პროცესის გასაგრძელებლად. საჭიროა დამატებითი რკალების შემოტანა, რომლებიც უზრუნველყოფს ბლოკირებული რესურსების გათავისუფლებას.

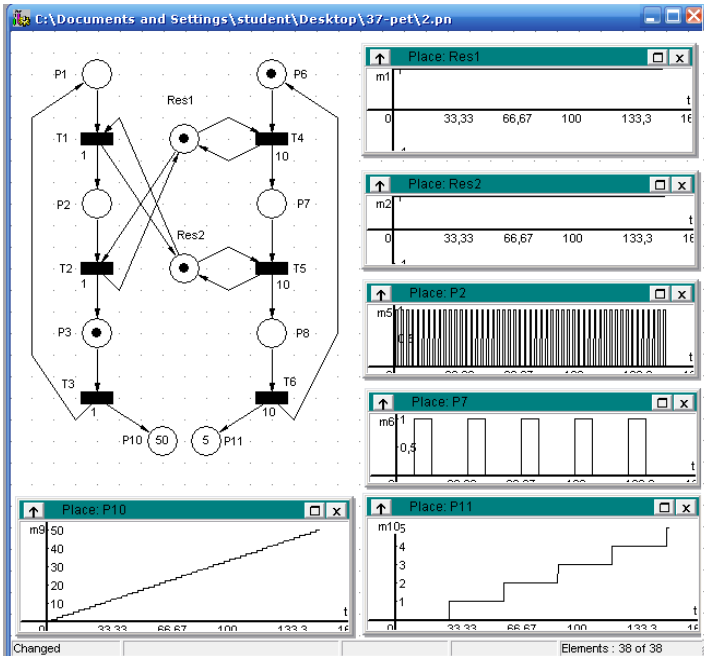
2.17 ნახაზზე დამატებულია აღნიშნული რკალები. აქვე ნაჩვენებია კონფლიქტურ გადასასვლელთა ჯგუფი.



ნახ.2.17. კონფლიქტური ჯგუფის გადასასვლელი

2.18 ნახაზზე ნაჩვენებია პეტრის ქსელის გრაფის იმიტაციის პროცესის შედეგები, მათი ცალკეული პოზიციების დროითი დიაგრამებით. სქემაზე ProZ1-ის გადასასვლელების (პროცედურათა შესრულების) დაყოვნების დრო არის, პირობითად, 1 წმ, ხოლო ProZ2-ის 10 წმ.

მონაცემთა განაწილებული ბაზების ადმინისტრირებისათვის ნებისმიერი ოპერაციული სისტემის პლატფორმაზე, ჩიხური პროცესების მართვა, ანუ აღმოჩენა და მისი დროული გამორიცხვა შესაძლებელია შესაბამისი პროცესების მოდელირებით პეტრის ქსელის გრაფო-ანალიზური ინსტრუმენტის საფუძველზე, რაც ზემოთ იყო ილუსტრირებული. აგებული მოდელის იმიტაციური პროცესის გამოკვლევა იძლევა შესაბამის დროითი მახასიათებლებს გარკვეული დასკვნების გასაკეთებლად.



ნახ.2.18. პროცესების შესრულების დროითი მასხასიათებლები

2.6. მონაცემთა განაწილებული ბაზების დაპროექტება კატეგორიული მიდგომით და ობიექტ-როლური მოდელის ინსტრუმენტით

განიხილება დაპროგრამების UML ტექნოლოგიის და ობიექტ-როლური მოდელირების ORM ინსტრუმენტის გამოყენებით არსთა-დამოკიდებულების ERM მოდელის ავტომატიზებულ რეჟიმში აგების ამოცანა. კატეგორიული მიდგომისა და ლოგიკურ-ალგებრული მეთოდების გამოყენებით განხორციელებულია საპრობლემო სფეროს მონაცემთა სტრუქტურებისა და ცოდნის ობიექტ-ორიენტირებული მოდელირება [23].

ინფორმაციული ტექნოლოგიების განვითარების თანამედროვე ეტაპზე განსაკუთრებით აქტუალურია სამეცნიერო და საინჟინრო სამუშაოების შესრულება განაწილებული მართვის ავტომატიზებული

სისტემების დასაპროექტებლად და სარეალიზაციოდ ობიექტ-ორიენტირებული მიდგომის საფუძველზე. უნიფიცირებული მოდელირების ენის (UML) ინსტრუმენტების გამოყენებით მიიღწევა დაპროექტებისა და დაპროგრამების ურთულეს პროცესთა ავტომატიზაცია, რაც საგრძნობლად ამცირებს საპროექტო-საინჟინრო ეტაპების შესრულების დროს და ამაღლებს საბოლოო პროდუქტის, მართვის სისტემის ხარისხს [24,25].

საუნივერსიტეტო განათლების სისტემა კომპლექსური და მეტად მნიშვნელოვანი ობიექტია მართვის პროცესების შემდგომი სრულყოფის თვალსაზრისით ახალი საინფორმაციო ტექნოლოგიების ბაზაზე [25,26]. აგრეთვე, ჩვენი ქვეყნის მრავალი დარგი, კერძოდ ენერგეტიკა, მრეწველობა, სოფლის მეურნეობა, ჯანმრთელობის დაცვა და ა.შ. ინფორმაციისა და მართვის ამოცანების გადასაჭრელად ფართოდ იყენებს რელაციური ბაზების მართვის სისტემებს, რომელთა შემდგომი სრულყოფა მათემატიკური მოდელირების სემანტიკური პრობლემების აღმოსაფხვრელად და ბაზების სტრუქტურების ავტომატიზებულ რეჟიმში დასაპროექტებლად – მეტად მნიშვნელოვანი საკითხია.

მონაცემთა რელაციური მოდელი ფორმალური მათემატიკური ობიექტია და მისი საშუალებით საგნობრივი სფეროს არაფორმალური თვისებების ასახვას მივყავართ რთულ სემანტიკურ პრობლემამდე. იგულისხმება სემანტიკის (შინაარსის) მათემატიკური მოდელირების სპეციფიკური პრობლემები. მათი სირთულე ძირითადად განისაზღვრება მონაცემთა ბაზის რეორგანიზაციის (განახლების) პროცედურების სირთულით, სტატიკური და დინამიკური შეზღუდვების (პრედიკატების) სისწორის შემოწმებით. შეზღუდვები, გამოხატავს მონაცემთა ბაზის მდგომარეობის (რელაციები) ზოგად, აბსტრაქტულ თვისებებს ანუ მონაცემთა ბაზის სემანტიკას. შეზღუდვების გამოხატვის ყველაზე ბუნებრივი ხერხია გამონათქვამები I-რივის პრედიკატების ენაზე, რომელიც განსაზღვრავს მონაცემთა ბაზის დასაშვებ მდგომარეობათა სიმრავლეს. ამ გამონათქვამების ერთობლიობა მოდელიდან ამოსარჩევი ელემენტის სახელებთან ერთად (ატრიბუტების დასახელება) ქმნის მონაცემთა ბაზის ე.წ. სტატიკურ სქემას. მონაცემთა ბაზა, გადადის რა ერთი მდგომარეობიდან მეორეში (რელაციური ცვალებადობა), აღწერს მონაცემთა მოდელში გარკვეულ ტრანსფორმაციას. ყოველი მომდევნო მდგომარეობა შეიძლება დამოკიდებული იყოს მის

წინა მდგომარეობაზე. ამ მდგომარეობებს შორის კავშირები აღიწერება მთლიანობის დინამიკური შეზღუდვებით (პრედისკატებით), რომლებიც ქმნის მონაცემთა ბაზის ე.წ. დინამიკურ სქემას.

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

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

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

ჩვენ შემოვიფარგლეთ ზოგიერთი ძირითადი პრობლემური საკითხების ანალიზით, რომლებიც დამახასიათებელია რელაციული

მოდელისათვის. შეიძლება აღინიშნოს, რომ ასეთი პრობლემების გადაჭრა შესაძლებელია როგორც რელაციური მოდელის შედარებით გართულებით (დამატებითი ინფორმაციის შემოტანით), ასევე პროგრამული და აპრატული საშუალებების სრულყოფით მომავალში.

კატეგორიები, რომლებსაც აქ ვინილავთ, 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. პრედიკატული ლოგიკა უნდა განვიხილოთ, როგორც ლოგიკური გრამატიკის გაფართოება.

ORM არის მოდელირება ფაქტების საფუძველზე, სადაც საპრობლემო არე განიხილება, როგორც ობიექტების ერთობლიობა, რომლებიც თამაშობს განსაზღვრულ როლებს [25]. კონცეპტუალური მოდელირება მიიღწევა არსთა დამოკიდებულების (ER) მოდელირება, თუმცა იგი შეიძლება გამოვიყენოთ მას შემდეგ, რაც დაპროექტების პროცესი დამთავრებულია. ER-დიაგრამა შორსაა ბუნებრივი ენისაგან, ვერ ხერხდება ამა თუ იმ მოვლენის ფაქტით შევსება, დამალულია ინფორმაცია იმ სემანტიკური დომენების შესახებ, რომლებიც ქმნის მოდელს.

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

დაპროექტების აღნიშნული პროცესი რამდენიმე ეტაპს მოიცავს:

1. ORM -ის კონცეპტუალური მოდელირების სქემის პროცედურა ანუ CSDP (conceptual schema design procedure) ყურადღებას ამახვილებს მონაცემების ანალიზზე და დაპროექტებაზე. ელემენტარული ფაქტების ფორმირება და მათი აღქვებურობის შემოწმება;

2. ფაქტების ტიპებისათვის დიაგრამის აგება და სისრულის შემოწმება;

3. იმ ობიექტთა ტიპების შემოწმება, რომლებიც უნდა გაერთიანდეს და მათი მათემატიკური წარმომავლობის დაფიქსირება;

4. დაემატოს უნიკალურობის შეზღუდვა და შემოწმდეს ფაქტების ტიპების ოპერანდების რაოდენობა;

5. დაემატოს როლების იძულებითი შეზღუდვები და შემოწმდეს მათი ლოგიკური წარმომავლობა;

6. დაემატოს ელემენტები, სიმრავლეთა შედარება და ქვეტიპის შეზღუდვები;

7. დაემატოს სხვა შეზღუდვები და მოხდეს საბოლოო შემოწმება.

პირველი ეტაპი ყველაზე მნიშვნელოვანია, ვინაიდან აქ ხდება სხვადასხვა სახის ინფორმაციის შეგროვება ბუნებრივ სალაპარაკო ენაზე. ასეთი ინფორმაცია ხშირად არის შემავალი და გამომავალი ფორმების ან ხელნაწერის სახით. შესაძლებელია მოდელის დამპროექტებელს მოუხდეს უშუალო კონსულტაციის მიღება კლიენტთან, რათა ზუსტად ჩამოაყალიბოს სისტემის მოთხოვნები.

ფაქტი ჩაიწერება შემდეგი ფორმით:

- *f1*: თანამშრომელს ნომრით 25 აქვს გვარი 'დოლიძე'

- *f2*: თანამშრომელი ნომრით 17 მუშაობს კონტრაქტით თარიღამდე '12.31.08'

თითოეული ფაქტი არის ბინარული დამოკიდებულება ორ ობიექტს შორის. პრედიკატს შეიძლება ჰქონდეს 1,2,3,.. ოპერანდი, თუმცა რადგან პრედიკატი ელემენტარულია 3-4 ოპერანდზე მეტი იშვიათად გვხვდება. უმრავლეს შემთხვევაში პრედიკატი არის ორობითი. ასეთი პრედიკატებისათვის არსებობს ინვერსული პრედიკატი. ისე, რომ ფაქტი შეიძლება წავიკითხოთ ორივე მიმართულებით.

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

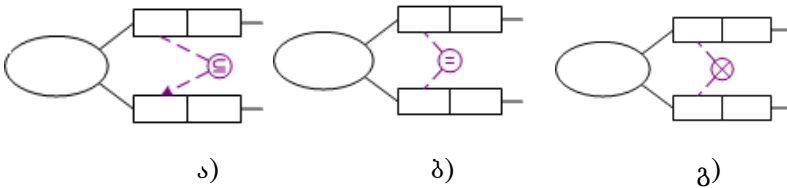
შემდეგ ბიჯებზე ხდება შეზღუდვების დაწესება. ობიექტ-როლური მოდელირება გამოიყენებს: უნიკალურობის, იძულების, წრიული, ქვესიმრავლეთა შედარების, სიხშირის, ინდექსურ და მნიშვნელობის სახის შეზღუდვებს.

ახლა განვიხილოთ ობიექტ-როლური მოდელის აგების ტიპური ამოცანები ალგებრულ-ლოგიკური შეზღუდვების გამოყენებით. ვინაიდან მონაცემთა ბაზების დაპროექტებისას გვიხდება კონკრეტულ საპრობლემო სფეროებთან ურთიერთობა (ინფორმაციის შინაარსის თვალსაზრისით), აქ ჩვენ განვიხილავთ სხვადასხვა მაგალითებს, განსხვავებული სფეროებიდან.

სიმრავლეთა შედარება: ქვესიმრავლის და ტოლობის შეზღუდვები.

თუ ორი სხვადასხვა როლი თამაშდება ერთი ობიექტის მიერ, მნიშვნელოვანი ხდება მათი შედარება. მონაცემთა ბაზებში არსებობს შედარების სამი ოპერატორი, რომლებიც ORM-დიაგრამაზე გამოისახება ასეთი სახით:

- ქვესიმრავლის შეზღუდვა – **Subset constraint (ნახ.2.19-ა);**
- ტოლობის შეზღუდვა – **Equality constraint (ნახ.2.19-ბ);**
- წყვილის გამორიცხვის შეზღუდვა – **pair-exclusion constraint (ნახ.2.19-გ)**

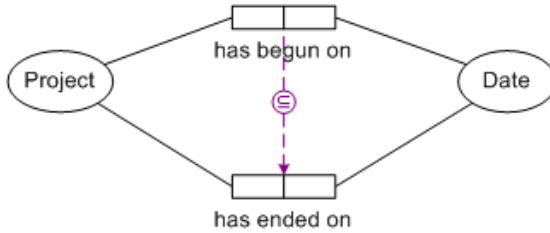


ნახ.2.19. ქვესიმრავლის, ტოლობის და წყვილის გამორიცხვის შეზღუდვები

ქვესიმრავლის შეზღუდვა – პირველი როლის ნაკრები არის მეორე როლის ნაკრების ქვესიმრავლე. მაგალითად,

- f1 : პროექტი (ნრ) დაიწყო განსაზღვრულ დღეს (მდყ)
- f2 : პროექტი (ნრ) დამთავრდა განსაზღვრულ დღეს (მდყ)

პროექტს ვერ ექნება დამთავრების თარიღი მანამ, სანამ არ ექნება დაწყების თარიღი (ნახ.2.20).

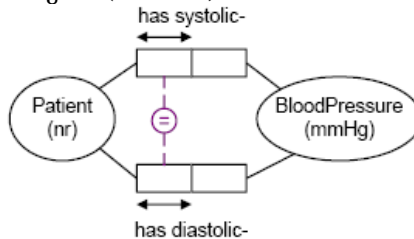


ნახ.2.20. ქვესიმრავლის შეზღუდვა

ტოლობის შეზღუდვა გამოიყენება მხოლოდ მაშინ, როდესაც როლები თავსებადია. იგი გვიჩვენებს, რომ ორი როლი პირობითად ტოლია.

- f1 : თუ პაციენტს აქვს არტერიული წნევა, მაშინ მას აუცილებლად ექნება გულის წნევა.

თუ ცნობილია ავადმყოფის არტერიული წნევა, მაშინ ცნობილია გულის წნევაც და პირიქით (ნახ.2.21).



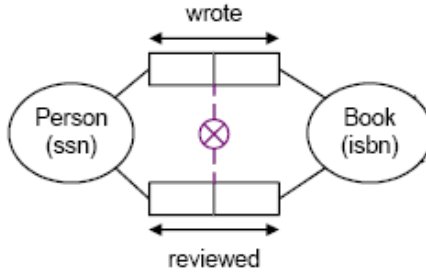
ნახ.2.21. ტოლობის შეზღუდვა

წყვილის გამორიცხვის შეზღუდვა აღწერს ორი როლის ურთიერთგამორიცხულობას. მაგალითად,

- f1 : პერსონამ (ssn) დაწერა წიგნი (ISBN)

- f2 : - პერსონამ (ssn) დაწერა რეცენზია წიგნზე (isbn)

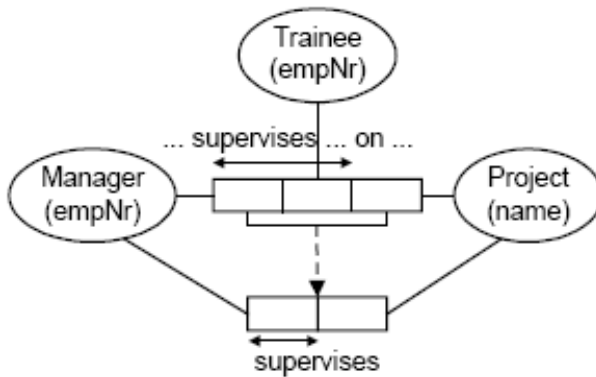
პერსონა საკუთარი წიგნის შესახებ ვერ დაწერს რეცენზიას (ნახ.2.22).



ნახ.2.22. წველის გამორიცხვის შეზღუდვა (მაგ.1)

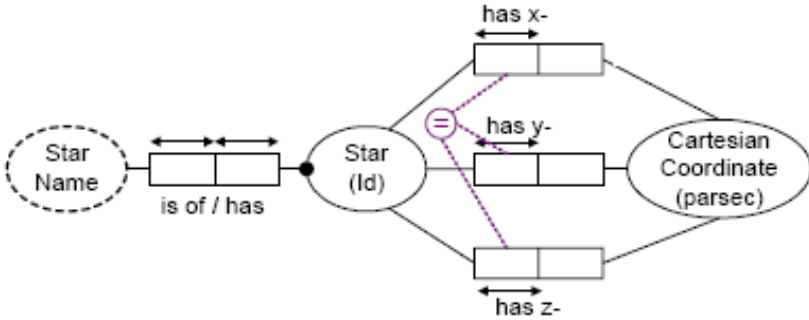
აღწერით ფაქტები:

- f1 : მენეჯერი ხელმძღვანელობს პროექტს (ნახ.2.23).
- f2 მენეჯერი ასევე აკონტროლებს ტრენინგს ამ პროექტისათვის.
- f3 თითოეული მენეჯერი რომელიც აკონტროლებს ტრენინგს ასევე ხელმძღვანელობს პროექტს.



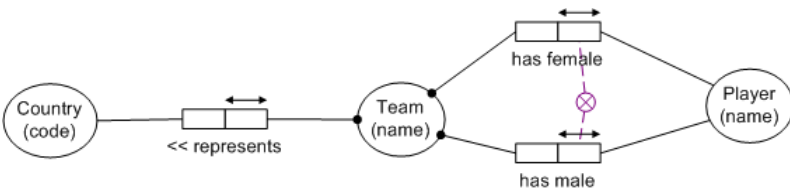
ნახ.2.23. წველის გამორიცხვის შეზღუდვა (მაგ.2)

- ა) - f1: ვარსკვლავს აქვს სახელი (ნახ.2.24).
 - f2: ვარსკვლავს აქვს x-დეკარტული კოორდინატი.
 - f3: ვარსკვლავს აქვს y-დეკარტული კოორდინატი.
 - f4 : ვარსკვლავს აქვს z-დეკარტული კოორდინატი.



ნახ.2.24. წყვილის გამორიცხვის შეზღუდვა (მაგ.3)

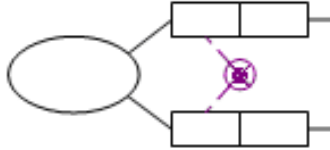
- გ) - f1: გუნდი წარმოადგენს ქვეყანას (ნახ.2.25).
 - f2: გუნდი შედგება ქალი მოთამაშეებისაგან.
 - f3: გუნდი შედგება კაცი მოთამაშეებისაგან.
 - f4: არ არებობს ისეთი გუნდი, სადაც იქნებიან როგორც ქალი, ისე კაცი მოთამაშეები.



ნახ.2.25 წყვილის გამორიცხვის შეზღუდვა (მაგ.4)

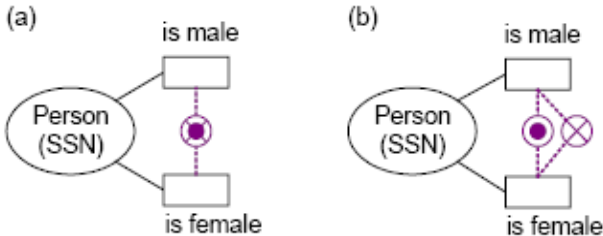
სიმრავლეთა შედარების შეზღუდვა: იძულების დიზუნქცია.

იძულების დიზუნქციის შეზღუდვაა – inclusive-or constraint (ნახ.2.26)



ნახ.2.26. იძულების დიზუნქციის შეზღუდვა (მაგ.1)

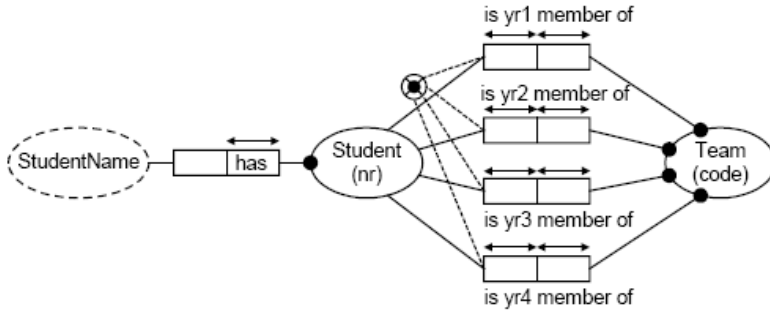
იგი გვჩვენებს, რომ ობიექტი ვალდებულია შეზღუდვაში ასრულებდეს მხოლოდ ერთ როლს და არა ორივეს ერთად. მაგალითად, პერსონა შეიძლება იყოს ქალი ან კაცი და არა ორივე ერთად ORM-დიაგრამაზე გამოისახება ასეთი სახით (ნახ.2.27).



ნახ.2.27. იძულების დიზუნქციის შეზღუდვა (მაგ.2)

აღწერით შემდეგი ფაქტები:

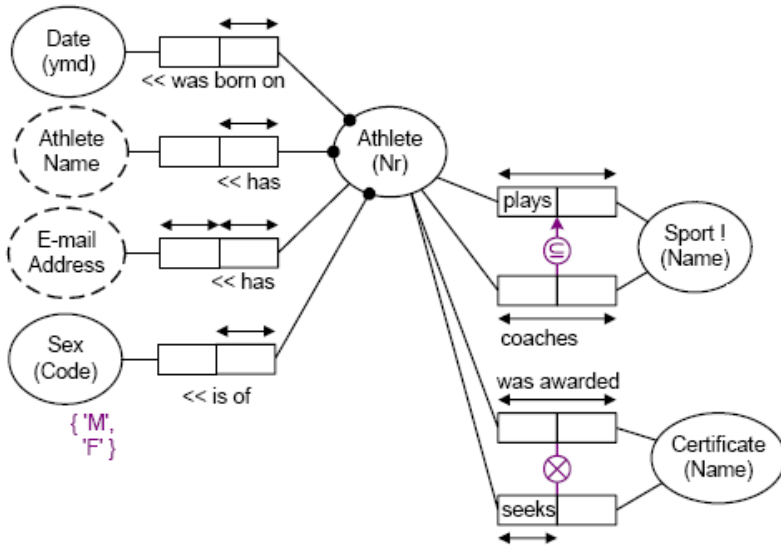
- ა) - f1 : სტუდენტს აქვს გვარი (ნახ.2.28).
- f2 : სტუდენტი 'აბაშიძე რ. №1' არის მხოლოდ და მხოლოდ '608836' ჯგუფში.
- f3 : სტუდენტი 'ბაქრაძე დ. №5' არის მხოლოდ და მხოლოდ '608835' ჯგუფში.
- f4 : სტუდენტი 'დვალი ს. №9' არის მხოლოდ და მხოლოდ '608837' ჯგუფში.
- f5 : სტუდენტი 'იობაძე რ. №35' არის მხოლოდ და მხოლოდ '608839' ჯგუფში.



ნახ.2.28. იძულების დიზუნქციის შეზღუდვა (მაგ.3)

- ბ) - f1: ათლეთს აქვს სახელი (ნახ.2.29).
- f2: ათლეთი დაიბადა 1967/01/18.
- f3: ათლეთს აქვს იმეილი.
- f4: ათლეთს აქვს სქესი მამრ. ან მდედრ. (.)
- f5: ათლეთი მონაწილეობს .. სახეობაში.
- f6: ათლეთი ვარჯიშობს .. სახეობაში.
- f7: თითოეული ათლეთი მონაწილეობს მხოლოდ იმ სახეობის შეჯიბრში, რომელშიც ვარჯიშობს.
- f8: ათლეთს მინიჭებული აქვს სერთიფიკატი.
- f9: ათლეთი იღწვის სერთიფიკატის მოსაპოვებლად.
- f10: ათლეთი, რომელმაც უკვე მოიპოვა სერთიფიკატი აღარ მოიპოვებს მას მეორედ.

მონაცემთა ბაზის სტრუქტურის დაპროექტების პროცესების ავტომატიზაცია ეფექტურად ხორციელდება ობიექტ როლური მოდელირების ინსტრუმენტით, რომლის თეორიულ საფუძველსაც კატეგორიალური მიდგომა წარმოადგენს. ლოგიკურ-ალგებრული მეთოდების გამოყენებით ხორციელდება საპრობლემო სფეროს მონაცემთა სტრუქტურებისა და ცოდნის ობიექტ-ორიენტირებული მოდელირება.



ნახ.2.29. იძულების დიზუნქციის შეზღუდვა (მაგ.4)

2.7. განაწილებული სისტემის კონცეპტუალური მოდელის დაპროექტების და მონაცემთა ბაზის აგების პროცედურების ავტომატიზაცია

მოცემულ პარაგრაფში განიხილება განაწილებული მართვის ორგანიზაციული სისტემის არსთა-დამოკიდებულების მოდელის (ERM) დაპროექტების ავტომატიზებული პროცესი წინა პარაგრაფში ჩამოყალიბებული ობიექტ-როლური მოდელის (ORM) საფუძველზე [27].

ამგვარად, საჭიროა დავაპროექტოთ რელაციური ბაზისთვის შესაბამისი საპრობლემო სფეროს კონცეპტუალური ER-მოდელი, მაგალითისათვის გამოვიყენოთ „უნივერსიტეტი“ (განათლების საგნობრივი სფერო). ამოცანის გადასაწყვეტად ვიყენებთ მოდელირების ობიექტ-ორიენტირებულ მეთოდს, დაპროგრამების თანამედროვე .NET-პლატფორმას, MS Visio ინსტრუმენტით, და MsSQL Server პაკეტს [23,27,32,35-39].

კომპიუტერული და საინფორმაციო ტექნოლოგიების განვითარების თანამედროვე ეტაპზე განსაკუთრებით აქტუალურია

სამეცნიერო და საინჟინრო სამუშაოების შესრულება განაწილებული მართვის ავტომატიზებული სისტემების დასაპროექტებლად და სარეალიზაციოდ ობიექტ-ორიენტირებული მიდგომის საფუძველზე. უნიფიცირებული მოდელირების ენის (UML) ინსტრუმენტების გამოყენებით მიიღწევა დაპროექტებისა და დაპროგრამების ურთულეს პროცესთა ავტომატიზაცია, რაც საგრძნობლად ამცირებს საპროექტო-საინჟინრო ეტაპების შესრულების დროს და ამაღლებს საბოლოო პროდუქტის, მართვის სისტემის ხარისხს. საუნივერსიტეტო განათლების სისტემა კომპლექსური და მეტად მნიშვნელოვანი ობიექტია მართვის პროცესების შემდგომი სრულყოფის თვალსაზრისით ახალი საინფორმაციო ტექნოლოგიების ბაზაზე [23,32].

საუნივერსიტეტო მართვის სისტემის საპრობლემო სფეროს მაგალითზე საჭიროა მონაცემთა ისეთი განაწილებული ბაზის დაპროექტება და აგება, რომელიც მომავალში მიუერთდება ქვეყნის ერთიანი საგანმანათლებლო სისტემის მონაცემთა საცავს. ასეთი საინფორმაციო სისტემის აგებისას აუცილებელია მიღებული საერთაშორისო სტანდარტების გათვალისწინება, რაც მნიშვნელოვნად შეუწყობს ხელს მის შემდგომ მასშტაბირებას და განვითარებას.

ასეთ სტანდარტს (უნიფიცირებული მოდელირების ენას, წესებს) წარმოადგენს -ტექნოლოგია, ხოლო მის ერთ-ერთ გავრცელებულ ინსტრუმენტს Ms Visio [1,40]. წინამდებარე ნაშრომის მიზანია ამ ინსტრუმენტების გამოყენებით მონაცემთა ბაზის კონცეპტუალური და ლოგიკური სტრუქტურების დამუშავება, შემდგომ კი მისი ფიზიკური ორგანიზაცია, რეალიზაცია მაგალითად, SQL Server მონაცემთა ბაზების მართვის სისტემაში [39].

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

ესაა ცოდნა სამართავი ობიექტის შესახებ, მისი სტატიკური (მდგომარეობათა სიმრავლე) და დინამიკური (ქცევათა სიმრავლე) მოდელებით. თუ ობიექტ-ორიენტირებული მოდელირების ტერმინებით ვისარგებლებთ, დასმული ამოცანის გადაწყვეტის „გასაღებს“ კლასების,

ობიექტების, კლასთაშორისი კავშირების, ობიექტ-როლური და არსთა-დამოკიდებულების მოდელებისა და სხვა სახის დიაგრამების აგება წარმოადგენს. ხოლო შემდეგ, კლასთა-ასოციაციებისა და არსთა-დამოკიდებულების დიაგრამათა საფუძველზე განხორციელდება მიზნობრივი სისტემის პროგრამული კოდების რეალიზაციის ავტომატიზებული პროცესი.

განვიხილოთ ეს საკითხები დეტალურად უნივერსიტეტის საპრობლემო სფეროს მაგალითზე. არაფორმალიზებული აღწერის ობიექტებია (ტერმინთა ლექსიკონი): ფაკულტეტები, კათედრები, სტუდენტები, ლექტორები (როგორც თანამშრომლები, ისე მოწვეულები, სამტატო და საათობრივი დატვირთვით), საგნები (აკადემიური დისციპლინები), რომლებიც იკითხება კათედრებისა და სპეციალობების მიხედვით. სასწავლო გეგმები, სილაბუსები (პროგრამები), ლექციები, პრაქტიკული და ლაბორატორიული სამუშაოები, გამოცდები, ტესტირება და ა.შ.

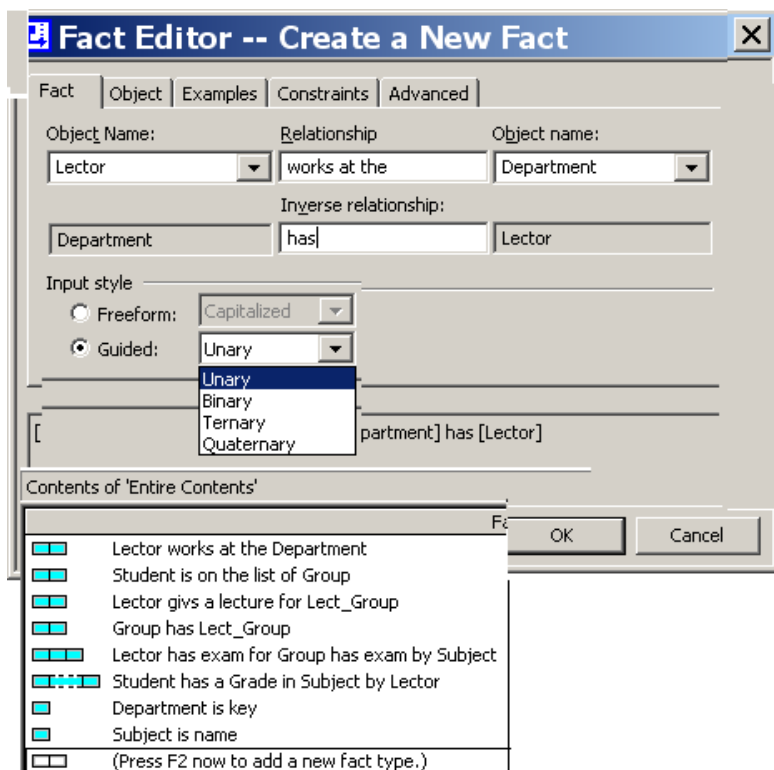
მეორე ეტაპზე საპრობლემო სფეროს შესახებ ცოდნა (კლასებისა და ობიექტების ძირითადი ტერმინები და ქცევის წესები) ჩვენს მიერ გადაიტანება Ms_Visio ინსტრუმენტის საშუალებით ობიექტ-როლურ მოდელში (ნახ.2.30-ა,ბ).

მომდევნო ეტაპზე განხორციელდება ობიექტ-როლური მოდელის ავტომატური გადაყვანა არსთა-დამოკიდებულების მოდელში. ამისათვის Ms_Visio ინსტრუმენტის (ORM-source) მენიუდან გამოვიძახებთ სტრიქონს :

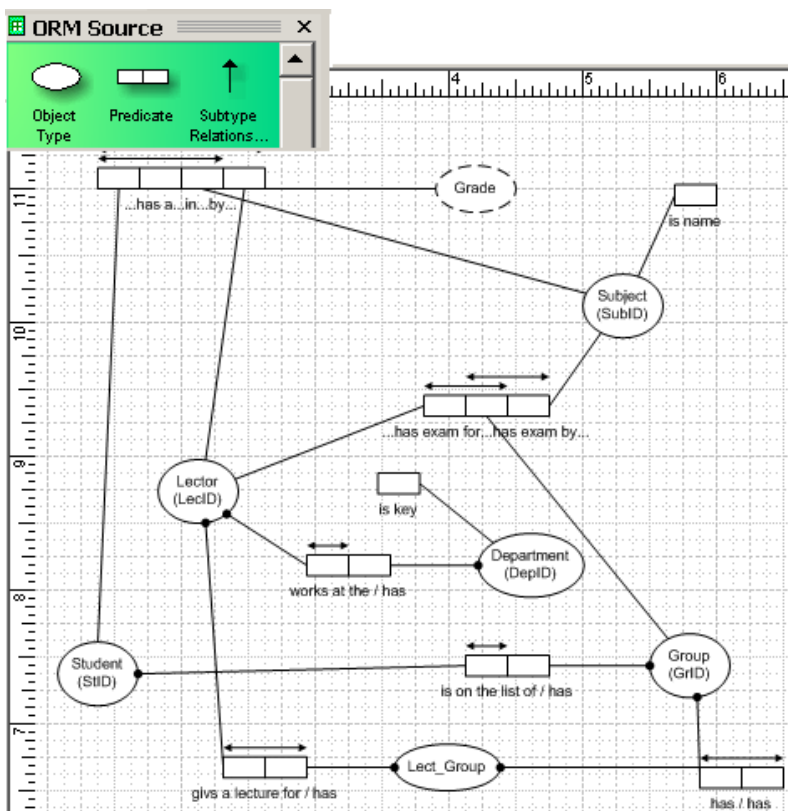
File | New | Database | Database Model Diagram

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

Database | Project | Add Existing Document

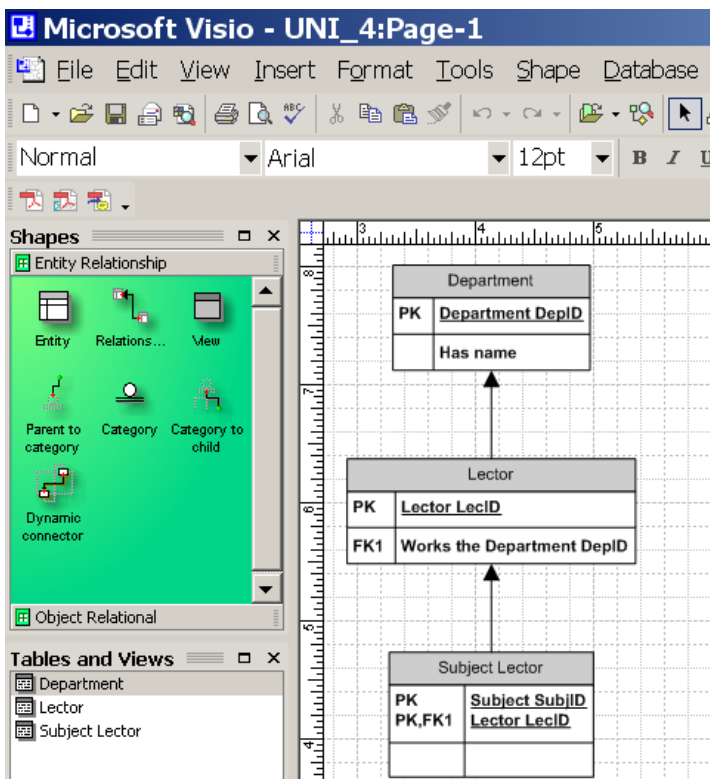


ნახ.2.30-ა. ობიექტებისა და მათ შორის კავშირების (წესების) აგების ინტერფეისი



ნახ.2.30-ბ. ობიექტ-როლური მოდელის მაგალითი

2.31 ნახაზზე ნაჩვენებია ფრაგმენტი ასეთი გზით მიღებული შედეგისათვის.



ნახ.2.31. არსთა-დამოკიდებულების მოდელის ფრაგმენტი

ნახაზზე ჩანს სამი ცხრილი : ლექტორი, კათედრა და საგანი, აგრეთვე მათ შორის კავშირები. თუ ყველა ცხრილს გამოვიყენებთ 2.30-ბ ნახაზიდან, მაშინ მივიღებთ რეალურ, მრავალცხრილურ კავშირებს.

მონაცემთა ბაზის აგების ავტომატიზაციის შემდეგი პროცედურა მდგომარეობს შუალედური ტექსტური ტიპის DDL-ფაილის მომზადებაში, რომელიც მომავალში SQL Server მონაცემთა ბაზების მართვის სისტემამ უნდა გამოიყენოს [39].

ქვემოთ მოცემულია ავტომატურად გენერირებადი DDL-ფაილის ტექსტის ფრაგმენტი ჩვენი მაგალითისთვის :

```

/* Driver Used : Microsoft Visual Studio - Microsoft SQL Server Driver.
*/
/* Create DB_Uni database.          */
use master
go
create database "DB_Uni"
go
use "DB_Uni"
go
/* Create new table "Subject Lector".          */
/* "Subject Lector" : Subject has a Lector / Lector gives a Subject */
/* "Subject SubjID" : Subject has a Lector          */
/* "Lector LecID" : Lector gives a Subject          */
create table "Subject Lector" (
    "Subject SubjID" char(10) not null,
    "Lector LecID" char(10) not null)
go
alter table "Subject Lector"
    add constraint "Subject Lector_PK" primary key ("Subject SubjID",
" Lector LecID")
go
/* Create new table "Lector".          */
/* "Lector" : Table of Lector          */
/* "Lector LecID" : Lector is identified by LecID          */
/* "Works the Department DepID" : Department has Lector */
create table "Lector" (
    "Lector LecID" char(10) not null,
    "Works the Department DepID" char(10) not null)
go
alter table "Lector"
    add constraint "Lector_PK" primary key ("Lector LecID")
go
/* Create new table "Department".          */
/* "Department" : Table of Department          */
/* "Department DepID" : Department is identified by DepID */
/* "Has name" : Department has name          */
create table "Department" (
    "Department DepID" char(10) not null,
    "Has name" bit not null)
go
alter table "Department"

```

```

add constraint "Department_PK" primary key ("Department
DepID")
go
/* Add foreign key constraints to table "Subject Lector". */
alter table "Subject Lector"
add constraint "Lector_Subject Lector_FK1" foreign key (
"Lector LecID")
references "Lector" (
"Lector LecID")
go
/* Add foreign key constraints to table "Lector". */
alter table "Lector"
add constraint "Department_Lector_FK1" foreign key (
"Works the Department DepID")
references "Department" (
"Department DepID")
go
/* This is the end of the Microsoft Visual Studio generated SQL DDL script. */

```

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

ახლა, ბოლო ეტაპზე განვიხილოთ კლასების დიაგრამასა და მონაცემთა ბაზის სტრუქტურას შორის მიმართების საკითხი [1,14,54].

კლასი, ობიექტ-ორიენტირებული მიდგომის ცნებაა და იგი მონაცემთა (ატრიბუტების) და ფუნქციათა (მეთოდების) ინკაპსულაციაა. აგრეგატული კავშირების საფუძველზე მეტაკლასს აქვს შემდეგი თვისებები: სახელი (კლასის იდენტიფიკატორი), ატრიბუტები (კლასის მდგომარეობა) და მეთოდები (კლასის ყოფაქცევა). მეტაკლასის

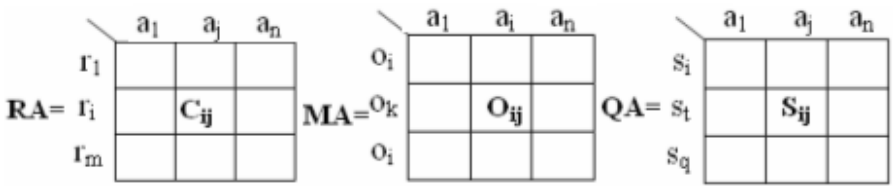
ეგზემპლარები კონკრეტული კლასებია, რომელთაც აღნიშნულ თვისებათა განსაზღვრული მნიშვნელობები აქვს და აგრეგატულად შედგება ერთტიპური ობიექტების ერთობლიობისგან.

კლასის რომელიმე მეთოდის გააქტიურება ამ კლასის მონაცემთა განსაზღვრული დამუშავების მიზნით, ხორციელდება კლასის გარედან მოსული შეტყობინების (Message) საფუძველზე. ესაა ინფორმაცია მართვის ავტომატიზებული სისტემის გარე მომხმარებელთა მოთხოვნების შესახებ. ამგვარად, მომხმარებელთა მოთხოვნის სტრუქტურის დინამიკურად ფორმირების საფუძველზე, ოპერატიულად, ავტომატურად განისაზღვრება მეთოდი (ან მეთოდები) და მისთვის საჭირო კლასის მონაცემები.

დასმული ამოცანის, ანუ მოთხოვნის საფუძველზე კლასების განსაზღვრისათვის ჯერ უნდა განისაზღვროს ბაზის მონაცემთა სტრუქტურები. ამისათვის სრულდება შემდეგი ეტაპები:

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

ლოგიკური, სემანტიკური და მიზეზ-შედეგობრივი დამოკიდებულებების საფუძველზე ზემოაღნიშნულ (A, R, M, Q) , სიმრავლეთა ელემენტებს შორის ვაგებთ სამ მატრიცას RA, MA, QA ატრიბუტების ერთი და იმავე სიმრავლეზე (ნახ.2.32).



ნახ.2.32. მატრიცები

$U=\{a_j\} \quad j=1,n$ ატრიბუტები, $R=\{r_i\} \quad i=1,m$ რელაციები,
 $M=\{O_k\} \quad k=1,p$ მეთოდები, $Q=\{S_t\} \quad t=1,q$ მოთხოვნები

- პირველი RA-მატრიცის ფორმირების მიზნით, რომელიც F_i დამოკიდებულებათა სიმრავლეა დომენების დეკარტული ნამრავლის ქვესიმრავლეზე ($R \subseteq a_1 \in A_1 \times a_2 \in A_2 \times \dots \times a_m \in A_m$), საჭიროა გამოვიყენოთ n -არულ მიმართებათა განსაზღვრის ლოგიკურ-ალგებრული მეთოდი. კორტეჟის ელემენტია:

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

1-ელი მატრიცის დამუშავების შემდეგ იცვლება რელაციურ დამოკიდებულებათა თავდაპირველი სტრუქტურა ანუ საწყის კლასთა დიაგრამაში იცვლება მონაცემთა სტრუქტურები (ცხრილების ნორმალიზაციის მეთოდის გამოყენების საფუძველზე: ჩნდება ახალი კლასები, იცვლება ატრიბუტთა რაოდენობა).

იმისათვის, რომ მიღებული ახალი სტრუქტურა ვაქციოთ კლასთა დიაგრამად, საჭიროა თითოეულ მათგანში განვსაზღვროთ მეთოდების ერთობლიობა. აქ ვიყენებთ მეორე, MA-მატრიცას, რომლის საფუძველზე კლასის ატრიბუტების სიმრავლეს შეესაბამება ესა თუ ის მეთოდი მეთოდების ბიბლიოთეკიდან. შესაძლებელია მეთოდების დუბლირება, რომელიც რეალიზებულ იქნება მეთოდების ინდექსების საშუალებით.

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

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

მესამე, ინფორმაციული QA-მატრიცის ფორმირების მიზნით, რომელიც წარმოადგენს დამოკიდებულებათა შეტყობინებების სიმრავლეს, რომელიც განსაზღვრულია შეტყობინებათა დომენების დეკარტული ნამრავლის ქვესიმრავლეზე ($Q \subseteq s_1 \in S_1 \times s_2 \in S_2 \times \dots \times s_m \in S_m$), საჭიროა გავაანალიზოთ ყველა შესაძლო შეტყობინება. QA მატრიცის ელემენტის კორტეჟია: $S_{ij} = \{(1, \text{if } a_j \in s_i \text{ OR } (0, \text{if } a_j \notin s_i))\}$.

ამ ცხრილის საფუძველზე განისაზღვრება მოთხოვნებისა და მეთოდების ურთიერთშეთანხმების შესაბამისობა.

2.8. ტრანსფორმაციისა და კონვერტაციის ალგორითმების დამუშავება

მონაცემთა საცავი მოიცავს ტრანსფორმაციისა და კონვერტაციის პროგრამებს, საბაზო მეტამონაცემთა სისტემას, არქივირებული შენახვის სისტემას და ინტეგრირებულ მონაცემთა საცავს [20]. განვიხილოთ მოკლედ მათი ფუნქციები და ის ინსტრუმენტები, რომლებიც დამუშავდა ჩვენს მიერ [23].

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 (იერარქიულ, რელაციურ ან ობიექტ-ორიენტირებულ) ან თანამიმდევრულ განსხვავებული ფორმატის ფაილებიდან (ASCII, ANSI, EBCDIC და ა.შ.) ექსტრადირდება. ტრანსფორმაციის წესის თანახმად ისინი ერთიანდება (Bridging), როგორც ინტეგრირებული, სუბიექტ-ორიენტირებული, მულტივი და დროში ცვალებადი სტრუქტურები. ტრანსფორმაციის პროგრამამ უნდა უზრუნველყოს ტრანსპორტირებისათვის ფუნქციათა წარმოდგენა და აგრეთვე მონაცემთა მომზადება საცავში გადასაგზავნად. მონაცემთა საცავში შიგა მონაცემების (ინფორმაციის) უდიდესი ნაწილი შესაძლოა მიღებული იქნას განაწილებული ოპერატიული სისტემიდან. მონაცემთა საცავში ისტორიული და მიმდინარე მონაცემების შევსება, ხდება ბაზებიდან, სადაც პერიოდულად მიმდინარეობს მონაცემთა აქტუალიზაცია. თუ ოპერატიულ მონაცემთა ჩანაწერების რეგისტრაციაში მონაცემის შეტანის თარიღი უფრო ახალია, ვიდრე ბოლო ტრანსფორმაციის შეტანის დრო, მაშინ მოხდება ამ უკანასკნელის ლიკვიდაცია. ე.ი. აუცილებლად

გათვალისწინებული უნდა იქნეს ცალკეულ ტრანსფორმაციის პროცესებს შორის ვადები.

მონაცემთა გარე მიმწოდებლებია შეტყობინებათა სამსახურები, ბირჟები, პოლიტიკური საინფორმაციო სამსახურები, სამეცნიერო კვლევითი ინსტიტუტები და ბაზარი. ინფორმაციის მოწოდებისათვის გამოიყენება ისეთი საშუალებები, 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 შეიძლება განვმარტოთ ასეთი სახით: ეს არის მონაცემები მონაცემების შესახებ. მეტამონაცემთა საბაზო სისტემა არის მონაცემთა საცავის შემადგენელი ნაწილი. 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 მენსიერების 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 დევექტების 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.

მონაცემთა საცავის ანალიზური დამუშავება OLAP – ტექნოლოგიის გამოყენებით. მონაცემთა მრავალგანზომილებიანი, კომპლექსური ანალიზის ტექნოლოგია **OLAP**-ის სახით ჩვენ წინა თავშიც განვიხილეთ. მონაცემთა საცავში იგი წარმოადგენს საკვანძო კომპონენტს.

OLAP – ინსტრუმენტი 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.

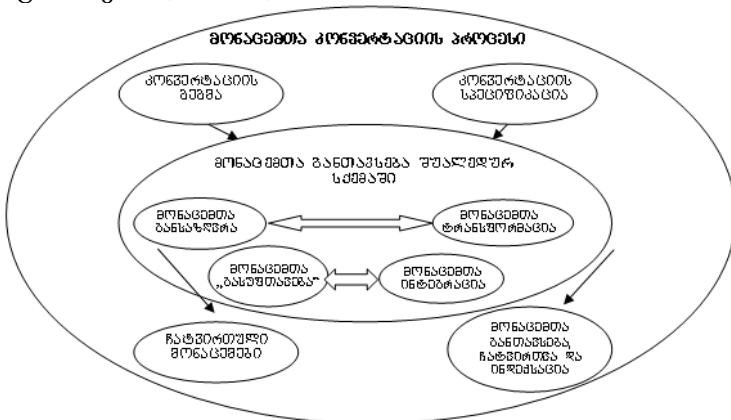
დიდი კორპორაციული სისტემების მონაცემთა საცავებისათვის ინფორმაციის კონვერტაცია საკმაოდ რთული პროცესია და ხასიათდება მაღალი შრომატევადობით. ნაშრომში წარმოდგენილია მონაცემთა საცავის კონვერტაციის პროცესის სტრუქტურა და გამოკვლეულია მისი კავშირი ორგანიზაციულ სისტემებში გადაწყვეტილების მიღების პროცესის დროით მახასიათებლებთან. მონაცემთა ხარისხიან კონვერტაციაზე დამოკიდებულია კორპორაიუციულ სისტემებში ოპტიმალური გადაწყვეტილების მიღება.

კონვერტაციის პროცესის დაწყების წინ საკონვერტაციო ჯგუფი ქმნის პროექტს და წარმოადგენს მონაცემთა ფიზიკურ მოდელს, რაც შემდგომ აისახება მიზნობრივ სქემაზე. მონაცემთა კონვერტაციის ჯგუფი უზრუნველყოფს ბიზნესისა და ტექნიკის სფეროს წარმომადგენლებთან შეთანხმებას და საცავის სპროექტო სტრუქტურის განსაზღვრას, სადაც უნდა მოხდეს საწყისი მონაცემების ანალიზი და ბოლოს განსაზღვრავს მონაცემთა კონვერტაციის ლოგიკას. ამას გარდა ეს ჯგუფი იყენებს მონაცემთა მიგრაციის, ტრანსფორმაციის და გაწმენდის ინსტრუმენტებს. მონაცემთა კონვერტაციის წარმატების მთავარი და კრიტიკული

ფაქტორი მდგომარეობს იმაში, რომ კონვერტაციის ჯგუფის ყველა წევრისათვის გასაგები უნდა იყოს სისტემის მიმართ დაყენებული მოთხოვნა. კონვერტაციის პროცესის დროს თავს იყრის რამდენიმე ალტერნატიული ვარიანტი, როდესაც მოთხოვნა სწორად არის გაგებული, მინიმალური დროის განმავლობაში შესაძლებელი ხდება კონვერტაციის სწორი ფორმის შერჩევა.

კონვერტაციის წინასწარ შედგენილი გეგმა, საცავში განსაზღვრავს საწყის მონაცემთა მიგრაციის ოპტიმალურ გზას, წინასწარი გეგმა ასევე ფლობს რესურსებს, რომლის დახმარებითაც შესაძლებელი ხდება საწყის მონაცემთა მოცულობის განსაზღვრა, ერთმანეთისგან განსხვავებულ პლატფორმათა და მონაცემთა მოპოვების მეთოდების რაოდენობისა და ტიპის განსაზღვრა, მონაცემთა საცავის სტრუქტურის და აუცილებელ მონაცემთა აგრეგაციის მოცულობის განსაზღვრა.

გეგმა ყოველი საწყისი სისტემისათვის ასაბუთებს პლატფორმას, რომელიც უზრუნველყოფს მონაცემთა მოპოვების მეთოდსა და პროგრამირების ენას. თუ ჩვენი ჯგუფი, რამდენიმე საწყისი სისტემიდან შეარჩევს საჭირო ინფორმაციას, მაშინ მონაცემთა კონვერტაციის გეგმა აღწერს მონაცემთა შერჩევის შესაბამის სტრატეგიას, რომელიც შემდგომში სისტემას ამზადებს მონაცემთა ტრანსფორმაციისა და ინტეგრაციისათვის. (ნახ.2.33)



ნახ.2.33. მონაცემთა კონვერტაციის პროცესი

მონაცემთა გადანაცვლების საერთო შუალედური სტრატეგიის შერჩევასა და გათვალისწინებული უნდა იყოს კომპიუტერის რესურსები, საკონვერტაციო ჯგუფის პროფესიონალიზმი და საწყის მონაცემთა მოცულობა.

მონაცემთა კონვერტაციის გეგმაში გათვალისწინებული უნდა იყოს მონაცემთა საცავის სტრუქტურა და მონაცემთა ბაზის მთლიანი სქემა. ჩვეულებრივად პროცესი ასეთია: მონაცემები თავის საწყისი სისტემის სქემიდან გადანაცვლებენ შუალედურ სქემაში. შუალედური სქემა არის სქემა, რომელიც ჩატვირთავს მონაცემებს საწყისი სისტემებიდან. შუალედური სქემა სიზუსტით არ შეესაბამება საწყის ან მთლიან სქემებს. იგი როგორც წესი არის შუალედური და შეიცავს დამატებით ველს, რომელიც შედგება ციფრისგან ან პროცენტისგან და გამოიყენება გამოსაანგარიშებლად. ან ეს ველი შესაძლოა იყოს გასაღებური ველი რომლის დახმარებითაც შესაძლებელია ცხრილების დათვალიერება, საჭირო ტექნიკური პირობების გაუმჯობესება, მონაცემთა გასუფთავება და ტრანსფორმაცია. საბოლოოდ მონაცემთა კონვერტაციის გეგმა უნდა ითვალისწინებდეს მონაცემთა ნაკადის ხასიათს კონვერტაციის მთელ გზაზე, სადც ყურადღება უნდა მიექცეს შემდეგ საკითხებს:

- საწყისი სისტემიდან როგორ ხდება მონაცემთა მიგრაცია შუალედურ სისტემაში;
- ჯგუფი როგორ ახორციელებს ტექნიკურ პირობებს: მონაცემთა გასუფთავება, ტრანსფორმაცია და ინტეგრაცია შუალედურ სისტემაში;
- ჯგუფი როგორ უზრუნველყოფს მონაცემთა საცავისათვის პირველად და შიგა გასაღებს;
- მონაცემები როგორ მიგრირდება შუალედური სისტემიდან საცავის სერვერზე;
- როგორ ინახება კონვერტირებული მონაცემები და როგორ ხდება მათი ექსპორტი რეპოზიტორებში;
- ჯგუფი როგორ ჩატვირთავს და აინდექსირებს მონაცემებს საცავის სერვერზე მონაცემებს;
- ჯგუფი როგორ უზრუნველყოფს დამატებით მონაცემთა წყარის, რომლის ყოფნაც შესაძლოა აუცილებელი გახდეს კონვერტაციის პროცესის ბოლო სტადიაზე;
- ჯგუფი როგორ უზრუნველყოფს კონვერტაციის პროცესი მონაცემთა ხარისხს.

გეგმის შედგენის შემდგომ უნდა განისაზღვროს მონაცემთა ატრიბუტები, თუ როგორი სახის მონაცემები მანიპულირება და როგორი სახის ლოგიკა გამოიყენება მონაცემთა მიგრაციისათვის.

მონაცემთა მიგრაციის ინსტრუმენტის გამოყენებით შესაძლებელია დავასაბუთოთ კონვერტაციის სპეციფიკა და წარმოვადგინოთ ანგარიში მომხმარებლის მოთხოვნის შესაბამისად. მონაცემთა მიგრაციის ინსტრუმენტის გამოყენება, მეტამონაცემთა კონვერტაციის შენახვის პროცესს ცენტრალურ დონეზე ხდის ავტომატიზებულს, საიდანაც რეპოზიტორში დროის ნებისმიერი შუალედში მეტამონაცემები ექსპორტირდება. იმ შემთხვევაში თუ კონვერტაციის პროცედურის გენერირებას ვახდენთ ხელით, ასეთივე წესით ვახდენთ მეტამონაცემთა კონვერტირების გენერაციასაც.

იმისდამიუხედავად კონვერტაციის სპეციფიკა პროგრამულ კოდში ჩატარებულია ხელით თუ ავტომატურად ის უნდა ემსახურობოდეს მონმხმარებლისთვის მონაცემთა მიწოდების სიღრმისეული გაგებასა და ლოგიკას. პროგრამული კოდი შედგება ექვსი პროცედურის ტიპისგან, რომელიც ასრულებს შემდეგ ფუნქციებს:

- საწყისი სისტემიდან მონაცემთა გადატანა შუალედურ სქემაში;
- მონაცემთა ჩატვირთისათვის შუალედური სქემის კონვერტაცია;
- ჩასატვირთი მონაცემების აგრეგაცია;
- ჩასატვირთი მონაცემების მიგრაცია შუალედური დონიდან მონაცემთა საცავის სერვერზე;
- დიდი მოცულობის მონაცემთა ბაზებიდან საცავში მონაცემთა ჩატვირთვა;
- მონაცემთა სისწორის შემოწმება.

ზოგიერთი პროგრამული კოდი რომელიც უზრუნველყოფს საწყისი სისტემიდან მონაცემთა გადატანას შუალედურ სქემაში გამოირჩევა უნიკალურობით თავისი მეთოდით პლატფორმითა და საწყისი სისტემის ენით. ამას გარდა ეს კოდი საწყისი მონაცემებიდან ამზადებს, წმენდს, ახდენს მონაცემთა ტრანსფორმაციასა და ინტეგრირებას საცავში. ამ კოდის მუშაობას დიდი მნიშვნელობა აქვს საცავისათვის, იგი ამცირებს საცავში მიგრირებულ მონაცემთა მოცულობას, რაც ამავე დროს ეკონომიურად იყენებს კომპიუტერის და ქსელის რესურსებს.

საწყისი მონაცემები, როგორც კი თავს მოიყრის შუალედურ დონეში, დეგბა კონვერტაციის პროცედური შესრულების დრო,

რომელსაც შეუძლია გაასუფთავოს მონაცემები. მონაცემთა გასუფთავება სპეციალური პროგრამის გამოყენებით უზრუნველყოფს მონაცემთა მთლიანობის დაცვას, გასწორებას და მათ სიზუსტეს ფართო გამოყენებისათვის. ასევე შესაძლებელია გამოვიყენოთ პროგრამები, რომელსაც შეაქვს შესწორებები და აუმჯობესებს ჩანაწერის ფორმატს.

მონაცემთა „გასუფთავების“ ძირითად კომპონენტებია:

- მონაცემთა კვლევა, მონაცემთა განმსაზღვრელი ხარისხი, მოდელის ხარისხი და ველების რაოდენობა;
- მონაცემთა სინტაქსის შემოწმება, შინაარსი და ყოველი კომპონენტის დანიშნულების ადგილი ყოველი ველისათვის;
- მონაცემთა კორექცია, ამ დროს კონვერტაციის ჯგუფი უნდა თანამშრომლობდეს დამკვეთებთან და კორექცია მოხდეს ინფორმაციის წყაროს ფარგლები;
- ჩანაწერების შედგენა, არის შემთხვევა, როდესაც ორი ჩანაწერი წარმოადგენს ერთსა და იმავე ობიექტს. ასეთ შემთხვევაში გამოიყენება რთული პროგრამული ინსტრუმენტი, რომელიც ახდენს მონაცემთა იდენტიფიკაციას.

კონვერტაციის გეგმაში და კორექციის პროცედურაში გათვალისწინებული უნდა იყოს დამკვეთის შეხედულება, რომელიც აუცილებლად შესაბამისობაში უნდა მოდიოდეს საწყის მონაცემებთან. ეფექტური კონვერტაციის სპეციფიკისათვის, საჭიროა რაც შეიძლება მეტი ინფორმაცია გექნოდეს საწყისი მონაცემების სტრუქტურასა და თითოეული ველის მნიშვნელობაზე. პირველი ნაბიჯის გადადგმა კონვერტაციის პროცესის ნებართვაზე იწყება მას შემდეგ, რაც საწყის მონაცემებს განვითავებთ შუალედურ სქემაში. შუალედურ სქემაში ხდება მონაცემთა გაფილტვრა, ტრანსფორმაცია და ინტეგრაცია. იმ შემთხვევაში თუ თავს იჩენს არასაიმედო ჩანაწერი შეტყობინება ეგზავნება დამკვეთს და ერთობლივად ხდება შეთანხმება საწყის სისტემაში მონაცემის შესწორებისა ან კონვერტაციის პროცესის გამოყენების შესახებ. იმ შემთხვევაში თუ შევასწორებთ მონაცემებს კონვერტაციის პროცესის დახმარებით, აუცილებლად უნდა მიეთითოს შესწორება მეტა მონაცემთა კონვერტაციაში.

კონვერტაციის პროცესის განხორციელება სისტემაში არსებული საბაზო ინფორმაციული ტექნოლოგიების ბაზაზე, არცთუ რთული პროცესს წარმოადგენს და აქვს ფართო გამოყენება.

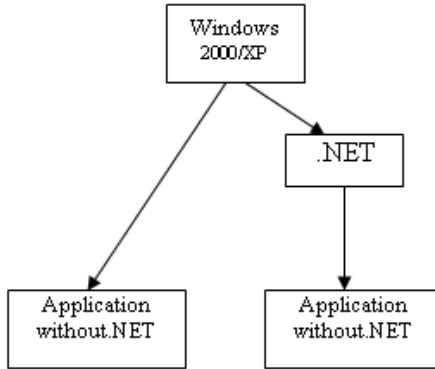
საბაზო ინფორმაციულ ტექნოლოგიას (სიბ) წარმოადგენს განაწილებული საინფორმაციო მართვის სისტემის ძირითადი პროგრამული პაკეტი. ესაა ოპერაციული სისტემა (მაგ., მაკროსოფტის Windows 2000/XP და დოტ-NET პლატფორმები) და მაღალი დონის პროგრამული ენები, რომელთაც აქვს ობიექტ-ორიენტირებული მეთოდები და ინსტრუმენტული საშუალებანი (მაგ., C#.NET, Visual Basic.NET, JavaScript.NET, Visual C++.NET, Borland C++ Builder და სხვ.).

საინფორმაციო ტექნოლოგიები (სბ-ი, შეიძლება განვიხილოთ ფართო სპექტრით. კონკრეტული ინტეგრირებული საინფორმაციო მართვის სისტემის დაპროექტებისას ობიექტ-ორიენტირებული ანალიზის ეტაპზე დაზუსტდება აუცილებელი პროგრამული პაკეტებისა და მათ შორის საკომუნიკაციო ინტერფეისების შედგენილობა. მაგ., დოტ-NET საბაზო პლატფორმის შემთხვევაში ინფორმაციული ტექნოლოგიები შეიძლება იყოს C#.NET, Visual Basic.NET, MsOffice, ASP.NET (Web-აპლიკაციისთვის), ADO.NET. მონაცემთა ბაზების მართვის სისტემების სახით გამოიყენება MsSQL Server-2000, Oracle, InterBase და ა.შ. ყოველ ინფორმაციულ ტექნოლოგიას აქვს საბაზო ტექნოლოგიასთან კომუნიკაციის საინტერფეისო ენა [20].

მონაცემთა საცავებისა და ბაზებიდან ინფორმაციის ამოსაღებად ფართოდაა მიღებული სტანდარტული სტრუქტურირებადი მოთხოვნების ენის ინსტრუმენტის გამოყენება (SQL). სხვადასხვა ინფორმაციული ტექნოლოგიების (მაგ., დაპროგრამების სხვადასხვა ენები, სხვადასხვა ბაზები) გამოყენებისას მონაცემების ან პროგრამული კოდების თავსებადობისათვის შეიძლება CORBA (Common Object Request Broker Architecture) ან DCOM (Distributed Component Object Model) ტექნოლოგიების გამოყენება [30].

დოტ-NET პლატფორმის შემთხვევაში მაკროსოფტმა ორიენტირებულად გადაწყვიტა და გაამარტივა ეს პროცესი. მან შექმნა შუალედური გარდაქმნის ენა (IL- Intermediate Language). პროგრამები, რომელთა საწყისი კოდები დაწერილია, მაგალითად C#, VB. ან .NET-ის IL-ენაზე, კომპილირდება მანქანურ კოდში. .NET-ის ყველა ენა ფლობს CTS (Common Type System) მონაცემთა შესათანხმებელ საერთო ტიპებს, რათა ენების სტანდარტიზაცია იქნეს მიღწეული [31,32]. ამგვარად, ობიექტური კოდები IL-ენის საშუალებით ისე მიიღება, რომ

მათში არაა დაფიქსირებული, თუ რომელ ენაზეა დაწერილი საწყისი კოდი. დოტ-NET პლატფორმა მდებარეობს Windows-ოპერაციულ სისტემასა და საავტომატიზაციო სისტემის აპლიკაციას შორის (იხ. ნახ.2.34).



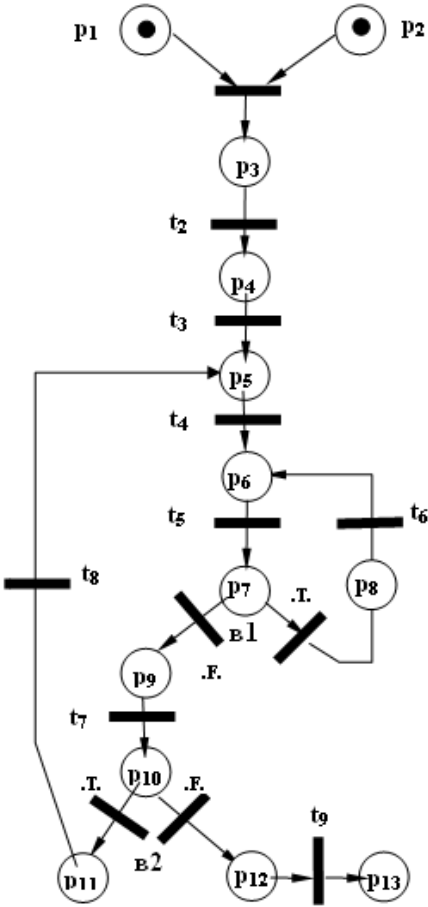
ნახ.2.34. პლატფორმა Windows-NET

ახლა განვიხილოთ ჩვენს მიერ დამუშავებული კონვერტირების ერთ-ერთი ინსტრუმენტი, რომლის გამოყენება ეფექტურია, მაგალითად საოფისე სისტემებისა და განაწილებული ბაზების მონაცემთა სტრუქტურების თავსებადობის განსახორციელებლად.

როგორც ანალიზმა გვიჩვენა, კორპორაციულ ორგანიზაციებში, ელექტრონული ბიზნესისა და კომერციის ობიექტებზე (მაგალითად, დიდი სავაჭრო ცენტრები) თითქმის 60-70% გამოიყენება MsExcel-პაკეტი აგებული ცხრილები (შეკვეთები, ფაქტურები და სხვა ანგარიშები). მონაცემთა საცავი რელაციური ტიპის ბაზებთან სამუშაოდაა ორიენტირებული, როგორებიცაა, მაგ., MsAccess, SQL Server, Oracle, InterBase და ა.შ. ამ მონაცემთა ბაზებს გააჩნიათ როგორც ერთმანეთთან, ასევე Ms Excel-თან მონაცემთა ცხრილების გაცვლის შესაძლებლობანი (Import/Export, MsQuery - ფუნქციების ან ინსტრუმენტების სახით და ODBC დრაივერით).

წინამდებარე პარაგრაფში ჩვენს მიერ შემუშავებულია ექსცელის ცხრილების კონვერტაციის ფუნქციის ალგორითმული სქემა მონაცემთა განაწილებული ბაზების მართვის სისტემის - InterBase-ის სტრუქტურაში. ამ მონაცემთა ბაზას ფართოდ იყენებს ბორლანდის ფირმა (Borland C++ Builder და Borland Delphi საბაზო

ტექნოლოგიების პროგრამული აპლიკაციებით) და OBDC დრაივერი აქვერ გამოიყენება. ამიტომაც ის აქტუალურად მიგვაჩნია. ამოცანა მდგომარეობს მონაცემთა საცავის განაწილებული რელაციური ბაზის ავტომატიზებულად შესავსებად მონაცემთა პირველადი წყაროების ელექტრონული ცხრილებიდან. 2.35 ნახაზზე ნაჩვენებია აღნიშნული ამოცანის გადაწყვეტის ალგორითმის შესაბამისი პეტრის ქსელის გრაფი, რომელშიც ასახულია შესასრულებელ პროცედურათა მიმდევრობა.



ნახ.2.35. პეტრის ქსელის გრაფი დილაკისათვის
ButtonClick: - Import

პოზიციებისა და გადასასვლელების სემანტიკა მოცემულია ქვემოთ:

პოზიციები:

- p₁ – შექმნილია F1Book - Excel ცხრილი;
- p₂ – შექმნილია InterBase ცხრილი;
- p₃ – გამოყოფილია Excel ცხრილის დასამუშავებელი არე;
- p₄ – გახსნილია InterBase ბაზის ალიასი;
- p₅ – გახსნილია InterBase ბაზის ალიასის ცხრილი;
- p₆ – InterBase –ში ჩამატებულია ცხრილის ცარიელი სტრიქონი;
- p₇ – InterBase-ს ცხრილი დამატებული ველით სტრიქონში;
- p₈ – პოზიცია მარკერით შემდეგი ველისათვის;
- p₉ – შევსებულია ცხრილის სტრიქონი;
- p₁₀ – InterBase-ის ცხრილში ჩაწერილი სტრიქონი;
- p₁₁ – პოზიცია მარკერით შემდეგი სტრიქონისთვის;
- p₁₂ – InterBase-ს შევსებული ცხრილი;
- p₁₃ – მონაცემთა ბაზის დახურული ფაილი.

გადასასვლელები:

- t₁ – Excel ცხრილის უბნის გამოყოფა;
- t₂ – InterBase ბაზაში ალიასის გახსნა;
- t₃ – InterBase ბაზაში ალიასით ცხრილის გახსნა;
- t₄ – InterBase ბაზის ცხრილში ცარიელი სტრიქონის ჩამატება;
- t₅ – Excel-ის გამოყოფილი უბნის კოპირება ბაზის ცხრილში;
- t₆ – Excel ცხრილის ახალი ველის არჩევა;
- t₇ – ოპერატიული მეხსიერებიდან ცხრილის სტრიქონის გადაგზავნა დისკზე;
- t₈ – Excel ცხრილში ახალი სტრიქონის არჩევა;
- t₉ – მონაცემთა ბაზაში ცხრილების დახურვა;
- B1 – არის კიდევ ველი ? B2 – არის კიდევ სტრიქონი ?

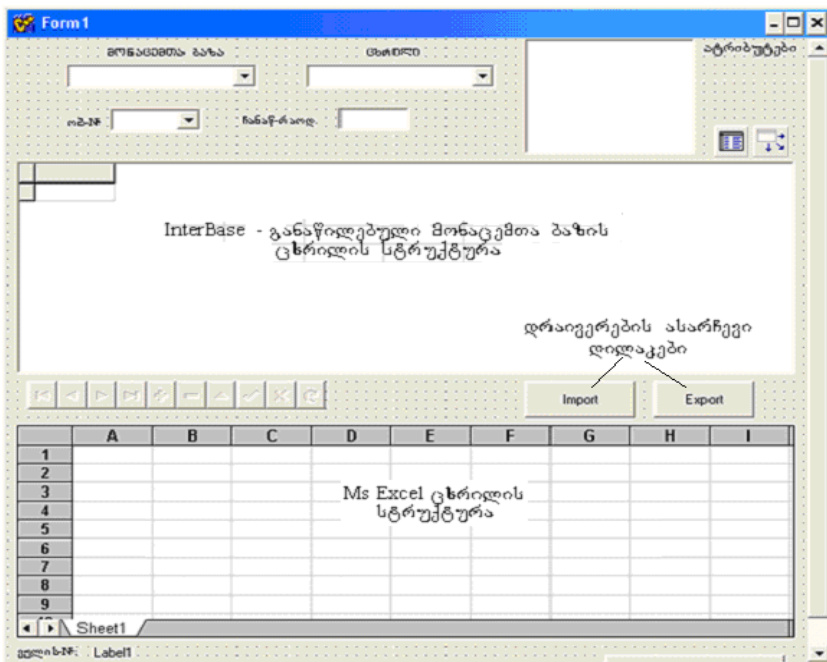
2.36 და 2.37 ნახაზებზე ილუსტრირებულია რეალიზებული C++პროგრამის ფრაგმენტი და შესაბამისი საინტერფეისო ფანჯარა მონაცემთა ბაზის ადმინისტრატორისათვის. სისტემა რეალიზებულია Borland C++Builder გარემოში.

```

//----- Unit1.cpp ----- პროგრამის ტექსის ფრაგმენტი -----
int i,j, Ob_Num, Dan_Num, Rec_Num;
void __fastcall TForm1::FormCreate(TObject *Sender)
{ Session->GetAliasNames(cbAlias->Items); }
void __fastcall TForm1::cbAliasChange(TObject *Sender)
{ Session->GetTableNames(cbAlias->Text,"",true,false,cbTable->Items);
  cbTable->ItemIndex = 0;
  if (cbTable->Text == "")
  { ShowMessage("Table not selected"); return; }
  Table1->Active = false;
  Table1->DatabaseName = cbAlias->Text;
  Table1->TableName = cbTable->Text;
  Table1->Active = true;
  if(Table1->Active) Table1->GetFieldNames(cbField->Items);
}
//----- Button of Import -----
void __fastcall TForm1::Button1Click(TObject *Sender)
{ AnsiString Info;
  if(Table1->Active)
  { int t=Table1->FieldCount;
    for(i=3; i<Rec_Num+3; i++) // str_num
    {
      Table1->Append();
      for(j = 1; j < t; j++) // col_num
      { Label3->Caption=t;
        Table1->FieldValues["OBJNUM"] = Ob_Num;
        Info = Table1->Fields->Fields[j]->FieldName;
        Label1->Caption=j;
        Excel->SetSelection(i,j,i,j); // choice of Range
        if(Excel->Text != 0)
        { Label2->Caption=Excel->Text;
          Table1->FieldValues[Info] = Excel->Text; }
        } // for j
      Table1->Post();
    } // for i
  } // for if()
}

```

ნახ.2.36. საინტერფეისო IMPORT-დრაივერის C++ პროგრამული კოდის ფრაგმენტი



ნახ.2.37. სანტერფეისო ფანჯარა

III თაზო

შედგენის პრაქტიკული რეალიზაცია:

3.1. ვირტუალური მარკეტინგული პროცესების მოდელირება და დისტრიბუტორ-რეალიზატორის ავტომატიზებული სამუშაო ადგილი

განხილულია ელექტრონული ბიზნესის და ელექტრონული კომერციის სისტემების დამკვიდრებისა და ვირტუალური მარკეტინგის პერსპექტიული განვითარების კონცეფცია და ამოცანები. საზღვარგარეთული პრაქტიკული გამოცდილების გამოყენების შესაძლებლობა ჩვენი ქვეყნის პირობებში, განსაკუთრებით უმუშევრობის შემცირებისა და ვირტუალური სამუშაო ადგილების შექმნის თვალსაზრისით. შემოთავაზებულია ვირტუალური მარკეტინგის სისტემაში პროდუქციის ვირტუალური დისტრიბუტორ-რეალიზატორის ავტომატიზებული სამუშაო ადგილის აგების ამოცანა, მონაცემთა აქტუალური ბაზით, საქმიანობის ეფექტურობის შეფასების პროცესის მოდელირებისა და ანალიზის ალგორითმით.

თანამედროვე სოციალურ-ეკონომიკური მდგომარეობა ქვეყნის მასშტაბით წინა პლანზე წამოწევს სოციალურად დაუცველი მოსახლეობის პრობლემებს, რომელთა შორისაც პირველ ადგილზეა „მასობრივი უმუშევრობა“. თუ ჩვენი ქვეყნის ეკონომიკური განვითარების დინამიკას დავუკვირდებით, მსოფლიოს სხვა ქვეყნებთან მიმართებაში, აშკარაა, რომ წინა და ახლანდელი თაობის პოლიტიკოსების „შეცდომები“ პირდაპირ კავშირშია დასმულ პრობლემასთან. პროდუქციის ადგილობრივ წარმოებას დიდ კონკურენციას უწევს მეზობელი და არამეზობელი ქვეყნებიდან იაფფასიანი, უსარისხო (ხშირად მავნებელი) პროდუქციის შემოტანა. ვაჭრობა (აღრე „სპეკულაციად“ წოდებული) გახდა პრიორიტეტული ფაქტორი პიროვნების, ოჯახის ან სხვა სუბიექტის გადარჩენისათვის (выживание, survival). ინტერნეტისა და მობილური ტელეფონების ეპოქაში, როცა ასე მწვავედ დაისვა გლობალიზაციის პრობლემა (არა მარტო განათლებისა და სოციალურ-ეკონომიკურ სფეროში), საქართველო და მისი მოსახლეობა უნდა შეეცადოს ფეხი აუწყოს ევროპისა და ამერიკის რითმს, რაც ალბათ რთული წარმოსადგენია. ერთ-ერთი აქტუალური და მნიშვნელოვანი ფაქტორი ამ პროცესში ელექტრონული ბიზნესისა და კომერციის

განვითარება და პიროვნებისათვის მასში ადგილის დამკვიდრებაა. აქ კი ნამდვილად გამოდგება ქართული ნიჭი, ტემპერამენტი, ამბიციაცია და კომფორტულად ცხოვრების იმედი თუ არა, პირველადი, აუცილებელი საგნებისათვის შესაძენი ფულის შოვნის (პატიოსანი გზით) შესაძლებლობა.

ელექტრონული ბიზნესი ისეთი საქმიანობაა, რომელიც იყენებს ინტერნეტ/ინტრანეტ საინფორმაციო ქსელებს კომპანიის შიგნით და გარე კავშირებისათვის, ახორციელებს სამეურნეო, პარტნიორულ და საშუამავლო ურთიერთობებს მოგების მიღების მიზნით. იგი მარკეტინგული სისტემაა ინტერნეტული ტექნოლოგიებით. ელექტრონული ბიზნესი ფირმებისათვის უზრუნველყოფს Web-საიტების შექმნას, კომპანიის ბიზნეს-პროცესების ინტეგრაციას, მათ კავშირს დამკვეთებთან და მომხმარებლებთან, ბაზრის მასშტაბების გაფართოებას [20].

ელექტრონული კომერცია ელექტრონული ბიზნესის მთავარი შემადგენელი ნაწილია. იგი ტექნიკური და ორგანიზაციული ფორმების ერთობლიობაა, რომლის საშუალებითაც შესაძლებელია მატერიალური და ფინანსური აქტივების გადაცემა ბიზნესის ერთი სუბიექტიდან მეორეზე. ბიზნეს-ურთიერთობები იყოფა ოთხ კატეგორიად: Business-to-Business (B2B) - ელექტრონული კომერციაა საწარმოებს შორის, რომელიც მოიცავს ყველა დონის ინფორმაციულ კავშირებს კომპანიათა შორის. ასეთი სისტემები ძირითადად გამოიყენება წარმოების მომარაგებასა და მზა პროდუქციის გასაღებაში; Business-to-Consumer (B2C)- ელექტრონული კომერციის ვარიანტია, სადაც მყიდველები წარმოადგენენ კერძო პირებს. მაგალითად, ინტერნეტ-მაღაზია, ფასიანი საინფორმაციო სამსახური და ა.შ.; Consumer-to-Consumer (C2C)- ელექტრონული კომერციაა სხვადასხვა კერძო პირებს შორის, მაგალითად ელექტრონული აუქციონი; Business-to-Government (B2G)- ელექტრონული კომერციაა საწარმოსა და სახელმწიფო ორგანოებს შორის. კომპანიის საბაზრო სტრატეგიის კუთხით ელექტრონული ბიზნესი და კომერცია გვთავაზობს ისეთი ფორმების არსებობას ინტერნეტში როგორცაა ელექტრონული სავიზიტო ბარათი, ელექტრონული კატალოგი, ელექტრონული ვაჭრობა (e-trading), ელექტრონული მაღაზიები, (e-shop), ელექტრონული ბუხჰალტერია და ა.შ.

ელექტრონული კომერცია არის სწრაფი და ბიზნესის ეკონომიური სახე, რომელიც არ ცნობს საზღვრებს. დღეს მსოფლიოში საკმაოდ პოპულარული თემაა ელექტრონული კომერცია – კომპიუტერის მონიტორიდან უშუალოდ მსოფლიო ბაზარზე ყიდვა–გაყიდვის წარმოება.

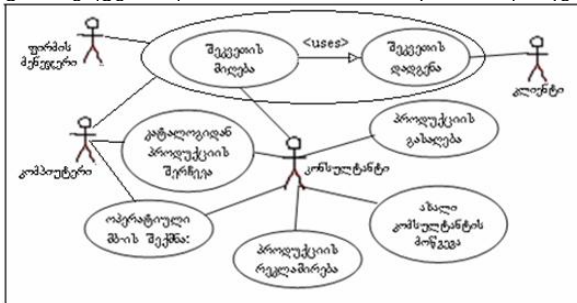
მთელი რიგი კომპანიების საშუალებით წარმატებით იქნა ორგანიზებული თანამედროვე გასაღების ბაზარი. სავაჭრო ორგანიზაციების ინტერნეტში განთავსების მიზეზი რამდენიმეა: ინტერნეტ მაღაზია ფუნქციონირებს 24 საათი. მისაწვდომია მსოფლიოს ყველა ქვეყნის მოსახლეობისთვის და უზრუნველყოფს მომხმარებლის ინდივიდუალურ მომსახურებასაც.

მაგალითისათვის, ჩვენ ყურადღებას გავამახვილებთ პარფიუმერიის შვედურ ფირმაზე Oriflame, რომელიც კარგად დამკვიდრდა მსოფლიოსა და ჩვენს ქვეყანაში და შემოგვთავაზა ვირტუალური კომერციისა და ვირტუალური თანამშრომლობის მრავალი შესაძლებლობა [21].

ფირმის ქართულ და რუსულ ვებ-გვერდებზე ასახულია არა მხოლოდ პროდუქციის კატალოგები, არამედ გადმოცემულია ის წესები, რომლებიც ნებისმიერ პიროვნებას ფირმასთან ვირტუალური თანამშრომლობის პირობებს სთავაზობს.

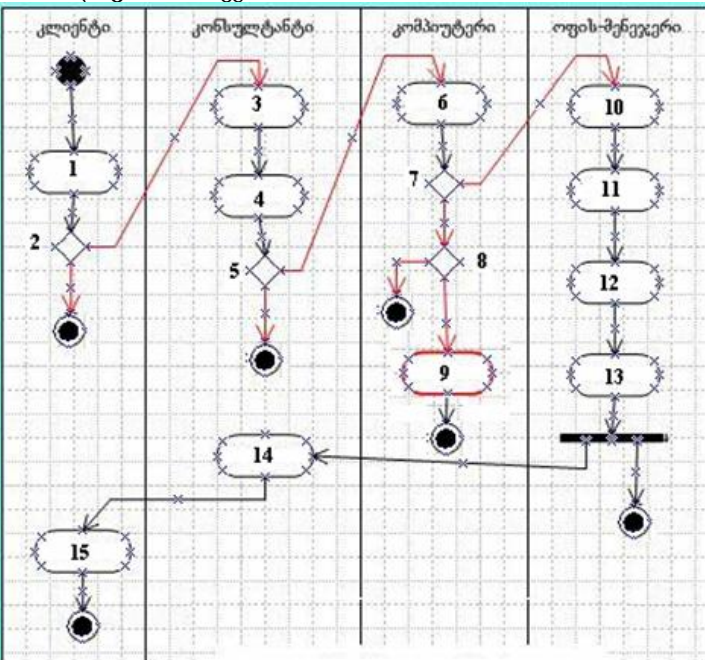
სტატიაში განიხილება ვირტუალური კონსულტანტის „საქმის წარმოების“ (როგორც ინდივიდუალური ფირმის) პროცესების კომპიუტერიზაცია, რომლის მიზანი მისი მოღვაწეობის ეფექტურობის ამაღლება, ანუ მოგების მაქსიმიზაციაა. პროგრამული სისტემის დასაპროექტებლად ვიყენებთ UML–ტექნოლოგიას.

3.1 ნახაზზე მოცემულია UseCase დიაგრამის ფრაგმენტი როლისთვის – კონსულტანტი, რომელზეც გაყიდვის ფუნქცია ორი ქვე-პროცედურისგან შედგება და მასში ოთხი როლი მონაწილეობს.



ნახ.3.1. UseCase დიაგრამის ფრაგმენტი

3.2 ნახაზზე ნაჩვენებია ამ როული პროცესის შესაბამისი აქტიურობის დიაგრამის სქემა.



ნახ.3.2. Activity - დიაგრამის ფრაგმენტი

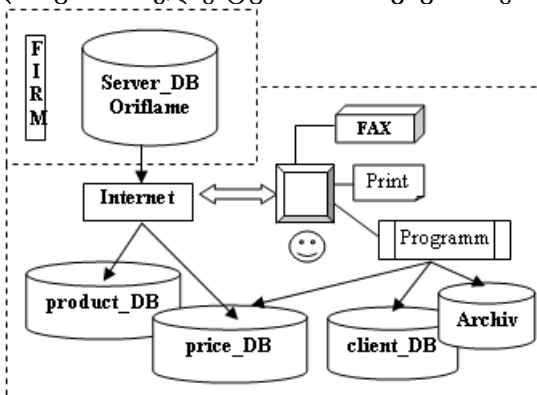
აქ ოთხი მართვის სფეროა გამოყოფილი: კლიენტის, კონსულტანტის, კომპიუტერის და ოფის-მენეჯერის. თითოეულში სპეციალური ოპერაციები სრულდება, ისინი განიხილება ზემოდან ქვემოთ და მარჯვნივ. მაგალითად,

1. კლიენტი ირჩევს პროდუქციას.
2. თუ არ მოეწონა არაფერი, მაშინ დასასრული. თუ აირჩია, მაშინ მოქმედება გადაეცემა კონსულტანტს, რომელიც
3. აფორმირებს შეკვეთას (რამდენიმე კლიენტი-სათვის).
4. კონსულტანტი კომპიუტერული ქსელის მანქანაში შეიტანს თავის User-name და Password-ს.
5. თუ კონსულტანტი უცხობა სისტემისთვის, მაშინ ის ამთავრებს მუშაობას, თუ ნაცნობია, მაშინ მას შეაქვს შეკვეთილი პროდუქციის სია.
6. კომპიუტერული პროგრამა ეძებს შეკვეთილი პროდუქციის არსებობას.
7. თუ არაა რომელიმე მათგანი, მაშინ
8. აზუსტებს, შემოიტანენ ამ პროდუქციას თუ არა (დასასრული).

9. თუ კი, მაშინ იქმნება სარეზერვო შეკვეთა.
10. ოფის-მენეჯერი იღებს შეკვეთილ პროდუქციის სიას და ანგარიშობს თანხებს,
11. ხსნის შეკვეთის გატარებას,
12. ღებულობს თანხას კონსულტანტიდან (უშუალოდ ან გადარიცხვით),
13. გადასცემს პროდუქციას კონსულტანტს.
14. კონსულტანტი გადასცემს პროდუქციას კლიენტს.
15. კლიენტი ამოწმებს შედეგს (და დასასრული). შეიძლება პროცესის გაგრძელება, თუ მაგალითად, კლიენტი არაა კმაყოფილი.

3.3 ნახაზზე წარმოდგენილია ვირტუალური კონსულტანტის ავტომატიზებული სამუშაო ადგილის (ვპასა) ზოგადი სტრუქტურა.

ვირტუალური კონსულტანტის ფუნქციები ორგანიზაციულ-ტექნიკური ხასიათისაა. ერთის მხრივ, იგი ყოველთვის უნდა ფლობდეს მთავარი ფირმის პროდუქციის აქტუალურ კატალოგს და ფასებს, მეორეს მხრივ, მან კარგად უნდა იცოდეს თავისი კლიენტურის მოთხოვნილება. ამავე დროს ის რეკლამირებას უკეთებს ფირმის პროდუქციას და მუშაობს კლიენტურის სიის გაფართოებაზე.



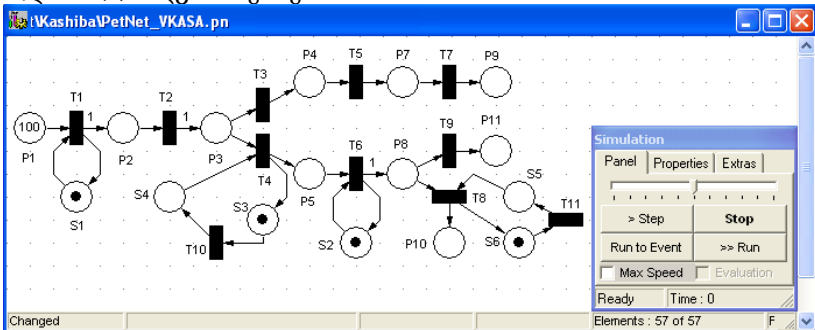
ნახ.3.3. ვპასა-ს სტრუქტურა

ტექნიკური თვალსაზრისით, ვპასა ის ინსტრუმენტია, რომელიც მას ოპერატიული მოქმედების საშუალებას აძლევს. ფირმის სერვერიდან (კონსულტანტს აქვს სტატუსი და პაროლი) იგი გადმოიტანს ახალ ინფორმაციას თავის კატალოგში (product_DB, price_DB). client_DB ბაზაში ინახება როგორც ინფორმაცია კლიენტების შესახებ, ასევე

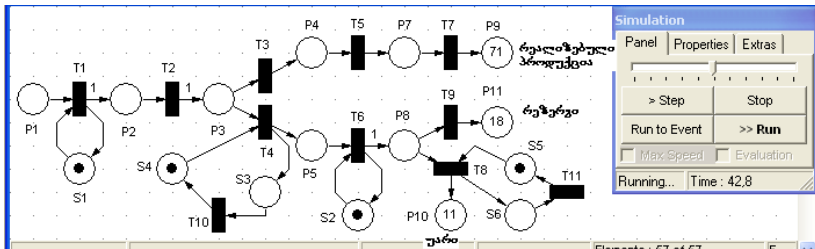
შეკვეთების ფაილები. კონსულტანტის Programm პაკეტში ინახება მისი ფუნქციური ამოცანების ბიბლიოთეკა, მაგალითად „კლიენტთა შეკვეთების ფაილების ანალიზით ერთიანი შეკვეთის ფაქტურის მომზადება სათაო ფირმისათვის“.

ამ დოკუმენტში ავტომატიზებულად იქნება დაანგარიშებული ფასები, მოგება, რენტაბელობა და სხვა ეკონომიკური მაჩვენებლები. ფირმიდან შეკვეთების მიღების, პროდუქციის განაწილების (რეალიზაციის), თანხების აღრიცხვის და სხვა ოპერაციების შემდეგ, სისტემა გააანალიზებს აღნიშნული შეკვეთების სეანსის შედეგებს.

Archiv-ში შეინახება დახურული შეკვეთები. შეუსრულებელი ან რეკლამირებული (დაწუნებული) შეკვეთების შემთხვევაში კონსულტანტი ოპერატიულად წყვეტს პრობლემას ისე, რომ მინიმალური ზარალით, არ დაკარგოს არც კლიენტი და არც ფირმის ნდობა. 3.4 ნახაზზე აგებულია ამ პროცესის შესაბამისი პეტრის ქსელის მოდელი, საწყისი (ა) და საბოლოო (ბ) მდგომარეობებით.



ნახ.3.4-ა. იმიტაციის საწყისი მდგომარეობა: 100 მოთხოვნით



ნახ.3.4-ბ. იმიტაციის საბოლოო მდგომარეობა: 71 / 18 / 11

მის საფუძველზე შესაძლებელია ჩავატაროთ ანალიზი ქსელის ოპტიმალური მახასიათებლების დასადგენად. ოპტიმალურობის ქვეშ იგულისხმება ვირტუალური კონსულტანტის მიერ კლიენტთა მოთხოვნების მაქსიმალურად დაკმაყოფილება შესაძლო მინიმალურ დროში, ამასთანავე კლიენტის, ფირმისა და თავისი ბიზნეს-მოთხოვნილებების (ხარჯი, შემოსავალი, მოგება, რენტაბელობა და ა.შ.) შეზღუდვების გათვალისწინებით.

პეტრის ქსელის შესასვლელზე მარკეტებით მოდელირდება კლიენტთა მოთხოვნების არსებობა. მათი დინამიკა სტოქასტიკურია. ასეთ სისტემაში რთული პროცესებისთვის მიღებულია პუასონის განაწილების კანონის გამოყენება [22].

შემუშავებული ტექნოლოგიის საფუძველზე შესაძლებელია ვირტუალური მარკეტინგის სისტემის ვირტუალური თანამშრომლისთვის საინფორმაციო ბაზისა და გადაწყვეტილების მიღების ხელშემწყობი კომპიუტერული სისტემის აგება. პეტრის ქსელით მოდელირებისა და ანალიზის საფუძველზე წარმოებს კომერციული პროცესების იმიტაცია, რაც საშუალებას იძლევა ექსპერიმენტულად წინასწარ შევაფასოთ პროცესის ეფექტურობა.

3.2. საწარმოო გაერთიანების მარკეტინგული უზრუნველყოფის დეპარტამენტის განაწილებული მონაცემთა ბაზის მთლიანობის დაცვა

განაწილებულ სისტემაში ვგულისხმობთ ამა თუ იმ გამოყენებითი სფეროს ინფორმაციის დამუშავებას ან მართვის ავტომატიზებულ სისტემას, რომელიც ორიენტირებულია პერსონალური კომპიუტერების ლოკალური ქსელისათვის. ასეთი სისტემის ერთ-ერთ ძირითად ნაწილს ფიზიკურად განაწილებული (რამდენიმე კომპიუტერულ კვანძში), მაგრამ ლოგიკურად ერთი მთლიანი (გლობალური) ინფორმაციული ბაზა შეადგენს. საყურადღებო და პრობლემურია ასეთი ბაზების დაპროექტების, აგებისა და კორექტირების ეფექტური პროგრამების დამუშავება, რომლებიც უზრუნველყოფს დროის სასურველ ინტერვალში დასმული ამოცანების გადაწყვეტას, მისი ერთ-ერთი მთავარი კრიტერიუმის – მონაცემთა ბაზის მთლიანობის ასპექტის დაცვის გათვალისწინებით.

განიხილება განაწილებული ბაზის გლობალური ლოგიკური სტრუქტურის დაპროექტების და მის საფუძველზე სისტემური კატალოგის აგების და კორექტირების პროცესების ავტომატიზაციის ამოცანები, მათი გადაჭრა, მათი გადაჭრის გზები და რეალიზაციის ინსტრუმენტული საშუალებანი [33].

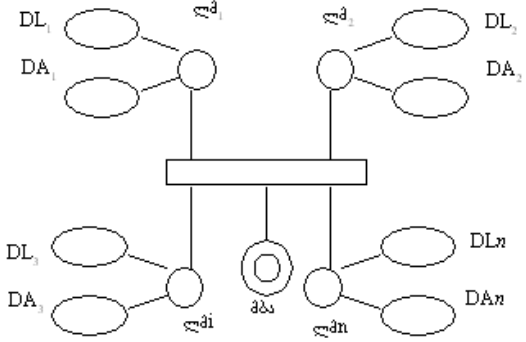
შედეგები ადაპტირებულია საწარმოო გაერთიანებათა მარკეტინგული უზრუნველყოფის ქვესისტემისათვის.

წინასწარ ვუშვებთ, რომ განხილული იქნება რელაციური ტიპის მონაცემთა მოდელი, ხოლო ბაზები – ე.წ. „გაფართოებული“ რელაციური. გაფართოებულში იგულისხმება სტანდარტული ლოგიკური სტრუქტურების „გამდიდრება“ სემანტიკური ინფორმაციით, რაც იძლევა მონაცემთა ლოგიკურ-კონცეპტუალურ მოდელზე ექსპერიმენტის ჩატარების საშუალებას.

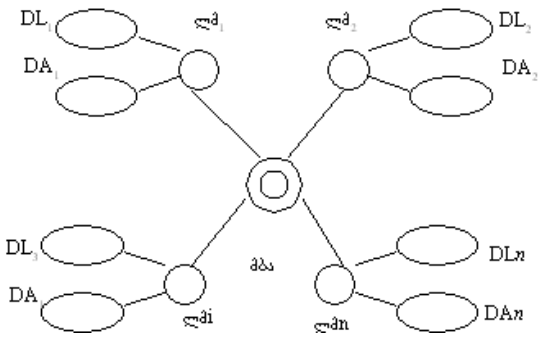
განაწილებული ბაზის მთლიანობის ასპექტის დაცვა მოითხოვს სპეციალური მაკონტროლებელი პროცედურების აგებას, რომელთა კლასიფიკაციის საფუძველზე ასხვავებენ ფიზიკურ ან ტექნიკურ, სტრუქტურულ ან ლოგიკურ, სემანტიკურ და პრაგმატულ მთლიანობათა დონეებს. ბაზის ფიზიკური მთლიანობა განიხილება როგორც ინფორმაციული მატარებლის საიმედოობა. მისი კონტროლის ამოცანა, ინფორმაციის ასახვის ფუნქციის დაცვა მანქანის ფიზიკურ მენსიერებაში.

სტრუქტურულ მთლიანობაში (რელაციური მოდელების თვალსაზრისით) განიხილება მონაცემთა დომენების, ტიპების, მანასიათებლების, გასაღებური ატრიბუტების (პირველადი და მეორეული), მაჩვენებლების (ინდექსური), ცხრილების მთლიანობის კონტროლის საკითხები. მონაცემთა სემანტიკური მთლიანობა დაკავშირებულია საგნობრივ სფეროს კონცეპტუალურ სქემასთან, რომლის შესაბამისი მთლიანობის პრედიკატის შეზღუდვა განაპირობებს კონტროლს სინტაქსურად შესაძლო ყველა ვარიანტიდან „გააზრებულის“ შესარჩევად ე.ი. უნდა გამოირიცხოს „შეუძლებელი“ და „უაზრო“ ინფორმაციული ფრაგმენტები. პრაგმატული მთლიანობა განიხილება დასაპროექტებელი სისტემის მიზნობრივ კონტექსტში. ამიტომაც მისი კონტროლი ხორციელდება ე.წ. დაუშვებელი ინფორმაციული ფრაგმენტების აღმოსაფხვრელად. სემანტიკურისგან განსხვავებით, დაუშვებელი (ანუ აკრძალული) ინფორმაციული ფრაგმენტი შეიძლება არ იყოს „შეუძლებელი“ ან „უაზრო“.

მთლიანობის ზემოთ განხილული დონეებიდან შევჩერდებით დეტალურად მეორეზე და ნაწილობრივ მესამეზე (ე.ი. სტრუქტურულ და სემანტიკურ დონეებზე). შევაჯამოთ მოცემულ პარაგრაფში განხილული ძირითადი საკითხები და განვიხილოთ განაწილებული ბაზების სისტემური კატალოგების აგების ამოცანა და მისი გადაწყვეტის ალგორითმები. 3.5 ნახაზზე ილუსტრირებულია ჰკ-ის ლოკალური ქსელის ტოპოლოგია (ა-საერთო სალტით, ბ-ვარსკლავისებრი), რომლისათვისაც დამახასიათებელია სერვერ-მანქანის არსებობა (მონაცემთა ბაზის ადმინისტრატორის სამუშაო ადგილი) და რამდენიმე ლოკალური მომხმარებლის სამუშაო ადგილი. საერთო გამოყენების რესურსებიდან ამჯერად ყურადღებას გავამახვილებთ მონაცემთა ბაზაზე.



ა) სალტი

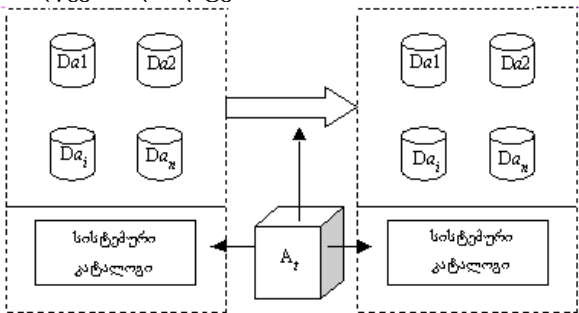


ნახ.3.5. განაწილებული ბაზების ტოპოლოგიური სქემები

ყოველ ლოკალურ მომხმარებელს (ლმი) გააჩნია საკუთარი მოხმარების ლოკალური ბაზა (DLi) და საერთო მოხმარების (ქსელის მასშტაბით) ბაზის ნაწილი (DAi), სწორედ DAi შეადგენს ერთიან განაწილებულ ბაზას, რომელსაც შეესაბამება გლობალური ლოგიკური სტრუქტურა. სისტემური კატალოგი (მაგალითად, სერვერ-მანქანაში) ფლობს ინფორმაციას განაწილებული ბაზების დამოკიდებულებების ატრიბუტების, კავშირების, დომენების და ა.შ. შესახებ, რათა სისტემის მართვის ოქმებით შესაძლებელი იყოს ქსელში მიმდინარე ტანზაქციების სინქრონიზებული შესრულება.

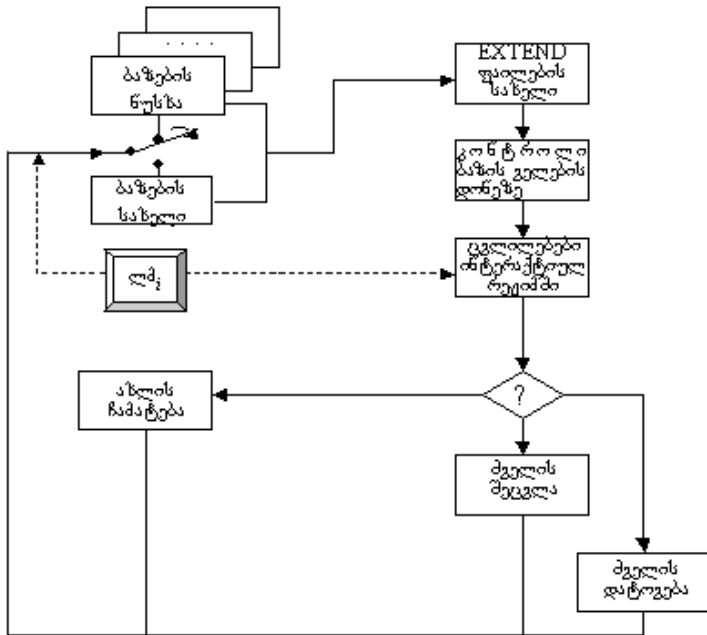
მონაცემთა განაწილებული (და ცენტრალიზებული) ბაზები ხშირად „არამთლიან“ მდგომარეობაშია, რაც განპირობებულია იმით, რომ მათი შექმნის და ექსპლუატაციის პროცესში მონაწილეობს სხვადასხვა კატეგორიის მომხმარებელი. მართალია, მათ მინიჭებული აქვთ ბაზასთან ურთიერთობის სტატუსი (მხოლოდ წაკითხვა, წაკითხვა და ჩასწორება და ა.შ.), მაგრამ განაწილებული სისტემის დიდი მოცულობის და სირთულის გამო, ადმინისტრატორისთვის შეუძლებელი ხდება მთლიანი ინფორმაციის ფლობა და მისი კონტროლი, ამიტომ აუცილებელია მას ჰქონდეს პროგრამული პაკეტი, რომლის დანიშნულებაც იქნება როგორც არამთლიანი მდგომარეობის განაწილებული ბაზის მთლიან მდგომარეობაში გადაყვანა (ბაზის აგების ეტაპი), ასევე მისი მთლიანობის უზრუნველყოფა (ექსპლუატაციის ეტაპი).

3.6 ნახაზზე ნაჩვენებია არამთლიან მდგომარეობაში მყოფი განაწილებული ბაზის მთლიან მდგომარეობაში გადაყვანის მოდელი. გადაყვანის ალგორითმი (At) უზრუნველყოფს განაწილებული ბაზის მთლიანობის მიღწევას და დაცვას.



ნახ.3.6. მონაცემთა განაწილებული ბაზების მთლიანობის მდგომარეობაში გადაყვანის მოდელი

იგი რეალურად ამუშავებს განაწილებული ბაზის ქვებაზებს და მათ შესაბამის სისტემურ კატალოგებს. საწყის ინფორმაციას მისთვის შეადგენს მომხმარებლის მიერ მოთხოვნაში ფორმირებული ქვებაზის სახელი (ან ქვებაზების დასახელებების სია). 3.7 ნახაზზე მოცემულია განხილული ალგორითმის გამსხვილებული ბლოკ-სქემა.



ნახ.3.7. მბ-ის მთლიან მდგომარეობაში გადაყვანის ალგორითმი

3.8 ნახაზზე ილუსტრირებულია სისტემური კატალოგის ერთ-ერთი ძირითადი კომპონენტის – ბაზის ატრიბუტის (ველის) სტრუქტურისა და ჩანაწერების ფრაგმენტი საწარმოო გაერთიანების მარკეტინგის განყოფილების საგნობრივი სფეროს მაგალითზე. კატალოგი რეალიზებულია ბაზის ფაილის (KATALOG.DBF) სახით შესაბამისი EXTEND ფაილის (EXT) გამოყენებით.

თანამედროვე საბაზრო ეკონომიკის განვითარების პირობებში, საწარმოებისა და ორგანიზაციების ეფექტური დაგეგმვისა და მართვისათვის აუცილებელი ხდება მარკეტინგის ფუნქციური როლების შექმნა [34].


```

USE KATALOG INDEX KAT-ATR
LIST STRU
DATABASE STRUCTURE C:\FOXPRO\KATALOGDBF
NUMBER OF RECORDS:          31
FIELD NAME                   TYPE          DEC
1. FIELD-NAME                C              10
2. FIELD-TYPE                C              1
3. FIELD-LEN                 N              10
4. FIELD-DEC                 N              3
5. A-NAME                    C              15
** SUM                        40
list off for a name>**

```

FIELD_NAME	FIELD-TYPE	FIELD-LEN	FIELD-DEC	A-NAME
AP	C	8	0	აპარატურა
APQ	N	6	1	აპ-მწარმოებელ
FARTI	N	4	0	მალაზიის-გამგე
GAMGE	C	8	0	მალ-ტელეფონი
IMP	N	1	0	იმპორტ-საქონელი
JGUFU	N	1	0	მალ-ჯგუფი
KATEG	N	1	0	მალ-კატეგორია
MISAM	C	20	0	მალ-მი სამართი
MNCM	N	3	0	მალ-№
MPASI	N	7	2	მასალის-ფასი
MSAX	C	18	0	მას-დასახელება
MSH	C	4	0	მასალის-შიფრი
MZOMERT	C	8	0	მას-ზომ-ერთი
N	N	3	0	რიგითი-№
PSAX	C	18	0	პროდ-დასახელება

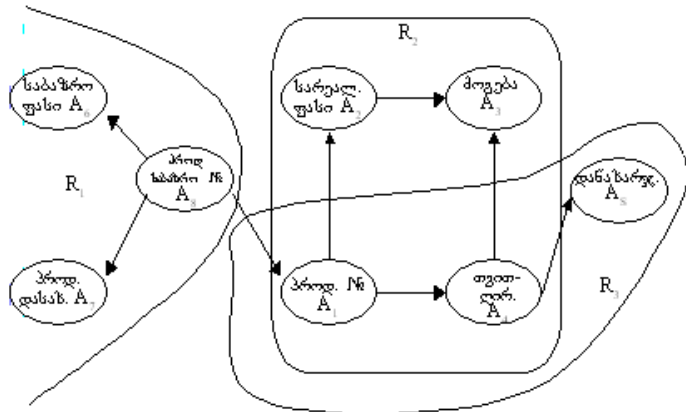
ნახ.3.8. კატალოგის ფრაგმენტის სტრუქტურა (ა) და ჩანაწერები (ბ)

საწარმოო გაერთიანებაში მარკეტინგი განიხილება როგორც პროდუქციის დაგეგმვის, წარმოების და გასაღების (ვაჭრობის) ერთიანი სისტემა. ამიტომაც საწარმოო გაერთიანების მართვის სისტემა შეიძლება

მიეკუთვნოთ განაწილებულ სისტემათა კლასს, რომლის ჰკ-ის ლოკალურ ქსელში მარკეტინგის სპეციალისტის ავტომატიზებული სამუშაო ადგილი იკავებს განსაკუთრებულ ადგილს [33].

მარკეტინგის ფუნქციური ამოცანების გადაჭრა მოითხოვს ისეთი საინფორმაციო ბაზების დამუშავებას, როგორცაა საწარმოთა პროდუქციის ასორტიმენტი, საბაზრო ფასები, პროდუქციის თვითღირებულება და სარეალიზაციო (საბითუმო, საცალო, სახელშეკრულებო) ფასები, სიმძლავრეები (პროდუქციის მოცულობისა და კონიუნქტურის ოპერატიული მართვისათვის), ტექნიკისა და ტექნოლოგიური პროცესების განვითარების (ახალი პროდუქციის ათვისების თვალსაზრისით), სასაწყობო მეურნეობის (ნედლეულისა და მზა პროდუქციის) მდგომარეობა, საბაზრო მოთხოვნების ანალიზის შედეგები და ა.შ. როგორც ჩანს, მონაცემთა ბაზები დანიშნულებისამებრ მიეკუთვნება საწარმოს სხვადასხვა ფუნქციურ რგოლებს (ქსელის ლმ i -ები), ბაზები იქმნება და ინახება მათი კორექტირების გათვალისწინებით.

მარკეტინგის ლმ კი მხოლოდ გამოიყენებს ამ ბაზებს (მას, როგორც წესი, არ გააჩნია მათი კორექტირების სტატუსი). 3.9 ნახაზზე ნაჩვენებია მარკეტინგის ლოკალური მომსახურებლის ერთი ფუნქციური ამოცანის (პროდუქციის რეალიზაციის ფასის ოპერატიული განსაზღვრა საბაზრო ფასებისა და მაქსიმალური მოგების გათვალისწინებით) მონაცემთა ბაზის ლოგიკური სტრუქტურის ფრაგმენტი.



ნახ.3.9. მოთხოვნის შესრულების სქემა კერძო მაგალითისათვის

1. $R_{12} = R_1[A_8 = A_1]R_2(A_6, A_1, A_2, A_4, A_3)$
2. $R_{12} \cdot A_6 - R_2 \cdot A_2$
3. $W = R_1 - R_2 \cdot A_4$
4. $W_1 - R_2 \cdot A_3 \geq Q_H$ (Q_H ნორმატივი)

3.3. ინფორმაციის უსაფრთხოებისა და საიდუმლოების დაცვის უზრუნველყოფის საკითხები

კლიენტ-სერვერული არქიტექტურის მქონე განაწილებული მონაცემთა ბაზები ძირითადად ფუნქციონირებს გლობალური ან ლოკალური ქსელის მასშტაბებით. ეჭვს არ იწვევს, რომ ნებისმიერი კორპორაციული მართვის სისტემას უნდა გააჩნდეს ინფორმაციის დაცვის საშუალებანი. აქ იგულისხმება როგორც არაკორექტული მონაცემების დამახსოვრების თავიდან აცილების შესაძლებლობა, ასევე მონაცემთა ბაზასთან მომხმარებელთა უნებართვო (არასანქცირებული) მიმართვების გამორიცხვა.

ინფორმაციის არაკორექტულობა შეიძლება გამოწვეული იყოს შემთხვევით, მონაცემების ან პროგრამის შეცდომით ჩაწერისას მანქანაში ან წინასწარი განზრახვით.

ამგვარად, ინფორმაციის დაცვა ორი ამოცანის გადაწყვეტას ითხოვს:

- მონაცემთა მთლიანობის უზრუნველყოფას და
- საიდუმლოების გარანტიას (მონაცემების მისაღებად შეზღუდვების დაყენებას).

მონაცემთა მთლიანობა. მონაცემთა ბაზის მთლიანობის უზრუნველსაყოფად იყენებენ შეზღუდვების შერჩევის მექანიზმს. ერთი მათგანია სტრუქტურული შეზღუდვები, მეორე კი უშუალოდ მონაცემთა მნიშვნელობების შეზღუდვები.

შეზღუდვები სტრუქტურულ დონეზე ეფუძნება მონაცემთა ბაზებში ფუნქციონალურ დამოკიდებულებათა (რელაციების, ატრიბუტების და ა.შ.) აღწერას. შემოიღებენ სპეციალური – გასაღებური ატრიბუტების, ინდექსების ცნებები (მარტივი ან შედგენილი). მათი საშუალებით

ხორციელდება რელაციურ ფაილებში ინფორმაციის მოწესრიგება და მონაცემთა ძებნა, ამორჩევა.

მთლიანობის შეზღუდვების მეორე დონეა მონაცემთა მნიშვნელობების ცვლილებები. მონაცემთა მნიშვნელობები შეიძლება განსაზღვრული იყოს გარკვეულ დიაპაზონებში ან მნიშვნელობები გაანგარიშებული უნდა იქნას რომელიმე მათემატიკური დამოკიდებულებით (ფორმულით). თუ მონაცემის მნიშვნელობა გამოვა განსაზღვრული დიაპაზონიდან ან არ შეესაბამება არსებულ მათემატიკურ დამოკიდებულებას, მაშინ ხორციელდება სპეციალური დამცველი ფუნქციების ამუშავება, რათა არ დაირღვეს ბაზის მთლიანობა.

მაგალითისათვის შეიძლება განვიხილოთ თანამედროვე მონაცემთა ბაზების სისტემებში (მაგალითად, `MySQL_Server`, `Oracle` მღა სხვ.) ინსტრუმენტული საშუალება `Triggers` (ჩართვა-გამორთვის ფუნქცია). ტრიგერები უზრუნველყოფს მონაცემთა მთლიანობას ლოგიკურად დაკავშირებულ ცხრილებში (რელაციებში). სისტემაში სტანდარტული ან კერძო ფუნქციების ამუშავება განისაზღვრება მომხმარებლის მიერ, ტრიგერები კი არაა დამოკიდებული პროგრამაზე, იგი ჩაირთვება ყოველთვის, როდესაც ადგილი აქვს მონაცემთა ბაზაში ინფორმაციის განახლებას: ახლის ჩამატებას, ძველის წაშლას ან შეცვლას. ამგვარად, ტრიგერების ერთ-ერთი მთავარი ფუნქციაა სისტემაში მონაცემთა ცვლილებების სტატისტიკის წარმოება.

მართვის ამოცანებში ამას დიდი მნიშვნელობა აქვს. მაგალითად, შესაძლებელია მონაცემთა ცვლილებები დამუშავდეს მათი მნიშვნელობების საკონტროლო მაჩვენებლებთან, რომელთა მიღწევის შემთხვევაში სისტემა სასწრაფოდ გააფრთხილებს მომხმარებელს მოცემულ სიტუაციაში შესაძლო გადაწყვეტილებების შესახებ.

მონაცემთა ბაზების მართვის სისტემაში მთლიანობის დაცვის მექანიზმის სახით, ზოგადად გამოიყენება:

Databases -> Referential Integrity

შემდეგ აირჩევა

Rules for{ Updating, Deleting, Inserting },

რომელშიც უნდა შეირჩეს ოფციები `Cascade` (ცვლილებების ნებართვა ჯერ „შვილ“ რელაციაში, შემდეგ „მშობლებში“), `Restrict` („მშობელში“ იკრძალება ცვლილებები, თუ არსებობს კავშირი „შვილებთან“) და `Ignore` (ყველაფერი დასაშვებია).

ამ მექანიზმის საშუალებით შეიქმნება ტრიგერების კოდები, რომლებიც მოთავსდება მონაცემთა ბაზის დასამახსოვრებელ პროცედურებში. მათი შემდგომი მიღება შესაძლებელია მენიუდან:

Data -> Stored Procedure.

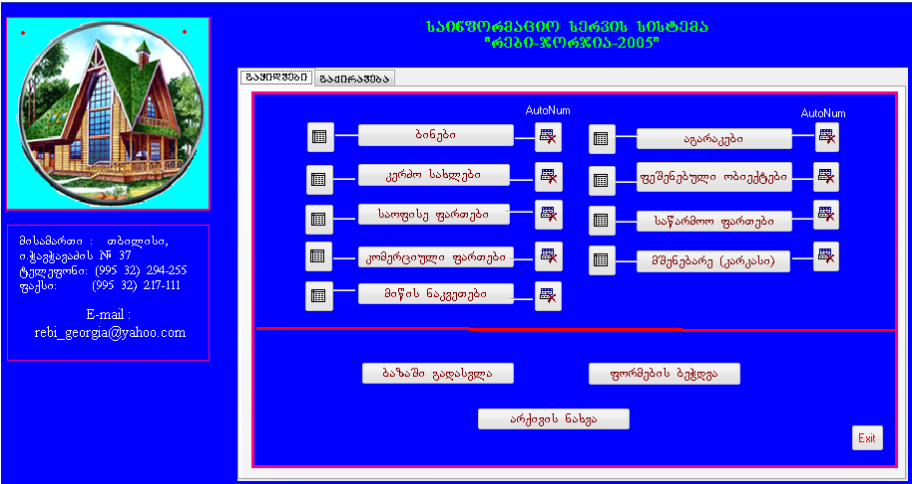
საიდუმლოების დაცვა. მონაცემთა ბაზები კოლექტიური მოხმარების რესურსია, ამიტომაც მათ მრავალი ელემენტი იყენებს. როგორც წესი, ეს ელემენტები განსხვავდება თავიანთი სტატუსებით. ყველას არა აქვს ყველაფრის მიღების უფლება. ამიტომაც მონაცემთა ბაზის ადმინისტრატორი ქმნის სპეციალურ საპაროლო სისტემას (ჟურნალს), რომელშიც ჩაწერილია თითოეული მომხმარებლის ვინაობა, სტატუსი და მონაცემთა გამოყენების უფლებები. პაროლის გარეშე მომხმარებელი ვერ შეუერთდება მონაცემთა ბაზას. სტატუსში იგულისხმება მომხმარებელთა კლასიფიკაცია. მაგალითად, კლიენტები, რომელთაც აქვთ მონაცემების მხოლოდ წაკითხვის (Read Only) უფლება. მათ არ შეუძლიათ შეიტანონ ახალი ან ჩაასწორონ მონაცემები ბაზაში. მონაცემთა მნიშვნელობის ჩასწორება ან ჩამატება უშუალოდ მნიშვნელობების ღონეზე შეუძლიათ სტატუსით ნებადართულ მომხმარებლებს. მონაცემთა ფაილების სტრუქტურების ცვლილების უფლება მხოლოდ ადმინისტრატორის ჯგუფის პრეროგატივაა.

3.4. მონაცემთა ბაზა და სერვის-პროგრამები უძრავი ქონების სააგენტოს მაგალითზე

ჩვენს მიერ შესრულდა პრაქტიკული საპროექტო სამუშაო, რომელიც ეხებოდა სერვის-ფირმის ტიპის სააგენტოს საინფორმაციო სისტემის აგებას უძრავი ქონების აღრიცხვისა და გაყიდვა-გაქირავების კონტროლის (მონიტორინგის) მიზნით.

დაისვა ამოცანა ასეთი სისტემის ასაგებად მაიკროსოფტის საოფისე Ms Access პაკეტის გამოყენებით (დამკვეთის კომპიუტერული ქსელი ფლობდა საოფისე პაკეტებს). სისტემის მიზანია სააგენტო ფირმის სერვის-სამსახურის ოპერატიულობის ამაღლება, საინფორმაციო ფონდის და არქივის ოპერატიული ანალიზის ამოცანის გადაწყვეტა, მონაცემთა ბაზების განახლების და სისტემის მომხმარებელთა ინტერფეისების დამსუავება. 3.10 ნახაზზე ნაჩვენებია სისტემის მთავარი გვერდი. უძრავი ქონების სააგენტო ფირმის საინფორმაციო სისტემა ძირითადად ორი ამოცანის, კერძოდ გაყიდვებისა და გაქირავების მომსახურებითაა

დაკავებული. ობიექტები, რომლებიც უძრავი ქონების აღრიცხვისა და მონიტორინგის სისტემაშია ჩადებული, წარმოადგენს: ბინებს, კერძო სახლებს, საოფისე, კომერციულ და საწარმოო ფართებს, მიწის ნაკვეთებს, აგარაკებს, ფეშენებულ ობიექტებს, მშენებარე კარკასებს.



ნახ.3.10. სააგენტო ფირმის სისტემის მთავარი გვერდი

3.11 ნახაზზე ნაჩვენებია ერთ-ერთი ინტერფეისის მაგალითი, „კერძო-სახლების გაყიდვა“. ყველა სხვა მიზნობრივ ფუნქციას (გაყიდვისა და გაქირავების ამოცანებს) აქვს საკუთარი ინტერფეისი, რომელიც მიბმულია მონაცემთა რელაციური, განაწილებული ბაზის ცხრილებთან.

კომპიუტერული სისტემა აგებულია ისე, რომ შესაძლებელია ტელეფონით, გვართ ან სხვა ატრიბუტით მოვებნით ჩვენთვის საჭირო ობიექტი. 3.12 ნახაზზე ნაჩვენებია „ტელ-№“ -ით იმ პიროვნებისა და მისი ობიექტების პოვნა, რომლებიც მონაცემთა ბაზაშია ჩადებული.

სისტემის მონაცემთა ბაზასთან კავშირი მომხმარებლისათვის მარტივია და მოითხოვს მხოლოდ თავისი სურვილის სწორად ჩამოყალიბებას. საინტერფეისო ფორმაზე შესაძლებელია სხვადასხვა კრიტერიუმების ამორჩევა და პასუხის მიღება.

ცხრილის სახით და შესაძლებელია რამდენიმე სტრიქონის ერთად დანახვა (შედარების, ამორჩევის თვალსაზრისით).

BinebisGakidva [X] [] []

ბინების გაყიდვა

№: უბანი: არჩევანი:

მისამართი: არასტანდარტული:

ტელეფონი: მშენებ-ობა:

მშენებელი: სახელი:

სართული: სულ-სართ: ამტკიცეს ღ შეამტკიცეს

ოთახ-სართ: 36/20/14 ღ შეამტკიცეს

სართო-ფართი: რეგონტი:

საფარველი: სვ-ფართ:

გომიწარმო-მომსახურ: ბათარია:

ანგარიშ-სართო/ფართი: გონივრება:

კოდი: სინგალიზ:

ლარი: ახალგაზრ:

სარეალ:

ფართობი:

ტარიფი:

სააქრო-ფორმა:

მნიშვნა: ფილტრი:

ფილტრიდან ამოწმვა:

გამომწოდებლის პარიფორმის მარტივი მარტივად: მარ-მარტივი: მარტივი-მარტივი:

მარტივი ფილტრის მარტივად: ფილტრიდან ამოწმვა:

Record: 1 of 78

ნახ.3.13. ინტერფეისის ფორმა „ბინების გაყიდვა“

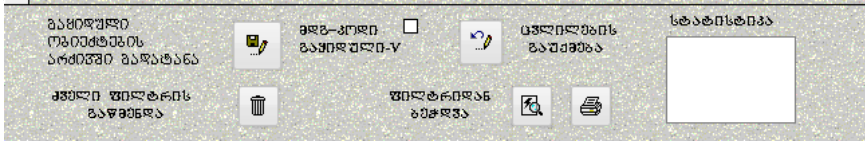
BinebisGakidva [X] [] []

უბანი	მისამართი	ტელეფონი	მობილური	სახელი	სართი	ს
ბაგე	ყოფშიძის ქ.8 II სად	29 4332	89 9144246	თამარიკო	9	
ბაგე	ახაშიძის ქ.43	222579	89 39 42752	ნინო	1	
ბაგე	ყოფშიძის ქ.12	2359 52		ნინო/დათო	7	
ბაგე	ბარნოვის ქ.82 ზიდთან	29179 3		ნინო	3	
ბაგე	ახაშიძის ქ.17	7449 41		ტარიცილი	1	
საბურთალო	ბანტრიონის ქ.24 I სს	3629 41		ნიკო	3	
ბაგე	ახაშიძის ქ.14 IIIსად	230877		ვალერი	6	
დიდუბე-ჩუღურეთი	თამარ მეფის გამზ.23	9 68024		დათო	5	
	მცხეთის I სვ. ბურჯია				0	
ბაგე	არაყიშვილის ქ.8ა	224641		დათო	2	

Record: 1 of 78

**ნახ.3.14. ინტერფეისის ფორმა „ბინების გაყიდვა“
ცხრილური ვარიანტი**

3.15 ნახაზზე გამოტანილია მენეჯერის სერვის ფუნქციების ლილაკები, რომელთა საშუალებითაც ხერხდება ფილტრების ორგანიზება, გაყიდული ობიექტების არქივში გადატანა, გამობეჭდვა და ა.შ., რაც მნიშვნელოვნად ამარტივებს მის მუშაობას. ბიზნეს-პროცესი აწყობილია ავტომატიზებულ რეჟიმში.



ნახ.3.15. სერვის ფუნქციების ლილაკები მენეჯერისათვის

Field Name	Data Type
#	AutoNumber
ubani	Text
misamarTi	Text
telefoni	Text
mobiluri	Text
saxeli	Text
sarTuli	Number
sul_sart	Number
penthausi	Yes/No
dupleqsi	Yes/No
oTax_raod	Number
oTax_farTebi	Text
Weri	Number
saerTo_farTi	Number
samzareulo	Number
komunal	Text
gaTboba	Yes/No
kondencia	Yes/No
signaliz	Yes/No
lifti	Yes/No
aivnebi_raod/farTi	Text
holi	Number
loji	Number
sv_wert	Text
proeqti	Text
mSeneb_org	Text
remonti	Text
sakuTr_forma	Text
a/sadgomi	Text
sardafi	Number
fasi	Text
TariRi	Date/Time
foto	OLE Object
statistika	Memo
SeniSvna	Text
mdg_kodi	Yes/No
filtri	Yes/No

3.17 ნახაზზე მოცემულია მონაცემთა ბაზის ერთ-ერთი საილუსტრაციო ცხრილის „ბინების გაყიდვა“ მაგალითი.

3.18 ნახაზზე ნაჩვენებია საბეჭდი ფორმის მომზადების ფრაგმენტი.

ნახ.3.17. ცხრილი „ბინების გაყიდვა“

Report Header											
პირველი პერიოდი											
Page Header											
№	უბანი	ტელეფონი	სახელი	სართლი	ქუჩა	კონტაქტი	ლუბი	რეზიდენტი	სერტიფიკატი	სტატუსი	საფურცელი
	მისამართი										
Detail											
№	უბანი	მისამართი	ტელეფონი	სახელი	სართლი	ქუჩა	საფურცელი	სტატუსი	სერტიფიკატი	სტატუსი	საფურცელი
Page Footer											
=Now()											=Page
Report Footer											

ნახ.3.18. რეპორტის მაგალითი დაპროექტების რეჟიმში

ჩვენს მიერ დაპროექტებული მონაცემთა ბაზების მართვის სისტემა რეალიზებულია და პრაქტიკულად გამოიყენება ფირმის RebiGeorgia მიერ, რომელსაც თბილისში აქვს რამდენიმე ოფის-ცენტრი [70].

ლიტერატურა:

1. Booch G., Jacobson I., Rumbaugh J. Unified Modeling Language for Object-Oriented Development. Rational Software Corporation, Santa Clara, 1996.
2. Oestereich B. Software Engineering mit UML. BRD. Springer Verlag. 1998.
3. Seemann J., Wolf J. UML – Unified Modeling Language. Informatik Spektrum 21: 89-90 S. Springer-Verlag. 1998.
4. სურგულაძე გ., შონია ო., ძამაშვილი ნ. მართვის ინფორმაციული სისტემების პროგრამული უზრუნველყოფის აგება UML-ტექნოლოგიით
5. Boggs W., Boggs M. Mastering UML with Rational Rose. Copyright 1999 SyBEX, California.
6. Paradigm Plus. Methods Manual. PLATINUM technology. 1997.
7. ჩოგოვაძე გ., სურგულაძე გ., შონია ო. მონაცემთა და ცოდნის ბაზების აგების საფუძვლები. თბ., სტუ, 1996.
8. გოგიჩაიშვილი გ., სურგულაძე გ., შონია ო. დაპროგრამების მეთოდები: სტრუქტურულ და ობიექტ-ორიენტირებულ მიდგომათა საფუძვლები C და C++ ენების ბაზაზე. თბ., სტუ, 1997.
9. ჩოგოვაძე გ., სურგულაძე გ. რელაციური ალგებრის ოპერაციების შესრულების ეფექტური პროცედურის აგების ერთი ინსტრუმენტის შესახებ მონაცემთა ბაზებში. საქ.მეცნ.აკად. „მოამბე“, 148-№3, 1993.
10. სუხიაშვილი თ., კაშიბაძე მ. სამუშაო პროცესების მოდელირება და ანალიზი მართვის განაწილებულ სისტემებში. სტუ შრ.კრ. „მართვის ავტომატიზებული სისტემები“, №1, 2006. 75-78 გვ.
11. Reisig W. Distributed Algorithm of Petri Net.
12. Горбатов В.А. Теория частично-упорядоченных систем. Москва, Радио, 1978.
13. სურგულაძე გ., გულუა დ. განაწილებული სისტემების ობიექტ-ორიენტირებული მოდელირება უნიფიცირებული პეტრის ქსელებით. სტუ. თბ., 2005.
14. ბოტჭე კ. (გერმ.), სურგულაძე გ., კაშიბაძე მ. შემკვიდრობითობა მართვის ინფორმაციული სისტემების დაპროგრამებაში: მონაცემთა ბაზებიდან UML-ტექნოლოგიამდე. საერთ.კონფ. შრ.კრ., №4(437). სტუ, თბილისი, 2001. 206-213 გვ.
15. კაშიბაძე მ. ოპერაციულ სისტემაში ჩიხური პროცესების მართვის ერთი ალგორითმის შესახებ. სტუ შრ.კრ. „მართვის ავტომატიზებული სისტემები“, №1(4), 2008. 119-122 გვ.
16. Дейтель Г. Введение в операционные системы. Пер. с англ., Мир, М., 1987

17. სურგულაძე გ., კაშიბაძე მ. ოპერაციული სისტემები: პროცესების მართვის კვლევა პეტრის ქსელების თეორიის გამოყენებით. სტუ. თბ., 1993
18. Питерсон Дж. Теория сетей Петри и моделирование систем. Пер. с англ., Мир, М., 1984
19. Reisig W. Elements of distributed Algorithms. Springer-Verlag. Berlin-Heidelberg. 1098.
20. სურგულაძე გ., პეტრიაშვილი ლ. მონაცემთა საცავის აგების ტექნოლოგია ინტერნეტული ბიზნესის სისტემებისათვის. სტუ, თბილისი, 2005.
21. <http://www.oriflame.ge/index.jhtml> - გადამოწმებულია 20.11.2007
22. ჩოგოვაძე გ., გოგიჩაიშვილი გ., სურგულაძე გ., შეროზია თ., შონია ო. მართვია ავტომატიზებული სისტემების დაპროექტება და აგება. თბილისი: ტექნიკური უნივერსიტეტი. 2001.
23. სურგულაძე გ., თოფურია ნ., კაშიბაძე მ., საღინაძე დ. პროგრამული აპლიკაციების მონაცემთა ბაზების დაპროექტება კატეგორიალური მიდგომით. სტუ შრ.კრ. „მართვის ავტომატიზებული სისტემები“, №1(4), 2008. 206-213 გვ.
24. სურგულაძე გ., ვედეკინდი ჰ., თოფურია ნ. განაწილებული ოფის-სისტემების მონაცემთა ბაზების დაპროექტება და რეალიზაცია UML-ტექნოლოგიით. მონოგრაფია. სტუ, თბილისი. 2006.
25. Halpin T.A., Information Modeling and relational Databases, Morgan Kaufmann Publishers, San Francisco, 2001. www.mkp.com/books_catalog/catalog.asp/ ISBN=1-55860-672-6.
26. გოგიჩაიშვილი გ., ფრანგიშვილი ა., სურგულაძე გ. ინფორმატიკა, პროგრამული ტექნოლოგიები და მათი განვითარების და სწავლების თანამედროვე მიმართულებანი. სტუ შრ.კრ.„მას“ №1(2), 2007.
27. Николаишвили В., Сургуладзе Г. Топурия Н., Кашибадзе М. Категориальный подход разработки абстрактных моделей данных для объектно-ориентированных, реляционных баз данных. Тез. Докл. Интерн. Конф. Киев, 2006.
28. Albrecht J. Anfrageoptimierung in Data – Warehouse-Systemen auf Grundlage des multidimensionalen Datenmodells. Friedrich-Alexander-Universität, Erlangen-Nürnberg, 2003.
29. Albrecht J. Grundlagen der Anfrageverarbeitung in relationalen Datenbanksysteme. Erlangen-Nürnberg, 2004.
30. ðæŋ Æ. C#: Æñõññõñ ðñðñññðñññ. Ýñõõññññ ðñðñññññ. ðð.ñ ãñæ., "DiaSoft", Ññ., 2002.
31. Robinson S., Cornes O., Glynn J., Harvey B., McQueen C., Moemeka J., Nagel C., Skinner M., Watson K. Professional C#. Birmingham, WroxPress, 2001.
32. სურგულაძე გ. დოლიძე თ., ყვავაძე ლ. კომპონენტურ-ვიზუალური დაპროგრამება. სტუ. თბილისი. 2006.

33. სურგულაძე გ., კაშიბაძე მ., ჩიქოვანი დ. ინფორმაციული ბაზის მოტივების დაცვა განაწილებულ სისტემებში საწარმოო გაერთიანებათა მარკეტინგული უზრუნველყოფის მაგალითზე. სტუ-ს შრ.კრ. №8(391). თბ., 1992
34. კოტლერი ფ. მარკეტინგის საფუძვლები. თარგ. ინგლ. მაცნე, თბ., 1993.
35. სურგულაძე გ., ქაჩიაია ვ., ილდიზ ი., თოფურია ნ. კონცეპტუალური მოდელის დაპროექტება UML-ტექნოლოგიით უნივერსიტეტის მონაცემთა ბაზების აგებისას. სტუ შრ.კრ. „მას“, №1, 2006. 154-158 გვ.
36. Цаленко М. Моделирование Семантики в базах данных. Наука, М., 1989.
37. Robinson S., Cornes O., Glynn J., Harvey B., McQueen C., Moemeka J., Nagel C., Skinner M., Watson K. Professional C#. Birmingham, Wrox Press, 2001.
38. Öððèíííà Ë., ìèáááá Á. MS ACCESS 2000: Ðaçðááíðèà ìðèíæáíèè. Ñ-Ìáððááððã, 2002.
39. Íóðáè Ì., Êííòá Ì. SQL Server 2000. Ñ-Ìáððááððã, 2002.
40. სურგულაძე გ. დაპროგრამების ვიზუალური მეთოდები და ინსტრუმენტები: UML, Ms Visio, Borland C++ Builder . სტუ. თბილისი, 2006.
41. ჩოგვაძე გ., გოგიანიშვილი გ., სურგულაძე გ., შეროზია თ., შონია ო. მართვის ავტომატიზებული სისტემების დაპროექტება და აგება (თეორიულ-პრაქტიკული ინფორმაცია). სტუ, თბილისი, 2001.
42. სურგულაძე გ., შონია ო., ყვავაძე ლ. მონაცემთა ბაზების მართვის სისტემები: Ms Access, SQL Server, InterBase, Oracle, Corba. სტუ, თბილისი, 2004.
43. Codd E.F. A relational model of data for large shared data banks. Communications of the ACM. 1970.
44. Codd E.F., Codd S.B., Salley C.T.: Providing OLAP to User-Analysts: An IT Mandate. Codd & Associates, Ann Arbor/Michigan 1993.
45. Альманах программиста: Microsoft ADO.NET, SQL Server, Доступ к данным из приложений. Сост Купцевич Ю.Е.. Москва, «Русская Редакция», 2003.
46. Halpin, T. A. 2001, 'Object Role Modeling: an overview', white paper, (online at www.orm.net). გადმოწმებულია 20.10.2007.
47. 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). გადმოწმებულია 20.10.2007.
48. Wedekind H. Objektorientierte Schemaentwicklung. Ein kategorialer Ansatz fuer Datenbanken und Programmierung. Wissenschaftsverlag, Manheim/Wien/Zuerich. 1991.

49. Barker R. CASE*Method. Entity-Relationship Modelling. Copyright Oracle Corporation UK Limited, Addison-Wesley Publishing Co., 1990.

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

51. Заложнев А. Модели и методы внутрифирменного управления - urss.ru/cgi-bin/db.pl?lang=sp&blang=ru&page=Book&list=189&id.

52. Леонтиев Б. MS Office Visio 2003: Не для Дилетантов. Построение проектов, диаграмм и бизнес-схем в ОС MS Windows XP. Москва, ЗАО "Новый издат.дом", 2005.

53. Буч Г., Рамбо Дж., Джакобсон А. Язык UML. Руководство пользователя. Пер. с англ., Питер, 2004.

54. Сургуладзе Г., Топурия Н., Петриашвили., Кикнадзе М. Автоматизированное проектирование классов и межклассовых отношений для распределенных, реляционных баз данных. Georgian Engineering News, No 3. Tbilisi, 2006.

55. Чоговадзе Г.Г., Качибая В.В., Сургуладзе Г.Г. Теория реляционных зависимостей и проектирование логической схемы базы данных. «Мецниереба», Тб., 1988.

56. Гогичаишвили Г.Г., Сургуладзе Г.Г. Разработка прикладного программного обеспечения интегрированных информационных систем управления на основе UML. Georgian Electronic Scientific Journal. 2002, №1. <http://gesj.internet-academy.org.ge>. გადამოწმებულია 20.10.2007.

57. Прангишвили А., Прокопьев С. Информационные технологии согласования управленческих решений по выбору целей и стратегий в конфликтологии. Georgian Electronic Scientific Journal. 2005, №3(7). <http://gesj.internet-academy.org.ge>. გადამოწმებულია 20.10.2007.

58. გოგიჩაიშვილი გ., ფრანგიშვილი ა., სურგულაძე გ. ინფორმატიკა, პროგრამული ტექნოლოგიები და მათი განვითარების და სწავლების თანამედროვე მიმართულებანი. სტუ შრ.კრ. „მას“ №1(2), 2007.

59. 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 - გადამოწმებულია 10.11.2007.

60. რეისიგი ვ., სურგულაძე გ., გულუა დ. დაპროგრამების სწავლებისა და სერტიფიცირების პროცესის მოდელირება სისტემური პეტრის ქსელებით. სტუ შრ.კრ. №4(437), 2001.

იბეჭდება ავტორთა მიერ
წარმოდგენილი სახით

გადაეცა წარმოებას 1.03.2009 წ. ხელმოწერილია დასაბეჭდად 10.04.2009 წ. ოფსეტური ქაღალდის ზომა 60X84 1/16. პირობითი ნაბეჭდი თაბახი 11,5. სააღრიცხვო საგამომცემლო თაბახი 10. ტირაჟი 100 ეგზ.

გამომცემლობა „ტექნიკური უნივერსიტეტი“
თბილისი, მ. კოსტავას 77