

საქართველოს ტექნიკური უნივერსიტეტი  
ხელნაწერის უფლებით

ირაკლი ბულია

**Web-აპლიკაციების დაპროექტება და რეალიზაცია  
ელექტრონული ბიზნესის მართვაში**

დოქტორის აკადემიური ხარისხის მოსაპოვებლად  
წარდგენილი დისერტაციის

ავტორეფერატი

თბილისი  
2012 წელი

სამუშაო შესრულებულია საქართველოს ტექნიკური უნივერსიტეტის  
ინფორმატიკისა და მართვის სისტემების ფაკულტეტის  
ორგანიზაციული მართვის დეპარტამენტის  
„მართვის ავტომატიზებული სისტემების“ მიმართულებაზე

სამეცნიერო ხელმძღვანელი: ტექნიკის მეცნიერებათა დოქტორი,  
სრული პროფ. გია სურგულაძე

რეცენზენტები:

სრული პროფ. ზურაბ ჩხაიძე,  
ასოცირებული პროფ. დავით გულუა

დაცვა შედგება 2012 წლის ” 20 ” ივნისს, 16.00 საათზე  
საქართველოს ტექნიკური უნივერსიტეტის - „ინფორმატიკისა და  
მართვის სისტემების“ ფაკულტეტის სადისერტაციო საბჭოს კოლეგიის  
სხდომაზე, კორპუსი მე-6, აუდიტორია 207-ა  
მისამართი: 0175, თბილისი, კოსტავას 77.

დისერტაციის გაცნობა შეიძლება სტუ-ს ბიბლიოთეკაში,  
ხოლო ავტორეფერატისა - სტუ-ს ვებგვერდზე

სადისერტაციო საბჭოს

მდივანი: სრული პროფ. თინათინ კაიშაური

## ნაშრომის ზოგადი დახასიათება:

**თემის აქტუალურობა.** ელექტრონული ბიზნესის მართვის თანამედროვე საინფორმაციო სისტემებში პროგრამული უზრუნველყოფების უმრავლესობა დაკავშირებულია ერთმანეთთან. მათ შორის წარმოებს ინფორმაციის მიმოცვლა. ეს აპლიკაციები შესაძლებელია განთავსებული იყოს როგორც კორპორაციის შიგა ქსელში, ასევე გლობალურ ქსელშიც. ორგანიზაციები ფლობს სხვადასხვა პროგრამებს, რომლებიც სხვადასხვა მიზნებს ემსახურება. ასევე კომპანიებს შორის ხდება ელექტრონული სახით მონაცემების გაცვლა (Electronic Data Interchange), სახელმწიფო სტრუქტურებთან, ასევე ტერიტორიულად დაშორებულ განყოფილებებთან. ყველა ამ პროგრამას ესაჭიროება გარკვეული სახის კავშირი სხვა სისტემებთან, რაშიც იგულისხმება მონაცემების გაცვლა-დამუშავება, გარკვეული ბიზნეს-ლოგიკის შესაბამისად.

ბიზნეს-პროცესების მართვის სხვადასხვა ტიპის აპლიკაციები, როგორცაა პროდუქციის მიწოდების (სასაწყობო მარაგების მართვის და მიწოდების ორგანიზება), კლიენტებთან ურთიერთქმედების სისტემები (არსებულ და პოტენციურ კლიენტებთან), ბუღალტრული აღრიცხვის, საწარმოთა რესურსების მართვის, Business Intelligence-ს სისტემებთან (OLAP), ადამიანური რესურსების მართვის, ჯანდაცვის და სხვა აპლიკაციებს, როგორც წესი, საჭიროებს ერთმანეთთან ურთიერთობას, მონაცემების გაცვლის და სხვადასხვა ბიზნეს-პროცესების მართვის კუთხით. ეს სისტემები მუშაობს სხვადასხვა ოპერაციული სისტემის სერვერებზე, სხვადასხვა სახის მონაცემთა ბაზებზე, შესრულებულია დაპროგრამების განსხვავებული ენების, განსხვავებული ტექნოლოგიების გამოყენებით, ან მოძველებულ სისტემებზე, რომელთა განვითარება და მხარდაჭერაც აღარ ხდება. რის გამოც რთულია მათ შორის პროცესების მართვა, ავტომატიზაცია. ასევე ხდება მონაცემების დუბლირება, მათი სხვადასხვა სისტემებში განთავსების გამო.

ზემოაღნიშნულიდან გამომდინარე განსაკუთრებით აქტუალურია კორპორაციაში ბიზნეს-პროცესების მართვის ელექტრონული სისტემების ინტეგრაციის (Enterprise Application Integration) პრობლემის გადაწყვეტა, რომლის მიზანია ორგანიზაციაში არსებული სისტემების დაკავშირება, აპლიკაციების ურთიერთკავშირის გამარტივება და ბიზნეს-პროცესების ავტომატიზაციის უზრუნველყოფა. ასეთი კავშირების სარეალიზაციოდ იქმნება სხვადასხვა პროგრამები, რომლებიც ასრულებს შუალედურ რგოლს ამ სისტემებს შორის. მსგავსი ბროკერული აპლიკაციების რაოდენობა იზრდება, რაც უფრო იზრდება ორგანიზაციაში სისტემების და ასევე გარე ორგანიზაციებთან არსებული კავშირები. საჭირო ხდება ამ ბროკერული აპლიკაციების შექმნა, იზრდება მათი ადმინისტრირების, მონიტორინგის, პროგრამული უზრუნველყოფის ცვლილებების მართვის სამუშაოები, რაც იწვევს ინფორმაციული ტექნოლოგიების რესურსების დიდი ოდენობით მოხმარებას. საჭირო ხდება როგორც ადამიანური რესურსების, ასევე ინფრასტრუქტურული, პროგრამული უზრუნველყოფების მუდმივი განახლება. თითოეული ბროკერული აპლიკაციის შექმნა არის ხანგრძლივი და შრომატევადი პროცესი. მაგალითად, თუ ორგანიზაციას სჭირდება N რაოდენობის აპლიკაციის ერთმანეთთან დაკავშირება, ამ აპლიკაციებს შორის ბროკერების რაოდენობამ შეიძლება მიაღწიოს  $(N-1)/2$  რიცხვს, რაც 10 აპლიკაციის შემთხვევაში არის 45. ეს კი საკმაოდ დიდი რიცხვია და რესურსების დიდ ხარჯებთანაა დაკავშირებული.

ელ-ბიზნესის სხვადასხვა სისტემები განსხვავებული ფორმატის მონაცემებს იძლევა გარე სისტემებთან დასაკავშირებლად. შესაძლებელია ეს იყოს ტექსტური ან ორობითი ფაილი, მონაცემები იყოს XML ფორმატში, ვებ-სერვისების საშუალებით იძლეოდეს მონაცემებზე წვდომის საშუალებას, ან ბაზის სხვადასხვა პროცედურების და ფუნქციების საშუალებით, ამასთანავე მონაცემთა გადაცემა ხდებოდეს TCP, HTTP პროტოკოლების გამოყენებით, თუ მონაცემთა გადაცემის სხვა სახის საშუალებებითაც. საჭიროა სხვა სახის ინტეგრაციული არქიტექტურის შექმნა, რათა კომპანიამ უფრო მარტივად უზრუნველყოს ინტეგრაციის ამოცანები, ნაკლები რესურსების გამოყენებით.

ამოცანა მდგომარეობს ისეთი სისტემის შექმნაში, რომელიც უზრუნველყოფს არსებული სისტემების ინტეგრაციას, როგორც სხვადასხვა ორგანიზაციებს შორის, ასევე ორგანიზაციის შიგნით, რომელიც უზრუნველყოფს სხვადასხვა ფორმატის მონაცემების მიღება-გადაცემა-დამუშავებას, მაღალ წარმადობას და საიმედოობას.

აპლიკაციების ინტეგრაციას ძირითადად აქვს სამი დანიშნულება:

- მონაცემთა ინტეგრაცია, რაც უზრუნველყოფს მონაცემების იდენტურობას სისტემებს შორის;
- აპლიკაციათა მიმწოდებლებზე დამოუკიდებლობა. უზრუნველყოფს, რომ აპლიკაციის ცვლილების შემთხვევაში, ბიზნეს-პროცესი და ბიზნეს წესების ხელახალი შექმნა არ იყოს საჭირო;
- ფასადის/ინტერფეისის შექმნა, რაც უზრუნველყოფს აპლიკაციებთან ერთიერთობის ერთიანი ინტერფეისის შექმნას, რომელიც საშუალებას იძლევა აპლიკაციებთან კომუნიკაცია შესრულდეს მათი შიგა სტრუქტურების შესწავლის გარეშე.

**სამუშაოს მიზანი და ამოცანები.** დისერტაციის მიზანია კორპორაციული და ინტერკორპორაციული ბიზნეს-პროცესების მართვის ვებ-აპლიკაციების დაპროექტება და რეალიზაცია ელექტრონული სისტემების ინტეგრაციის შესაძლებლობით და სერვის-ორიენტირებული არქიტექტურით.

მიზნის მისაღწევად ნაშრომში განიხილება შემდეგი ძირითადი ამოცანები:

- არსებული თანამედროვე ინტეგრაციის სისტემების ანალიზი და შესაბამისი ინფორმაციული ტექნოლოგიების კლასიფიკაცია, ობიექტ-, პროცეს- და სერვის-ორიენტირებული დაპროექტების პრინციპებით;
- შიგაკორპორაციული და კორპორაციათაშორისი ბიზნეს-პროცესების ტრადიციული და სერვის-ორიენტირებული მოდელების აგება სისტემური ანალიზის საფუძველზე, BPMN და UML ტექნოლოგიების ბაზაზე. მათი შედარებითი ანალიზი;
- ვებ-აპლიკაციების დასაპროექტებლად ორგანიზაციის სერვისული ბიზნეს-პროცესების ობიექტ-ორიენტირებული მოდელირება კლასების, ობიექტების და ვებ-მეთოდების ფორმალიზაციის საფუძველზე, პოლიმორფიზმის, მემკვიდრეობითობის და ინტერფეისული თვისებების გათვალისწინებით;
- სერვის-ორიენტირებული კორპორაციული ვებ-აპლიკაციების მონაცემთა განაწილებული ბაზების სტრუქტურების დასაპროექტებლად ობიექტ-როლური მოდელების (ORM) აგება და კვლევა რევერსიული CASE ტექნოლოგიების გამოყენებით;
- სერვის-ორიენტირებული არქიტექტურის კორპორაციული სისტემის ბიზნეს-პროცესების უნიფიცირებული მოდელების ასახვის (BPMN -> Activity-D -> PetNet) ალგორითმების შემუშავება სტოქასტური, დროითი პეტრის ქსელის გრაფებში,

მათი პროცესების შესრულების ეფექტურობის შეფასების და სერვისული უზრუნველყოფის შემდგომი სრულყოფის გადაწყვეტილების მიღების მიზნით;

- სერვის-ორიენტირებული არქიტექტურის ინტერკორპორაციული სისტემის ინფორმაციული ნაკადების გაცვლის სერვისული ბიზნეს-პროცესების იმიტაციური მოდელის აგება და მისი ფუნქციონირების დროითი მახასიათებლების კვლევა ფერადი პეტრის ქსელების (CPN) გამოყენებით;

- პროექტის შედეგების საფუძველზე ესპერიმენტული პროგრამული სისტემის რეალიზაცია Ms Visual Studio .NET პლატფორმაზე, ASP.NET, ADO.NET, MsSQL\_Server, C#.NET, Natural ORM Architect და BizTalk პროგრამული პაკეტების გამოყენებით.

**კვლევის ობიექტი.** საფინანსო-საბანკო ორგანიზაციების, ფინანსთა სამინისტროს შემოსავლების სამსახურის, დიდი სავაჭრო ცენტრების და სხვა ობიექტების სერვის-პროცესებისა და ელექტრონული კომერციის მართვის საინფორმაციო სისტემები.

**კვლევის მეთოდები.** სისტემური ანალიზის მეთოდი. ობიექტ-ორიენტირებული მოდელირების, ანალიზის და პროექტირების მეთოდები. უნიფიცირებული მოდელირების ენა და მისი რეალიზაციის ინსტრუმენტები. პეტრის ქსელების მათემატიკური მოდელი. გრაფთა თეორია. იმიტაციური მოდელირება ფერადი პეტრის ქსელებით. ობიექტ-ორიენტირებული დაპროგრამების თეორია. მონაცემთა განაწილებული ბაზების თეორია. ობიექტ-როლური მოდელირება. კლიენტ-სერვერული არქიტექტურა. სერვის-ორიენტირებული არქიტექტურა.

**მეცნიერული სიახლე.** საფინანსო-საბანკო ინტერკორპორაციული მენეჯმენტის ბიზნეს-პროცესების მართვის ვებ-აპლიკაციების დასაპროექტებლად და სარეალიზაციოდ სერვის-ორიენტირებული, უნიფიცირებული მოდელების და პროცეს-ორიენტირებული მეთოდების დამუშავება ელექტრონული სისტემების ინტეგრაციის მიზნით.

1. პირველად დისერტაციაში აგებულ იქნა ელექტრონული ბიზნესის მართვის სერვის-ორიენტირებული მოდელები პროცეს-ორიენტირებული ტექნოლოგიების გამოყენებით და ჩატარდა მათი შედარებითი ანალიზი თანამედროვე ტრადიციულ მოდელებთან;

2. სერვის-ორიენტირებული მოდელების ბაზაზე დაპროექტებული ვებ-აპლიკაციისთვის აგებულ იქნა სტოქასტიკურ-დროითი პეტრის ქსელის მათემატიკური მოდელი, რომლის საფუძველზე ჩატარდა ბიზნეს-პროცესების უნიფიცირებული მოდელების კვლევა;

3. პირველად დისერტაციაში შემუშავებულ იქნა საფინანსო-საბანკო ორგანიზაციების სერვის-ორიენტირებული ბიზნეს-პროცესების მართვის მოდელები კლასების პოლიმორფიზმითა და მემკვიდრეობითობით, ასევე ობიექტ-როლური მოდელები მონაცემთა ბაზების ავტომატიზებული აგების მიზნით რევერსული CASE ტექნოლოგიების გამოყენებით.

**შედეგების გამოყენების სფერო.** დისერტაციის შედეგებს აქვს პრაქტიკული ღირებულება, კერძოდ ის შეიძლება გამოყენებულ იქნას როგორც საფინანსო-საბანკო და კომერციულ დაწესებულებებში, ასევე ნებისმიერი დარგის კორპორაციულ ორგანიზაციებში, რომლებიც სერვის-ორიენტირებული არქიტექტურის პრინციპებით ფუნქციონირებას გამოიყენებს. განსაკუთრებით

ღირებულია ასეთი მიდგომა ინტერკორპორაციული მენეჯმენტის ბიზნეს-პროცესების მართვისთვის.

**ნაშრომის აპრობაცია:** დისერტაციის ძირითადი შინაარსი მოხსენებული იყო ინფორმაციისა და მართვის სისტემების ფაკულტეტის მართვის ავტომატიზებული სისტემების კოლეგიის სამეცნიერო სემინარების სხდომაზე. დისერტაციის შინაარსი და ძირითადი შედეგები წარდგენილი იყო საერთაშორისო სამეცნიერო-ტექნიკურ კონფერენციებზე (2010 და 2011 წლებში.). პუბლიკაციები: დისერტაციის ძირითადი შედეგები გამოქვეყნებულია 12 სამეცნიერო ნაშრომში, მათი ჩამონათვალი მოყვანილია ავტორეფერატის ბოლოს.

**ნაშრომის მოცულობა და სტრუქტურა:** დისერტაციის სრული მოცულობა შეადგენს 152 ნაბეჭდ გვერდს; დისერტაცია შედგება რეზიუმესაგან (ორ ენაზე), სარჩევის, შესავლის, სამი თავის და ილუსტრაციის სახით მოყვანილი 80 ნახაზის, 6 ცხრილის, დასკვნებისა, 61 გამოყენებული ლიტერატურისაგან და ავტორის მიერ გამოქვეყნებული 12 სამეცნიერო ნაშრომებისაგან, რომლებშიც ასახულია დისერტაციის ძირითადი შედეგები.

### ნაშრომის შინაარსი

**შესავალში** გადმოცემულია დისერტაციის ზოგადი დახასიათება, თემის აქტუალობა, მიზანი და გადასაწყვეტი ამოცანები, სამეცნიერო სიახლე და პრაქტიკული ღირებულება. აგრეთვე ნაშრომის მოკლე შინაარსი თავების მიხედვით.

**პირველი თავი** ეხება ელექტრონული ბიზნესის მართვის თანამედროვე საინფორმაციო სისტემების განხილვას. გადმოცემულია დღეისათვის მსოფლიოში არსებული ტრადიციული, ობიექტ-ორიენტირებული, პროცეს-ორიენტირებული და სერვის-ორიენტირებული მიდგომებით შექმნილი საინფორმაციო სისტემები და მათი შესაბამისი ინფორმაციული ტექნოლოგიები. ჩატარებულია მათი კრიტიკული ანალიზი კორპორაციული და ინტერკორპორაციული მენეჯმენტის სერვისული ბიზნეს-პროცესების მოდელირების, დაპროექტების და ვებ-აპლიკაციების პროგრამული რეალიზაციის მიზნით, მათი მოქნილი ინტეგრაციის თვალსაზრისით. ამ ამოცანის რეალიზაციისათვის საჭიროა გამოყენებულ იყოს ინტეგრაციის აპრობირებული მეთოდები, პროგრამული უზრუნველყოფის არქიტექტურის მოდელები, სტანდარტები, რომლებიც უზრუნველყოფს მოქნილობას, მასშტაბირებას, არაერთგვაროვანი და კომპლექსური სისტემების ურთიერთკავშირს.

**ორგანიზაციის აპლიკაციების ინტეგრაცია (Enterprise Application Integration).** ამ ტექნოლოგიის დანიშნულებაა საწარმოს აპლიკაციების ერთმანეთთან დაკავშირება, მათი ერთიან ბიზნეს-პროცესში ჩართვა, ასევე მათ შორის გადაცემული მონაცემთა ფორმატის ტრანსფორმაცია. EAI მეთოდის გამოყენება ხდება როდესაც საჭიროა: აპლიკაციის აპლიკაციასთან დაკავშირება (Application to Application); ადამიანის სისტემასთან კავშირი (Person to System); ბიზნესის ბიზნესთან კავშირი (Business to Business).

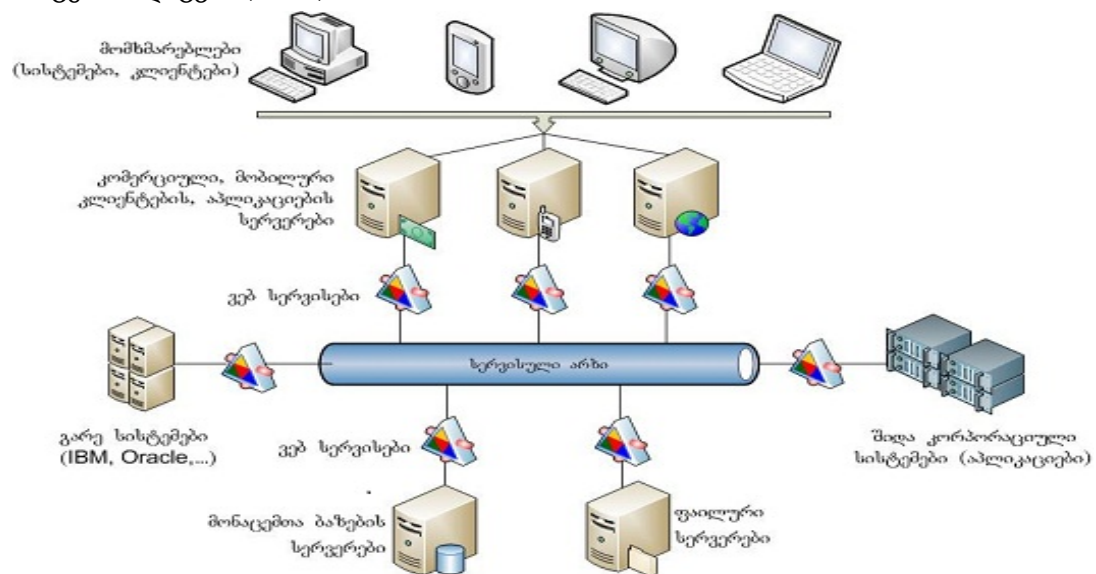
EAI მეთოდის საშუალებით ხდება აპლიკაციებს შორის პირდაპირი კავშირების დამყარების აუცილებლობის გამორიცხვა. მისი რეალიზაციისას ხდება სხვადასხვა მეთოდების, სქემების და სხვა პროგრამული უზრუნველყოფების გამოყენებით სხვადასხვა აპლიკაციების პროცეს-ორიენტირებული ინტეგრაცია. განსხვავებული ფორმატის მონაცემთა მიღება და ტრანსფორმაცია სპეციალური

ადაპტერების საშუალებით სრულდება. EAI-ში შეიძლება ცენტრალიზებულად განისაზღვროს ბიზნეს-პროცესების ლოგიკა, რაც საშუალებას იძლევა გარკვეული ბიზნეს-ლოგიკის ცვლილების შემთხვევაში, აპლიკაციების გადაკეთების გარეშე განხორციელდეს ახალი წესების იმპლემენტაცია. ინტეგრაციის არხი ზრუნავს მონაცემთა წინასწარ დამუშავებაზე განსაზღვრული წესების შესაბამისად და შედეგების გადამისამართებაზე კონკრეტული ბიზნეს-პროცესის შესასრულებლად.

**სერვის ორიენტირებული არქიტექტურა (Service Oriented Architecture)** ერთმანეთთან ურთიერთმოქმედი სისტემების პროექტირების მეთოდოლოგიაა. EAI-ის შესაძლებლობები ფართოვდება SOA-ს გამოყენებით, რომელიც სხვადასხვა აპლიკაციებსა და სხვადასხვა ორგანიზაციებს შორის სისტემების ინტეგრაციას და ბიზნეს ამოცანების შესრულებას უზრუნველყოფს. SOA-ს შემთხვევაში სხვადასხვა კომპონენტები განიხილება, როგორც გარკვეული მომსახურების მიმწოდებელი ან მომხმარებელი სერვისები. მისი საშუალებით ხდება განაწილებული მონაცემთა ბაზების, ვებ-სერვისების, სხვა სერვისების სტრუქტურირება გარკვეული ბიზნეს-პროცესების შესასრულებლად.

მაგალითად, ბანკში კლიენტისთვის სესხის გაცემა შედგება სხვადასხვა პროცესებისგან: ბანკის კლიენტის შექმნა, ანგარიშის გახსნა, სასესხო ხელშეკრულების შექმნა და სხვა. ეს პროცესები შესაძლებელია სრულდებოდეს სხვადასხვა აპლიკაციების მიერ, SOA-ს საშუალებით კი ეს პროცესი შესრულდება გარკვეული მიმდევრობის და ბიზნეს-პირობების დაცვის საშუალებით, რასაც ორკესტრაცია უზრუნველყოფს. ერთიდაიგივე სერვისები შესაძლებელია გამოყენებულ იქნას სხვადასხვა ამოცანებისთვის. საბოლოოდ, ასეთი მოდგომა საშუალებას იძლევა დაიზოგოს რესურსები პროგრამულ უზრუნველყოფის შექმნაზე, ახალი ამოცანებისთვის უკვე არსებული სერვისების გამოყენების საშუალებით.

**ორგანიზაციის სერვისული არხი (Enterprise Service Bus).** ESB არის განაწილებული სისტემების არქიტექტურული მოდელი. სხვადასხვა აპლიკაციებს შორის კავშირის დასამყარებლად ამ მოდელში გამოიყენება შუალედური პროგრამული საშუალება, რომელიც უზრუნველყოფს ტრანზაქციული ოპერაციების შესრულებას, შეტყობინებათა მარშრუტიზაციას, ტრანსფორმაციას, მონაცემთა დაცვას (ნახ.1).



ნახ.1

მისი საშუალებით შესაძლებელია სერვის-ორიენტირებული არქიტექტურის რეალიზაცია. იგი შედგება დამოუკიდებელი კომპონენტებისგან, რომლებიც განთავსებულია სერვისულ არხში და საჭიროების შემთხვევაში ურთიერთქმედებს ერთმანეთთან. სხვადასხვა სისტემები შეტყობინებებს გადასცემს ორგანიზაციის სერვისულ არხს, რომელიც ახდენს მათ მიწოდების დაყოვნებას, პრიორიტეტიზაციას, გადავადებას ბუფერში მიმღები სერვისის განთავსუფლებამდე, მონაცემების სისწორის შემოწმებას, მონაცემების ტრანსფორმაციას, პროცესების კონტროლსა და შეცდომებზე რეაგირებას.

**მონაცემთა ელექტრონული გაცვლა (Electronic Data Interchange).** EDI აღნიშნავს მონაცემების ელექტრონულ დამუშავებას ელექტრონული ტრანსპორტირების მეთოდების გამოყენებით. მასში მონაწილეობს სხვადასხვა ორგანიზაციის გამოყენებითი სისტემები. EDI სტანდარტები განსაზღვრავს მონაცემთა გაცვლის მეთოდებს და რეგულაციას ორგანიზაციებს ან მათ ფილიალებს შორის. მისი დანიშნულებაა ციფრული ინფორმაციის გადაცემის სტანდარტიზება, სხვადასხვა დანიშნულების კომპიუტერული სისტემების პროგრამული ურთიერთქმედების უზრუნველყოფა. EDI გულისხმობს რამდენიმე ორგანიზაციას შორის შეთანხმებას გარკვეული მნიშვნელობის, ფორმატის, კავშირების საშუალებით მონაცემთა გაცვლაზე. EDI-ის გამოყენების ძირითადი სარგებლობა არის მონაცემების ელექტრონული გადაცემის მაღალი სიჩქარე, ადამიანის ჩარევის გარეშე ბიზნესს მონაცემების გადაცემა, ადამიანური შეცდომების გამორიცხვა.

არსებობს მონაცემთა გადაცემის საერთაშორისო სტანდარტები, მაგალითად SWIFT-საბანკო გადარიცხვებისთვის, GAEB-მშენებლობაში, VDA-საავტომობილო ინდუსტრიაში და სხვ. მონაცემების გადაცემა სრულდება ასევე გავრცელებული პროტოკოლებით: SMTP, HTTP, FTP და სხვ.

**ბიზნეს პროცესების მენეჯმენტი (Business Process Management).** BPM მოიცავს საწარმოო პროცესების იდენტიფიკაციას, დოკუმენტირებას, დაგეგმვას, მართვას და გაუმჯობესებას. იგი მიმართულია არა მხოლოდ ტექნიკურ, არამედ ასევე ორგანიზაციული საკითხების მოსაგვარებლად, სტრატეგიულ მიმართულებათა, ორგანიზაციული კულტურის, მონაწილე მხარეების ინტეგრაციის და მართვის საკითხების დასახვეწად.

ბიზნეს-პროცესების ავტომატიზაციის პრობლემები იყოფა 3 ჯგუფად:

- კომპანიის შიგნით პროგრამული უზრუნველყოფების დაკავშირება Enterprise Application Integration (EAI);
- სხვადასხვა კომპანიების პროგრამულ უზრუნველყოფათა შორის კავშირის დამყარება Business-to-Business B2B ინტეგრაცია;
- ბიზნეს-პროცესების ავტომატიზაციის განზოგადებული მიდგომის უზრუნველყოფა, რომელიც განსაზღვრულია ბიზნეს-პროცესების მენეჯმენტის (BPM) მიხედვით.

ბიზნეს-პროცესების მენეჯმენტის მიზანია, ორგანიზაციაში არსებული პროცესების გაუმჯობესება, უკეთესი მომსახურების, კლიენტების კმაყოფილების, პროდუქციის ხარისხის გაუმჯობესების გზით, წარმოების ეფექტურობის ამაღლება და ორგანიზაციის მიზნების მიღწევა.

**აპლიკაციათა ინტეგრაციის სისტემები.** სისტემების დაკავშირებისთვის ცალკეული ბროკერული აპლიკაციების შექმნა და მათი მართვის არაეფექტურობის გამო, უმჯობესია გარკვეული სისტემის გამოყენება, რომელიც უზრუნველყოფს სხვადასხვა ფორმატის და არხით მიღებული



მონაცემების დამუშავებას, ტრანსლიაციას, და მის გადაცემას. ამ სისტემის დანიშნულებაა სხვადასხვა ტექნოლოგიებზე, პლატფორმებზე შექმნილი განსხვავებული ბიზნეს-აპლიკაციების დაკავშირება, ბიზნეს-პროცესების ავტომატიზაცია და მათი მონიტორინგი.

ინტეგრაციის ამოცანების გადაწყვეტა შესაძლებელია სხვადასხვა სისტემების გამოყენებით. ეს სისტემები უზრუნველყოფს ორგანიზაციის სერვისული არხის (ESB), სერვის ორიენტირებულ არქიტექტურის (SOA), მონაცემთა ელექტრონული გაცვლის (EDI), ორკესტრაციის, პროცესების მართვის ინსტრუმენტების საშუალებით სერვისების ინტეგრაციას, ბიზნეს პროცესების დამუშავებას და ავტომატიზაციას (BPM), სერვისების, მონაცემების, პროგრამული უზრუნველყოფების დაკავშირებას. ასეთ სისტემებს შორის შესაძლებელია გამოიყოს მსოფლიოში წამყვანი პროგრამული უზრუნველყოფის მწარმოებელ კომპანიების პროდუქტები, როგორიცაა: Jboss–Enterprise SOA Platform - RedHat-სგან, Oracle Enterprise Service Bus - Oracle-სგან, BizTalk Server – Microsoft-სგან, WebSphere Process Server – IBM-სგან, SAP Exchange Infrastructure – SAP AG-სგან, webMethods - Software-სგან და ა.შ.

ეს პროდუქტები ძირითადად განსხვავდება ოპერაციული სისტემების, მონაცემთა ბაზების ტიპების მხარდაჭერის შესაძლებლობებით. ამ სისტემებში ინტეგრირებულია სხვადასხვა მზა ადაპტერები, რომლებიც სხვადასხვა აპლიკაციების დაკავშირების საშუალებას იძლევა, რომელთაც მონაცემთა სხვადასხვა ფორმატი გააჩნია. ასევე არის ბიზნეს პროცესების სქემების აგების, ავტომატიზაციის შესაძლებლობა. აგრეთვე პროცესებზე მონიტორინგის, სხვადასხვა პარამეტრების გაზომვის ინსტრუმენტები (Business Activity Monitoring); მონაცემების გადაცემის, ტრანსფორმაციის დროს წარმოშობილი შეცდომების დიაგნოსტიკა. ასევე კონფიგურაციების საშუალებით სისტემის ფუნქციონირების, ბიზნეს წესების პარამეტრიზაცია. მათ ფუნქციებს მიეკუთვნება ასევე მონაცემების დაცვის უზრუნველყოფა, უსაფრთხოება, მაღალი წარმადობა, საიმედოობა. ამ სისტემების გამოყენების უპირატესობებია: სისტემის ელემენტების შექმნის და ბიზნეს-ლოგიკის დამუშავების სიმარტივე; მონაცემების და პროცესების მონიტორინგის შესაძლებლობა, ასევე მონაცემების ანალიზი; ბიზნეს-მომხმარებლის მიერ ბიზნეს-ლოგიკის ცვლილების შესაძლებლობა; მონაცემთა უსაფრთხო მიღება-გადაცემა; სერვერების ჯგუფების შექმნა დატვირთვების გასანაწილებლად და საიმედოობის გასაზრდელად და სხვა.

ამავე თავში მოცემულია ინფორმაციული ტექნოლოგიების იმ ინსტრუმენტების მოკლე აღწერა, რომლებიც გამოიყენება დისერტაციაში დასმული პრობლემის გადასაწყვეტად. კერძოდ განიხილება BPMN, UML, ASP.NET ტექნოლოგიები ბიზნეს-პროცესების აღწერის და პროგრამირების მიზნით, PetEdit და CPN პეტრის ქსელების ინსტრუმენტები, ბიზნეს-პროცესების იმიტაციური მოდელირების მიზნით, ობიექტ როლური მოდელირების Natural ORM Architect პაკეტი მონაცემთა ბაზის სტრუქტურის დასაპროექტებლად და ა.შ.

**მეორე თავში** განიხილება საფინანსო-საბანკო კორპორაციული მენეჯმენტის სერვისული ბიზნეს-პროცესების მოდელირება ტრადიციული და ვებ-სერვისული მეთოდებით. კერძოდ შემუშავებულია ინტერკორპორაციული აპლიკაციების ჰორიზონტალური და ვერტიკალური ინტეგრაციის მართვის მოდელები და მეთოდები სერვის-ორიენტირებული არქიტექტურის ბაზაზე.

კორპორაციული სტრუქტურის კომპანიების ფუნქციონირების ერთ-ერთი ძირითადი მახასიათებელია ბიზნეს მოთხოვნების მუდმივი ცვლილება, რასაც იწვევს სერვისების გაუმჯობესების, შიგა იერარქიული სტრუქტურის რეორგანიზაციის, გარე სისტემებთან თანამშრომლობის და სხვ. ფაქტორები.

ამ ტიპის ცვლილებების მხარდაჭერა და ასახვა კომპანიის ფუნქციონალური მართვის პროგრამულ პროდუქტებში არის ურთულესი პროცესი, რადგან მუშა პროგრამულ აპლიკაციაში ახალი ბიზნეს-მოთხოვნების ფუნქციონალის ჩაშენებასთან ერთად, დიდია რისკ-ფაქტორები ცვლილებების გავლენის საზღვრების დადგენაში, რომელიც უშუალოდ ეხება იერარქიულად დამოკიდებულ კლასებს, მეთოდებსა და მონაცემებს.

საქმიანი პროცესების მართვა ხშირ შემთხვევაში წარმოებს როგორც კომპანიის შიგა სტრუქტურის ფარგლებში, ისე მრავალორგანიზაციული მასშტაბით. ბიზნეს-ფუნქციების გადანაწილების ამგვარი სპეციფიკა საჭიროებს სხვადასხვა აპლიკაციებში დამუშავებული ინფორმაციის კომპოზიციასა და სინქრონიზაციას, რაც ჰორიზონტალური და ვერტიკალური ინტეგრაციის პრობლემებს მიეკუთვნება. ჰორიზონტალური და ვერტიკალური ინტეგრაციის კონცეფცია მნიშვნელოვანი ფაქტორია ინტრა- და ინტერკორპორაციული აპლიკაციების ბიზნეს-პროცესების მართვის პროცესში.

ობიექტ-ორიენტირებული დაპროგრამების მეთოდი არასაკმარისი აღმოჩნდა სხვადასხვა ტიპის აპლიკაციების ინტეგრაციის პოზიციების გადასაჭრელად. ვებ-აპლიკაციების პროგრამული დანართების, ბიზნეს-პროცესების მართვის ჰორიზონტალური და ვერტიკალური ინტეგრაციის პრობლემების გადასაწყვეტად მნიშვნელოვანია შემდეგი ინფოტექნოლოგიები: SOA, XML ენა, მოდელებით მართვადი არქიტექტურა (MDA), კომპოზიციური დანართების აგების ტექნოლოგია და სხვ.

სერვის-ორიენტირებული არქიტექტურა, რომლის მთავარი არსია პროცესზე ორიენტირება, აერთიანებს სხვადასხვა კომპაქტურ ბიზნეს-პროცესებს წარმოდგენილს ვებ-სერვისების სახით პოლიმორფიზმის პრინციპით და ახდენს მათ ორკესტრირებას. განსაკუთრებით მოქნილს და რაც მთავარია შესაძლებელს ხდის ინფორმაციის კომპოზიციისა და სინქრონიზაციის მართვას მრავალაპლიკაციური მუშაობის რეჟიმში. სერვის-ორიენტირებული არქიტექტურის ბირთვია ვებ-სერვისი. ვებ-სერვისის ფორმირება ხდება განზოგადებული სახით, რაც ლოგიკურად მსგავს სხვადასხვა აპლიკაციებში გამოყენების შესაძლებლობას იძლევა. ვებ-სერვისის გამოყენება აღმოფხვრავს პროგრამული კოდის განმეორებადობას (ტიპური დავალებების ავტონომიურად არსებობის თვალსაზრისით), აჩქარებს პროგრამული სისტემების შექმნის დროს და ამცირებს ცვლილებების გავლენის რისკებს.

სერვის-ორიენტირებული არქიტექტურა აფართოებს ობიექტ-ორიენტირებული ტექნოლოგიის გამოყენების საზღვრებს და ობიექტ-ორიენტირებული სისტემების ინტეგრაციის შესაძლებლობას ქმნის. მთავარი განსხვავება ამ ორ ტექნოლოგიას შორის, პრაქტიკულად, მეთოდების განზოგადებისა და წვდომის ასპექტში იკვეთება.

პრინციპული განსხვავება ობიექტ-ორიენტირებულ და სერვის-ორიენტირებულ მიდგომებს შორის არის ორიენტირება ბიზნეს-პროცესზე და არა ობიექტზე. ამ პრინციპით, ვებ-სერვისში ინკაფსულირებულია ბიზნეს-ფუნქცია აბსტრაქციის მაღალი დონით, რაც უნიფიცირებული პროცესის შაბლონად შექმნის შესაძლებლობას ქმნის. სერვის-ორიენტირებული

არქიტექტურის ერთ-ერთი პრინციპია ტიპიზირებული ვებ-სერვისების ბიბლიოთეკის არსებობა, მასში ამ შაბლონების შენახვა ვებ-სერვისის სახით და მისი მრავალჯერადად გამოყენება. ეს არის ობიექტ-ორიენტირებული ტექნოლოგიის ობიექტების ბიბლიოთეკის ანალოგია, სადაც ობიექტი წარმოდგენილია პროცესის სახით და გატანილია სერვერზე. ამავდროულად, სერვის-ორიენტირებულ არქიტექტურის პრაქტიკული რეალიზაციაა არსებული ვებ-სერვისების ნაკრებისგან (უმეტესად ობიექტ-ორიენტირებულ სისტემებში შექმნილი) კონკრეტული ბიზნეს-სცენარის აწყობა, რაც ორკესტრირებისა და ქორეოგრაფიის კონცეფციით ხორციელდება.

ორკესტრირება, პროცეს-ორიენტირებული მიდგომის თვალსაზრისით სისტემის სხვადასხვა სახის ბიზნეს-პროცესების ანუ ვებ-სერვისების კომპოზიციას, რომელთა შესრულების მიმდევრობას მართავს ბიზნეს-ლოგიკა (მიზნობრივი ფუნქცია). ორკესტრირების აღწერის ენად გამოყენებაშია XPDL და BPEL ენები. ქორეოგრაფია, სხვადასხვა პლატფორმაზე და პროგრამულ ენაზე აღწერილი ბიზნეს-პროცესების (ვებ-სერვისების) აღქმადობასა და გამოყენებას უზრუნველყოფს XML და WSDL ენების საშუალებით.

მაგალითად, მონაცემთა ბაზასა და პროგრამულ დანართს შორის ტრიგერული ფუნქციების (დამატება, განახლება, ძებნა, წაშლა) წარმოებისთვის საჭიროა მონაცემთა ბაზის მხარეს შეიქმნას პროცედურა, ხოლო პროგრამული სისტემის მხარეს დაიწეროს ამ პროცედურის წაკითხვისა და აპლიკაციასთან ურთიერთობის მეთოდები. ეს პროცესი იდენტურია ყოველი ცხრილისთვის და უფრო ზედა დონეზე ყოველი აპლიკაციისთვის, რომელიც მონაცემთა ბაზასთან მუშაობს. ობიექტ-ორიენტირებულ პლატფორმებში ამ ტიპის პროცესისთვის იქმნება ზოგადი კლასი, რომელიც მართავს მონაცემთა ბაზასა და აპლიკაციას შორის მონაცემთა მანიპულირებას ერთი პროექტის ფარგლებში (პოლიმორფიზმი). ამ პროცესის ვებ-სერვისის სახით გარდაქმნისას, უკვე შესაძლებელია ნებისმიერი პროექტისა და ნებისმიერი პლატფორმის დონეზე (XML-WSDL ენების საშუალებით) ერთხელ შექმნილი კლასის გამოყენება.

პრაქტიკული გამოყენების თვალსაზრისით ვებ-სერვისი შეიძლება შედარდეს სკრიპტულ ენაზე დაწერილ პორტლეტს, სხვადასხვა ტიპის მარტივი ფუნქციონალობის კომპონენტებს (მაგ., კალკულატორი, ინტერნეტით თანხის გადახდის ფუნქცია, აპლიკაციის მართვის ელემენტები) შინაარსობრივად უფრო გაფართოებული (ბიზნეს-ფუნქციაზე აყვანილი) და გამოსაყენებლად საკმაოდ მოქნილი (პლატფორმადამოუკიდებელი) შესაძლებლობებით.

სერვის-ორიენტირებული მიდგომა ობიექტ-ორიენტირებულ მეთოდოლოგიაზე რეალიზებული სისტემის გაფართოების ხერხია, ამდენად პროგრამული უზრუნველყოფის სისტემის მოდელირებისას ძირითადად, ისევ გამოიყენება ობიექტ-ორიენტირებული დაპროექტებისა და მოდელირების ტექნოლოგია (პროტოტიპული პრინციპი, ვერსიების მართვა და ა.შ.), როგორც სისტემის შინაარსობრივი, ისე ვებ-სერვისის ფუნქციონალური აღწერისთვის. სისტემის სერვის-ორიენტირებული არქიტექტურის მხარე აღიწერება ბიზნეს-პროცესების მოდელირების ნოტაციისა და ბიზნეს-პროცესების შესრულების ენის საფუძველზე. ბიზნეს-პროცესების მოდელირების ნოტაცია ობიექტ-ორიენტირებულ მოდელირებას აფართოებს სისტემის სერვის-ორიენტირებული ფუნქციონირების აღწერით, როგორც მრავალაპლიკაციური, ისე ჰორიზონტალური და ვერტიკალური დაპროექტების ჭრილში.

ვებ-სერვისების გამოყენებით, მრავალაპლიკაციურ რეჟიმში ბიზნეს-პროცესების წარმოების ერთიანი ელექტრონული სისტემის ფუნქციონირების თვალსაჩინოებისთვის, განვიხილავთ ფინანსთა სამინისტროს შემოსავლების სამსახურისა და საფინანსო ბანკების ინფორმაციული კომპოზიციის მაგალითს. ამ ორი მასშტაბური სტრუქტურის ფუნქციონალური ურთიერთობა იკვეთება გადასახადის გადამხდელი კლიენტის ბანკში ანგარიშის გახსნის, დახურვის, ცვლილებების, საგადასახადო დავალებების ოპერაციების წარმოების დროს.

იმ შემთხვევაში თუ ბანკი კლიენტის ანგარიშზე ახდენს გახსნა/დახურვა/ცვლილებების ოპერაციებს (გადასახადის გადამხდელი კლიენტის შემთხვევაში), ვალდებულია აცნობოს და შემდგომ მიიღოს ინფორმაცია ფინანსთა სამინისტროს შემოსავლების სამსახურიდან კლიენტის მიმდინარე ვალდებულების შესახებ. გამომდინარე მიღებული ინფორმაციიდან ბანკი ანიჭებს კლიენტის ანგარიშს შესაბამის სტატუსს, რის საფუძველზეც შესაძლებელია განხორციელდეს ან დარჩეს უცვლელად გახსნა/დახურვა/ცვლილებების ოპერაცია.

შემოსავლების სამსახურის მხრიდან საბანკო სისტემასთან ურთიერთობა აისახება სხვადასხვა ტიპის საგადასახადო დავალებების ფინანსური ოპერაციების უზრუნველსაყოფად (მაგ., ყადაღა, ინკასო და სხვ.). პერიოდულად, შემოსავლების სამსახურიდან ეგზავნება ბანკს კლიენტის ფინანსური ვალდებულების ნოტიფიკაცია (დაკისრება ან განთავისუფლება), რის საფუძველზეც ბანკი ახდენს კლიენტის ანგარიშის დაბლოკვა/განბლოკვის ოპერაციებს.

მონაცემებისა და ნოტიფიკაციების გაცვლა ამ ორ სტრუქტურას შორის წარმოებს XML ენაზე. აპლიკაციებმა უნდა უზრუნველყოს როგორც ამ ინფორმაციის მიღება, ისე სინქრონიზაცია და შედეგებზე ასახვა, მაქსიმალურად ავტომატიზებულ რეჟიმში.

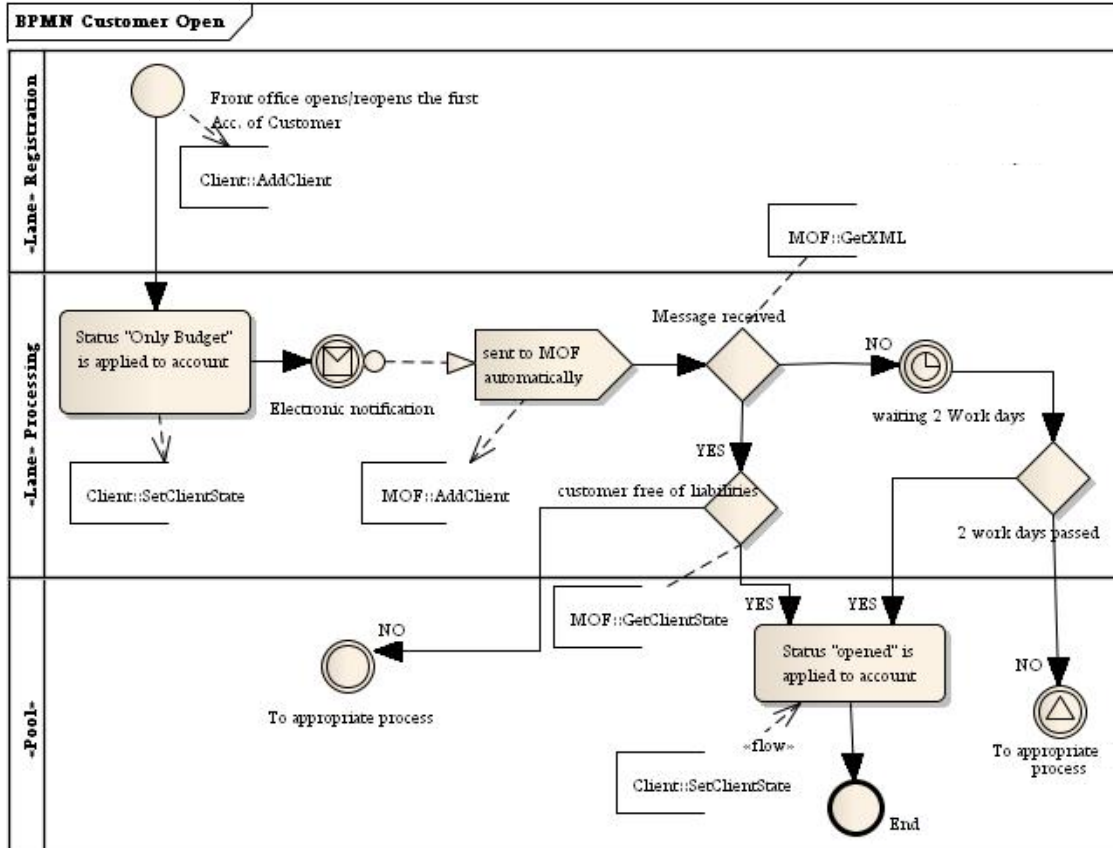
სერვის-ორიენტირებული მიდგომით, აღწერილი ბიზნეს-პროცესების სარეალიზაციოდ, გამოყენებულია შემდეგი სახის ვებ-სერვისები (ცხრ.1):

| ვებ-სერვისი | ვებ-მეთოდი                                                                                                                                                                                                                                                                                              | ცხრ.1 |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| MOF         | <b>AcceptMessageFromMOF</b> - ბანკის სერვისი, რომელსაც იძახებს შემოსავლების სამსახური. გამოიყენება ბანკებისთვის ინფორმაციის მისაწოდებლად. მონაცემები გადაეცემა XML-ის სახით, XML-შეიცავს მონაცემებს ახალი/შეცვლილი/გასაწვევი ინკასოების, ახალი/გამოსათხოვი ყადაღების, დასტურს კლიენტის გახსნის შესახებ. |       |
|             | <b>AcceptMessageFromBank</b> - შემოსავლების სამსახურის სერვისი, რომლითაც ბანკიდან მიეწოდება მონაცემები შემოსავლების სამსახურს                                                                                                                                                                           |       |
|             | <b>AddClient</b> - აგზავნის შემოსავლების სამსახურში შეტყობინებას ახალი კლიენტის ბანკში დარეგისტრირების შესახებ. ინფორმაცია იწერება დროებით XML-ში                                                                                                                                                       |       |
|             | <b>CloseClient</b> - აგზავნის შემოსავლების სამსახურში შეტყობინებას კლიენტის და მისი ყველა ანგარიშის დახურვის შესახებ                                                                                                                                                                                    |       |
|             | <b>CloseClientResponse</b> - ტექნიკური შეტყობინება შემოსავლების სამსახურიდან კლიენტის დახურვის შეტყობინებაზე                                                                                                                                                                                            |       |

|             |                                                                                                                                                                                                                                                                                                                                 |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | <b>ClientTypeChange</b> - აგზავნის შემოსავლების სამსახურში შეტყობინებას კლიენტის ტიპის შეცვლის შესახებ (ანუ თუ კლიენტი გახდება გადამხდელი/მეწარმე)                                                                                                                                                                              |
|             | <b>IsClientHonestPayer</b> - შემოსავლების სამსახურიდან მიღებული XML-იდან, ამოიღება დასტური კლიენტის გახსნის შესახებ. (თუ მასზე შემოსავლების სამსახურში არ არის რეგისტრირებული რაიმე დავალიანება ყადაღა/ინკასო)                                                                                                                  |
|             | <b>GetOrderRecalls</b> - იძახებს შემოსავლების სამსახურიდან მიღებული ინკასოების გაწვევების ან ცვლილებების შესახებ მონაცემებს (კლიენტის იდენტიფიკატორებს და ინკასოს რედაქტირებულ თანხას, რა თანხის გადარიცხვაც უნდა მოხდეს კლიენტის ანგარიშებიდან. შესაძლოა ვალდებულების თანხა განუღდეს, რაც ნიშნავს, რომ კლიენტს მოეხსნა ინკასო) |
|             | <b>RequestAmountFromMOF</b> - სისტემა ამ სერვისის საშუალებით ეკითხება შემოსავლების სამსახურს, თუ რა თანხა უნდა ჩამოჭრას კლიენტს. უგზავნის XML კლიენტების იდენტიფიკატორებით                                                                                                                                                      |
|             | <b>ReceiveApprovedAmount</b> - შემოსავლების სამსახურიდან მიღებული ინფორმაციიდან (XML), რომელიც დაბრუნდა GetXML მეთოდით, კლიენტის თანხების დამუშავება                                                                                                                                                                            |
| Client      | <b>AddClient</b> - გადაეცემა ახალი კლიენტის მონაცემები, ამატებს კლიენტს ბაზაში და აბრუნებს დამატებული კლიენტის იდენტიფიკატორს                                                                                                                                                                                                   |
|             | <b>ChangeClientStatus</b> - გადაეცემა კლიენტის იდენტიფიკატორი და კლიენტის სტატუსი გახდება „აქტიური“, „დახურული“                                                                                                                                                                                                                 |
|             | <b>ChangeType</b> - გადაეცემა კლიენტის იდენტიფიკატორი და კლიენტის ტიპი ხდება „ფიზიკური პირი“ ან „იურიდიული ტიპი“                                                                                                                                                                                                                |
| Accounts    | <b>CloseAccount</b> - გადაეცემა ანგარიშის იდენტიფიკატორი და ხდება მისი სტატუსის შეცვლა „დახურულზე“                                                                                                                                                                                                                              |
|             | <b>CheckSaldoAvailable</b> - გადაეცემა ანგარიშის იდენტიფიკატორი და აბრუნებს ინფორმაციას თუ რა თანხაა ამ ანგარიშზე                                                                                                                                                                                                               |
|             | <b>UnblockAccounts</b> - გადაეცემა კლიენტის იდენტიფიკატორი და ახდენს მისი ანგარიშების სტატუსის ცვლილებას „აქტიურზე“. კლიენტი ჩვეულებრივად შეძლებს ანგარიშების გამოყენებას                                                                                                                                                       |
|             | <b>BlockAccounts</b> - გადაეცემა კლიენტის იდენტიფიკატორი და სისტემა ასრულებს ამ კლიენტის ანგარიშების სტატუსის გადაყვანას „მხოლოდ ბიუჯეტურში“, რის შემდეგაც კლიენტი ვეღარ გამოიყენებს ამ ანგარიშებს, გარდა თანხის ბიუჯეტში გადასარიცხად                                                                                          |
| Transaction | <b>AddTransaction</b> - ახალი ტრანზაქციის დამატება სტატუსით „დაუდასტურებული“                                                                                                                                                                                                                                                    |
|             | <b>ApproveTransaction</b> - გადაეცემა ტრანზაქციის იდენტიფიკატორი და ახდენს მოცემული ტრანზაქციის სტატუსის გადაყვანას „დადასტურებული“                                                                                                                                                                                             |
|             | <b>MakeTransaction</b> - გადაეცემა ტრანზაქციის სტატუსი და ახდენს მოცემული                                                                                                                                                                                                                                                       |

|       |                                                                                                                                                              |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | ტრანზაქციის სტატუსის გადაყვანას „დასრულებული“-ში                                                                                                             |
| Cards | <b>UnblockCard</b> - პლასტიკური ბარათის დაბლოკვა ბარათების საპროცესინგო ცენტრში (ბარათზე შეიზღუდება ოპერაციების წარმოება: გადარიცხვა, თანხის გატანა და ა.შ.) |
|       | <b>BlockCard</b> - პლასტიკურ ბარათზე აღდგება ოპერაციების წარმოების შესაძლებლობა                                                                              |

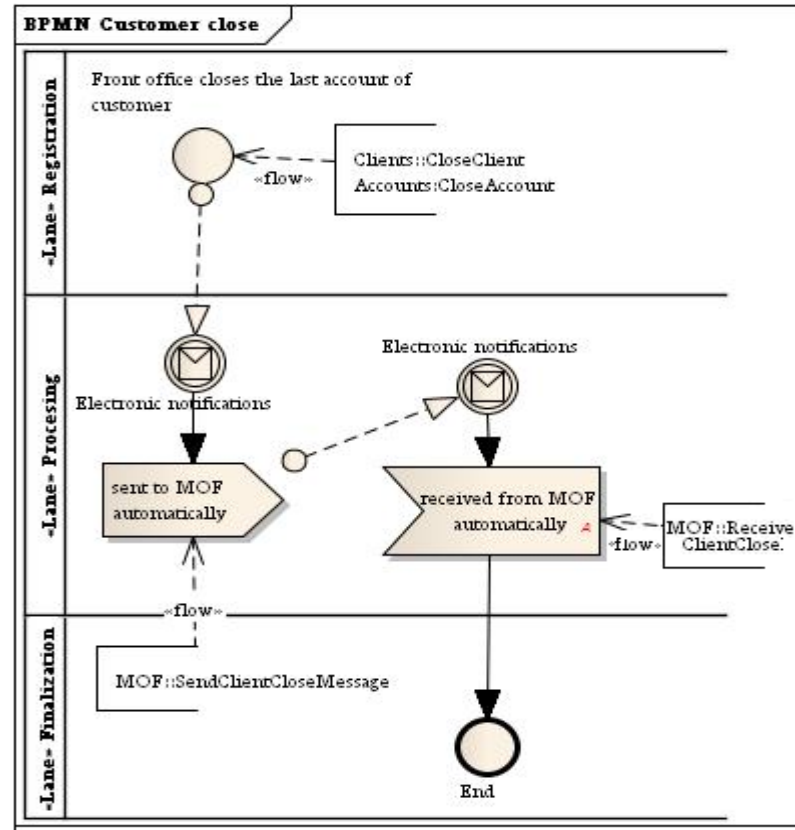
განვიხილოთ მაგალითად, კლიენტის ანგარიშის გახსნა/დახურვის ბიზნეს-პროცესის მოდელირება ვებ-სერვისის გამოყენებით (ნახ.2,3).



ნახ.2. Client Open() ვებ-სერვისებით

ბანკის კლიენტის გახსნა შესაძლებელია სხვადასხვა პროგრამის საშუალებით. საჭიროა განისაზღვროს კლიენტის პარამეტრების მინიმალური ოდენობა. იქმნება კლიენტების ვებ-სერვისი, რომელსაც გააჩნია რამდენიმე მეთოდი, კლიენტების მონაცემების სამართავად. მათ შორის არის კლიენტების შექმნის მეთოდი. მისი გამოძახება შესაძლებელია სხვადასხვა პროგრამის მიერ, დამოუკიდებლად მათი ტექნოლოგიებისა. ახალი კლიენტის დამატების მეთოდი იძახებს კლიენტების სტატუსის ცვლილების მეთოდს, რომელიც კლიენტის სტატუსს აყენებს: „მხოლოდ ბიუჯეტურზე“. ასევე იძახებს მეთოდს, რომელიც ამზადებს კლიენტის მონაცემებს შემოსავლების სამსახურში მონაცემების გასაგზავნად. ეს მონაცემები იწერება ცხრილში. შემდეგ Windows-სერვისი გარკვეული პერიოდულობით კითხულობს ახალ მონაცემებს ამ ცხრილიდან და იძახებს შემოსავლების სამსახურის ვებ-სერვისს და გადასცემს კლიენტის მონაცემებს შესამოწმებლად. შემოსავლების

სამსახური, ამ მონაცემების გადამოწმების შემდეგ, ბანკის მხარეს იძახებს ვებ-სერვის Accept\_Message, რომელსაც გადმოცემა შემოწმებული კლიენტის მონაცემები, ამ მონაცემების ჩაწერა ხდება ცხრილში და შემდეგ Windows-სერვისი მიღებული მონაცემების საფუძველზე შეუცვლის კლიენტს, ვებ-სერვისის გამოძახების საშუალებით, სტატუსს – „აქტიურზე“.



ნახ.3. Client Close() ვებ-სერვისებით

კლიენტის ანგარიშის დახურვის შემდეგ, მოწმდება დარჩა თუ არა კლიენტს ღია ანგარიშები. თუ არცერთი ანგარიში აღარაა აქტიური, ხდება კლიენტის მონაცემების ჩაწერა ცხრილში და Windows-სერვისი გადააგზავნის შემოსავლების სამსახურში შეტყობინებას კლიენტის დახურვის თაობაზე. შემოსავლების სამსახური უგზავნის დასტურს, ბანკის ვებ-სერვისის გამოძახების საშუალებით.

ვებ-სერვისების გამოყენებით, ინტერკორპორაციულ აპლიკაციებში მომქმედი გლობალური ბიზნეს-პროცესების BPMN ნოტაციით აგებული მოდელი ნაჩვენებია მე-4 ნახაზზე. მნიშვნელოვანია, რომ რამდენიმე ოპერაციის (მაგ., კლიენტის ანგარიშის გახსნა, დახურვა, ცვლილება, ყადაღის დადება, ყადაღის მოხსნა, ინსკასაციის წარმოება) წარმოებისთვის, გამოყენებაშია ერთი ტიპის ბიზნეს-პროცესები (მაგ., setclientState, blockaccount, unblockaccount), რაც სერვის-ორიენტირებული მიდგომის გამოყენებით შესაძლოა გავაერთიანოთ ერთ ვებ-სერვისში.

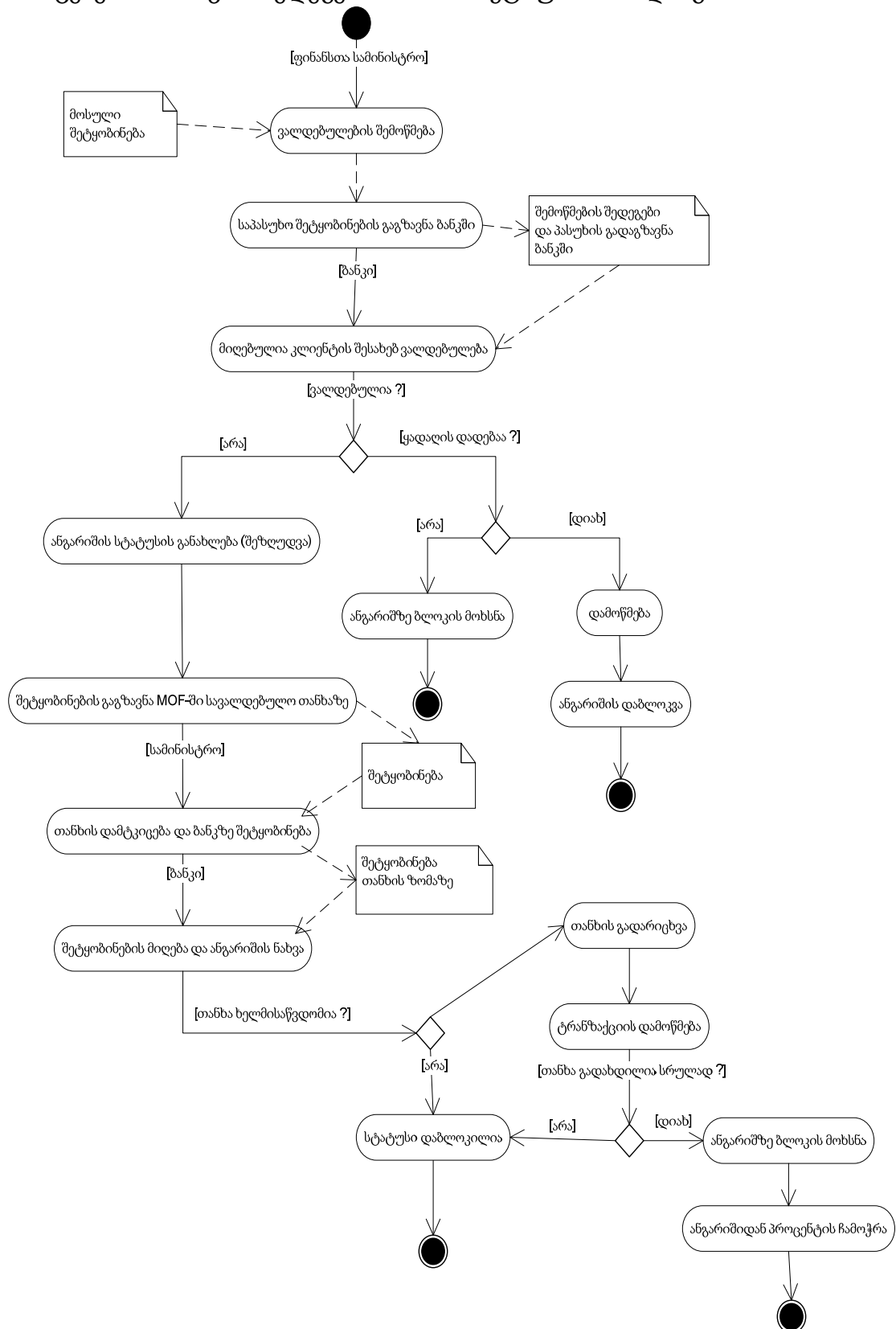
ამავდროულად, უკვე რეალიზებული ვებ-სერვისი (მაგ., კლიენტის ანგარიშის მოდიფიკაცია), შესაძლებელია ასევე გამოვიყენოთ სხვა აპლიკაციის რეალიზაციის (მაგ., კლიენტისთვის ინტერნეტ-ბანკინგის ჩართვა) ან სხვადასხვა აპლიკაციების ერთობლივი მუშაობის დროს (მაგ., ბანკის საბაჟო სტრუქტურასთან ჰორიზონტალური ინტეგრაცია).







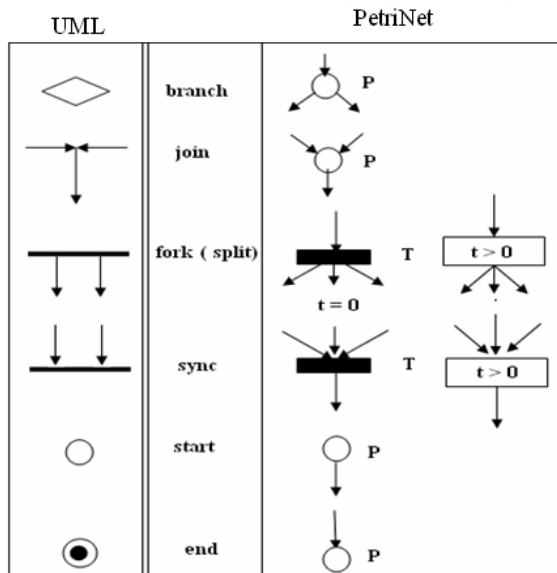
მე-5 ნახაზზე წარმოდგენილია BPMN მოდელის (ნახ.4) ბიზნეს-პროცესების ასახვის შედეგები UML-ის აქტიურობის დიაგრამით.



ნახ.5. Activity-D: კლიენტის რეგისტრაცია და ინკასაცია (ვარიანტი\_1)

არსებობს კიდევ ერთი პრობლემა, როგორიცაა ჩვენ მიერ აღწერილი ბიზნეს-პროცესების აქტიურობის დიაგრამის ეფექტურობის შეფასების შესაძლებლობა. აქტიურობათა დიაგრამა დინამიკური პროცესის აღწერის მოდელია. მას არ გააჩნია რაოდენობრივი შეფასების კრიტერიუმები და ინსტრუმენტი. თეორიაში ასეთი ამოცანები წყდება სისტემების იზომორფიზმზე დაყრდნობით, კერძოდ ისეთი მათემატიკური მოდელის საფუძველზე, როგორიცაა პეტრის დროით-სტოქასტიკური ქსელები.

საჭიროა აქტიურობის დიაგრამის (ნახ.5) შესაბამისი პეტრის ქსელის გრაფის აგება, ანუ მოდელირება და შემდეგ ამ მოდელის დინამიკის ანალიზი



დროით პარამეტრებში. აქტიურობის დიაგრამაზე გამოიყენება გრაფიკული ელემენტები: საწყისი და საბოლოო კვანძები, ქმედება, შედგენილი-ქმედება (იერარქიულად ჩადგმული პროცესი), განშტოება, შეერთება და ა.შ. ეს ელემენტები პეტრის ქსელებში მოდელირდება გრაფის პოზიციებით (Pi). დაყოფისა (fork, split) და გაერთიანების (join, sync) ელემენტები კი მოდელირდება პეტრის ქსელის გადასასვლელებით (Tj). მე-6 ნახაზზე ნაჩვენებია ეს იზომორფული ელემენტები.

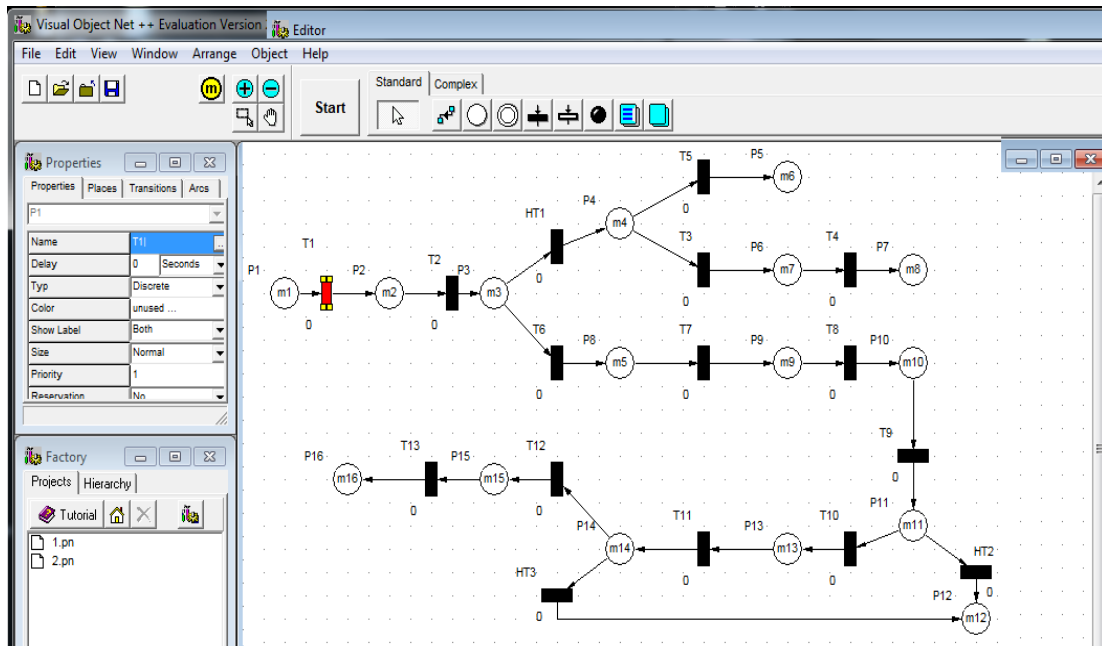
ნახ.6. UML და PetriNet იზომორფული ელემენტები

პეტრის ქსელის პოზიციებზე, გადასასვლელებზე ან/და რკალებზე დროითი დაყოვნების განსაზღვრას (მაგალითად, თითოეული ბიზნეს-პროცესის შესრულების დრო, მოთხოვნების მოსვლის ინტენსივობა და ა.შ.) დროითი პეტრის ქსელის ტიპი შემოაქვს, დაყოვნების დროთა ალბათურ განაწილებას – სტოქასტიკური პეტრის ქსელის ტიპი. ამასთანავე, პეტრის ქსელის გადასასვლელი შეიძლება იყოს ორი სახის: მყისიერად შესრულებადი (დროის დაყოვნების გარეშე) და დაყოვნებით (დროითი).

მე-7 ნახაზზე ილუსტრირებულია PetEdit რედაქტორში მე-5 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 შესაკვრელად, როდესაც მოსაზღვრეა ორი პოზიცია ან ორი გადასასვლელი.

ქვემოთ ცხრილებში მოცემულია პეტრის ქსელის პოზიციებისა (ცხრ.1) და გადასასვლელების (ცხრ.2) შინაარსობრივი მნიშვნელობები.

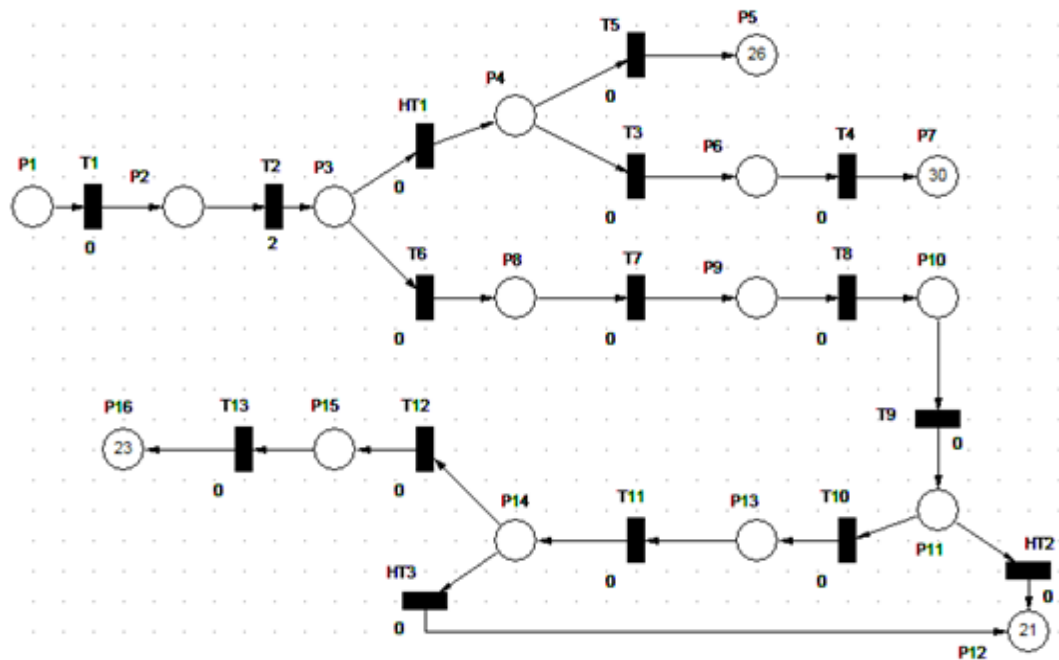
მიღებული პეტრის ქსელის ანალიზმა მისი იმიტაციური მოდელირების შესაძლებლობების ფარგლებში, როდესაც ქსელის საწყისი მდგომარეობა იყო 100 მოთხოვნით შესასვლელზე, მოგვცა საბოლოო მდგომარეობა (ყველა მოთხოვნა დამუშავდა), რომელიც ასახულია მე-8 ნახაზზე.



ნახ.7. პეტრის ქსელის გრაფის ფრაგმენტი PetEdit რედაქტორში

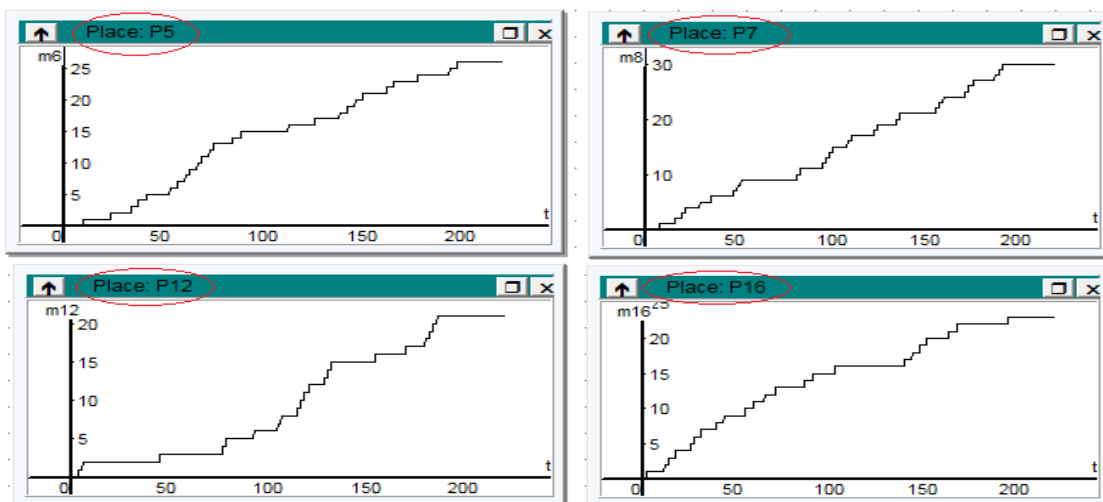
| P_N | პოზიციის დანიშნულება                                    | ცხრ.1 |
|-----|---------------------------------------------------------|-------|
| P1  | MOF-ში შემოსულია მოთხოვნა ვალდებულებებზე                |       |
| P2  | MOF -ში მიღებულია შემოწმების შედეგები                   |       |
| P3  | ბანკში მიღებულია კლიენტის შესახებ ვალდებულება           |       |
| P4  | კლიენტისთვის მიღებულია მოთხოვნა "ყადაღის" შესახებ       |       |
| P5  | „ყადაღა“ მოიხსნება ანუ კლიენტის ანგარიშზე ბლოკის მოხსნა |       |
| P6  | მოთხოვნა "ყადაღის" დადებაზე დამოწმებულია                |       |
| P7  | კლიენტის ანგარიში დაბლოკილია                            |       |
| P8  | MOF-ში შემოსულია კლიენტზე სავალდებულო თანხის მოთხოვნა   |       |
| P9  | სავალდებულო თანხა დადგენილია                            |       |
| P10 | ბანკში შემოსულია სავალდებულო თანხის ზომა                |       |
| P11 | ანგარიშზე თანხის რაოდენობა                              |       |
| P12 | "დაბლოკილი" ანგარიში                                    |       |
| P13 | გადარიცხული თანხა                                       |       |
| P14 | ანგარიშზე დარჩენილი თანხა                               |       |
| P15 | "ბლოკმოსხნილი" ანგარიში                                 |       |
| P16 | კლიენტის საბოლოო ანგარიშის მდგომარეობა                  |       |

| T_N | გადასასვლელის დანიშნულება                                               | ცხრ.2 |
|-----|-------------------------------------------------------------------------|-------|
| T1  | კლიენტის ვალდებულების შემოწმება MOF-ში                                  |       |
| T2  | შეტყობინების გადაგზავნა ბანკში კლიენტის შესახებ                         |       |
| HT1 | 1-ელი დამხმარე გადასასვლელი, როცა არის მოთხოვნა "ყადაღა"                |       |
| T3  | ბანკში კლიენტისათვის "ყადაღის" დამოწმება                                |       |
| T4  | კლიენტის ანგარიშის დაბლოკვა                                             |       |
| T5  | კლიენტის ანგარიშზე ბლოკის მოხსნა                                        |       |
| T6  | კლიენტის ანგარიშის სტატუსის განახლება (შეზღუდვა)                        |       |
| T7  | შეტყობინების გადაგზავნა MOF-ში სავალდებულო თანხაზე                      |       |
| T8  | MOF-ში თანხის დამტკიცება და ბანკზე შეტყობინება                          |       |
| T9  | შეტყობინების მიღება და კლიენტის ანგარიშის ნახვა                         |       |
| HT2 | მე-2 დამხმარე გადასასვლელი, როცა ანგარიშზე მოწმდება ხელმისაწვდომი თანხა |       |
| T10 | თანხის გადარიცხვა                                                       |       |
| T11 | ტრანზაქციის დამოწმება                                                   |       |
| T12 | ანგარიშზე ბლოკის მოხსნა                                                 |       |
| HT3 | მე-3 დამხმარე გადასასვლელი, როცა ანგარიშზე არაა სრული თანხა             |       |
| T13 | ანგარიშის პროცენტის ჩამოჭრა                                             |       |



ნახ.8. საბოლოო მდგომარეობა: ყველა მოთხოვნა შესრულდა

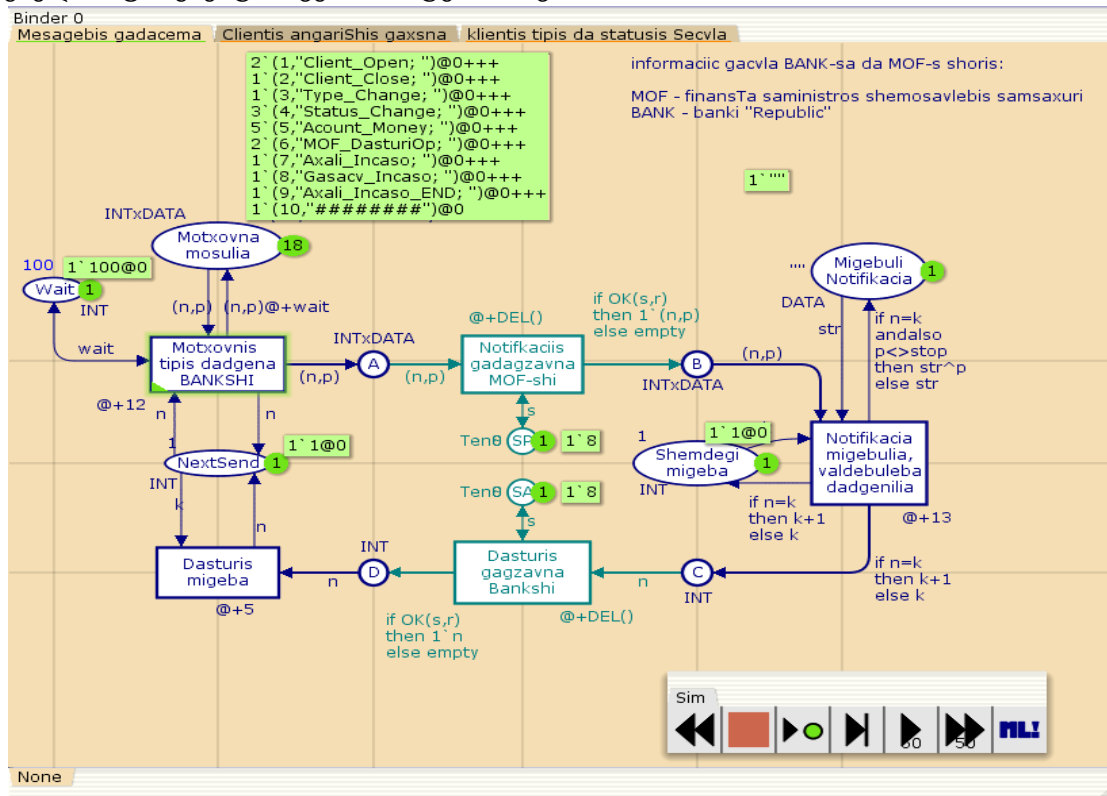
მოთხოვნის ტიპები, რომლებიც გამოიყენება საბანკო კორპორაციაში შემდეგია: კლიენტის ანგარიშის გახსნა, ანგარიშის დახურვა, კლიენტის სტატუსის შეცვლა, კლიენტის ტიპის შეცვლა, ინკასოს ბლოკის დადება/მოხსნა, ყადაღის დადება/მოხსნა, კლიენტის ანგარიშზე თანხის დადგენა, ინკასოს თანხის გადარიცხვა ბიუჯეტში და ა.შ. სისტემაში შემოსული 100 მოთხოვნა პეტრის ქსელის გრაფში მოძრაობისას გადანაწილდა სტოქასტურად (P3, P4, P11 P14 კონფლიქტურ პოზიციებში) და საბოლოო შედეგში დაფიქსირდა: P5=26 (ყადაღა მოხსნილია 26 მოთხოვნაში), P7=30 (ანგარიში დაიბლოკა 30 შემთხვევაში), P12=21 („დაბლოკილი ანგარიშის“ დატოვება 21-ჯერ, როცა არასაკმარისი თანხაა ბლოკის მოსახსნელად), P16=23 (ანგარიშის საბოლოო მდგომარეობის ნახვა 23-ჯერ, %-ის ჩამოჭრის შემდეგ) და ა.შ. მე-9 ნახაზზე გამოტანილია საკონტროლო პოზიციების (P5, P7, P12 და P16) დროითი დიაგრამები, აგებული პეტრის ქსელის სიმულატორით.



ნახ.9. P5, P7, P12 და P16 პოზიციების დროითი დიაგრამები

ახლა განვიხილოთ ინტერკორპორაციული საინფორმაციო სისტემის სერვის-აპლიკაციებს შორის ინფორმაციის გაცვლის პროცესის მოდელირების და ანალიზის საკითხები, კერძოდ საფინანსო ბანკებს და ფინანსთა სამინისტროს შემოსავლების სამსახურს შორის სერვის-ორიენტირებული არქიტექტურის ბაზაზე. ასეთი ქსელის ანალიზისათვის ვიყენებთ იმიტაციური მოდელირების ფერადი პეტრის ქსელის, CPN (Colored Petri Net) ინსტრუმენტს. საკვლევი სისტემის მოდელის ასაგებად საჭირო მოთხოვნილებათა განსაზღვრა მოხდა სერვისების ბიზნეს-პროცესების და ბიზნეს-წესების შესწავლის საფუძველზე და შესაბამის აქტიურობათა დიაგრამების აგებით. სერვის-ფუნქციები, რომლებიც განთავსებულია შემოსავლების სამსახურისა და საფინანსო ბანკების სერვერებზე, შეიძლება წარმოვადგინოთ ობიექტ-ორიენტირებული დაპროგრამების საფუძველზე როგორც კლასები და მათი მეთოდები (ცხრ.1).

როგორც ბიზნეს-პროცესების ანალიზმა გვიჩვენა, საფინანსო ბანკებსა და შემოსავლების სამსახურს შორის, როგორც ინტერკორპორაციული ორგანიზმისთვის, დამახასიათებელია შეტყობინებათა და მონაცემთა გაცვლის სერვისების ხშირი გამოყენება (ყოველდღიურად ბანკმა შეიძლება მიიღოს (ან გადასცეს) 1000-ზე მეტი მოთხოვნა კლიენტების შესახებ). ასეთი ინფორმაციის მენეჯმენტი მოითხოვს საიმედო აღრიცხვისა და რისკების გამორიცხვის პროცედურების გათვალისწინებას. შეტყობინებათა ერთობლიობა, რომელიც მუდმივად გადაიცემა ქსელის საშუალებით, არ უნდა დაიკარგოს და ყოველი მათგანი უნდა ექვემდებარებოდეს მკაცრ კონტროლს, უნდა შეიძლებოდეს აღდგენა, ანუ განმეორებითი პროცედურის შესრულება. კორპორაციებს შორის ასეთი სერვისული მოთხოვნების დამუშავების მართვის პროცესის მოდელირება ჩატარდა CPN ინსტრუმენტით. მე-10 ნახაზზე მოცემულია ასეთი ქსელის ფრაგმენტი ჩვენი სისტემისათვის.



ნახ.10. MOF-BANK კორპორაციული კავშირების პროცესების იმიტაციური მოდელი CPN -ის გარემოში: „მოთხოვნების დამუშავება“

აქ გადასასვლელ ბლოკებში ნაჩვენებია, მაგალითად, *მოთხოვნის ტიპის დადგენა ბანკში*, ნოტიფიკაციის გადაგზავნა MOF-შემოსავლების სამსახურში, ნოტიფიკაცია მიღებულია და ვალდებულება დადგენილია, MOF-დან დასტურის გადაგზავნა ბანკში, დასტურის მიღება ბანკში. თითოეული მათგანი უნდა გაიშალოს დამოუკიდებელი პეტრის ქსელით და მოხდეს მათი ანალიზი, ამასთანავე შეიქმნება ერთიანი იერარქიული სისტემა ჩადგმული პეტრის ქვექსელებით. CPN-ინსტრუმენტი იყენებს ობიექტ-ორიენტირებული, ვიზუალური დაპროგრამების პრინციპებს, მისი ენა ML საშუალებას იძლევა აღწეროს ქსელის ფერადი კომპონენტები (მარკერები), ცვლადები, კონსტანტები და თვით პოზიციების, გადასასვლელებისა და რკალების ტექსტური აღწერები, რაც ერთგვარ კომფორტს ქმნის ქსელის წასაკითხად და გასაგებად.

ქსელის ყოველი პოზიციის გვერდით შეიძლება აისახოს მოცემულ მომენტში შემავალი ფერადი მარკერები. საინიციალიზაციო მარკირება ხაზგასმული ტექსტის სახით გამოითქმება. მაგალითად, საწყის მდგომარეობაში პოზიცია „მოთხოვნა მოსულია“ შეიცავს INTxDATA ტიპის ფერად მარკერთა 9-ელემენტიან სიმრავლეს (საინიციალიზაციო მარკირება):

{ 2'(1, 'კლიენტის\_გახსნა'), 1'(2, 'კლიენტის დახურვა'), 1'(3, „ტიპის ცვლილება“), 3'(4, „სტატუსის ცვლილება“), 5'(5, „ანგარიშზე თანხა“), 2'(6, „MOF-დან დასტური“) და ა.შ. }. აქ ბოლო, მე-10 ელემენტი შეესაბამება დასასრულის იდენტიფიკაციას - stop.

პირველი რიცხვი სტრიქონში: კოფიციენტია, რომელიც მიუთითებს, რომ პოზიციაში არის არაუმეტეს 1 ცალი მოცემული ფერის მონაცემი (ანუ არსებობს მხოლოდ ერთი მოთხოვნა ნომრით „ტიპის ცვლილება“, რომლის ფერია - რიგითი ნომერი 3). ამ შემთხვევაში გვაქვს მონაცემთა ელემენტების სიმრავლე.

მეორე მაგალითი, პოზიცია „შემოსული\_მოთხოვნები“ შედგება 18 ელემენტისგან (2+1+1+3+5+2+1+1+1), რომლებიც 9 სხვადასხვა (მარკერების ფერის) მოთხოვნათა ტიპების რაოდენობას, ანუ მულტისიმრავლეს ასახავს.

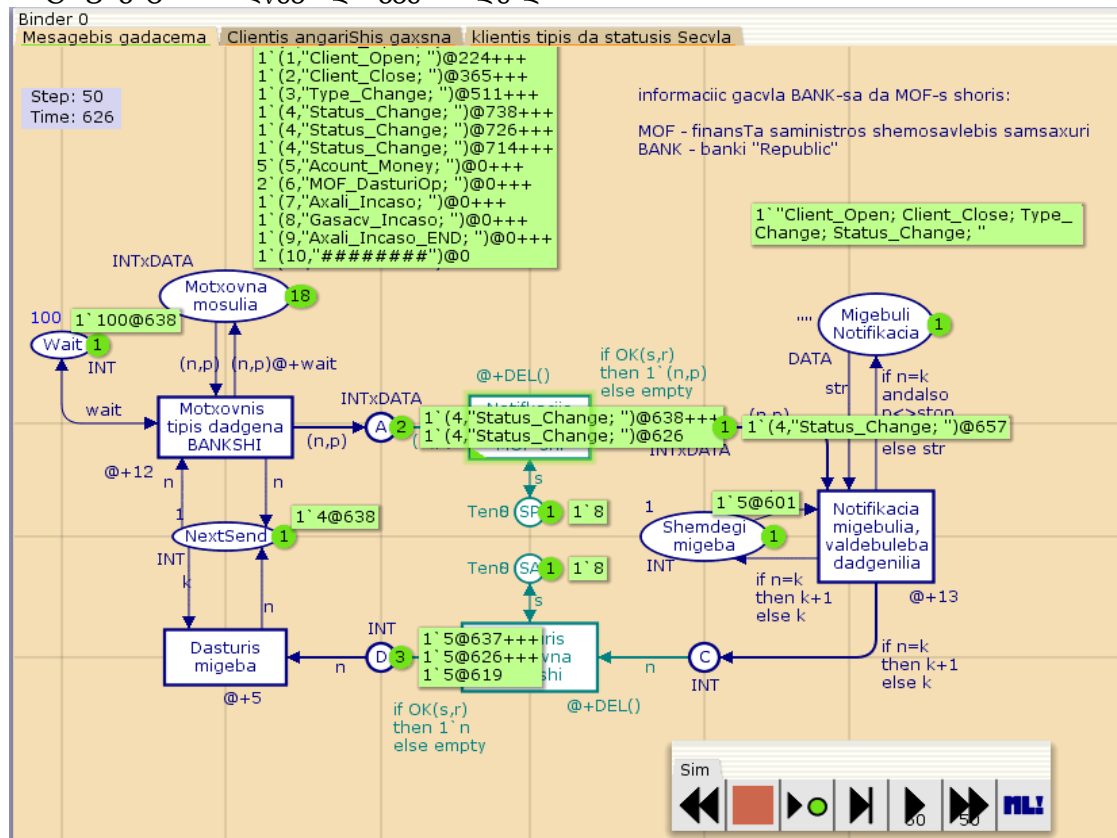
პროცესების შესრულების დრო (დაყოვნება) აისახება გადასასვლელთან სიმბოლოს და დროის ერთეულის (მაგალითად, @+7, @+wait) მითითებით, სადაც wait წინასწარ განსაზღვრული კონსტანტაა.

ამავე ნახაზზე ასახულია არადეტერმინირებული ლოგიკური გამოსახულება (პირობის ბლოკი) ფერადი პეტრის ქსელის რკალებზე, რომლებიც გადასასვლელთა გაშვების სხვადასხვა პირობებს და შედეგებს ასახავს, ანუ ლოგიკური პირობის ჭეშმარიტებისას გადასასვლელს განსხვავებული მნიშვნელობა მიეწოდება (ან გადასასვლელიდან განსხვავებული მნიშვნელობა გამოვა), მცდარობისას – განსხვავებული. მაგალითად, გადასასვლელს „ნოტიფიკაციის გადაგზავნა MOF-ში“ გამოსასვლელ რკალზე აქვს ლოგიკური პირობა - თუ გამოგზავნილი ნოტიფიკაციის ნომერი (n) ემთხვევა კლიენტის ანგარიშის ნომერს (k), მაშინ გვაქვს „true“, წინააღმდეგ შემთხვევაში „false“, რაც იმას ნიშნავს, რომ საჭირო ნოტიფიკაცია არაა მისული MOF-ში. თუ ყველაფერი წესრიგშია, მაშინ MOF (მიმღები) უგზავნის ბანკს შეტყობინებას გადასასვლელით „დასტურის\_გამოგზავნა“. ნოტიფიკაციის და შეტყობინების გადაცემათა ქსელში შემთხვევითი პროცესის არსებობა განპირობებულია დაყოვნების ცვლადი დროის გამო, რაც აისახება colset NetDelay=int with 25..75, fun DEL( ) =NetDelay.ran( ) random-ფუნქციით. ლოგიკური პირობის მნიშვნელობა სხვადასხვა შემთხვევებში სხვადასხვანაირად განისაზღვრება. ინტერაქტიულ სიმულატორებში ჭეშმარიტება-მცდარობას თავად



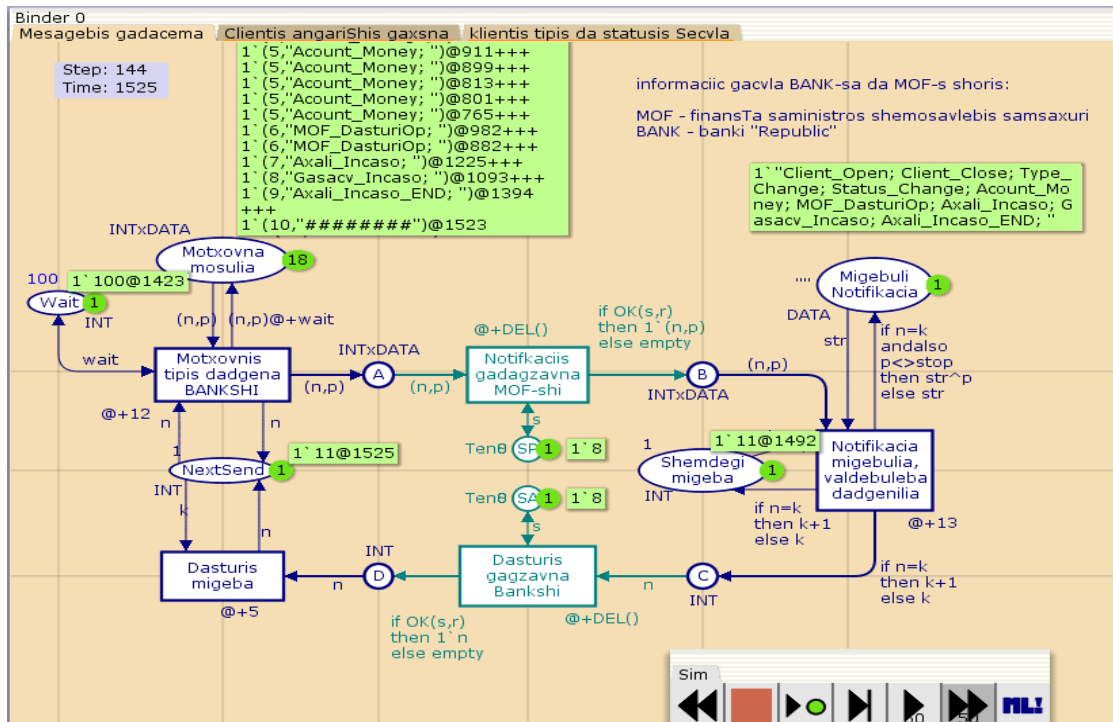
მომხმარებელი განსაზღვრავს, ავტომატური სიმულაციისას – შემთხვევით სიდიდეთა გენერატორი.

მე-11 ნახაზზე ნაჩვენებია გვაქვს ჩვენი ამოცანის პეტრის ქსელის ფრაგმენტი 50-ე ბიჯის და პროცესის დასრულების შემდეგ. ჩანს მარკერების შეცვლილი მდგომარეობა. თავიდან გაიშვება გადასასვლელი „მოთხოვნის ტიპის დადგენა ბანკში“ (ნახ.10, გადასასვლელი გააქტიურებულია - მწვანე ფერის ჩარჩო), ვინაიდან მის შესასვლელ პოზიციაში „მოთხოვნა\_მოსულია“ მზადაა მარკერები. ესაა სიგნალი იმის შესახებ, რომ 1-ელი მოთხოვნით გათვალისწინებული შეტყობინება გაიშვება ქსელში „ნოტიფიკაციის გადაცემა\_MOF-ში“. გააქტიურდება ეს გადასასვლელი და მარკერი გადავა „გაგზავნის“ A-პოზიციაში (n=1, p=„Motxovna\_1“). ტრანსპორტირების გარკვეული დროის შემდეგ (სტოქასტიკური დრო: @+DEL( )) ნოტიფიკაცია მიაღწევს დამკვეთამდე და ა.შ.



ნახ.11. იმიტაციური მოდელირების შუალედური ეტაპი (ბიჯი=50)

ჩვენი დროითი CPN -მოდელით შეიძლება გამოვიკვლიოთ კორპორაციათა შორის (მაგალითად, ბანკსა და შემოსავლების სამსახურს შორის) შეტყობინებათა გაცვლის პროცესის მახასიათებლები. შეტყობინებათა განმეორებითი გადაცემის დაყოვნების დროის (wait) სხვადასხვა მნიშვნელობისათვის. ხანმოკლე დაყოვნება ზრდის შანსს განმეორებითი გადაგზავნების თავიდან ასაცილებლად. იგი ასევე ზრდის შანსს, რომ ოპერაცია Dasturis\_migebe გადაიდოს, რადგან პროცესი Motxovnis tipis dadgena bankshi დაკავებულია განმეორებითი გადაგზავნით. გრძელი დაყოვნება ნიშნავს, რომ საჭირო იქნება დიდხანს ცდა, სანამ ბანკი დარწმუნდება, რომ შეტყობინება ან დასტური იქნა დაკარგული. სიმულაციის პროცესში, სხვადასხვა wait-მნიშვნელობით შეიძლება დადგინდეს ოპტიმალური მნიშვნელობა განმეორებითი გადაცემის დაყოვნებისათვის. მე-12 ნახაზზე პროცესი დასრულებულია.



ნახ.12. იმიტაციური მოდელირების დასასრული (ბიჯი=144)

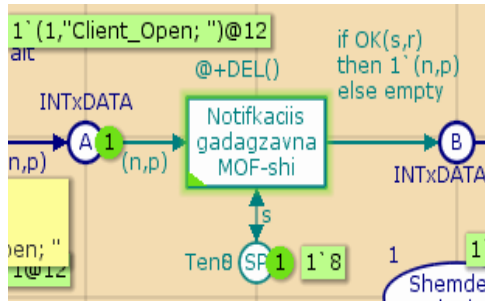
როგორც აღვნიშნეთ, არაა გამორიცხული შემთხვევები, რომ შეტყობინება ვერ მივიდეს დროულად დანიშნულების ადგილას (გარკვეული ობიექტურ-სუბიექტური მიზეზების გამო), ან დაიკარგოს დასტურის შეტყობინება. ასეთ შემთხვევებში საჭიროა ინფორმაციის დროულად გამოკვლევა და არშესრულებული პროცედურის გამეორება.

პეტრის ქსელის გადასასვლელები, როგორებიცაა Motxovnis\_tipis\_dadgena\_bankshi, Notifikaciis\_gadagzavna\_MOF-shi, Notifikacia\_migebulia\_valdebuleba\_dadgenilia, dasturis\_gagzavna\_bankshi და ა.შ. ხასიათდება დროითი დაყოვნებით, რომლებიც ან კონსტანტური მნიშვნელობისაა, ან შემთხვევითი რიცხვების დიაპაზონიდან აიღება სისტემის მიერ. ამგვარად, CPN-ინსტრუმენტით შესაძლებელია მდგომარეობათა სივრცის ანგარიშის მთლიანი პროცესის სრული ავტომატიზაცია, რაც მნიშვნელოვნად აჩქარებს ქსელის დიაგნოსტიკის პროცესს მისი რეალურ ობიექტთან ადეკვატურობის შესახებ, ანუ რამდენად სწორად ასახავს მოდელი რეალური ობიექტის ყოფაქცევას.

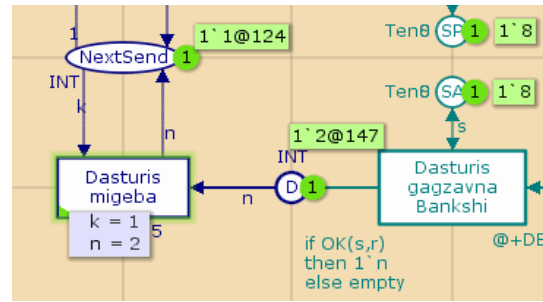
მდგომარეობათა სრული სივრცე – ორიენტირებული გრაფით აისახება, რომელშიც მწვერვალები შეესაბამება ქსელის დასაშვებ მარკირებებს, ხოლო რკალები – მოვლენებს დამაკავშირებელი ელემენტებით. ე.ი. M1 მდგომარეობიდან (მარკირებიდან) სისტემა გადადის M2 მდგომარეობაში, როდესაც არსებობს რკალი დამაკავშირებელი (n,p)-ელემენტით, სადაც n-ფერადი მარკერია, ხოლო p-ინფორმაციული ნაწილი. პეტრის ქსელის ახალ მარკირებისთვის პირველი ბიჯის შემდეგ A-პოზიციაში გაჩნდა ახალი, 1-მარკერი, რომლის ფერი=1, მონაცემი="Client\_Open". ამასთანავე ეს მარკერი მოვიდა ქსელის ამუშავებიდან t=12 დროითი ერთეულის (მაგ., წუთი) შემდეგ (ვინაიდან Motxovnis\_tipis\_dadgena\_bankshi გადასასვლელის დროითი დაყოვნებაა @+12). ახლა გააქტიურდა Notifikaciis\_gadagzavna გადასასვლელი და შესაძლებელია ასევე Motxovnis\_tipis\_dadgena\_bankshi გადასასვლელის



ხელახალი გაშვებაც. ეს ორივე პროცესი შეიძლება შესრულდეს პარალელურად, ისინი ერთმანეთს ხელს არ უშლის.



ნახ.13



ნახ.14

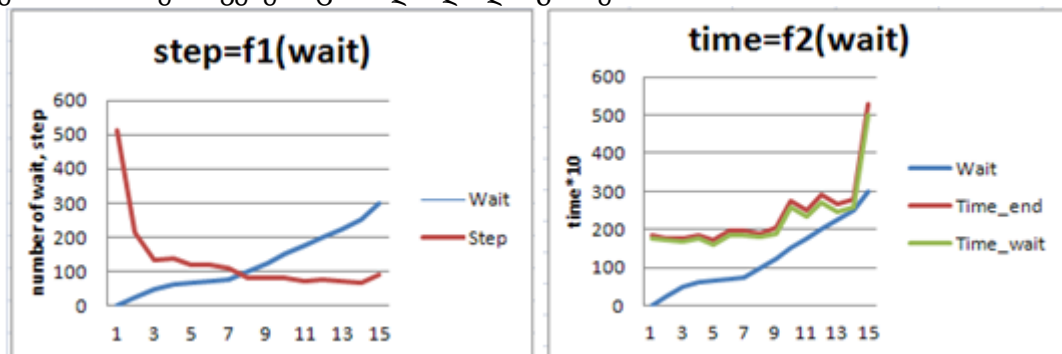
Notifikaciis\_gadagzavna გადასასვლელიდან B-პოზიციაში შემავალი რკალი აკონტროლებს ლოგიკურ პირობას, ანუ დასაშვებია ორი შემთხვევა (ნახ.13):

TP + =( Notifikaciis\_gadagzavna, <n=1,p="Client\_Open", success=true>),  
 TP - = ( Notifikaciis\_gadagzavna, <n=1,p=" Client\_Open", success=false>).

ეს ორი დამაკავშირებელი ელემენტი TP+ და TP- იმყოფება კონფლიქტში ერთმანეთთან, ანუ ერთის შესრულება მეორეს გამორიცხავს. პირველით მოდელირდება ქსელში ნოტიფიკაციის წარმატებით გადაცემა, ხოლო მეორეთი კი – ამ შეტყობინების დაკარგვა.

მარკერი გადადის D-პოზიციაში და გააქტიურდება Dasturis\_Migebe გადასასვლელი, რაც იმის მაუწყებელია, რომ ბანკის ამ ნოტიფიკაციაზე მოვიდა პასუხი MOF-დან (ნახ.14). შესაბამისად, ბანკი გააგრძელებს დასტურით მიღებული ინფორმაციის საფუძველზე მოქმედებას. მაგალითად, თუ ამ კლიენტს აქვს “ვალდებულება”, მაშინ მას უხსნიან ანგარიშს, ოღონდ სტატუსით “მხოლოდ ბიუჯეტური”. თუ არ აქვს ვალდებულება, მაშინ – “ღია” სტატუსით, და ა.შ. ქსელში გრძელდება სხვა მოთხოვნების (ნოტიფიკაციების) დამუშავება.

პეტრის ქსელის იმიტაციური მოდელის საფუძველზე (ნახ.12) პროცესების კვლევის შედეგად, როდესაც ცვლით მოთხოვნების ანალიზის და შესრულების დასაწყისის დაყოვნების დროს (wait - პოზიცია) ინტერვალში [0-:300], მივიღეთ მე-15 ნახაზზე ნაჩვენები ცხრილი და დიაგრამები.



ნახ.15

შემდეგ ეტაპზე ვიხილავთ კორპორაციული მართვის საინფორმაციო სისტემის მოზნაცემთა ბაზების დამუშავების საკითხს. კერძოდ, კონცეპტუალური მოდელირების ახალი ტექნოლოგიის სახით ვიყენებთ ობიექტ-როლურ მოდელირებას (ORM). იგი პრაქტიკულად სემანტიკური მოდელირების ინსტრუმენტია ფაქტებზე დაყრდნობით. ბუნებრივი ენის და ინტუიციური დიაგრამების (რომელთა შევსებაც ხდება მაგალითებით) გამოყენება და ასევე საპრობლემო სფეროს აღწერა ელემენტარული ფაქტების საფუძველზე,

საგრძნობლად ამარტივებს მოდელის აგების და დაპროექტების პროცესს. კონცეპტუალური მოდელის ავტომატური დაპროექტებისათვის წარმოდგენილია ობიექტ-როლური მოდელირების მეთოდი (ORM), Natural ORM Architect მოდელირების ინსტრუმენტი. თავდაპირველად ხდება საპრობლემო არის, კორპორაციათაშორის არსებული სისტემის მოთხოვნილებათა ანალიზი და ტექნიკური დავალების განსაზღვრა. არსებული ბიზნეს-პროცესებისა და ბიზნეს-წესების საფუძველზე ჩამოყალიბდება ფაქტები, ანუ იმ კანონზომიერებათა ერთობლიობა, რომელიც უნდა აისახოს მონაცემთა ბაზის მოდელში. ამგვარად, მიღებული სემანტიკური ფაქტების სიმრავლის ფიზიკური გადატანით ავტომატიზებული დაპროექტების NORMA ინსტრუმენტით Ms Visual Studio.NET გარემოში, განისაზღვრება ჩვენი სისტემის ობიექტ-როლური მოდელი, რომლის მიხედვითაც შემდგომ აიგება ERM-დიაგრამა.

საფინანსო ბანკების და შემოსავლების სამსახურის კორპორაციის ურთიერთობების საპრობლემო სფერო აღიწერება, მაგალითად, შემდეგი სახის ინფორმაციით: „ბანკში კლიენტის რეგისტრაცია ხდება ოპერატორის მიერ, ”კლიენტების მართვის“ პროგრამის საშუალებით. ახალ კლიენტს ენიჭება სტატუსი „მხოლოდ ბიუჯეტური“. ამ კლიენტის შესახებ შეტყობინება ეგზავნება შემოსავლების სამსახურს. შემოსავლების სამსახური ამოწმებს კლიენტის მონაცემებს. კერძოდ, განსაზღვრავს აქვს თუ არა ამ კლიენტს რაიმე ვალდებულება (დავალიანება ბიუჯეტის მიმართ ან სხვ.) და შემდეგ ბანკს უგზავნის შესაბამის საპასუხო შეტყობინებას. ამ შეტყობინების შესაბამისად ხდება კლიენტის სტატუსის შეცვლა „აქტიურზე“ (თუ არ აქვს ვალდებულება).

კლიენტის ანგარიშებზე ინკასოს (ან ყადაღას) წარმოდგენისთვის შემოსავლების სამსახური აფორმირებს შეტყობინებას, რომელიც შეიცავს კლიენტის მონაცემებს და თანხის ოდენობას. ეს შეტყობინება ეგზავნება ბანკის ვებ-სერვისს. ოპერატორი ინკასოების (ან ყადაღების) მართვის მოდულში ნახულობს შემოსული ინკასოების სიას და ამოწმებს მათ. დადასტურებულ ინკასოებზე Windows სერვისი კლიენტის ანგარიშებს ანიჭებს „მხოლოდ ბიუჯეტურ“ სტატუსს და ა.შ. ელემენტარული ფაქტების ერთობლიობა შეიძლება ასე წარმოვადგინოთ: f1- კლიენტს აქვს გვარი; f2 - კლიენტს აქვს ანგარიში ; f3-კლიენტს აქვს ტიპი; f4-ანგარიშს აქვს ანგარიშის ნომერი ; f5-ანგარიშს აქვს ვალუტა; f6-კლიენტს აქვს ინკასო; f7-ინკასოს აქვს ინკასოს სტატუსი; f8-ანგარიშებს აქვს ანგარიშების სტატუსი; f9-ანგარიშს აქვს ფილიალი; f10-კლიენტს აქვს სტატუსი; f11-ანგარიშიდან ხდება ინკასოში მითითებული თანხის მოხსნა და ა.შ.

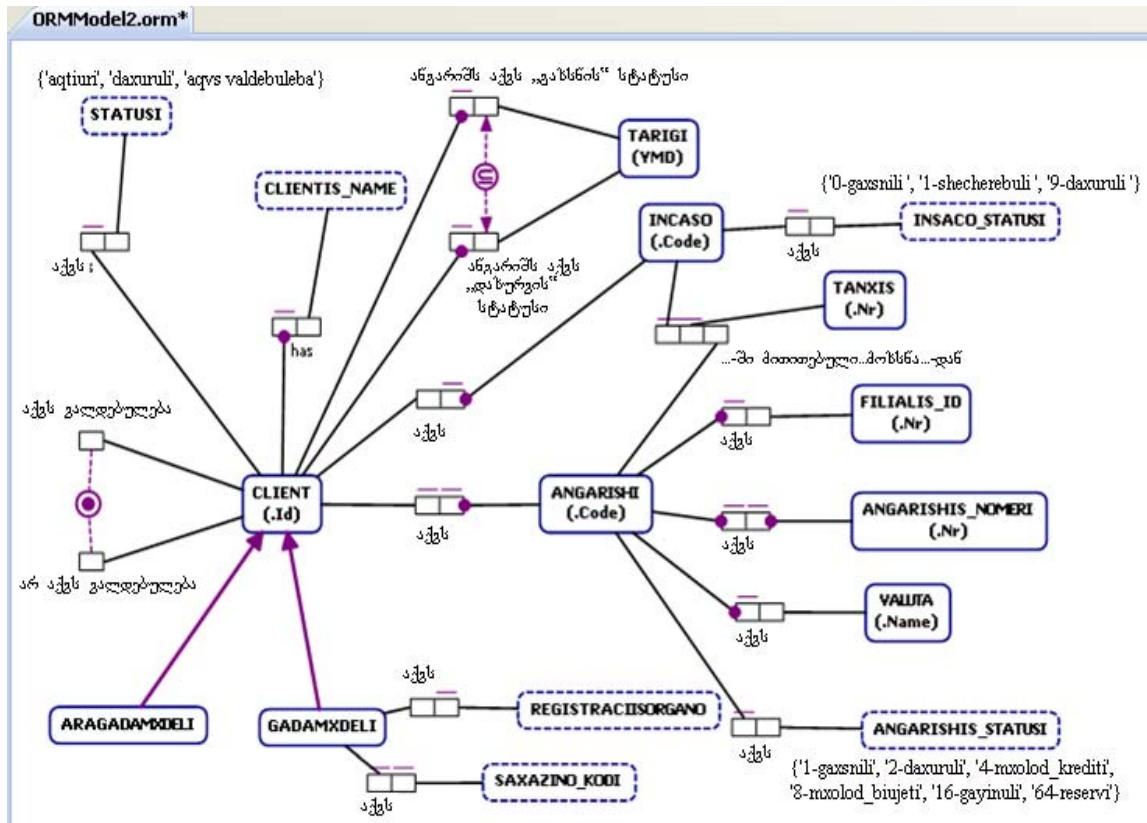
ამ ფაქტების საფუძველზე NORMA პროგრამული ინსტრუმენტით აგებული შესაბამისი ORM-დიაგრამა წარმოდგენილია მე-16 ნახაზზე.

ORM-დიაგრამაზე ობიექტები გამოსახულია ელიფსებში, წყვეტილი ელიფსი აღნიშნავს მნიშვნელობას (value), პრედიკატები გამოსახულია მართკუთხედებში. გამოყენებულია ბინარული და ტერნარული პრედიკატები. შეზღუდვები ნახაზზე გამოსახულია სპეც-აღნიშვნებით.

ჩვენ მიერ პროგრამულ ინსტრუმენტში შეტანილ ფაქტებს ობიექტებისა და მათი სემანტიკური კავშირების შესახებ აქვს მე-4 ცხრილში ნაჩვენები სახე.

მე-17 ნახაზზე ჩანს სისტემის მიერ აგებული პრედიკატები.

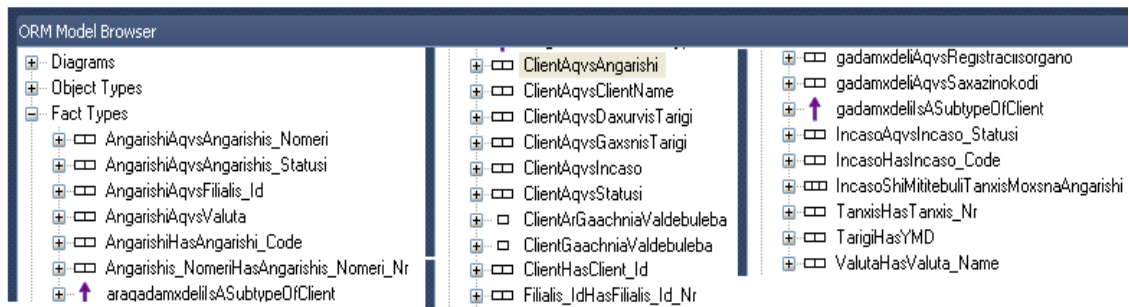
MsVisualStudio.Net გარემოში NORMA პროგრამული პაკეტის საშუალებით ORM-დიაგრამიდან ავტომატურად ვღებულობთ ERM არსთა დამოკიდებულებათა მოდელს (ნახ.18).



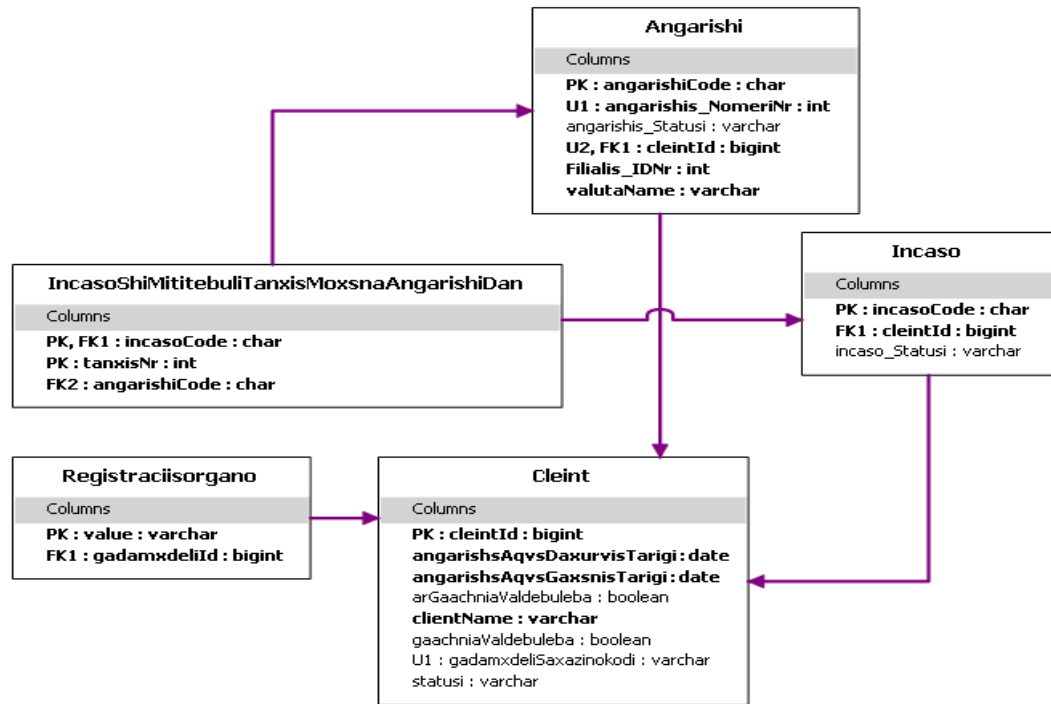
ნახ.16. ORM-მოდელის ფრაგმენტი

ცხრ.4

|                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Fact Types:</b><br/>         Client aqvs ClientName.<br/> <b>Each</b> Client aqvs <b>exactly one</b> ClientName.<br/> <b>It is possible that more than one</b> Client aqvs <b>the same</b> ClientName.<br/>         Gadamxdeli <b>is an entity type</b>.<br/> <b>Reference Scheme:</b> Client has Client_Id.<br/> <b>Reference Mode:</b> .Id.</p> | <p><b>Fact Types:</b><br/>         Gadamxdeli aqvs Registraciisorgano.<br/> <b>For each</b> Registraciisorgano, <b>at most one</b> gadamxdeli aqvs <b>that</b> Registraciisorgano.<br/> <b>It is possible that the same</b> gadamxdeli aqvs <b>more than one</b> Registraciisorgano.<br/>         Tarigi <b>is an entity type</b>.<br/> <b>Reference Scheme:</b> Tarigi has YMD.<br/> <b>Reference Mode:</b> YMD.</p>                                                                  |
| <p><b>Fact Types:</b><br/> <b>Each</b> gadamxdeli <b>is an instance of</b> Client.<br/>         Gadamxdeli aqvs Saxazinokodi.<br/>         Gadamxdeli aqvs Registraciisorgano.<br/>         Aragadamxdeli <b>is an entity type</b>.<br/> <b>Reference Scheme:</b> Client has Client_Id.<br/> <b>Reference Mode:</b> .Id.</p>                            | <p><b>Fact Types:</b><br/>         Gadamxdeli aqvs Saxazinokodi.<br/> <b>Each</b> gadamxdeli aqvs <b>at most one</b> Saxazinokodi.<br/> <b>For each</b> Saxazinokodi, <b>at most one</b> gadamxdeli aqvs <b>that</b> Saxazinokodi.<br/>         Registraciisorgano <b>is a value type</b>.<br/> <b>Portable data type:</b> Text: Variable Length.</p>                                                                                                                                  |
| <p><b>Fact Types:</b><br/> <b>Each</b> aragadamxdeli <b>is an instance of</b> Client.<br/>         Saxazinokodi <b>is a value type</b>.<br/> <b>Portable data type:</b> Text: Variable Length.</p>                                                                                                                                                      | <p><b>Fact Types:</b><br/>         Filialis_Id has Filialis_Id_Nr.<br/>         Angarishi aqvs Filialis_Id.<br/>         Angarishi aqvs Filialis_Id.<br/> <b>Each</b> Angarishi aqvs <b>exactly one</b> Filialis_Id.<br/> <b>It is possible that more than one</b> Angarishi aqvs <b>the same</b> Filialis_Id.<br/>         Angarishis_Nomeri <b>is an entity type</b>.<br/> <b>Reference Scheme:</b> Angarishis_Nomeri has Angarishis_Nomeri_Nr.<br/> <b>Reference Mode:</b> .Nr.</p> |
| <p>და ა.შ.</p>                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |



ნახ.17



ნახ.18. ERM-დიაგრამის ფრაგმენტი

შემდეგ MsVisualStudio.Net გარემოში სისტემის დახმარებით ავტომატურად ვაგენერირებთ DDL–ფაილს, რომლის ფრაგმენტი მოცემულია ქვემოთ:

— ERM დიაგრამის შესაბამისი DDL ფაილის ტექსტი -----

```
CREATE SCHEMA ORMModel1
GO
CREATE TABLE ORMModel1.Cleint
(
    cleintId BIGINT CHECK (cleintId IN ('aqturi', 'daxuruli', 'aqvs
valdebuleba'))
NOT NULL,
    clientName NATIONAL CHARACTER VARYING(MAX) NOT NULL,
    aqvsDaxurvisTarigi DATETIME NOT NULL,
    aqvsGaxsnisTarigi DATETIME NOT NULL,
    gadamxdeliSaxazinokodi NATIONAL CHARACTER VARYING(MAX),
    statusi NATIONAL CHARACTER VARYING(MAX) CHECK (statusi IN ('aqturi',
'daxuruli', 'aqvs valdebuleba')),
    gaachniaValdebuleba BIT,
    arGaachniaValdebuleba BIT,
    CONSTRAINT Cleint_PK PRIMARY KEY(cleintId)
)
GO
CREATE TABLE ORMModel1.Angarishebi
(
    angarishebiCode NATIONAL CHARACTER(4000) NOT NULL,
    angarishis_NomeriNr INTEGER NOT NULL,
```

```
cleintId BIGINT CHECK (cleintId IN ('aqturi', 'daxuruli', 'aqvs  
valdebulba'))  
. . . და ა.შ.
```

ORM2 ნოტაცია MsVisualStudio.Net პაკეტის საშუალებით ითვალისწინებს რევერსული დაპროექტების პროცესსაც. შესაძლებელია ERM მოდელიდან აღვადგინოთ ORM მოდელი და გამოვიკვლიოთ მათი სინტაქსურ-სემანტიკური სისრულე. შემდეგ შევადაროთ მიღებული და ადრე დაპროექტებული დიაგრამები ერთმანეთს. ექსპერტული შეფასებების საფუძველზე შესაძლებელი იქნება ცდომილებების მიგნება და შემდეგ მათი კორექტირება.

**მესამე თავში** გადმოცემულია Web-აპლიკაციების პროგრამული რეალიზაციის მაგალითები, ექსპერიმენტული სადემონსტრაციო ელექტრონული ბიზნესის სისტემები. კერძოდ, განხილულია საფინანსო-საბანკო სისტემები, სავაჭრო-კომერციული ცენტრების, პროდუქციის შესყიდვა-მიწოდების კონტრაქტების რეალიზაციის სისტემები და ა.შ. ელექტრონული ბიზნესის მომხმარებელთა ინტერფეისების პროგრამული გადაწყვეტიტ Ms Visual Studio.NET პლატფორმაზე და ASP.NET, ADO.NET, C#, XML, Ms SQL Server, BizTalk, Data Warehouse, Enterprise Architect პაკეტების ბაზაზე, ობიექტ ორიენტირებული, პროცეს-ორიენტირებული და სერვის-ორიენტირებული არქიტექტურის მოთხოვნათა გათვალისწინებით.

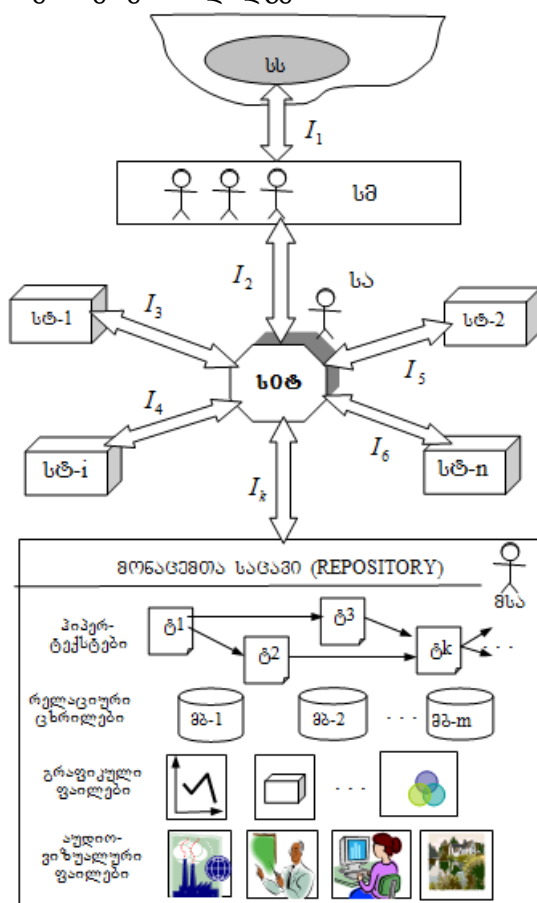
განაწილებული კორპორაციული სისტემების დაპროექტების ტექნიკური რეალიზაციის მხარე მოითხოვს მისი ცალკეული კვანძების ფუნქციური ანალიზის საფუძველზე აპარატული და პროგრამული უზრუნველყოფების ამოცანების გადაწყვეტას. აპარატულში იგულისხმება კომპიუტერული და ქსელური ტექნიკა, რომლის საფუძველზედაც უნდა მოხდეს სისტემის გლობალურ / ლოკალურ ქსელში ფიზიკურად ჩართვის ორგანიზება. პროგრამული კი შეესაბამება ქსელში ჩართულ ოპერაციული სისტემის, პლატფორმის, საერთო-სერვისული გარემოს და კერძო-ფუნქციური პაკეტების ერთობლიობას.

ინტერკორპორაციულ ქსელებში ელექტრონული ფუნქციური კავშირების განსახორციელებლად საჭიროა შიგა და გარე ინფორმაციული ნაკადების ანალიზის ჩატარება. არსებობს შემდეგი სახის ინფორმაციული ნაკადები (Workflow): კორესპოდენცია და წერილები; ნორმატიული აქტები და კანონები; კონტრაქტები (ხანგრძლივი) და შეკვეთები (ერთჯერადი); კრედიტები და განვადებები, მონიტორინგი და კონტროლი, აგრეთვე ვალდებულებათა ფაქტობრივი შესრულებები და მათი ანალიზის მასალები; ინტერნეტიდან მიღებული ინფორმაცია (ფურცლები); აუდიო და ვიდეო ინფორმაცია (ელექტრონული გამოფენები, სალონები, პროდუქციის კატალოგები); საბანკო ანგარიშები, ბუღალტრული აღრიცხვა; კადრების აღრიცხვისა და შრომითი დასაქმების დოკუმენტაცია; ინფორმაცია პარტნიორებისა და კონკურენტების შესახებ; ინფორმაცია შიგა და გარე სერვისების შესახებ; სტატისტიკური ანალიზის მასალები და სხვ.

ინფორმაციული ნაკადების მოცულობათა საანგარიშოდ ინფორმაციის ერთეულად მიღებულია:  $I_T$ : ერთი ნაბეჭდი A4 ფორმატის ფურცლის ტექსტური ინფორმაციის სიდიდე;  $I_A$ : ერთი აუდიო ინფორმაციის სიდიდე;  $I_V$ : ერთი ვიდეო ინფორმაციის სიდიდე; პირობითად მივიღოთ, რომ  $I_T=4$  Kb,  $I_A=20$  Kb,  $I_V=30$  Kb. ინფორმაციული ნაკადების ზომები დამოკიდებული იქნება კომერციული ობიექტების მასშტაბებზე (ზომები, კონიუნქტურა, წლიური



ფონდბრუნვა და საქონელბრუნვა, ფილიალების რაოდენობა, სეზონი, რეგიონი და ა.შ.). ინფორმაციული ნაკადების მოცულობების საანგარიშოდ შეიძლება ჩავატაროთ მიახლოებითი, გასაშუალებული გათვლები (თვის, კვარტლის, წლის და ხანგრძლივი პერიოდისთვის), რომელთა საფუძველზე შესაძლებელი იქნება საერთო ინფორმაციული ფონდის მოცულობის შეფასება და მონაცემთა განაწილებული საცავის (Warehouse) ფიზიკური მოწყობილობების საჭირო მეხსიერების დადგენა.



ნახ.19

მე-19 ნახაზზე მოცემულია ინტეგრირებული საინფორმაციო მართვის სისტემის ზოგადი სქემა. საპრობლემო სფერო (სს), ჩვენ შემთხვევაში განაწილებული კორპორაციული სისტემა - კომპლექსური ობიექტია. სისტემის მომხმარებლები (სმ) კლასიფიცირდება მათი ფუნქციური დანიშნულების მიხედვით (მაგალითად, სისტემის ადმინისტრატორი, მონაცემთა საცავის ადმინისტრატორი, საბოლოო მომხმარებლები: ხელმძღვანელები და ფუნქციურ ქვედანაყოფთა სპეციალისტები და ა.შ.). II - ინტერფეისია საპრობლემო სფეროს და სისტემის მომხმარებლებს შორის.

საანგარიშო ფორმულებად ვიყენებთ შემდეგ გამოსახულებებს:

$$V_{IT} = k_j * \sum_{i=1}^n R_i * I_T, \text{ სადაც}$$

$V_{IT}$  არის ტექსტური სახის ინფორმაციის თვიური, კვარტალური და წლიური დოკუმენტების ჯამური მოცულობა მეგაბაიტებში;  $k_j$  - თვიური,

კვარტალური და წლიური კოეფიციენტი (1, 3, 12);

$R_i$  - ტექსტური დოკუმენტის A4-ფურცლების რაოდენობა;

აუდიო ინფორმაციული ნაკადებისათვის შესაბამისად გვექნება:

$$V_{jA} = k_j * \sum_{i=1}^m A_i, \text{ სადაც}$$

$A_i$  - აუდიო ინფორმაციის ფაილის ზომა;

საჭიროა გავითვალისწინოთ ხმის გადაცემის მახასიათებელი, რომელიც საშუალოდ წარმოშობს 64 bit/sec წარმადობის ინფორმაციულ ნაკადებს.

ვიზუალურისათვის შესაბამისად გვექნება:

$$V_{jV} = k_j * \sum_{i=1}^m V_i, \text{ სადაც}$$

$V_i$  - ვიდეო ინფორმაციის ფაილის ზომა;

საჭიროა გავითვალისწინოთ, რომ ვიდეო გამოსახულების გადაცემა არქივირების გარეშე წარმოშობს 9.216 ბიტ/სეც, ხოლო არქივირებით 1.5 ბიტ/სეც წარმადობის ინფორმაციულ ნაკადებს.

მთლიანად ინფორმაციული ნაკადების ჯამური მოცულობა იქნება:

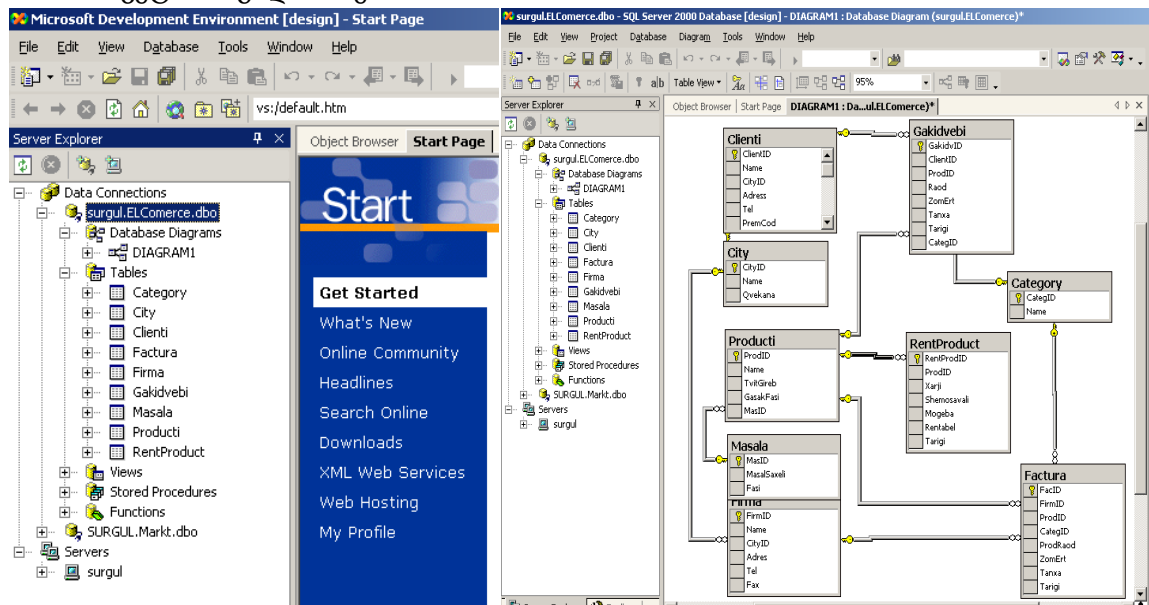
$$S = T_i * K_i * \sum_{j=1}^n V_j^i, \text{ სადაც}$$

$T_i$  -  $i$ -ური კომერციული ობიექტის არსებობის მთლიანი პერიოდია (წლები);

$K_i$  -  $i$ -ურ კომერციულ ობიექტზე ფილიალების რაოდენობა;

$V_j^i$  -  $i$ -ური კომერციული ობიექტის  $J$ -ური სახის ინფორმაციული ნაკადის მოცულობა.

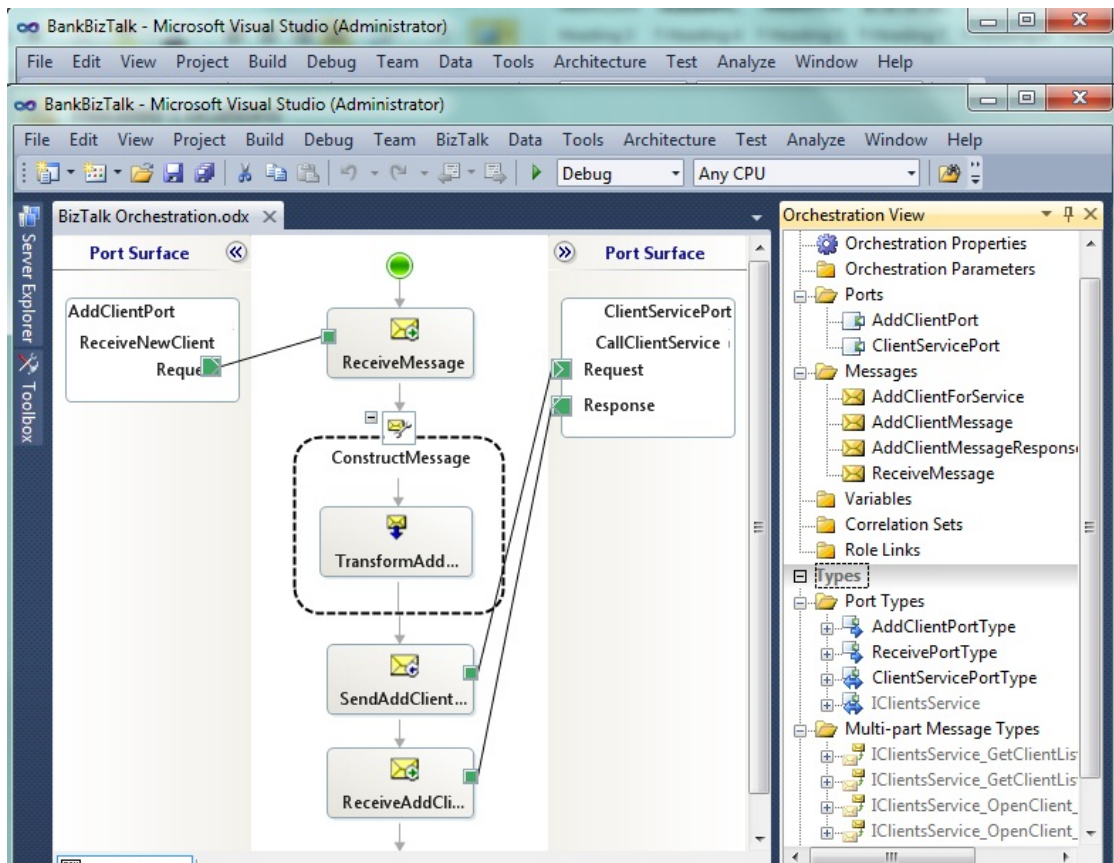
ექსპერტული ინფორმაციის საფუძველზე, როგორც ჩვენი პირობითი გათვლებიდან გამომდინარეობს, ერთ კორპორაციულ ობიექტზე დაახლოებით 10 წლიანი არსებობის პერიოდში მონაცემთა საცავისათვის საშუალოდ დაგეგმირდება 250 GB-იანი მეხსიერება (გაითვალისწინება ძირითადი სტატისტიკური და ისტორიული ფაილების შენახვაც). დიდი კორპორაციებისთვის ეს რიცხვი უნდა გამრავლდეს ობიექტების რაოდენობაზე და დაემატოს მათი მენეჯმენტისათვის საჭირო ინფორმაციული ნაკადი. ამგვარად, მონაცემთა საცავისათვის საჭირო ფიზიკური მეხსიერება მიაღწევს რამდენიმე ტერაბაიტ-მოცულობას. მონაცემთა ბაზების ცხრილებიდან საჭირო ინფორმაციის ამოსაღებად გამოიყენება SQL (Structured Query Language-სტრუქტურირებული მოთხოვნების ენა). ამისათვის იწერება მოთხოვნათა (Query) ფორმა მაიკროსოფტის მიერ დამუშავებულ SQL-ენის სტანდარტის შესაბამისად. მე-20 ნახაზზე მოცემულია ელექტრონული კომერციის სისტემის (ELCommerce) რეალიზებული მონაცემთა ბაზის ერთ-ერთი ფრაგმენტი MsSQL Server პაკეტის მაგალითზე.



ნახ.20

სერვის-აპლიკაციების პროგრამული რეალიზაცია და გამოყენება ხდება Ms BizTalk Server გარემოში, რაც საშუალებას იძლევა მიიღოს და დაამუშაოს შეტყობინებები, დამატებული ორკესტრაციის ფაილით Bank\_orchestration.odx.





ნახ.21

შეტყობინების მიღება ხდება XML ფორმატში, რომლის დასამუშავებლად იქმნება სქემის ფაილი Clients.xsd. მონაცემები მიიღება პორტების საშუალებით. შემდეგ უნდა მოხდეს XML-ის ფორმირება, რა ფორმატიც საჭიროა მისი გადაცემა ვებ-სერვისზე. ამისათვის გამოიყენება Transform კომპონენტი. მის შემავალ შეტყობინების ტიპში მიეთითება Clients.xsd და Clients4Service.xsd. ამ სქემებს შორის ურთიერთკავშირი აიკვება სპეციალური ინსტრუმენტით. მიღებული XML-ის გადაცემა ხდება ClientsService ვებ-სერვისისთვის.

ინტერნეტ ტექნოლოგიების გამოყენება, რომლის ერთ-ერთ სახესაც ვებ-გვერდების აგების ახალი ASP.NET ტექნოლოგია წარმოადგენს საშუალებას გვაძლევს შევქმნათ ისეთი საინფორმაციო გვერდი რომელიც მომხმარებელს საშუალებას აძლევს შეარჩიოს მისთვის სასურველი სერვისი/პროდუქცია. ამ მოდულში ხდება მომხმარებლის მიერ სხვადასხვა სერვისის/პროდუქციის ნახვა, ძებნა, არჩევა, შეკვეთა, მომხმარებლის რეგისტრაცია სისტემაში და პირადი მონაცემები შეტანა.

დისერტაციის მესამე თავში წარმოდგენილია აგრეთვე ჩვენ მიერ აგებული სისტემების მაგალითები: კომერციული ბანკის ინტეგრირებული სისტემა, ნავთობპროდუქტების ტრანსპორტირების მონიტორინგის ქსელური სისტემა, ინტერკორპორაციული ავტომატიზებული სისტემა BizTalk პლატფორმაზე და სხვ.

## დასკვნები:

1. მზარდი მოთხოვნა ინტეგრაციულ სისტემებზე, მონაცემთა დამუშავების ავტომატიზაციის ამოცანების, სქემების, მიდგომების, პრინციპების მუდმივი ცვლილება, ასევე ქსელური და აპარატურული ტექნოლოგიების შესაძლებლობების ზრდა, განაპირობებს Web-აპლიკაციების დაპროექტების და რეალიზაციის სისტემების პროგრამული უზრუნველყოფის განვითარებას სერვის-ორიენტირებული არქიტექტურით;

2. რთულ კორპორაციულ და კორპორაციათაშორის სისტემებში აუცილებელია თანამედროვე ინტეგრაციული საშუალებების გამოყენება, რაც უზრუნველყოფს სერვის-ორიენტირებული არქიტექტურის რეალიზაციას, რომელიც დაფუძნებულია ორგანიზაციის ინტეგრაციის არხზე. მსგავსი მიდგომა თავის მხრივ განაპირობებს ელექტრონული ბიზნეს-პროცესების შესრულების სისწრაფის ზრდას, ეფექტურობას, საიმედოობას და უსაფრთხოებას;

3. ობიექტ-ორიენტირებული აპლიკაციების ვებ-სერვისებით ფორმირება განსაკუთრებით მოქნილი, მოსახერხებელი და გაცილებით საიმედოა, როგორც ახალი პროგრამული პროდუქტების შექმნისას, ისე არსებულ სისტემაში ცვლილებების გატარებისას. ვებ-სერვისი, შეიძლება განვიხილოთ, როგორც ხიდი ობიექტ-ორიენტირებულ და პროცეს-ორიენტირებულ ტექნოლოგიებს შორის, რაც კომპანიათაშორისი და კომპანიის მსხვილ სტრუქტურათაშორისი საქმიანი პროცესების ინტეგრაციასა და მრავალაპლიკაციურ მართვას უზრუნველყოფს;

4. სერვის-ორიენტირებულ არქიტექტურას ახლავს რიგი სირთულეები, რაც უკავშირდება ვებ-სერვისების ტიპიზირებას და ტიპიზირებული ვებ-სერვისების მასშტაბურ რაოდენობას, არასაკმარის ინსტრუმენტულ საშუალებებს, სტანდარტიზაციისა და ცენტრალიზაციის პრობლემების წარმოქმნას. მიუხედავად ამ სირთულეებისა, აპლიკაციების ინტეგრაციის თვალსაზრისით სერვის-ორიენტირებული არქიტექტურა და პროცეს-ორიენტირებული მიდგომა აქტუალური, მომქმედი და განვითარებადია, რასაც მხარს უჭერს Microsoft და Java ტექნოლოგიები;

5. სისტემა BizTalk, როგორც ორგანიზაციის სერვისული არხის იმპლემენტაციის პლატფორმა, უზრუნველყოფს სხვადასხვა აპლიკაციებს შორის კავშირის დამყარებას. გრაფიკული ინტერფეისის საშუალებებით ახორციელებს ბიზნეს-პროცესების ორკესტრაციას, მონიტორინგს, როგორც პროცესების რაოდენობრივი შეფასებისთვის, ასევე გარკვეულ მოვლენებზე რეაგირებისთვის. BizTalk-ის დახმარებით შესაძლებელია პროცესების ავტომატიზაციის და ინდუსტრიული სტანდარტების უზრუნველყოფა, რაც საშუალებას იძლევა შემცირდეს დანახარჯები და კომპლექსურობა B2B კავშირების დასამყარებლად;

6. კორპორაციული მართვის ბიზნეს-პროცესების მოდელირებისა და ანალიზისთვის ეფექტურად გამოიყენება სტოქასტურ-დროითი პეტრის ქსელები, როგორც დინამიკური პროცესების იმიტაციური მოდელირების ინსტრუმენტი. აგებული პეტრის ქსელის მოდელების ვარიანტების ანალიზის საფუძველზე შესაძლებელია გარკვეული დასკვნების გაკეთება ობიექტზე ბიზნეს-პროცესების ეკვივალენტურ აქტიურობათა დიაგრამების პროგრამული რეალიზაციის ეფექტიანობის შესახებ. აგრეთვე, პეტრის ქსელზე გადასასვლელთა (აქტიურობის დიაგრამაზე შესრულებადი პროცესების) დროითი პარამეტრების შერჩევით და „კონფლიქტური“ გადასასვლელების გარკვეული ლოგიკური პირობების

შემოტანით, შესაძლებელი ხდება გასაანალიზებელი შედეგების ხარისხის ამაღლება და ცალკეული სერვის-პროცესების დროითი მაჩვენებლების ანალიზი.

7. კორპორაციათაშორის საინფორმაციო მართვის სისტემების რეალიზაციის მიზნით სერვის-ორიენტირებული არქიტექტურით, განსაკუთრებით მნიშვნელოვანია მონაცემთა გაცვლის საშუალებების ფუნქციონირების ეფექტურობის ანალიზი, რაც წარმატებით ხორციელდება ფერადი პეტრის ქსელების (CPN) ინსტრუმენტით. მასში კარგადაა შერწყმული პეტრის სტოქასტური ქსელებისა და ობიექტ-ორიენტირებული დაპროგრამების თეორიის ძირითადი პრინციპები (იერარქიულობა, მოდულურობა, მემკვიდრეობითობა, პოლიმორფიზმი – დიდი სისტემების მოდელირებისთვის), რაც მის დიდ პრაქტიკულ ღირებულებასაც განაპირობებს თანამედროვე ინფორმაციულ ტექნოლოგიათა გამოყენების მრავალ სფეროში;

8. თანამედროვე ინფორმაციული ტექნოლოგიების, კერძოდ CASE-ინსტრუმენტების გამოყენებით კორპორაციული მენეჯმენტისა და ინტერკორპორაციული ვებ-აპლიკაციების აგების პროცესში, მნიშვნელოვნად უმჯობესდება პროგრამული უზრუნველყოფის ხარისხი და საგრძნობლად მცირდება დაპროექტების, მისი იმპლემენტაციისა და რეინჟინერინგის პერიოდები. დიდი საინფორმაციო სისტემების მონაცემთა ბაზების სტრუქტურების დაპროექტებისა და აგების პროცესების ავტომატიზება, აგრეთვე მისი შემდგომი რესტრუქტურიზაციის პრობლემების მოქნილად გადაწყვეტის საშუალებას იძლევა;

#### **გამოქვეყნებული ლიტერატურა:**

1. ბულია ი. თანამედროვე სისტემებში ინტეგრაციის, მონაცემთა გადაცემის და დამუშავების ტექნოლოგიები. სტუ შრ.კრ. „მართვის ავტომატიზებული სისტემები“. N2(11), თბილისი, 2011, გვ. 139–144.
2. ბულია ი., თურქია ე., გიუტაშვილი მ. ინტერკორპორაციული აპლიკაციების ჰორიზონტალური და ვერტიკალური ინტეგრაციის მართვა სერვის-ორიენტირებული არქიტექტურის ბაზაზე. სტუ შრ.კრ. „მართვის ავტომატიზებული სისტემები“. N1(12). გვ. 57–62.
3. ბულია ი. ბიზნეს-აპლიკაციების აგების თანამედროვე ინფორმაციული ტექნოლოგიები. საერთაშ. სამეცნ. ტექნ. კონფ.: „მართვის ავტომატიზ. სისტ. და ახალი ინფორმაციული ტექნოლოგიები“. სტუ, თბილისი, 20-22 მაისი, 2011
4. Bulia I., Surguladze G., Topuria N., Burchuladze A. Modelling and automation of monitoring and control processes for transporting oil products. Intern.Conf. of Comp. Inf.and Comp. Technologies, Modelling, Control. Tbilisi, Georgia, Nov.1-4, 2010. p.37-39
5. ბულია ი., სურგულაძე გ., კაშიბაძე მ., შურღია ი. სერვის-ორიენტირებული არქიტექტურის ინტერკორპორაციული ვებ-აპლიკაციების ბიზნეს-პროცესების მოდელირება და კვლევა პეტრის ქსელებით. სტუ შრ.კრ. „მართვის ავტომატიზებული სისტემები“. N1(12). გვ. 63–72.
6. ბულია ი., სურგულაძე გ., ოხანაშვილი მ., ქრისტესიაშვილი ხ. კორპორაციული მენეჯმენტის ბიზნეს-პროცესების მოდელირება და კვლევა ფერადი პეტრის ქსელებით. სტუ შრ.კრ. „მართვის ავტომატიზებული სისტემები“. N1(12). გვ. 73–82.
7. ბულია ი., სურგულაძე გ., თოფურია ნ. ინტერკორპორაციული სერვის-ორიენტირებული სისტემის მონაცემთა ბაზების დაპროექტება და რეალიზაცია. სტუ შრ.კრ. „მართვის ავტომატიზებული სისტემები“. N1(12). გვ. 83–89.
8. ბულია ი., სურგულაძე გ., თოფურია ნ., მარკოსიანი დ. სოციალური მომსახურების სისტემაში მონიტორინგის პროცესების ავტომატიზაცია. სტუ, შრ.კრ. „მართვის ავტომატიზებული სისტემები“. - N 1(8), 2010. გვ. 175–180.
9. ბულია ი., სურგულაძე გ., ბერძენიშვილი ი., ვაჭარაძე ი., ხელაძე ნ. კორპორაციული მართვის სისტემის ვებ-აპლიკაციის დამუშავება ინტერნეტ-ინტრანეტ

გარემოში .Net-პლატფორმაზე. სტუ, შრ.კრ. „მართვის ავტომატიზებული სისტემები“. - N1, 2006. გვ. 159–162.

10. ბულია ი., სურგულაძე გ., პეტრიაშვილი ლ., ყვავაძე ლ. ბიზნეს-პროცესების მართვის ინტერნეტული სისტემის დაპროექტება და რეალიზაცია ASP და C# .NET-ტექნოლოგიებით. საქ.მეცნ.აკად., “მეცნიერება და ტექნოლოგიები”, N10-12. თბილისი, 2005. გვ. 45–49.

11. ბულია ი., სურგულაძე გ., თურქია ე. ვებ-აპლიკაციების აგება ASP & ADO & C# პაკეტებით .NET პლატფორმაზე. ISBN 978-9941-14-365-6. სტუ, თბ., 2009. 86 გვ.

12. ბულია ი., სურგულაძე გ., თურქია ე. ვებ-აპლიკაციების დამუშავება მონაცემთა ბაზების საფუძველზე (ADO.NET, ASP.NET, C#). ISBN 978-9941-14-289-5. სტუ, თბ., 2009. 172 გვ.

## ABSTRACT

The work considers modern means of information exchange on the basis of their integration in the enterprise and inter-enterprise systems. There are discussed methods, architectures, principles, models that facilitate the integration issues, making them more efficient. They serve in data transfer and conversion, management and automation of business processes between different systems and applications. Establishing of the integration between the existing software in enterprises, is offered by modern methods, in particular through the implementation of Service Oriented Architecture, which is realized by means of different technologies such as: Electronic Data Interchange(EDI), Enterprise Service Bus (ESB), XML, BPEL. These technologies enable the integration between interacting applications across multiple platforms, data interchange of different formats, reliability, flexibility, scalability.

The work presents the problems of inter-corporate business process management. The questions of business processes integration, information composition and synchronization related to multi-application environment are suggested. Considering the inter and intra-corporate management aspects, main attention is given to the strategies of software product development in the organizational systems. To provide the Integration of the horizontal and vertical management in the inter-corporate applications, the possibilities of service-oriented architecture and its practical realization examples are described. Based on the business-processes model, which occurs between the Tax Service of Ministry of Finance and the Banks, formation of the encapsulated business-functions in the web-services and the inter-corporate application process functioning examples are shown.

Questions of a unified modeling of inter-corporate business processes with service-oriented architecture and display them in the time-stochastic Petri nets, with the aim of further research. The task of transformation models is proposed based on the tools BPMN -> Activity-D -> PetNet. Investigated the temporal characteristics of business process execution system that provides the basis for correct decision making to further improve the service providing organization. In order to implement such a system using BizTalk platform in the working environment MsVisualStudio.NET.

The problems of modeling and studying the processing of requests in the banking corporation based on service applications and systems analysis. Introduced the concept of building an integrated automated UML-based standards and service-oriented architecture. It is proposed the use of colored Petri nets (CPN) to construct a simulation model of inter-corporate service processes and study the temporal characteristics of its functioning.

Object-role modeling problem for intercorporate automated systems and its implementation is considered in the article. The creation concept for the distributed databases for various financial structures on the basis of Service-oriented architecture is offered. The web-application forms on Ms Visual Studio.NET platform by The NORMA software and Ms SQL Server package are realized as a result.

Proposed for financial banks, large shopping centers and other objects the service- and e-commerce- processes information flows required for the object - oriented analysis and design, the user interfaces of software implementation of the integrated use of modern information technology. Information objects for meta description is given in accordance with the SQL-standard. The deployment of the technology described in the distributed data warehouse.