

თ. სტურუა

საინფორმაციო სისტემების დაპროექტება

„ტექნიკური უნივერსიტეტი“

საქართველოს ტექნიკური უნივერსიტეტი

თ. სტურუა

საინფორმაციო სისტემების დაპროექტება



დამტკიცებულია სახელმძღვანელოდ
სტუ-ის სარედაქციო-საგამომცემლო
საბჭოს მიერ. 05.06.2012, ოქმი №2

თბილისი
2012

მოცემულია საინფორმაციო სისტემის ცნება, კლასიფიკაცია და უზრუნველყოფის ქვესისტემები, საინფორმაციო სისტემის სასიცოცხლო ციკლი და მისი დამუშავების ეტაპები. განსაკუთრებული ყურადღებაა გამახვილებული საინფორმაციო სისტემების დაპროექტების თანამედროვე საშუალებებზე - CASE-სისტემებზე. განხილულია საინფორმაციო სისტემების სტრუქტურასა და ობიექტზე ორიენტირებული დაპროექტების საშუალებანი, კერძოდ შესაბამისი დიაგრამები, მათი აგების წესები და ამ დიაგრამების ასაგებად გამოყენებულია Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio.

განკუთვნილია ინფორმატიკისა და მართვის სისტემების ფაკულტეტის ბაკალავრების, მაგისტრების, დოქტორანტებისა და საინფორმაციო სისტემების დაპროექტებით დაინტერესებული სხვა სპეციალისტებისათვის.

რეცენზენტები: სრული პროფესორი ი. როდონაია,
სრული პროფესორი ბ. მეფარიშვილი

© საგამომცემლო სახლი "ტექნიკური უნივერსიტეტი", 2012

ISBN 978-9941-20-102-8

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



Verba volant,
scripta manent

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

საავტორო უფლებების დარღვევა ისჯება კანონით.

შესავალი

ეკონომიკური ობიექტების (წარმოება, ბანკი, სავაჭრო ორგანიზაცია, სახელმწიფო დაწესებულება და სხვ.) მართვისათვის საინფორმაციო სისტემების გამოყენების ეფექტურობა დიდად არის დამოკიდებული იმაზე, თუ რამდენად მოიცავს და მასთან არის ინტეგრირებული მართვის ფუნქცია.

პირველი ელექტრონული გამოთვლითი მანქანების გამოჩენისთანავე, ინფორმაციულმა სისტემებმა არსებითი ცვლილებები განიცადა.

50-იან წლებში ელექტრონულ გამოთვლით მანქანებზე ძირითადად ხდებოდა ისეთი ეკონომიკური ამოცანების გადაწყვეტა, რომლებიც დიდი მოცულობის ინფორმაციული მასივების გადამუშავებასთან იყო დაკავშირებული.

60-იან წლებში, საწარმოებისა და დაწესებულებების მართვის კომპლექსური ავტომატიზაციისა და მონაცემთა ბაზების ინფორმაციული უზრუნველყოფის ინტეგრაციის იდეა გაჩნდა. მართვის ავტომატიზებული სისტემების დამუშავება 70-იან წლებში მესამე თაობის ელექტრონული გამოთვლითი მანქანების ბაზაზე რეალობად იქცა, რაც საშუალებას იძლეოდა გამოთვლითი სისტემები ტერმინალური ქსელით შეგვექმნა. მაგრამ გამოთვლითი მანქანების არასაკმარისმა სწრაფქმედებამ და საიმედოობამ, აგრეთვე მომხმარებლის ინფორმაციული მოთხოვნილების რეალიზაციის მოქნილი საშუალებების არქონამ, საინფორმაციო სისტემები წარმოების მართვის ეფექტურობის ამაღლების ძირითად იარაღად ვერ აქცია.

80-იანი წლები მართვის სფეროში დასაქმებული მუშაკების მიერ პერსონალური კომპიუტერების ფართოდ გამოყენებით აღინიშნება, მეოთხე თაობის ალგორითმული ენების გამოყენებით ავტომატიზებული სამუშაო ადგილები იქმნება, რომელიც საშუალებას გვაძლევს მომხმარებლისათვის საჭირო ფორმები სწრაფად დავამუშაოთ. მაგრამ ეს ავტომატიზებული სამუშაო ადგილები მხოლოდ ლოკალურ ავტომატიზაციას წარმოადგენს და წარმოების ეფექტურ მართვაზე არსებით გავლენას ვერ ახდენს.

90-იანი წლებისათვის დამახასიათებელია ტელეკომუნიკაციური საშუალებების განვითარება, რამაც მოქნილი ლოკალური და გლობალური

გამოთვლითი ქსელების შექმნამდე მიგვიყვანა, რაც კორპორაციული საინფორმაციო სისტემების დამუშავების წინაპირობაა.

კორპორაციული საინფორმაციო სისტემა თავისთავში აერთიანებს 70-იანი წლების მართვის კომპლექსური ავტომატიზებული სისტემებისა და 80-იანი წლების ლოკალური ავტომატიზაციის შესაძლებლობებს.

ნახევარსაუკუნოვანი არსებობის მანძილზე პროგრამულმა უზრუნველყოფამ უდიდესი ცვლილება განიცადა: მხოლოდ მარტივი ლოგიკური და არითმეტიკული ოპერაციების შესრულების პროგრამებიდან, წარმოების მართვის რთულ სისტემებამდე. პროგრამული უზრუნველყოფის განვითარების პროცესში ყოველთვის შეიძლება ორი ძირითადი მიმართულების გამოყოფა:

- გამოთვლების შესრულება;
- ინფორმაციის დაგროვება და გადამუშავება.

თუმცა, დასაწყისში კომპიუტერი უმთავრესოდ რთული მათემატიკური გამოთვლების შესასრულებლად (პირველ რიგში, ატომური იარაღისა და სარაკეტო ტექნიკის შექმნასთან დაკავშირებული გამოთვლებისათვის) იყო გათვალისწინებული, ხოლო ამჟამად დომინირებს მეორე მიმართულება. გამოთვლითი ტექნიკის მიერ შესრულებული ძირითადი ფუნქციების ასეთი გადანაწილება სავსებით გასაგებია - სამოქალაქო ბიზნესი უფრო მეტად გავრცელებულია, ვიდრე სამხედრო და სამეცნიერო გამოთვლები, ხოლო ღირებულების შემცირების გამო, კომპიუტერი როგორც მცირე საწარმოებისათვის, ისე კერძო პირებისათვისაც ხელმისაწვდომი გახდა.

ამგვარად, თანამედროვე საინფორმაციო სისტემები კომუნიკაციის ოპერატიულობასა და ბიზნეს-პროცესების მონაწილეთა ინტეგრაციას, მართვის ნებისმიერ დონეზე მიღებული გადაწყვეტილებების ხარისხს უზრუნველყოფს.

ინფორმაციის როლი ეკონომიკაში სწრაფად იზრდება. ინფორმაციული ეკონომიკის ერთ-ერთი ძირითადი რესურსი, რომელსაც ორგანიზაციები იყენებენ, არის ინფორმაცია. ორგანიზაციის არსებობა იმაზეა დამოკიდებული, თუ რამდენად ზუსტად, დროულად და ხარისხიანად ადევნებს იგი თვალყურს იმ ბაზრისა და საქონლის შესახებ ინფორმაციას, რომელთანაც მას უხდება მუშაობა.

Internet/Intranet ტექნოლოგიების გამოჩენამ არსებითად გაგვიმარტივა საინფორმაციო სისტემების დამუშავება და ექსპლუატაცია აგრეთვე, უფრო რთული საინფორმაციო სისტემების დამუშავებისა და გამოყენების საშუალება მოგვცა, მათ შორის ისეთების, რომლებიც მთელ მსოფლიოში ვრცელდება.

Internet/Intranet ტექნოლოგიების გამოყენებით ძალისა და სხვა რესურსების დაახლოებით იმავე დანახარჯების მეშვეობითაა ეს შესაძლებელი, რაც ადრე გაცილებით მცირე მასშტაბის სისტემების დამუშავებას ესაჭიროებოდა. მაგრამ დიდი საინფორმაციო სისტემების დამუშავებელთა კვალიფიკაციის დონე გაცილებით მაღალი უნდა იყოს. დაპროექტებისას დაშვებული შეცდომა შეიძლება ძალიან გვიან გამოვლინდეს, ხოლო მის გამოსასწორებლად გაწეული ხარჯები შეიძლება საკმაოდ აღემატებოდეს საინფორმაციო სისტემის პირველადი დამუშავების ხარჯებს.

გაცილებით გაიზარდა ხელმისაწვდომი ინფორმაციის მოცულობა. ამჟამად, აშშ-ში სამუშაო ძალის 65%-ზე მეტი ამა თუ იმ ხარისხით ინფორმაციის დამუშავების სფეროშია დასაქმებული, მაშინ როდესაც 1950 წელს ამ საქმიანობით მხოლოდ 17% იყო დაკავებული. ანალოგიური მდგომარეობაა მსოფლიოს წამყვან ქვეყნებშიც, რაც მიუთითებს, რომ ინდუსტრიული ეკონომიკიდან მსოფლიო ინფორმაციულზე გადადის.

თანამედროვე კომპიუტერული ტექნოლოგიები საშუალებას გვაძლევს დიდი მოცულობის ინფორმაციული ბაზა შევქმნათ. მენეჯერები ხშირად ძალიან დიდი მოცულობის ინფორმაციას იღებენ. პრობლემა უკვე საერთოდ ინფორმაციის ნაკლებობა კი არ არის, არამედ იმ ინფორმაციის ნაკლებობაა რომელიც კონკრეტულ მომხმარებელს კონკრეტული სახით ესაჭიროება.

დიდი ძალისხმევა ესაჭიროება ყველაზე მარტივი მონაცემების მოძებნასაც კი. საჭირო ინფორმაცია შეიძლება მომხმარებელს დაგვიანებით და ამასთან, დამახინჯებული სახით მიეწოდოს. ზოგჯერ შეიძლება ინფორმაცია ძალიან ბევრი იყოს, მაგრამ მარტივი არ არის მასში ის ინფორმაცია ამოვარჩიოთ, რომელიც მოცემულ მომენტში მომხმარებელს ესაჭიროება. სამწუხაროდ, ჩვენთვის ხელმისაწვდომი ინფორმაციის შეგროვებასა და მის ხარისხიან დამუშავებაზე გაწეული დანახარჯები უსასრულოდ იზრდება.

ამიტომ, ინფორმაციული ნაკადების ოპტიმიზაციის ამოცანა სულ უფრო აქტუალური ხდება.

საინფორმაციო სისტემები

მართალია, საინფორმაციო სისტემები ჩვეულებრივი პროგრამული პროდუქტია, მაგრამ სტანდარტული გამოყენებითი პროგრამებისა და სისტემებისაგან რიგი არსებითი თვისებებით განსხვავდება.

საგნობრივ არეზე დამოკიდებულების მიხედვით, საინფორმაციო სისტემები ერთმანეთისაგან შეიძლება თავისი ფუნქციებით, არქიტექტურით, რეალიზაციით ძალიან განსხვავდებოდეს. მაგრამ, შეიძლება გამოვყოთ რიგი თვისებები, რომლებიც ყველასათვის საერთოა:

- საინფორმაციო სისტემები ინფორმაციის შეგროვების, შენახვისა და გადამუშავებისათვისაა განკუთვნილი. ამიტომ, ნებისმიერ მათგანს საფუძვლად უდევს შენახვისა და მონაცემთა წვდომის გარემო;
- საინფორმაციო სისტემები გამოთვლითი ტექნიკის გამოყენების სფეროში დაბალი კვალიფიკაციის საბოლოო მომხმარებლებზეა ორიენტირებული. ამიტომ, საინფორმაციო სისტემის მომხმარებლის კლიენტური დანართები უნდა მოიცავდეს უბრალო, მოსახერხებელ და მარტივად ასათვისებელ ინტერფეისს, რომელიც საბოლოო მომხმარებელს ფუნქციებთან სამუშაო ყველა აუცილებელ ინსტრუქციას მიაწვდის, მაგრამ ამასთანავე, მას უფლებას არ მისცემს შეასრულოს რაიმე ზედმეტი მოქმედება.

ამგვარად, საინფორმაციო სისტემების დამუშავების დროს, ორი ძირითადი ამოცანის გადაწყვეტაა საჭირო:

- ინფორმაციის შესანახად განკუთვნილ მონაცემთა ბაზის დამუშავების ამოცანა;
- კლიენტური დანართების მომხმარებლის გრაფიკული ინტერფეისის დამუშავების ამოცანა.

დღესდღეობით, საწარმოს მართვა კომპიუტერის გარეშე უბრალოდ წარმოუდგენელია. კომპიუტერი დიდი ხანია მყარად შემოვიდა მართვის ისეთ

არეებში, როგორცაა საბუღალტრო აღრიცხვა, საწყობის, ასორტიმენტისა და შესყიდვების მართვა. მაგრამ თანამედროვე ბიზნესი წარმოების მართვაში საინფორმაციო ტექნოლოგიების ბევრად უფრო ფართო გამოყენებას მოითხოვს. საინფორმაციო ტექნოლოგიების სიცოცხლისუნარიანობა და განვითარება იმით აიხსნება, რომ თანამედროვე ბიზნესი მართვაში დაშვებული შეცდომების მიმართ ძალზე მგრძობიარეა. ინტუიცია, ხელმძღვანელის პირადი გამოცდილების და კაპიტალის სიდიდე უკვე მცირეა იმისათვის, რომ იყო პირველი. განუსაზღვრელობისა და რისკის პირობებში ნებისმიერი საქმიანი მმართველი გადაწყვეტილების მისაღებად აუცილებელია მუდმივად აკონტროლებდეს საფინანსო-სამეურნეო საქმიანობის ცალკეულ ასპექტებს, მაგალითად, ვაჭრობას, წარმოებას თუ რაიმე მომსახურების სფეროს. ამიტომ, მართვის მიმართ თანამედროვე მიდგომა საინფორმაციო ტექნოლოგიებში გარკვეული სახსრების დაბანდებას ითვალისწინებს. და რაც უფრო მსხვილია წარმოება, დაბანდებაებიც მით უფრო სერიოზული უნდა იყოს. ეს აუცილებელია - მკაცრ კონკურენტულ ბრძოლაში მხოლოდ ის გაიმარჯვებს, ვინც უფრო უკეთესადაა აღჭურვილი და ყველაზე უფრო ეფექტურადაა ორგანიზებული.

მთელი ინფორმაცია, რომელიც საინფორმაციო სისტემების ჩარჩოებში გროვდება, შეიძლება სამ ჯგუფად დაიყოს:

1. მოცემული ორგანიზაციის სხვა საინფორმაციო სისტემებში არსებული ინფორმაცია;
2. შესასყიდი ინფორმაცია;
3. დამოუკიდებლად შესაგროვებელი ინფორმაცია.

განვიხილოთ ინფორმაციის ეს წყაროები უფრო დაწვრილებით:

1. მოცემული ორგანიზაციის სხვა საინფორმაციო სისტემებში არსებული ინფორმაცია. როგორც წესი, სხვადასხვა სახის ინფორმაცია (ფინანსური, კადრების და სხვ.) უკვე არსებობს ელექტრონული სახით. თუ ეს ინფორმაცია ესაჭიროება დასამუშავებელ საინფორმაციო სისტემას, მაშინ მასზე მიმართვა უნდა უზრუნველვყოთ. მაგრამ ეს მონაცემები სრულიად სხვა მიზნებისათვის გროვდებოდა და ამიტომაც, მათი წარმოდგენის ფორმა შეიძლება სრულიად განსხვავებული იყოს. თუ საჭიროა, დასამუშავებელმა საინფორმაციო სისტემამ

ამ ინფორმაციაზე მიმართვა უნდა უზრუნველყოს და აგრეთვე, იგი მომხმარებლისათვის საჭირო ფორმით წარმოადგინოს.

მოცემული ორგანიზაციის სხვა საინფორმაციო სისტემებში არსებული ინფორმაცია შეიძლება იყოს:

- ბუღალტერიაში – საფინანსო საინფორმაციო სისტემები (მაგ.: სისტემა "ბუღალტერია", "ხელფასი");
- საწარმოო განყოფილებაში – საწარმოო საინფორმაციო სისტემები (მაგ.: საამქროს მართვის ავტომატიზებული სისტემა, სისტემა "საწყობი");
- კადრების განყოფილებაში – კადრების საინფორმაციო სისტემები (მაგ.: სისტემა "კადრები");
- მარკეტინგის განყოფილებაში – მარკეტინგული საინფორმაციო სისტემები (მაგ.: ინფორმაცია საქონელზე, ბაზარზე, მომხმარებელზე, ფასები და სხვ.).

2. შესასყიდი ინფორმაცია. პირველ რიგში, ეს არის მონაცემთა ბაზები, რომელთა შექმნა საინფორმაციო სისტემებისათვის უნდა მოხდეს. შეიძლება შემდეგი სახის ინფორმაცია აღვნიშნოთ, რომლებიც მონაცემთა ბაზების სახით უკვე არსებობს:

- სტატისტიკური ინფორმაცია;
- სამისამართო ინფორმაცია;
- იურდიული ინფორმაცია;
- კომერციული ინფორმაცია;
- რეკლამა, პრესის განხილვა და სხვა.

ზემოთ ჩამოთვლილი ინფორმაციებიდან, ამჟამად, უმეტესობის მოპოვება Internet-ის საშუალებითაა შესაძლებელი.

3. დამოუკიდებლად შესაგროვებელი ინფორმაცია თითოეული კონკრეტული საინფორმაციო სისტემისათვის უნიკალურია. მაგალითად, მარკეტინგული საინფორმაციო სისტემისათვის ეს იქნება ინფორმაცია საქონელზე, ბაზარზე, მომხმარებელზე, კონკურენტზე, ფასებზე, გასაღების არხებზე და სხვ. საინფორმაციო სისტემისათვის უნდა ჩატარდეს მონიტორინგი

და ის კონკრეტული ინფორმაცია განისაზღვროს, რომელიც უნდა მოვიპოვოთ და რომელიც მოცემული დარგისათვის იქნება აუცილებელი.

დამოუკიდებლად შესაგროვებელი მონაცემების მოცულობა დაახლოებით შეიძლება მონაცემების საერთო მოცულობის 20%-ს, ხოლო მის შესაგროვებლად გაწეული დანახარჯები მთელი ხარჯის 80%-ს შეადგენდეს.

პროექტების მართვის თეორიის თვალსაზრისით, საინფორმაციო სისტემების დამუშავება და ექსპლუატაცია შეიძლება შემდეგ ეტაპებად დაიყოს:

1. საინფორმაციო სისტემის კონცეფციის დამუშავება (წინასწარი ანალიზი);
2. საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის დაგეგმვა;
3. საინფორმაციო სისტემის დამუშავება;
4. საინფორმაციო სისტემის ექსპლუატაციის დეტალური გეგმა;
5. საინფორმაციო სისტემის ექსპლუატაცია.

განსაკუთრებით დიდი მნიშვნელობა აქვს პირველი და მეორე ეტაპის სწორად წარმართვას, რაც შემდგომში საინფორმაციო სისტემის ექსპლუატაციასა და მის შედეგებზე განსაკუთრებით დიდ გავლენას მოახდენს.

მონაცემთა ბაზები

როგორც ზემოთ იყო აღნიშნული, მონაცემთა ბაზის მართვის სისტემა (მბმს) ნებისმიერი საინფორმაციო სისტემის განუყოფელი ნაწილია. გამოყენებული მბმს-ის ტიპი ჩვეულებრივ საინფორმაციო სისტემის მასშტაბით განისაზღვრება - მცირე საინფორმაციო სისტემებმა შეიძლება ლოკალური მბმს გამოიყენონ, კორპორატიულ საინფორმაციო სისტემებში საჭიროა მძლავრი კლიენტ-სერვერული მბმს, რომელიც უზრუნველყოფს მრავალმომხმარებლიან მუშაობას.

ამჟამად, უფრო ფართოდაა გავრცელებული რელაციური მბმს-ები. ობიექტზე ორიენტირებული მბმს-ის (ObjectStore, Objectivity, O2, Jasmin) აშკარა მიმზიდველობისა და მზარდი პოპულარობის მიუხედავად, ჯერჯერობით ის რელაციური მონაცემთა ბაზები ფლობენ უპირატესობას, რომლებიც კარგადაა

გამართული, განვითარებული, თანმდევი სისტემებით და რომლებიც SQL-92 სტანდარტს აკმაყოფილებენ (ასეთ სისტემებს მიეკუთვნება, მაგალითად, Oracle, Informix, Sybase, DB2, MS SQL Server).

საინფორმაციო სისტემის ორგანიზაციის ტრადიციული მეთოდი არის კლიენტ-სერვერის ორრგოლიანი არქიტექტურა. ამ შემთხვევაში, საინფორმაციო სისტემის მთელი გამოყენებითი ნაწილი სამუშაო სადგურებზეა განთავსებული, ხოლო სერვერის მხარეს წვდომა მხოლოდ მონაცემთა ბაზაზე ხორციელდება. კლიენტის სამუშაო სადგურის განტვირთვისა და ქსელის დატვირთვის შემცირებისათვის გამოიყენება კლიენტ-სერვერის სამრგოლიანი არქიტექტურა. ამ არქიტექტურაში სისტემის კლიენტური ნაწილისა და მონაცემთა ბაზის სერვერის გარდა, დანართის საშუალოდ სერვერი შემოგვყავს. კლიენტის მხარეს მხოლოდ ინტერფეისული მოქმედება სრულდება, ხოლო ინფორმაციის დამუშავების მთელი ლოგიკა დანართის სერვერში ხორციელდება.

მონაცემთა ბაზის დამუშავების დროს აუცილებელია იმ მბმს სპეციფიკის გათვალისწინება, რომლისათვისაც ეს დამუშავება ხორციელდება. ANSI SQL 92 სტანდარტის არსებობის მიუხედავად, პრაქტიკულად ყველა SQL-სერვერი იყენებს თავის SQL რეალიზაციას, რომელიც სტანდარტის გაფართოებას შეიცავს. მიუხედავად ამისა, საწყის ეტაპზე, მონაცემთა ბაზის საერთო სტრუქტურის დამუშავების დროს (კონცეპტუალური მოდელის დონეზე) გამოყენებული მბმს-ს თავისებურებანი შეიძლება არ იქნება გათვალისწინებული.

CASE-საშუალებები

საინფორმაციო სისტემის დაპროექტების პირველი ნაბიჯია საგნობრივი არის ფორმალური აღწერის მიღება, საინფორმაციო სისტემის სრული, არაურთიერთსაწინააღმდეგო ფუნქციური და საინფორმაციო მოდელების აგება. ეს ლოგიკურად რთული, შრომატევადი და ხანგრძლივი სამუშაოა, რომელიც მასში მაღალკვალიფიციური სპეციალისტების მონაწილეობას საჭიროებს. უნდა აღინიშნოს, რომ საინფორმაციო სისტემის შექმნისა და

ფუნქციონირების პროცესში მომხმარებელთა მოთხოვნები შეიძლება შეიცვალოს ან დაზუსტდეს, რაც ასეთი სისტემის დამუშავებას კიდევ უფრო ართულებს.

მითითებულმა სირთულეებმა იმ პროგრამულ-ტექნოლოგიური საშუალებების სპეციალური კლასის, ეგრეთ წოდებული CASE-საშუალებების წარმოშობას შეუწყო ხელი, რომლებიც პროგრამული უზრუნველყოფის დამუშავების ეფექტურობის ამაღლებისთვისაა განკუთვნილი.

ამჟამად, ტერმინი CASE (Computer Aided Software/System Engineering) ძალიან ფართო გაგებით გამოიყენება. პირველად კი ტერმინი CASE მხოლოდ პროგრამული უზრუნველყოფის დამუშავების ავტომატიზაციის საკითხებით შემოისაზღვრებოდა, ხოლო ახლა – მთლიანად რთული საინფორმაციო სისტემების დამუშავების პროცესებით. CASE-საშუალებების ტერმინში იგულისხმება საინფორმაციო სისტემების დამუშავების პროგრამული საშუალებები, მათ შორის ინფორმაციაზე მოთხოვნილების ანალიზი და ფორმულირება, გამოყენებითი პროგრამული უზრუნველყოფის დაპროექტება და მონაცემთა ბაზა, კოდების გენერაცია, ტესტირება, დოკუმენტირება, ხარისხის უზრუნველყოფა, პროექტის მართვა და სხვა პროცესები. CASE-საშუალებები სისტემურ პროგრამულ უზრუნველყოფასა და ტექნიკურ საშუალებებთან ერთად, საინფორმაციო სისტემების დამუშავების სრულ გარემოს ქმნის.

CASE-ტექნოლოგია საინფორმაციო სისტემის დაპროექტების მეთოდოლოგიური და ინსტრუმენტული საშუალებების ნაკრებია, რომელიც საშუალებას გვაძლევს თვალსაჩინო ფორმით მოვახდინოთ საგნობრივი მოდელირება, საინფორმაციო სისტემის დამუშავების ყველა ეტაპზე გავაანალიზოთ ეს მოდელი, დავამუშაოთ მომხმარებლის ინფორმაციული მოთხოვნილების თანახმად მიღებული შესაბამისი დანართები.

დამუშავების საშუალებები

საინფორმაციო სისტემის დაპროექტების დროს, კიდევ ერთი კლასის გადასაწყვეტ ამოცანას მიეკუთვნება საინფორმაციო სისტემის მიზნის მოსახერხებელი და შესაბამისი მომხმარებლის ინტერფეისი. უნდა ვიცოდეთ, რომ ერგონომიული ინტერფეისის ამოცანის ფორმალიზაცია არ ხდება, მაგრამ ამავე დროს, ძალიან მნიშვნელოვანია. მომხმარებლები ხშირად, მთლიანად სისტემის ხარისხზე მსჯელობენ ინტერფეისის ხარისხით. უფრო მეტიც, ინტერფეისის ხარისხზეა დამოკიდებული სისტემის გამოყენების ეფექტურობა.

ინტერფეისის დამუშავება ყოველთვის ითვლებოდა შრომატევად სამუშაოდ, რომელიც დამმუშავებლებს (დეველოპერებს) საკმაოდ დიდ დროს ართმევს. მაგრამ, ბოლო დროს ეგრეთ წოდებული *დანართის დამუშავების ვიზუალური საშუალება* გამოჩნდა, რომელიც მომხმარებლის გრაფიკული ინტერფეისის დამუშავების ამოცანას საკმაოდ ამარტივებს. ამჟამად, პროგრამული უზრუნველყოფის ბაზარზე გამოჩნდა საინფორმაციო სისტემის დამუშავებისათვის განკუთვნილი მრავალი დანართის დამუშავების ვიზუალური საშუალებანი. ისინი პირობითად შეიძლება ორ კლასად გაიყოს:

- სპეციალიზებული საშუალებანი - განსაკუთრებულად განსაზღვრულ მბმს-სთან სამუშაოდ ორიენტირებულია და განკუთვნილია ისეთი ჩვეულებრივი დანართების დასამუშავებლად, რომლებიც მონაცემთა ბაზებს არ იყენებს. ასეთი საშუალებების მაგალითია ფირმა Sybase-ის Power Builder სისტემა;
- უნივერსალური საშუალებანი, რომლებიც შეიძლება გამოყენებულ იქნეს როგორც მონაცემთა ბაზებთან ურთიერთმოქმედ, ისე მონაცემთა ბაზების გამოყენების გარეშე მომუშავე საინფორმაციო დანართების დამუშავებისათვის. ასეთი საშუალებებიდან ყველაზე მეტად ცნობილია ფირმა Borland-ის Borland Delphi და ფირმა Microsoft-ის Visual Basic სისტემები.

თითოეულ მათგანს გარკვეული ღირსება და ნაკლი აქვს, ამიტომ ძნელია რომელიმე მათგანს რაიმე უპირატესობა მივანიჭოთ.

ობიექტზე ორიენტირებული დაპროგრამება საშუალებას იძლევა ნებისმიერი სისტემა უფრო მოქნილი, დინამიკური გავხადოთ და ამით მონაცემთა ბაზისა და დანართების სტრუქტურის მუდმივი გადაწერა გამოირიცხოს.

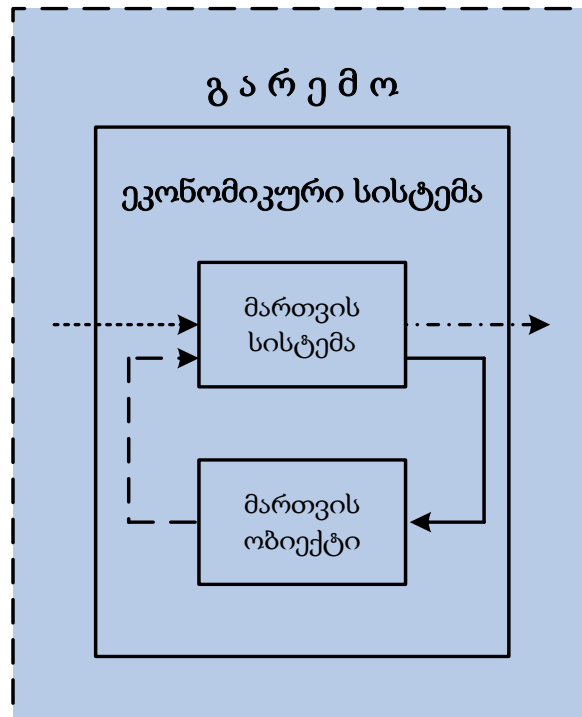
ობიექტზე ორიენტირებული დაპროექტების მთავარი ღირსებაა ადრე დაწერილი კოდების განმეორებითი გამოყენება. ამის გარდა, შესაძლებელია ობიექტური სისტემების მოდიფიკაცია და განვითარება. მონაცემთა ბაზების ეს თვისება საშუალებას იძლევა მომავალი სისტემის დაპროექტება ისე დავიწყოთ, რომ საგნობრივ არეზე ამომწურავი წარმოდგენა არ გვქონდეს. ვინაიდან, საგნობრივ არეზე დეტალური ინფორმაციის მიღება საკმაოდ შრომატევადი პროცესია, მაშინ ობიექტზე ორიენტირებული მიდგომის გამოყენება საშუალებას მოგვცემს სისტემის დამუშავების ვადები და ღირებულება შევამციროთ.

საინფორმაციო სისტემის ცნება

საინფორმაციო სისტემა ორგანიზაციული, ტექნიკური, პროგრამული და ინფორმაციული საშუალებების ერთობლიობაა, რომლებიც ერთიან სისტემაშია გაერთიანებული აუცილებელი ინფორმაციის შეგროვების, შენახვის, გადამუშავებისა და გაცემის მიზნით, მართვის ფუნქციის განსახორციელებლად.

კიბერნეტიკის პოზიციიდან, ეკონომიკური სისტემის მართვის პროცესი, როგორც სისტემის ელემენტებზე მიზნის მისაღწევად მიზანმიმართული ზემოქმედება, შეიძლება ინფორმაციული პროცესის სახით წარმოვადგინოთ, რომელიც გარემოს, მართვის ობიექტსა და მართვის სისტემას ერთმანეთთან აკავშირებს. ამასთან, გარემო და მართვის ობიექტი ინფორმაციას უგზავნის მართვის სისტემას, მისი მდგომარეობის შესახებ. მართვის სისტემა მიღებულ ინფორმაციას აანალიზებს, მართვის ობიექტზე მმართველ ზემოქმედებას გამოიმუშავებს, პასუხობს გარემოს ზემოქმედებაზე და საჭიროების შემთხვევაში, მთელი სისტემის მიზნისა და სტრუქტურის მოდიფიკაციას ახდენს.

კიბერნეტიკის პოზიციიდან ეკონომიკური სისტემის სტრუქტურა ნაჩვენებია 1-ელ ნახატზე, სადაც ძირითადი ინფორმაციული ნაკადები გარემოს, მართვის ობიექტსა და მართვის სისტემას შორის სპეციალური ისრებითა აღნიშნული.



ნახ. 1. ეკონომიკური სისტემის სტრუქტურა

საინფორმაციო სისტემის ინფორმაციული ნაკადები, რომლებიც ერთმანეთთან გარემოს, მართვის ობიექტსა და მართვის სისტემას აკავშირებს შემდეგია:

.....▶ – ინფორმაციული ნაკადი მიმართული გარედან მართვის სისტემაში, რომელიც ერთი მხრივ არის კანონის სახით სახელმწიფო დაწესებულებების მიერ შექმნილი ნორმატიული ინფორმაციის ნაკადი, ხოლო მეორე მხრივ – კონკურენტების მიერ შექმნილი ბაზრის კონიუნქტურის, მომხმარებლების, მიმწოდებლების ინფორმაციული ნაკადი.

- - -▶ – ინფორმაციული ნაკადი მიმართული მართვის სისტემიდან გარეთ, სახელდობრ საანგარიშო ინფორმაცია: არის უპირველეს ყოვლისა საფინანსო ინფორმაცია სახელმწიფო ორგანოებში, ინვესტორების, კრედიტორებისა და მომხმარებლებისათვის; მარკეტინგული ინფორმაცია პოტენციური მომხმარებლებისათვის;

—> – ინფორმაციული ნაკადი მიმართული მართვის სისტემიდან მართვის ობიექტზე (პირდაპირი კიბერნეტიკული კავშირი), რომელიც გეგმური, ნორმატიული და განმკარგულებელი ინფორმაციის ერთობლიობაა და რომელიც სამეურნეო პროცესების განსახორციელებლადაა განკუთვნილი;

—> – ინფორმაციული ნაკადი მიმართული მართვის ობიექტიდან მართვის სისტემაში (უკუკიბერნეტიკული კავშირი), რომელშიც ეკონომიკური სისტემის მართვის ობიექტის მდგომარეობის შესახებ ანგარიში (ნედლეული, მასალები, ფულადი, ენერგეტიკული და შრომითი რესურსები, მზა პროდუქცია და შესრულებული მომსახურება) სამეურნეო პროცესების შესრულების შემდეგ აისახება.

ეკონომიკურ სისტემაში მართვის ობიექტი არის ეკონომიკური საქმიანობის მატერიალური ელემენტებისა (სამრეწველო საწარმოში: ნედლეული და საქონელი, მოწყობილობები, მზა პროდუქცია, მუშახელი და სხვ.) და სამეურნეო პროცესების (სამრეწველო საწარმოში: ძირითადი და დამხმარე წარმოება, მომარაგება, გასაღება და სხვ.) ქვესისტემა.

მართვის სისტემა არის ეკონომიკური სისტემის სტრუქტურული განყოფილებების (მაგალითად, სამრეწველო საწარმოში: დირექცია, საფინანსო, საწარმოო, მომარაგება-გასაღების და სხვა განყოფილებები) ურთიერთშეთანხმებული მუშაობა და რომლებიც მართვის შემდეგ ფუნქციას ახორციელებენ:

- დაგეგმვა არის ფუნქცია, რომელიც ეკონომიკური სისტემის ფუნქციონირების მიზანს დროის სხვადასხვა პერიოდში განსაზღვრავს;
- აღრიცხვა არის ფუნქცია, რომელიც მართვის ობიექტის მდგომარეობას სამეურნეო პროცესების შესრულების შემდეგ ასახავს;
- კონტროლი არის ფუნქცია, რომლის მეშვეობითაც განისაზღვრება აღრიცხვის მონაცემების გადახრა გეგმური და ნორმატიული მონაცემებისაგან;
- ოპერატიული მართვა არის ფუნქცია, რომელიც ყველა სამეურნეო პროცესის რეგულირებას ახორციელებს, რათა განსხვავება გეგმურ და ნორმატიულ მონაცემებს შორის აღმოიფხვრას;
- ანალიზი არის ფუნქცია, რომლის საშუალებით შეიძლება ეკონომიკური სისტემის მუშაობასა და რეზერვს შორის ის ტენდენცია

განვსაზღვროთ, რომელიც შემდგომში ახალი გეგმის შედგენის დროს გამოიყენება.

საინფორმაციო სისტემა აგროვებს, გადაამუშავებს შემოსულ საანგარიშო და არსებულ ნორმატიულ და გეგმურ ინფორმაციებს ანალიტიკურ ინფორმაციად, რომელიც შემდგომ ეკონომიკური სისტემის განვითარების პროგნოზირებისათვის, მისი მიზნების კორექტირებისა და კვლავწარმოების ახალი ციკლისათვის გეგმების შესადგენად გამოიყენება.

საინფორმაციო სისტემებში ინფორმაციის დამუშავებას შემდეგი მოთხოვნები წარედგინება:

- მართვის ფუნქციების რეალიზაციისათვის ინფორმაციის სისრულე და საკმარისობა;
- ინფორმაციის დროული წარმოდგენა;
- მმართველობის დონიდან გამომდინარე, ინფორმაციის საიმედოობის აუცილებელი ხარისხით უზრუნველყოფა;
- ინფორმაციის დამუშავების ეკონომიურობა: მონაცემების დამუშავებაზე გაწეული ხარჯები მიღებულ ეფექტს არ უნდა აღემატებოდეს;
- მომხმარებლის ინფორმაციული ცვალებადი მოთხოვნილებების მიმართ ადაპტურობა (შეგუებადობა).

საინფორმაციო სისტემებში ინფორმაციის დამუშავების ხასიათის შესაბამისად, ეკონომიკური სისტემის მართვის სხვადასხვა დონეზე (ოპერატიული, ტაქტიკური და სტრატეგიული), საინფორმაციო სისტემების შემდეგი ტიპები გამოიყოფა:

- მონაცემთა დამუშავების სისტემები (EDP – Electronic Data Processing);
- მართვის საინფორმაციო სისტემები (MIS – Management Information System);
- გადაწყვეტილების მიღების სისტემა (DSS – Decision Support System).

მონაცემების დამუშავების სისტემა სამეურნეო ოპერაციების აღრიცხვასა და ოპერატიულ რეგულირებას ასრულებს, გარემოსათვის სტანდარტულ დოკუმენტებს (ანგარიშები, ზედდებულები, საგადასახადო ვალდებულებები) ამზადებს. სამეურნეო პროცესების ოპერატიული მართვის ჰორიზონტი ერთიდან რამდენიმე დღემდე შეადგენს, ახდენს რეგისტრაციისა და

ხდომილობის დამუშავების აღრიცხვას, მაგალითად, შეკვეთათა შესრულების გაფორმებას და მონიტორინგს, საწყობში მატერიალური ფასეულობის შემოსავალს და გასავალს, სამუშაო დროის აღრიცხვის ტაბელის წარმოებას და სხვ. ამ ამოცანებს იტერაციული, რეგულარული ხასიათი აქვს, უშუალოდ სამეურნეო საქმიანობის შემსრულებლების მიერ (მუშა, მეკუჭნავე, ადმინისტრატორი და სხვ.) სრულდება და მკაცრად განსაზღვრული ალგორითმების შესაბამისად დოკუმენტების გაფორმებასა და გადაგზავნასთან არის დაკავშირებული. სამეურნეო ოპერაციების შესრულების შედეგების შეტანა მონაცემთა ბაზებში ეკრანული ფორმების საშუალებით ხდება.

მართვის საინფორმაციო სისტემები მართვის ტაქტიკურ დონეზეა ორიენტირებული: სამუშაოს საშუალო ხანგრძლივობის დროით დაგეგმვა, ანალიზი და ორგანიზაცია რამდენიმე კვირის (თვის) განმავლობაში. მაგალითად, ნედლეულის მოწოდება, პროდუქციის გასაღების ანალიზი და დაგეგმვა, საწარმოო პროგრამების შედგენა. მოცემულ ამოცანათა კლასისათვის დამახასიათებელია საშედეგო დოკუმენტების ფორმირებისა და ამოცანის ამოხსნის მკაცრად განსაზღვრული ალგორითმის რეგლამენტირება (პერიოდული გამეორება). მსგავსი ამოცანების ამოხსნის შედეგები წარმოების სხვადასხვა სამსახურების ხელმძღვანელებისათვისაა (მატერიალურ-ტექნიკური მომარაგებისა და გასაღების, საამქროებისა და სხვა უფროსები) განკუთვნილი. ამოცანები ოპერატიული მონაცემების დაგროვებული ბაზის საფუძველზე ამოიხსნება.

გადაწყვეტილების მიღების სისტემა ძირითადად გამოიყენება მართვის ზედა დონეებზე (ფირმის, წარმოების, ორგანიზაციის ხელმძღვანელობა), რომელთაც სტრატეგიული ხანგრძლივი მნიშვნელობა მთელი წლის ან რამდენიმე წლის განმავლობაში აქვთ. ასეთ ამოცანებს სტრატეგიული მიზნის ფორმირების, რესურსების მოზიდვის დაგეგმვის, დაფინანსების წყაროების და სხვა ამოცანები მიეკუთვნება. აღნიშნულ ამოცანებს, როგორც წესი, არარეგულარული ხასიათი აქვს.

იდეალურად ითვლება საინფორმაციო სისტემა, რომელიც ზემოთ ჩამოთვლილი ყველა ტიპს მოიცავს.

საინფორმაციო სისტემების კლასიფიკაცია

საინფორმაციო სისტემების კლასიფიკაცია შესაძლებელია რამდენიმე ნიშნის მიხედვით. განვიხილოთ უფრო ხშირად გამოყენებული კლასიფიკაცია.

კლასიფიკაცია მასშტაბის მიხედვით

საინფორმაციო სისტემები მასშტაბის მიხედვით შემდეგ ჯგუფებად იყოფა (ნახ. 2):

- ერთეული;
- ჯგუფური;
- კორპორატიული.



ნახ. 2. საინფორმაციო სისტემების კლასიფიკაცია მასშტაბის მიხედვით.

ერთეული საინფორმაციო სისტემები

ერთეული საინფორმაციო სისტემების რეალიზაცია, როგორც წესი, ავტონომიური პერსონალური კომპიუტერის საშუალებით ხორციელდება (ქსელი არ გამოიყენება). ასეთი სისტემა შეიძლება რამდენიმე უბრალო დანართს შეიცავდეს, რომლებიც საერთო ინფორმაციული ფონდითაა ურთიერთდაკავშირებული, და ერთი ან ერთ სამუშაო ადგილზე მომუშავე მომხმარებელთა ჯგუფის მუშაობისათვისა არის გათვალისწინებული. მსგავსი დანართები ეგრეთ წოდებული სამაგიდო ანუ ლოკალურ მონაცემთა ბაზების მართვის სისტემების (მბმს) მეშვეობით იქმნება. ლოკალურ მბმს-ს შორის ყველაზე ცნობილია Clarion, Clipper, FoxPro, Paradox, dBase და Microsoft Access.

ჯგუფური საინფორმაციო სისტემები

ჯგუფური საინფორმაციო სისტემები სამუშაო ჯგუფის წევრების მიერ ინფორმაციის კოლექტიურ გამოყენებაზეა ორიენტირებული და ყველაზე ხშირად ლოკალური გამოთვლითი ქსელის ბაზაზე იქმნება. ასეთი დანართების დამუშავების დროს სამუშაო ჯგუფებისათვის მონაცემთა ბაზების სერვერი (ხშირად, SQL-სერვერები ეწოდება) გამოიყენება. არსებობს, საკმაოდ დიდი რაოდენობის სხვადასხვა როგორც კომერციული, ისე თავისუფალი ტიპის SQL-სერვერები. მათ შორის ყველაზე უფრო ცნობილი მონაცემთა ბაზების სერვერებია Oracle, DB2, Microsoft SQL Server, InterBase, Sybase, Inforqix.

კორპორატიული საინფორმაციო სისტემები

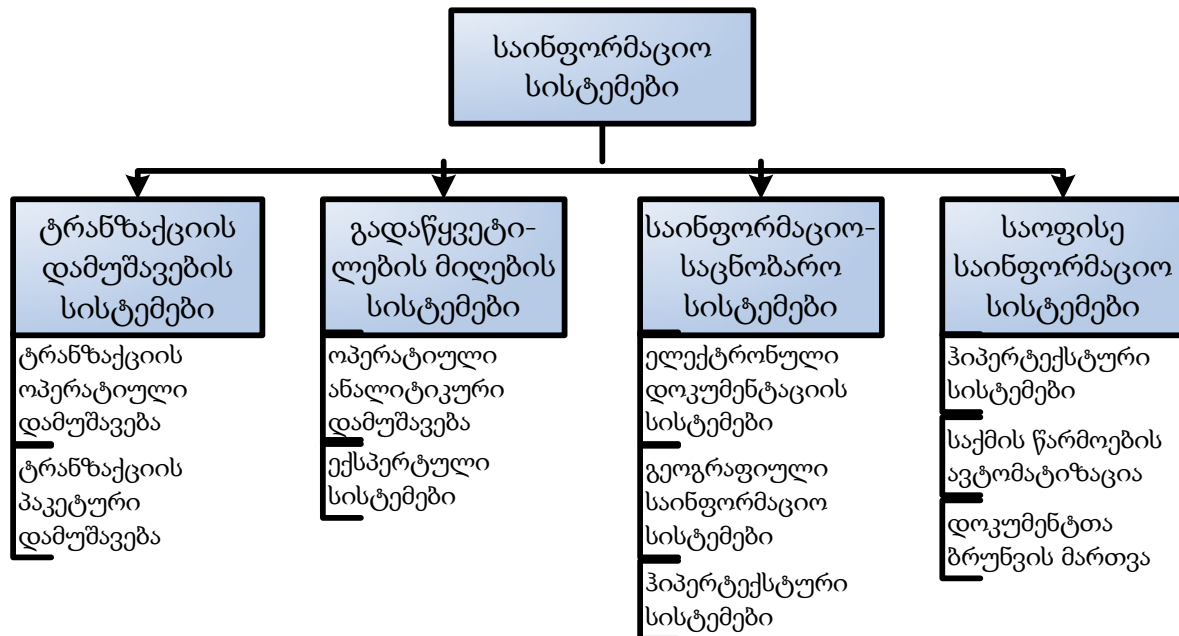
კორპორატიული საინფორმაციო სისტემები სამუშაო ჯგუფებისათვის სისტემის განვითარებას წარმოადგენს, მსხვილ კომპანიებზეა ორიენტირებული, ტერიტორიულად გაბნეული კვანძებისა და ქსელების მხარდაჭერა შეუძლია; ძირითადად რამდენიმე დონისაგან შემდგარი იერარქიული სტრუქტურა აქვს. ასეთი სისტემებისათვის დამახასიათებელია სპეციალიზებული სერვერით კლიენტ-სერვერული ან მრავალდონიანი არქიტექტურა. ასეთი სისტემების დამუშავების დროს შეიძლება გამოყენებულ იქნეს მონაცემთა ბაზების იგივე სერვერები, რომლებიც ჯგუფური საინფორმაციო სისტემების დამუშავების დროს გამოიყენებოდა. მაგრამ, მსხვილი საინფორმაციო სისტემების დამუშავების დროს ყველაზე უფრო გავრცელება პოვა Oracle, DB2 და Microsoft SQL Server სერვერებმა.

ჯგუფური და კორპორატიული საინფორმაციო სისტემებისათვის განსაკუთრებით დიდია ფუნქციონირებისა და მონაცემთა შენახვის საიმედოობის მიმართ მოთხოვნები. ეს თვისებები უზრუნველყოფილია მონაცემთა ბაზების სერვერებში მონაცემთა მთლიანობით, მიმართვითა და ტრანზაქციით მხარდაჭერილი.

კლასიფიკაცია გამოყენების სფეროს მიხედვით

საინფორმაციო სისტემები გამოყენების სფეროს მიხედვით, ჩვეულებრივ ოთხ ჯგუფად იყოფა (ნახ. 3):

- ტრანზაქციის დამუშავების სისტემები;
- გადაწყვეტილების მიღების სისტემები;
- საინფორმაციო-საცნობარო სისტემები;
- საოფისე საინფორმაციო სისტემები.



ნახ. 3. საინფორმაციო სისტემების კლასიფიკაცია გამოყენების სფეროს მიხედვით

ტრანზაქციის დამუშავების სისტემები, თავის მხრივ, მონაცემთა დამუშავების ოპერატიულობის მიხედვით იყოფა პაკეტურ და ოპერატიულ ინფორმაციულ სისტემებად. ორგანიზაციული მართვის საინფორმაციო სისტემებში სჭარბობს ტრანზაქციის ოპერატიული დამუშავების რეჟიმი - OLTP (OnLine Transaction Processing), დროის ნებისმიერ მომენტში საგნობრივი არის მდგომარეობის აქტუალური ასახვისათვის, ხოლო ტრანზაქციის პაკეტურ დამუშავებას საკმაოდ შეზღუდული ნაწილი უჭირავს. OLTP სისტემისათვის საკმაოდ მარტივი ტრანზაქციების რეგულარული (შესაძლებელია, ინტენსიური) ნაკადია დამახასიათებელი, რომელიც შეკვეთების, ფულის გადახდის, შეკითხვების და ა. შ. როლს ასრულებს. მათთვის მნიშვნელოვან მოთხოვნას წარმოადგენს:

- ტრანზაქციის დამუშავების მაღალი მწარმოებლობა;

- დიდ მანძილზე დაცილებული მონაცემთა ბაზებისათვის ინფორმაციის ტელეკომუნიკაციის არხებით გარანტირებული მიწოდება.

გადაწყვეტილების მიღების სისტემები - DSS (Decision Support System) წარმოადგენს სხვა ტიპის საინფორმაციო სისტემებს, რომლებშიც, საკმაოდ რთული შეკითხვების ბაზაზე, მონაცემთა ამორჩევა და ანალიზი სხვადასხვა ჭრილში ხდება: დროითი, გეოგრაფიული და სხვა მაჩვენებლებით.

საინფორმაციო-საცნობარო სისტემების ფართო კლასის საფუძველს ჰიპერტექსტური დოკუმენტები და მულტიმედია წარმოადგენს. ასეთი ტიპის საინფორმაციო სისტემებმა ყველაზე დიდი გავრცელება პოვა ინტერნეტის ქსელში.

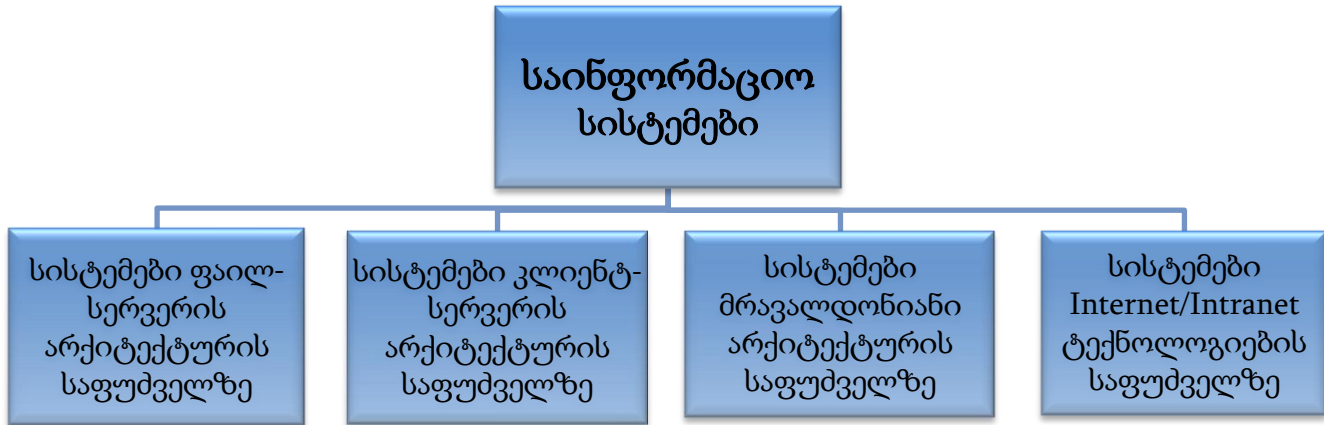
საოფისე საინფორმაციო სისტემების კლასის მიზანია ქაღალდზე არსებული დოკუმენტების ელექტრონული სახით წარმოდგენა, საქმის წარმოების ავტომატიზაცია და დოკუმენტთა ბრუნვის მართვა.

შენიშვნა. უნდა აღინიშნოს, რომ საინფორმაციო სისტემების კლასიფიკაცია, გამოყენების სფეროს მიხედვით, საკმაოდ პირობითია. მსხვილ საინფორმაციო სისტემებს ხშირად ყველა ზემოთ ჩამოთვლილი თვისება აქვს. ამის გარდა, ხშირად, კორპორატიული საინფორმაციო სისტემების შემადგენელი სხვადასხვა ქვესისტემა გამოყენების სხვადასხვა სფეროს მიეკუთვნება.

კლასიფიკაცია ორგანიზაციის წესის მიხედვით

ორგანიზაციის წესის მიხედვით ჯგუფური და კორპორატიული საინფორმაციო სისტემები შემდეგი ტიპის კლასებად იყოფა (ნახ. 4):

- სისტემები ფაილ-სერვერის არქიტექტურის საფუძველზე;
- სისტემები კლიენტ-სერვერის არქიტექტურის საფუძველზე;
- სისტემები მრავალდონიანი არქიტექტურის საფუძველზე;
- სისტემები Internet/Intranet-ტექნოლოგიების საფუძველზე.



ნახ. 4. საინფორმაციო სისტემების კლასიფიკაცია ორგანიზაციის წესის მიხედვით

ნებისმიერ საინფორმაციო სისტემაში შეიძლება გამოიყოს აუცილებელი ფუნქციონალური კომპონენტები (ცხრილი 1), რომლებიც სხვადასხვა არქიტექტურის საინფორმაციო სისტემის შეზღუდვების გაგებაში მოგვეხმარება. საინფორმაციო დანართების აგების ვარიანტების თავისებურებანი უფრო დაწვრილებით განვიხილოთ.

არქიტექტურა ფაილ-სერვერი

არქიტექტურა ფაილ-სერვერს არა აქვს დიალოგ PS და PL კომპონენტების ქსელური გამყოფები და კომპიუტერს გამოსახვის ფუნქციით იყენებს, რაც გრაფიკული ინტერფეისის აგებას ამარტივებს. ფაილ-სერვერი მონაცემებს ფაილიდან მხოლოდ იღებს; ასე რომ, დამატებითი მომხმარებლები და დანართები ცენტრალურ პროცესორზე მხოლოდ უმნიშვნელო დატვირთვას უმატებენ. ყოველი ახალი კლიენტი ქსელში გამოთვლით სიმძლავრეს უმატებს.

ფაილ-სერვერულ დანართში დამუშავების ობიექტები დანართის კომპონენტებია, რომლებიც განსაზღვრავს დიალოგის ლოგიკას PL, ასევე დამუშავების ლოგიკას BL და მონაცემთა მართვას DL. დამუშავებული დანართების რეალიზაცია ან დამთავრებული ჩატვირთვის მოდულის სახით, ან ინტერპრეტაციის სპეციალური კოდის სახით ხდება.

საინფორმაციო სისტემების ტიპური ფუნქციური კომპონენტები

აღნიშვნა	დასახელება	დახასიათება
PS	Presentation Services (წარმოდგენის საშუალებანი)	უზრუნველყოფილია იმ მოწყობილობებით, რომლებიც ინფორმაციას მომხმარებლისაგან იღებენ და იმას ასახავენ, რასაც მას წარმოდგენის ლოგიკის PL კომპონენტი შესაბამისი მხარდამჭერი პროგრამის გამოყენებით მიაწვდის.
PL	Presentation Logic (წარმოდგენის ლოგიკა)	მომხმარებელსა და კომპიუტერს შორის ურთიერთქმედებას მართავს. მომხმარებლის ქმედებას ამუშავებს მენიუში ბრძანების შერჩევის, ღილაკზე დაჭერის ან სიიდან ელემენტის ამორჩევის დროს.
BL	Business or Application Logic (გამოყენებითი ლოგიკა)	დანართის მიერ შესასრულებელი გადაწყვეტილების მიღების, გამოთვლებისა და ოპერაციების წესების ერთობლიობა.
DL	Data Logic (მონაცემთა მართვის ლოგიკა)	მონაცემთა ბაზის ოპერაციები (SQL-ოპერატორები), რომლებიც მონაცემთა მართვის გამოყენებითი ლოგიკის რეალიზაციისათვის უნდა შესრულდეს.
DS	Data Services (ოპერაციები მონაცემთა ბაზასთან)	მონაცემთა მართვის ლოგიკის შესრულებისათვის გამოძახებული მბმს მოქმედებები, როგორცაა მონაცემებით მანიპულირება, მონაცემთა განსაზღვრა, ტრანზაქციის ფიქსაცია და სხვ. მბმს ჩვეულებრივ SQL-წინადადების კომპილაციას ახდენს.
FS	File Services (ფაილური ოპერაციები)	მონაცემთა წაკითხვისა და ჩაწერის დისკური ოპერაციები მბმს-სა და სხვა კომპონენტისათვის. ჩვეულებრივ, ოპერაციული სისტემის ფუნქციებია.

მაგრამ ასეთ არქიტექტურას მნიშვნელოვანი ნაკლი აქვს: მონაცემთა ბაზის მიმართ ზოგიერთი შეკითხვის შესრულების დროს კლიენტს შეიძლება მონაცემთა დიდი მოცულობა გადაეცეს, გადატვირთოს ქსელი და გაუთვალისწინებელი რეაქცია გამოიწვიოს. მნიშვნელოვანი ქსელური ტრაფიკი ფაილ-სერვერზე განსაკუთრებით მკვეთრ ზემოქმედებას მონაცემთა ბაზებზე დიდ მანძილზე დაცილებული, წვდომის დაბალსიჩქარიანი კავშირგაბმულობის არხების საშუალებით ორგანიზების დროს ახდენს. ამ ნაკლის გამოსწორება შეიძლება, თუ დიდ მანძილზე დაცილებული კლიენტისაგან დიალოგის შეტანა-გამოტანა ტელეკომუნიკაციის საშუალებით განხორციელდება. დანართები ძალზე რთული არ უნდა იყოს, წინააღმდეგ შემთხვევაში დიდია სერვერის გადატვირთვის ალბათობა, ან დანართის სერვერისათვის საკმაოდ მძლავრი პლატფორმა იქნება აუცილებელი.

შენიშვნა. ფაილ-სერვერული სისტემის შექმნის ტრადიციული საშუალება არის ლოკალური მბმს, მაგრამ ასეთი სისტემები, როგორც წესი, მონაცემთა მთლიანობას (კერძოდ, მხარს არ უჭერს ტრანზაქციას) არ უზრუნველყოფს. ამიტომ, მათი გამოყენების დროს მონაცემთა მთლიანობის უზრუნველყოფა კლიენტის პროგრამებს ეკისრება, რასაც კლიენტთა დანართების გართულებამდე მივყავართ. მაგრამ ეს ინსტრუმენტები თავისი გამოყენების სიმარტივით, მოხერხებულობით და ხელმისაწვდომობით გამოირჩევა. ამიტომ, ფაილ-სერვერული საინფორმაციო სისტემები, ამჟამად, მცირე სამუშაო ჯგუფებისათვის საკმაოდ საინტერესოა, და უფრო მეტიც, ხშირად საწარმოს საინფორმაციო სისტემების მასშტაბითაც გამოიყენება.

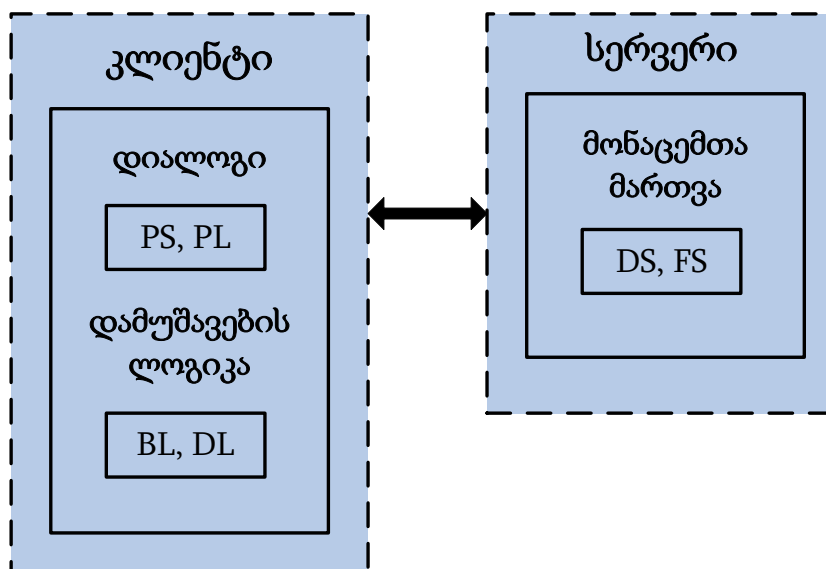
არქიტექტურა კლიენტ-სერვერი

არქიტექტურა კლიენტ-სერვერი, ფაილ-სერვერული დანართების პრობლემების გადასაჭრელად არის განკუთვნილი დანართის კომპონენტების დაყოფისა და მათი იქ განთავსების გზით, სადაც უფრო ეფექტურად იფუნქციონირებს. არქიტექტურა კლიენტ-სერვერის თავისებურებებია მონაცემთა ბაზის გამოყოფილი სერვერის გამოყენება, რომელსაც SQL (Structured Query Language) სტრუქტურირებული შეკითხვების ენაზე

დაწერილი შეკითხვა "ესმის" და ინფორმაციის ძებნას, დალაგებასა და აგრერირებას ახორციელებს.

მონაცემთა ბაზის სერვერის განმასხვავებელი ნიშანია იმ საცნობარო მონაცემების არსებობა, რომელშიც მონაცემთა ბაზის სტრუქტურა, მონაცემთა მთლიანობის შეზღუდვები, ფორმატები და პროგრამაში გამოძახებით ან ხდომილობებით მონაცემთა დამუშავების სერვერული პროცედურებიც კია ჩაწერილი. ასეთ დანართებში დამუშავების ობიექტებს, მონაცემთა დამუშავების დიალოგისა და ლოგიკის გარდა, უპირველეს ყოვლისა, მონაცემთა რელაციური მოდელი და მონაცემთა ბაზის მიმართ ტიპური შეკითხვებისათვის მასთან დამაკავშირებელი SQL-ოპერატორების ნაკრები წარმოადგენს.

კლიენტ-სერვერი კონფიგურაციის უმეტესობა ორდონიან მოდელს იყენებს, რომელშიც კლიენტი სერვერის მომსახურებას მიმართავს. იგულისხმება, რომ დიალოგური კომპონენტები PS და PL კლიენტის მხარეს განთავსდება, რაც გრაფიკულ ინტერფეისს უზრუნველყოფს. მონაცემთა მართვის კომპონენტები DS და FS სერვერზე, ხოლო დიალოგი (PS, PL) და ლოგიკა (BL, DL) - კლიენტის მხარეს განთავსდება. არქიტექტურა კლიენტ-სერვერის ორდონიანი მოდელი ამ ვარიანტს იყენებს: დანართი კლიენტთან მუშაობს, ხოლო მბმს - სერვერზე (ნახ. 5).



ნახ. 5. კლიენტ-სერვერული საინფორმაციო სისტემის კლასიკური ვარიანტი

ვინაიდან ეს სქემა სერვერს უმცირეს მოთხოვნებს უყენებს, იგი ყველაზე კარგ მასშტაბირებას ფლობს. მაგრამ, რთულ დანართებს, რომლებიც მონაცემთა ბაზებთან დიდ ურთიერთქმედებას იწვევს, შეუძლია როგორც კლიენტი, ასევე ქსელი ძლიერ დატვირთოს. SQL-შეკითხვის შედეგები დამუშავებისათვის კლიენტს უნდა დაუბრუნდეს, რადგან იქ გადაწყვეტილების მიღების ლოგიკა განთავსებული. ასეთ სქემას სხვადასხვა კლიენტების კვანძებში გაბნეული დანართების ადმინისტრირების დამატებით გართულებამდე მივყავართ.

ქსელზე დატვირთვის შესამცირებლად და დანართის ადმინისტრირების გასამარტივებლად BL კომპონენტი შეიძლება სერვერზე განთავსდეს. ამასთან, გადაწყვეტილების მიღების მთელი ლოგიკა შენახული პროცედურის სახით ფორმდება და მონაცემთა ბაზის სერვერზე სრულდება.

შენახული პროცედურა არის SQL ოპერატორებით იმ მონაცემთა ბაზაზე წვდომისათვის განკუთვნილი პროცედურა, რომელთა გამოძახება საჭირო პარამეტრების მიწოდებით სახელის საშუალებით ხდება და მონაცემთა ბაზის სერვერზე სრულდება. შეიძლება მოხდეს შენახული პროცედურების კომპილაცია, რაც მის შესრულებას აჩქარებს და სერვერის დატვირთვას ამცირებს.

შენახული პროცედურები დანართებისა და მონაცემთა ბაზის მთლიანობას აუმჯობესებს, კოლექტიურად გამოყენებული ოპერაციებისა და გამოთვლების აქტუალურობის გარანტიას იძლევა. ასეთი პროცედურების თანხლება და აგრეთვე, უსაფრთხოება (არ არის მონაცემების პირდაპირი წვდომა) უმჯობესდება.

შენიშვნა. უნდა გვახსოვდეს, რომ გამოყენებითი ლოგიკით შენახული პროცედურების გადატვირთვამ შეიძლება სერვერის გადატვირთვა გამოიწვიოს, რაც ნაყოფიერების შემცირებამდე მიგვიყვანს. ეს პრობლემა განსაკუთრებით მსხვილი საინფორმაციო სისტემების დამუშავების დროსაა აქტუალური, რომელშიც სერვერს ერთდროულად შეიძლება დიდი რაოდენობის კლიენტებმა მიმართოს. ამიტომ, უმეტეს შემთხვევაში შეიძლება კომპრომისული გადაწყვეტილება მივიღოთ: დანართის ლოგიკის ნაწილი სერვერის მხარეს, ნაწილი კი კლიენტის მხარეს განთავსდეს. ასეთ კლიენტ-სერვერულ სისტემას, ეწოდება სისტემა განაწილებული ლოგიკით. მოცემული

სქემა წარმატებული განაწილებული ლოგიკით საშუალებას იძლევა უფრო ბალანსირებულად დაიტვირთოს კლიენტი და სერვერი.

ამჟამად, კლიენტ-სერვერულმა არქიტექტურამ, როგორც სამუშაო ჯგუფებისა და კორპორატიული დონის საინფორმაციო სისტემებისათვის დანართების ორგანიზაციის საშუალებამ, აღიარება და ფართო გავრცელება პოვა. სამუშაოს ასეთი ორგანიზება, მონაცემთა ბაზის სერვერის შესაძლებლობების გამოყენების, ქსელის განტვირთვისა და მონაცემთა მთლიანობის კონტროლის უზრუნველყოფის საფუძველზე, დანართების შესრულების ეფექტურობას ამაღლებს.

კლიენტ-სერვერული არქიტექტურის ორდონიან სქემებს, მრავალი მომხმარებლითა და ჩახლართული ლოგიკით, რთულ საინფორმაციო დანართებში ზოგიერთი პრობლემის გამოწვევა შეუძლია. ამ პრობლემების გადაწყვეტა მრავალდონიანი არქიტექტურის გამოყენებითაა შესაძლებელი.

მრავალდონიანი არქიტექტურა

მრავალდონიანი არქიტექტურა კლიენტ-სერვერული არქიტექტურის განვითარების შედეგია და თავისი კლასიკური ფორმით, სამი დონისაგან შედგება:

- ქვედა დონე წარმოადგენს კლიენტის დანართებს, რომელიც PS და PL კომპონენტების ფუნქციისა და ლოგიკის შესასრულებლად, ხოლო პროგრამული ინტერფეისი საშუალო დონის დანართების გამოსაძახებლადაა გამოყოფილი;
- საშუალო დონე დანართის სერვერია, რომელზეც სრულდება BL გამოყენებითი ლოგიკა და რომლიდანაც DL მონაცემთა დამუშავების ლოგიკა DS მონაცემთა ბაზასთან ოპერაციას იძახებს;
- ზედა დონე მონაცემთა ბაზის დაცილებული სპეციალიზებული სერვერია, რომელიც DS მონაცემთა დამუშავებისა და FS ფაილებზე ოპერაციის (შენახული პროცედურების გამოყენების რისკის გარეშე) მომსახურებისათვისაა გამოყოფილი.

მონაცემთა დამუშავების მსგავსი კონცეფციის პროპაგანდას ეწევიან Oracle, Sun, Borland და სხვა ფირმები.

სამდონიანი არქიტექტურა სხვადასხვა კვანძებსა და ქსელებზე დატვირთვის ბალანსირების საშუალებას იძლევა, აგრეთვე დანართების დამუშავებისათვის ინსტრუმენტების სპეციალიზაციასა და კლიენტ-სერვერული ორდონიანი მოდელების ნაკლოვანებათა აღმოფხვრას უწყობს ხელს.

დანართის ლოგიკის ცენტრალიზება ადმინისტრირებასა და თანხლებას ამარტივებს. ინტერფეისისა და გამოყენებითი ლოგიკის რეალიზაციისათვის პლატფორმები და ინსტრუმენტები მკაფიოდ არის ერთმანეთისაგან გამოყოფილი, რაც ვიწრო პროფილის სპეციალისტებს საშუალებას აძლევს მთელი ძალისხმევით მოახდინონ მათი რეალიზება. და - ბოლოს, გამოყენებითი ლოგიკის ცვლილება ინტერფეისს არ ეხება, და პირიქით. მაგრამ, რადგანაც საზღვრები PL, BL და DL კომპონენტებს შორის თითქმის არ არის, ამიტომ გამოყენებითი ლოგიკა შეიძლება ნებისმიერ სამ დონეზე გამოჩნდეს. დანართის სერვერი ტრანზაქციის მონიტორის საშუალებით ინტერფეისს უზრუნველყოფს კლიენტებთან და სხვა სერვერებთან, შეუძლია ტრანზაქციის მართვა და მოგვცეს განაწილებული მონაცემთა ბაზის მთლიანობის გარანტია. პროცედურების დაშორებული გამოძახების საშუალებანი განაწილებული გამოთვლების იდეას ყველაზე მეტად შეესაბამება: სხვა კვანძზე განთავსებული გამოყენებითი პროცედურის ქსელის ნებისმიერი კვანძიდან გამოძახებას, პარამეტრების გადაცემას, დაშორებულ დამუშავებასა და შედეგების დაბრუნებას უზრუნველყოფს.

კლიენტ-სერვერული სისტემის ზრდასთან ერთად, სამი დონის არსებობის აუცილებლობა კიდევ უფრო თვალნათლივ ჩანს. სამდონიანი არქიტექტურისათვის, პროდუქტი ეგრეთ წოდებული ტრანზაქციის მონიტორები, შედარებით ახალია. პრაქტიკაში, მონაცემთა ბაზის ერთსა და იმავე სერვერთან შეიძლება შერეული (ორდონიანი და სამდონიანი) არქიტექტურის გამოყენება. გლობალური კავშირების გათვალისწინებით, არქიტექტურას შეიძლება სამ დონეზე მეტი ჰქონდეს. კლიენტ-სერვერული

დანართის ქსელის სხვადასხვა კვანძებში მოქნილი სეგმენტაციისათვის გამოჩნდა ახალი ინსტრუმენტული საშუალებანი.

ამგვარად, განაწილებული დანართების მრავალდონიანი არქიტექტურა კორპორატიული საინფორმაციო სისტემების მუშაობის ეფექტურობის ამაღლებისა და მისი პროგრამულ-აპარატურული რესურსების განაწილების ოპტიმიზაციის საშუალებას იძლევა.

Internet/Intranet-ტექნოლოგიები

ჯერჯერობით Internet/Intranet ტექნოლოგიების განვითარებაში ძირითადი აქცენტი ინსტრუმენტული პროგრამული საშუალებების დამუშავებაზე კეთდება. ამავე დროს, მონაცემთა ბაზებთან მომუშავე დანართების დასამუშავებელი განვითარებული საშუალებანი არ არსებობს. მონაცემთა ბაზებთან ეფექტურად მომუშავე, მოსახერხებელი და გამოყენებაში მარტივი საინფორმაციო სისტემების დამუშავების კომპრომისული გადაწყვეტა Internet/Intranet ტექნოლოგიებისა და მრავალდონიანი არქიტექტურის გაერთიანება გახდა. ამასთან, საინფორმაციო დანართის სტრუქტურა შემდეგ სახეს ღებულობს: ბრაუზერი - დანართების სერვერი - მონაცემთა ბაზის სერვერი - დინამიკური გვერდის სერვერი - Web-სერვერი.

Internet/Intranet ტექნოლოგიებისა და კლიენტ-სერვერული არქიტექტურის ინტეგრაციის წყალობით კორპორატიული საინფორმაციო სისტემების დანერგვა და თანმხლები პროცესები, საკმაოდ მაღალი ეფექტურობისა და მარტივი ერთობლივად გამოსაყენებელი ინფორმაციის შენახვის დროს, მნიშვნელოვნად მარტივდება.

საინფორმაციო სისტემის უზრუნველყოფის ქვესისტემები

საინფორმაციო სისტემის უზრუნველყოფის ქვესისტემები ყველა ტიპის საინფორმაციო სისტემისათვის საერთოა, იმისგან დამოუკიდებლად, თუ როგორ მართვის ფუნქციასა და დონეს მოიცავს ეს საინფორმაციო სისტემა.

უზრუნველყოფის ქვესისტემების შემადგენლობა არ არის დამოკიდებული იმაზე, თუ რა დარგის საინფორმაციო სისტემა უნდა დამუშავდეს. მის შემადგენლობაში შედის: ორგანიზაციული, სამართლებრივი, ტექნიკური, მათემატიკური, პროგრამული, ინფორმაციული, ლინგვისტიკური და ტექნოლოგიური უზრუნველყოფა.

ორგანიზაციული უზრუნველყოფის ქვესისტემა

ორგანიზაციული უზრუნველყოფის ქვესისტემა საინფორმაციო სისტემის ერთ-ერთი ძირითადი ქვესისტემაა, რომელზეც დიდად არის დამოკიდებული სისტემის მიზნებისა და ფუნქციების წარმატებული განხორციელება. ორგანიზაციული უზრუნველყოფის ქვესისტემის შემადგენლობას მიეკუთვნება: მეთოდური მასალები, რომლებიც ახდენს საინფორმაციო სისტემის შექმნისა და ფუნქციონირების პროცესის რეგლამენტირებას; ტიპური პროექტები; ტექნიკური დოკუმენტაცია, რომელიც სისტემის გამოკვლევის, დაპროექტებისა და დანერგვის პროცესში მიიღება; ტექნიკურ-ეკონომიკური დასაბუთება; ტექნიკური დავალება; ტექნიკური და სამუშაო პროექტები; ტიპური გამოყენებითი პროგრამების პაკეტი; სისტემის ექსპლუატაციაში ეტაპობრივი ჩაბარების გაფორმებისათვის აუცილებელი დოკუმენტები; კონსტრუქტორებისა და პერსონალის სამტატო სტრუქტურა და სხვ.

სამართლებრივი უზრუნველყოფის ქვესისტემა

სამართლებრივი უზრუნველყოფის ქვესისტემა საინფორმაციო სისტემების შექმნისა და ექსპლუატაციის პროცესების რეგლამენტირებისათვის არის განკუთვნილი. ის თავისთავში მოიცავს იურდიული დოკუმენტების ერთობლიობას, რომელიც სისტემის საშუალო და საბოლოო ინფორმაციის ფორმირების, შენახვისა და გადამუშავების რეგლამენტს განსაზღვრავს. სამართლებრივ დოკუმენტებს მიეკუთვნება: ხელშეკრულება დამკვეთსა და შემსრულებელს შორის; დასამუშავებელი სისტემის სტატუსის დახასიათება; საინფორმაციო სისტემის ქვესისტემების იურდიული უფლება-მოვალეობანი;

ინფორმაციის დამუშავების პროცესის ცალკეული სახეობების სამართლებრივი უფლებანი; მომხმარებლის ტექნიკური საშუალებების გამოყენებისადმი იურდიული უფლებები და სხვ.

ტექნიკური უზრუნველყოფის ქვესისტემა

ტექნიკური უზრუნველყოფის ქვესისტემა, ინფორმაციულ სისტემაში ინფორმაციის დამუშავებისათვის განკუთვნილი ტექნიკური საშუალებების ერთობლიობაა. მის შემადგენლობაში შედის: კომპიუტერები, რომელთა მეშვეობით ინფორმაციის დამუშავება ხორციელდება; ინფორმაციის შეგროვებისა და რეგისტრაციის საშუალებები; კავშირგაბმულობის არხებით მონაცემების გადაცემის საშუალებები; მონაცემების დაგროვებისა და შენახვის, საშედეგო ინფორმაციის გაცემის საშუალებები; დამხმარე მოწყობილობები და ორგანიზაციული დანიშნულების სხვა ტექნიკა.

მათემატიკური უზრუნველყოფის ქვესისტემა

მათემატიკური უზრუნველყოფის ქვესისტემა არის ამოცანების ამოხსნისა და კომპიუტერის საშუალებით ინფორმაციის დამუშავებისათვის განკუთვნილი მათემატიკური მოდელებისა და ალგორითმების ერთობლიობა, აგრეთვე, მართვის ამოცანის ეკონომიკურ-მათემატიკური მოდელების ასაგები საშუალებებისა და მეთოდების კომპლექსი. მათემატიკური უზრუნველყოფის შემადგენლობაში შედის: მათემატიკური უზრუნველყოფის საშუალებები (მართვის ტიპური ამოცანების მოდელირების საშუალებები, მრავალკრიტერიუმული ოპტიმიზაციის მეთოდები, მათემატიკური სტატისტიკა, მასობრივი მომსახურების თეორია და სხვ.); ტექნიკური დოკუმენტაცია (ამოცანის აღწერა, ამოცანების ამოხსნის ალგორითმები, ეკონომიკურ-მათემატიკური მოდელები); მათემატიკური უზრუნველყოფის შერჩევის მეთოდები (ამოცანათა ტიპების შერჩევის, ალგორითმების შესრულების სირთულის შეფასების და შედეგების სანდოობის შეფასების მეთოდები).

პროგრამული უზრუნველყოფის ქვესისტემა

პროგრამული უზრუნველყოფის ქვესისტემა კომპიუტერული პროგრამების ერთობლიობას, აღწერებსა და მათი გამოყენების ინსტრუქციებს მოიცავს.

პროგრამული უზრუნველყოფა იყოფა ორ ჯგუფად: საერთო (ოპერაციული სისტემები, ოპერაციული გარსები, კომპილატორები, ინტერპრეტატორები, მონაცემთა ბაზები, ქსელური პროგრამები და სხვ.) და სპეციალური (გამოყენებითი პროგრამების ერთობლიობა, რომლებიც დამუშავდა კონკრეტული ამოცანებისათვის და საკონტროლო მაგალითები) დანიშნულების.

ინფორმაციული უზრუნველყოფის ქვესისტემა

ინფორმაციული უზრუნველყოფის ქვესისტემა არის ტექნიკურ-ეკონომიკური ინფორმაციის კლასიფიკაციისა და კოდირების ერთიანი სისტემის ერთობლიობა, დოკუმენტაციისა და ინფორმაციული ბაზის უნიფიცირებული სისტემები.

ინფორმაციული უზრუნველყოფის შემადგენლობა იყოფა ორ ჯგუფად: ინფორმაციული უზრუნველყოფის არაკომპიუტერული კომპონენტები (ტექნიკურ-ეკონომიკური ინფორმაცია და დოკუმენტები) და ინფორმაციული უზრუნველყოფის შიგა კომპიუტერული კომპონენტები (კომპიუტერში შესაყვანი პირველადი მონაცემებისა და საშედეგო ინფორმაციის მისაღები ეკრანული ფორმები და მაკეტები, ინფორმაციული ბაზის სტრუქტურა: ინფორმაციის შესატანი და გამოსატანი ფაილები, მონაცემთა ბაზები).

ინფორმაციული უზრუნველყოფის მთავარი კომპონენტია მონაცემთა ბაზები, რომელთა მეშვეობით სხვადასხვა ამოცანებს შორის მონაცემთა მიმოცვლა ხორციელდება.

ლინგვისტიკური უზრუნველყოფის ქვესისტემა

ლინგვისტიკური უზრუნველყოფის ქვესისტემა მოიცავს საინფორმაციო სისტემაში გამოყენებული სამეცნიერო-ტექნიკური ტერმინებისა და ენის სხვა საშუალებების ერთობლიობას, აგრეთვე ჩვეულებრივი ენის ფორმალიზაციის წესებს, რათა მოხდეს ინფორმაციის ეფექტური დამუშავება და მომხმარებელსა და ინფორმაციულ სისტემას შორის გავამარტივოთ ურთიერთობა გამარტივდეს. ენის საშუალებები, რომლებიც ლინგვისტიკურ უზრუნველყოფაში გამოიყენება, იყოფა ორ ჯგუფად: ტრადიციული ენები (ჩვეულებრივი სალაპარაკო ენა, მათემატიკური, ალგორითმული ენები, მოდელირების ენა) და ენები, რომლებიც კომპიუტერთან დიალოგისათვისაა (ინფორმაციულ-საძიებო ენები, მონაცემთა ბაზების ენა, ოპერაციული გარემოს ენა და სხვ.) განკუთვნილი.

ტექნოლოგიური უზრუნველყოფის ქვესისტემა

ტექნოლოგიური უზრუნველყოფის ქვესისტემა საინფორმაციო სისტემის სხვადასხვა სახის ინფორმაციის დამუშავების ტექნოლოგიური ეტაპების ქვესისტემებად დაყოფას შეესაბამება:

- პირველადი და საბოლოო ინფორმაცია – პირველადი ინფორმაციის შეგროვების, გადაცემის, დაგროვების, შენახვის, დამუშავების, საბოლოო ინფორმაციის მიღებისა და გაცემის ტექნოლოგიურ პროცესს;
- საორგანიზაციო-განმკარგულებელი დოკუმენტაციები – შესატანი დოკუმენტაციის მიღების, შესასრულებლად გადაცემის, საქმეების ფორმირებისა და შენახვის, შიგა დოკუმენტებს, ანგარიშების შედგენისა და გამრავლების ეტაპებს;
- ტექნოლოგიური დოკუმენტაცია და ნახაზები – საწყისი მონაცემების შეტანისა და ახალი პროდუქციის გამოშვებისათვის საპროექტო დოკუმენტაციის შედგენის, ტექნიკური პირობებისა და ნორმატიული მონაცემების ეტაპებს;

- მონაცემთა და ცოდნის ბაზები – მონაცემთა და ცოდნის ბაზების ფორმირების, ამონახსნის საპოვნელად შეკითხვის შეტანისა და დამუშავების, ამონახსნის ვარიანტის გაცემის ეტაპებს;
- სამეცნიერო-კვლევითი ინფორმაცია, სტანდარტები და ტექნიკური პირობები, სამართლებრივი დოკუმენტები და საქმეები – ინფორმაციული ფონდის ფორმირების, შეკითხვის კოდირების, ძეგნის შესრულებისა და დოკუმენტის გაცემის ეტაპებს.

ყველა ზემოთ ჩამოთვლილი საინფორმაციო სისტემის ქვესისტემები დაკავშირებულია ერთმანეთთან.

საინფორმაციო სისტემების გამოყენების სფერო და მისი რეალიზაციის მაგალითები

ბოლო პერიოდში კომპიუტერი საწარმოს მართვის სისტემის განუყოფელი ნაწილი გახდა. მაგრამ მართვის მიმართ თანამედროვე მიდგომა საინფორმაციო ტექნოლოგიებში თანხის დაბანდებასაც გულისხმობს. ამასთანავე, რაც უფრო მსხვილია წარმოება, მით უფრო დიდია მსგავსი დაბანდებანი.

საინფორმაციო ტექნოლოგიების სწრაფმა განვითარებამ, მათი გამოყენების სფეროს გაფართოება გამოიწვია. თითქმის ერთადერთი სფერო, რომელშიც საინფორმაციო სისტემა გამოიყენებოდა, საბუღალტრო აღრიცხვის ავტომატიზაცია იყო, ხოლო ამჟამად საინფორმაციო სისტემები თითქმის ყველა სფეროში გამოიყენება. კორპორატიული საინფორმაციო სისტემების ეფექტური გამოყენება უფრო ზუსტი პროგნოზის გაკეთების, მართვაში შესაძლო შეცდომების და რისკების თავიდან აცილებისა და მაქსიმალური სარგებლის მიღების საშუალებას იძლევა.

წარმოების მართვისათვის სპეციალური პროგრამული საშუალებებით გადასაწყვეტი ზოგიერთი მნიშვნელოვანი ამოცანა განვიხილოთ.

საბუღალტრო აღრიცხვა

საბუღალტრო აღრიცხვის ამოცანა საინფორმაციო ტექნოლოგიების გამოყენების კლასიკური სფეროა და დღესდღეობით ყველაზე ხშირად რეალიზებულ ამოცანას წარმოადგენს. ასეთი მდგომარეობა სავსებით გასაგებია. პირველ რიგში იმიტომ, რომ ბუღალტრის შეცდომა ძალზე ძვირი ჯდება და ამიტომ, ბუღალტერიაში ავტომატიზაციის გამოყენების სარგებელი ცხადია. მეორე მხრივ, საბუღალტრო აღრიცხვის ამოცანის ფორმალიზება საკმაოდ მარტივია, რის გამოც საბუღალტრო აღრიცხვის ავტომატიზებული სისტემის დამუშავება ტექნიკურად რთულ პრობლემას არ წარმოადგენს.

შენიშვნა. მიუხედავად ამისა, საბუღალტრო აღრიცხვის ავტომატიზებული სისტემის დამუშავება საკმაოდ შრომატევადია. ეს დაკავშირებულია

იმასთან, რომ საბუღალტრო აღრიცხვის სისტემის საიმედოობის მიმართ განსაკუთრებულად მაღალი მოთხოვნილებაა და ამასთანავე, მისი ექსპლუატაცია მაქსიმალურად მარტივი და მოსახერხებელი უნდა იყოს.

ფინანსური ნაკადების მართვა

ფინანსური ნაკადების მართვაში საინფორმაციო ტექნოლოგიების დანერგვა ასევე, წარმოების მართვაში დაშვებულ შეცდომებზე ამ სფეროს დიდი გავლენით არის გამოჩვეული. მომწოდებლებსა და მომხმარებლებს შორის ანგარიშსწორების სისტემის არასწორი აგების შემთხვევაში, შესყიდვებისა და გასაღების გამართული ქსელისა და კარგი მარკეტინგის პირობებშიც კი, შეიძლება კრიზისი იქნეს პროვოცირებული. და პირიქით, ზუსტად გათვლილი და მკაცრად კონტროლირებადი ფინანსური გათვლების პირობებში, ფირმის ბრუნვის საშუალებანი შეიძლება მკვეთრად გაიზარდოს.

საწყობის, ასორტიმენტის, შესყიდვების მართვა

შემდეგ შეიძლება საქონლის მოძრაობის ანალიზის პროცესის ავტომატიზება ამით თვალი მივადევნოთ და დავაფიქსიროთ ასორტიმენტის ის ოცი პროცენტი, რომელსაც მთელი მოგების ოთხმოცი პროცენტი მოაქვს. ეს საშუალებას მოგვცემს პასუხი გავცეთ მთავარ კითხვას - საშუალებების მუდმივი ნაკლებობის პირობებში როგორ მივიღოთ მაქსიმალური მოგება?

საწარმოო პროცესების მართვა

საწარმოო პროცესების მართვა ძალზე შრომატევადი ამოცანაა. აქ ძირითად მექანიზმს წარმოადგენს საწარმოო პროცესების დაგეგმარება და ოპტიმალური მართვა.

მსგავსი ამოცანის ავტომატიზებული გადაწყვეტა საშუალებას იძლევა საქმიანად დაიგეგმოს, გათვალისწინებულ იქნეს ხარჯები, განხორციელდეს საწარმოს ტექნიკური მომზადება, პროდუქციის გამოშვების პროცესის ოპერატიული მართვა წარმოების პროგრამისა და ტექნოლოგიის შესაბამისად განხორციელდეს.

ცხადია, რაც დიდია საწარმო, მოგების მიღებაში მით უფრო მეტი ბიზნეს-პროცესი მონაწილეობს, ეს კი ნიშნავს, რომ აუცილებელია საინფორმაციო სისტემის გამოყენება.

მარკეტინგის მართვა

მარკეტინგის მართვა გულისხმობს ფირმა-კონკურენტებზე მონაცემების შეგროვებას, მათი პროდუქციისა და ფასების პოლიტიკის ანალიზს, აგრეთვე ფასების ოპტიმალური დონის განსაზღვრის, მოგების პროგნოზირებისა და სარეკლამო კამპანიის დაგეგმარების მიზნით, გარე პარამეტრების მოდელირებას. შესაძლებელია ამ ამოცანების უმეტესობის ფორმალიზება და მათი საინფორმაციო სისტემის სახით წარმოდგენა, რაც მარკეტინგის მართვის ეფექტურობას მნიშვნელოვნად გაზრდის.

დოკუმენტთა ბრუნვა

დოკუმენტთა ბრუნვა ნებისმიერი საწარმოს საქმიანობის ძალზე მნიშვნელოვანი პროცესია. დოკუმენტთა ბრუნვის კარგად გამართული სისტემა ასახავს წარმოებაში მიმდინარე რეალურ მდგომარეობას და მმართველ რგოლს მასზე ზემოქმედების საშუალებას აძლევს. ამიტომ, დოკუმენტთა ბრუნვის ავტომატიზება ეფექტური მართვის ამაღლების საშუალებას იძლევა.

წარმოების ოპერატიული მართვა

საინფორმაციო სისტემა, რომლის საშუალებითაც წყდება წარმოების ოპერატიული მართვის ამოცანა, აიგება მონაცემთა ბაზაზე, რომელშიც წარმოების შესახებ ყველა შესაძლო ინფორმაცია ფიქსირდება. ასეთი საინფორმაციო სისტემა არის ბიზნესის მართვის ინსტრუმენტი და ჩვეულებრივ მას, კორპორატიულ საინფორმაციო სისტემას უწოდებენ.

ოპერატიული მართვის საინფორმაციო სისტემა თავის თავში მოიცავს კონკრეტულ წარმოებაში ბიზნეს-პროცესების ავტომატიზების მრავალ პროგრამულ ამოცანას. ასეთი საინფორმაციო სისტემების მიმართ ერთ-ერთ ყველაზე მნიშვნელოვანი მოთხოვნაა მოქნილობა, ადაპტაციისა და შემდგომი განვითარების უნარი.

ფირმის შესახებ ინფორმაციის მიწოდება

ინტერნეტის ქსელის აქტიურმა განვითარებამ საწარმოს შესახებ სხვადასხვა სახის ინფორმაციის მიწოდებისათვის, კორპორატიული სერვერების შექმნამდე მიგვიყვანა. პრაქტიკულად, თითქმის ყველა საწარმოს ამჟამად თავისი web-სერვერი აქვს. საწარმოს web-სერვერი მრავალ ამოცანას წყვეტს, მათ შორის შეიძლება ორი ძირითადი გამოვეყნოთ:

- საწარმოს იმიჯის შექმნა;
- კომპანიის საცნობარო სამსახურის, არსებული და პოტენციური მომხმარებლებისათვის ფირმის, სავარაუდო საქონლის, მომსახურებისა და ფასების შესახებ აუცილებელი ინფორმაციის მიწოდების გზით, მაქსიმალური განტვირთვა.

ამის გარდა, web-ტექნოლოგიების გამოყენება ელექტრონული კომერციისა და ინტერნეტის საშუალებით, მყიდველთა მომსახურებისათვის ძალზე პერსპექტიულია.

საინფორმაციო სისტემის სასიცოცხლო ციკლი

კორპორატიული საინფორმაციო სისტემის დამუშავება, როგორც წესი, სრულიად გარკვეული საწარმოსათვის სრულდება. ამ საწარმოს საქმიანობა, რა თქმა უნდა, საინფორმაციო სისტემის სტრუქტურაზე გარკვეულ ზემოქმედებას მოახდენს, მაგრამ ამასთან, სხვადასხვა საწარმოს სტრუქტურა მთლიანობაში ერთმანეთს ჰგავს. ნებისმიერი ორგანიზაცია, მისი საქმიანობის მიუხედავად, შედგება რამდენიმე ქვედანაყოფისაგან, რომლებიც ამ კომპანიის ამა თუ იმ საქმიანობას უშუალოდ ახორციელებს. ამგვარად, ნებისმიერი ორგანიზაცია შეიძლება განვიხილოთ როგორც ურთიერთმოქმედი ელემენტების (ქვედანაყოფების) ერთობლიობა, რომელთაგან თითოეულს შეიძლება საკმაოდ რთული სტრუქტურა ჰქონდეს. ამ ქვედანაყოფებს შორისაც, ურთიერთკავშირი შეიძლება საკმაოდ რთული იყოს. ზოგადად, საწარმოს ქვედანაყოფებს შორის შეიძლება სამი სახის კავშირი არსებობდეს:

- *ფუნქციური კავშირი* - თითოეული ქვედანაყოფი, ერთიანი ბიზნეს-პროცესის ჩარჩოში რაიმე სახის სამუშაოს ასრულებს;
- *ინფორმაციული კავშირი* - ქვედანაყოფები ინფორმაციას (დოკუმენტებს, ფაქსებს, წერილობით და ზეპირ განკარგულებებს და ა. შ.) ერთმანეთში ცვლიან;
- *გარე კავშირები* - ზოგიერთი ქვედანაყოფი გარე სისტემებთან ურთიერთობს, ამასთან, მათი კავშირი როგორც ფუნქციური, ისე ინფორმაციული შეიძლება იყოს.

საინფორმაციო სისტემა, როგორც რაიმე პროექტი მუშავდება. პროექტის მართვისა და დამუშავების ფაზების (სასიცოცხლო ციკლის ფაზების) მრავალი თავისებურება საერთოა და არც პროექტის ხასიათსა და არც მის სფეროზე არ არის დამოკიდებული. ამიტომ, პირველ რიგში პროექტების მართვის ზოგადი საკითხები განვიხილოთ.

პროექტი არის ცალკეული სისტემის მკაცრად განსაზღვრული მიზნითა და დროში შეზღუდული მიზანმიმართული ცვლილება, რომლის მიღწევა ვადების, შედეგების, რისკის, საშუალებებისა და რესურსების ხარჯვის,

ორგანიზაციული სტრუქტურის მიმართ დადგენილი მოთხოვნების მიხედვით განსაზღვრავს პროექტის დამთავრებას.

შენიშვნა. პროექტის განმარტებისათვის საკმაოდ ძნელია ცალსახა ფორმულირება, რომელიც მის განმარტებას სრულად მოგვცემს, ამიტომ მოცემული განმარტება ერთადერთი და სრულყოფილი არ არის.

შეიძლება პროექტის, როგორც მართვის ობიექტის, შემდეგი ძირითადი განმასხვავებელი ნიშნები გამოიყოს:

- ცვალებადობა - სისტემის მიზანმიმართული არსებული მდგომარეობიდან პროექტის მიზნის ტერმინებში აღწერით რაიმე სასურველ მდგომარეობაში გადაყვანა;
- საბოლოო მიზნის შეზღუდულობა;
- ვადებში შეზღუდულობა;
- ბიუჯეტის შეზღუდულობა;
- საჭირო რესურსების შეზღუდულობა;
- სიახლე საწარმოსათვის, რომლისთვისაც ხდება პროექტის რეალიზება;
- კომპლექსურობა - არსებობა დიდი რაოდენობის ფაქტორების, რომლებიც პირდაპირ ან ირიბად პროექტის პროგრესსა და შედეგზე ზემოქმედებს;
- სამართლებრივი და ორგანიზაციული უზრუნველყოფა - პროექტის რეალიზაციის პერიოდში სპეციფიკური ორგანიზაციული სტრუქტურის შექმნა.

პროექტის დაგეგმვისა და მართვის პერიოდში გაცნობიერებულ უნდა იქნეს, რომ საქმე ეხება რაიმე დინამიკური ობიექტის მართვას. ამიტომ, პროექტის მართვის სისტემა საკმაოდ მოქნილი უნდა იყოს, რომ სამუშაო პროგრამაში გლობალური ცვლილებების გარეშე მისი მოდიფიკაცია იქნეს შესაძლებელი.

პროექტების კლასიფიკაცია

პროექტები ერთმანეთისაგან დიდად განსხვავდება სფეროს, შემადგენლობის, მასშტაბების, ხანგრძლივობის, მონაწილეთა შემადგენლობის,

სირთულის, შედეგების მნიშვნელობითა და სხვათა მიხედვით. პროექტების კლასიფიკაცია შეიძლება სხვადასხვა ნიშნის მიხედვით მოხდეს. მოვიყვანოთ ზოგიერთი მათგანი.

პროექტის კლასი პროექტის შემადგენლობითა და სტრუქტურით განისაზღვრება. ჩვეულებრივ განასხვავებენ:

- მონოპროექტს (ცალკეული პროექტი, რომელიც შეიძლება ნებისმიერი ტიპის, სახისა და მასშტაბის იყოს);
- მულტიპროექტს (კომპლექსური პროექტი, რომელიც მონოპროექტები-საგან შედგება და მრავალპროექტიანი მართვის გამოყენებას საჭიროებს).

პროექტის ტიპი განისაზღვრება საქმიანობის ძირითადი სფეროს მიხედვით, რომელშიც პროექტის განხორციელება ხდება. შეიძლება პროექტის ხუთი ძირითადი ტიპი გამოვყოთ:

- ტექნიკური;
- ორგანიზაციული;
- ეკონომიკური;
- სოციალური;
- შერეული.

შენიშვნა. საინფორმაციო სისტემის დამუშავება ტექნიკურ პროექტებს მიეკუთვნება, რომლებსაც შემდეგი თავისებურებანი ახასიათებს:

- პროექტის მთავარი მიზანი მკაცრადაა განსაზღვრული, მაგრამ ცალკეული მიზანი კერძო შედეგების მიღწევის შემდეგ შეიძლება დაზუსტდეს;
- პროექტის დამთავრების ვადები და ხანგრძლივობა წინასწარაა განსაზღვრული; სასურველია მათი ზუსტი დაცვა, მაგრამ მიღებული საშუალებდო შედეგებისა და პროექტის საერთო პროგრესის მიხედვით შეიძლება მათი კორექტირება.

პროექტის მასშტაბი ბიუჯეტის სიდიდისა და მონაწილეთა რაოდენობით განისაზღვრება:

- მცირე პროექტები;
- პატარა პროექტები;
- საშუალო პროექტები;

- დიდი პროექტები.

შეიძლება ასევე უფრო კონკრეტულად განისაზღვროს პროექტის მასშტაბები - დარგობრივი, კორპორატიული, უწყებრივი პროექტები, ერთი საწარმოს პროექტი.

საინფორმაციო სისტემის დაპროექტების ძირითადი ფაზები

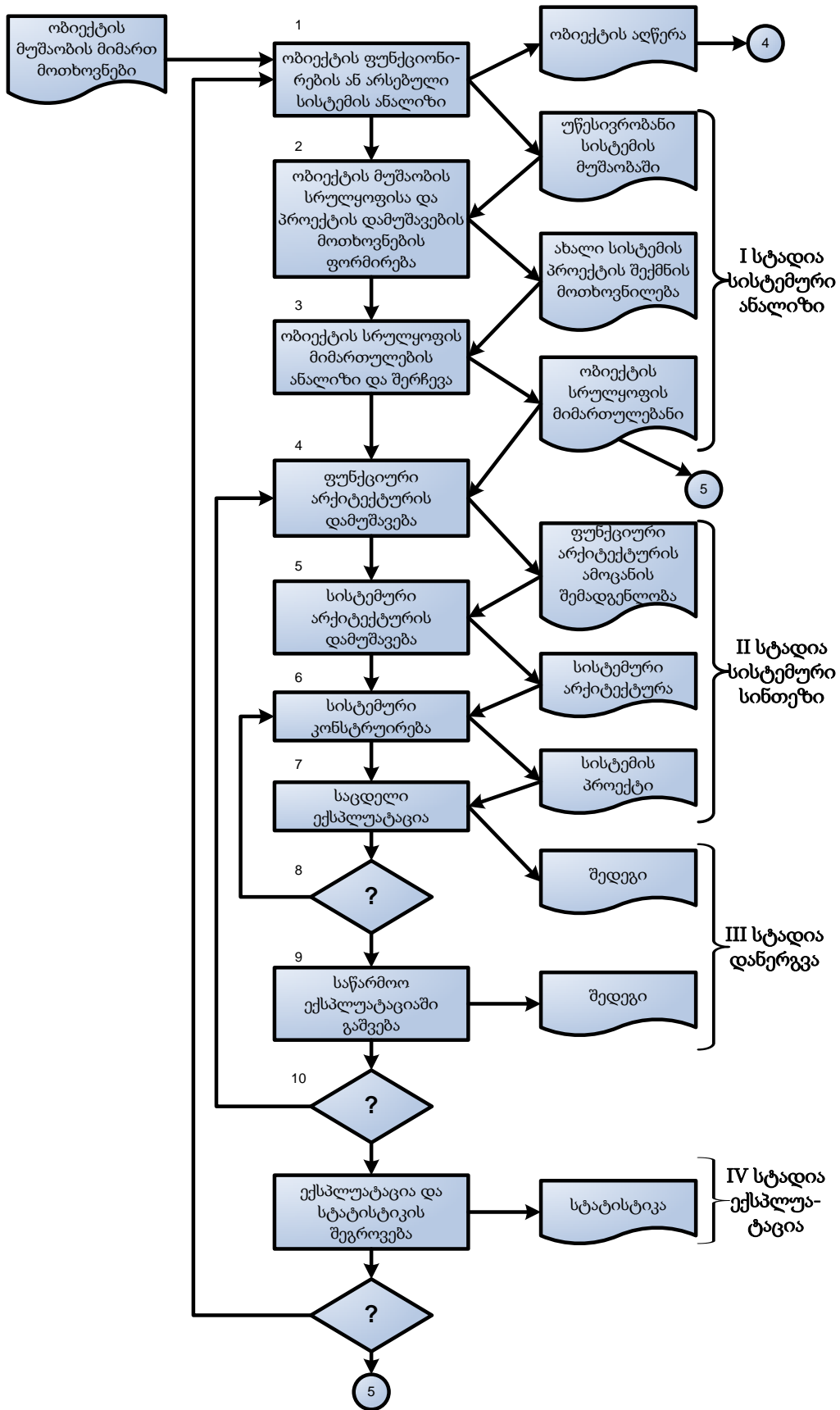
საინფორმაციო სისტემის დაპროექტება ძალზე შრომატევადი, ხანგრძლივი და დინამიკური პროცესია. თანამედრივე პირობებში, დაპროექტების ტექნოლოგია სისტემის ეტაპობრივ დამუშავებას ითვალისწინებს. ეტაპები თავის მიზნის მიხედვით შეიძლება სტადიებად გაერთიანდეს. სტადიებისა და ეტაპების ერთობლიობას, რომელსაც საინფორმაციო სისტემა თავისი განვითარების მანძილზე, სისტემის შექმნის გადაწყვეტილების მიღებიდან სისტემის ფუნქციონირების სრული შეწყვეტის მომენტამდე გაივლის, საინფორმაციო სისტემის სასიცოცხლო ციკლი ეწოდება.

საინფორმაციო სისტემის შექმნის აუცილებლობის დროს, პირველ რიგში უნდა განისაზღვროს კონკრეტულად რა მიზნის მისაღწევად მუშავდეს სისტემა; მეორე, რა დროისათვის არის მიზანშეწონილი სისტემის დამუშავება და მესამე, რა დანახარჯებთანაა დაკავშირებული სისტემის დაპროექტება.

საინფორმაციო სისტემის დამუშავების სასიცოცხლო ციკლის არსი ერთი და იგივეა სხვადასხვა მიდგომისათვის და შემდეგი სტადიების შესრულებამდე დადის:

1. დაგეგმვა და მოთხოვნების ანალიზი (დაპროექტების სტადია) – სისტემური ანალიზი. არსებული საინფორმაციო სისტემის კვლევა და ანალიზი, დასამუშავებელი საინფორმაციო სისტემისადმი მოთხოვნების განსაზღვრა, საინფორმაციო სისტემის დამუშავების ტექნიკურ-ეკონომიკური დასაბუთება და ტექნიკური დავალება;

2. დაპროექტება (ტექნიკური და ლოგიკური დაპროექტება). ავტომატიზებული ფუნქციებისა (ფუნქციური არქიტექტურა) და უზრუნველყოფის ქვესისტემების შემადგენლობის (სისტემური არქიტექტურა) დამუშავება, საინფორმაციო სისტემის ტექნიკური პროექტის გაფორმება;



ნახ. 6. საინფორმაციო სისტემების დამუშავების სასიცოცხლო ციკლი

3. რეალიზაცია (სამუშაო დაპროექტება, ფიზიკური დაპროექტება, დაპროგრამება). პროგრამების დამუშავება, მონაცემთა ბაზის შევსება, პერსონალისათვის სამუშაო ინსტრუქციების დამუშავება, სამუშაო პროექტის გაფორმება;

4. დანერგვა (ტესტირება, საცდელი ექსპლუატაცია). საინფორმაციო სისტემის ქვესისტემების გამართვა, პერსონალის სწავლება, საინფორმაციო სისტემის ექსპლუატაციაში ეტაპობრივი გაშვება, საინფორმაციო სისტემის მიღება-ჩაბარების აქტის გაფორმება;

5. საინფორმაციო სისტემის ექსპლუატაცია. საინფორმაციო სისტემის ფუნქციონირების შესახებ რეკლამაციისა და სტატისტიკის შეგროვება, შეცდომების გასწორება, საინფორმაციო სისტემის მოდერნიზაციის მოთხოვნის გაფორმება და მათი შესრულება (2-5 სტადიის გამეორება).

ხშირად, მეორე და მესამე ეტაპს აერთიანებენ ერთ სტადიად, რომელსაც ტექნიკურ-სამუშაო დაპროექტება ან სისტემური სინთეზი ეწოდება. მე-6 ნახ-ზე საინფორმაციო სისტემის სასიცოცხლო ციკლის ზოგადი ბლოკ-სქემაა მოცემული:

მე-8 ბლოკში უნდა გაირკვეს პროექტის ელემენტებში ცალკეული შეცდომები, რომელთა გასწორება მე-6 ბლოკიდან იწყება.

მე-10 ბლოკში – საწარმო ექსპლუატაციაში გაშვების შემდეგ, უნდა გაირკვეს, თუ რამდენად შეესაბამება დამკვეთის მოთხოვნები პროექტს ან გამოვლინდა თუ არა ფუნქციური არქიტექტურის შეცდომები და სხვ.

მე-12 ბლოკში პირველ პუნქტზე გადასვლა განხორციელდება მაშინ, როდესაც სისტემის მორალური მოძველება მოხდება და მისი მთლიანად შეცვლაა საჭირო, ხოლო მე-5 პუნქტზე გადასვლა იმ შემთხვევაში მოხდება, როცა სისტემური არქიტექტურის მოდიფიკაციაა საჭირო.

ამრიგად, სასიცოცხლო ციკლი უწყვეტი პროცესია სისტემის შექმნის გადაწყვეტილების მიღებიდან დაწყებული, ექსპლუატაციიდან მისი სრული ამოღებით დამთავრებული.

მთავარი ნორმატიული დოკუმენტი, რომელიც სასიცოცხლო ციკლის რეგლამენტირებას ახდენს, არის საერთაშორისო სტანდარტი ISO/IEC 12207 (ISO – International Organization of Standardization – სტანდარტების საერთაშორისო

ორგანიზაცია, ICE – International Electrotechnical Commission – ელექტროტექნიკის საერთაშორისო კომისია). ის სასიცოცხლო ციკლის სტრუქტურას, შემადგენელ პროცესებს, მოქმედებებსა და დასამუშავებელ ამოცანებს განსაზღვრავს. ეს სტანდარტები სასიცოცხლო ციკლის კონკრეტულ მოდელს და მისი დამუშავების მეთოდებს თავს არ გვახვევს. მისი რეგლამენტი ნებისმიერი სასიცოცხლო ციკლის მოდელის, მეთოდოლოგიისა და დამუშავების ტექნოლოგიისათვის საერთოა. ზემოთ მოყვანილი სტანდარტების თანახმად, სასიცოცხლო ციკლის სტრუქტურა პროცესების სამ ჯგუფს ეფუძნება:

- სასიცოცხლო ციკლის ძირითადი პროცესები (შექმენა, მიწოდება, დამუშავება, ექსპლუატაცია, დოკუმენტაციის თანხლება);
- დამხმარე პროცესები, რომლებიც ძირითადი პროცესების შესრულებას უზრუნველყოფს (დოკუმენტირება, კონფიგურაციის მართვა, ხარისხის უზრუნველყოფა, ვერიფიკაცია, ატესტაცია, შეფასება, აუდიტი, პრობლემის გადაწყვეტა);
- ორგანიზაციული პროცესი (პროექტის მართვა, პროექტის ინფრასტრუქტურის შექმენა, სასიცოცხლო ციკლის განსაზღვრა, შეფასება და გაუმჯობესება, სწავლება).

საინფორმაციო სისტემის სასიცოცხლო ციკლის მოდელები

ამჟამად არსებული სასიცოცხლო ციკლის მოდელებს შორის შეიძლება სამი ტიპის მოდელი გამოვყოთ: კასკადური, იტერაციული და სპირალური მოდელები.

კასკადური მოდელი

საინფორმაციო სისტემის დამუშავების კასკადური მოდელი ფართოდ გამოიყენებოდა 70-იან წლებში და 80-იანი წლების დასაწყისში. მისი ძირითადი თვისებაა მოდელის დამუშავების პროცესის ეტაპებად დაყოფა; ამასთან, ერთი ეტაპიდან მეორეზე გადასვლა მხოლოდ და მხოლოდ მას შემდეგ ხორციელდება, როცა წინა ეტაპზე მუშაობა დამთავრდება (ნახ. 7). ყოველი ეტაპი დოკუმენტაციის სრული კომპლექტის გამოშვებით სრულდება. ასეთი

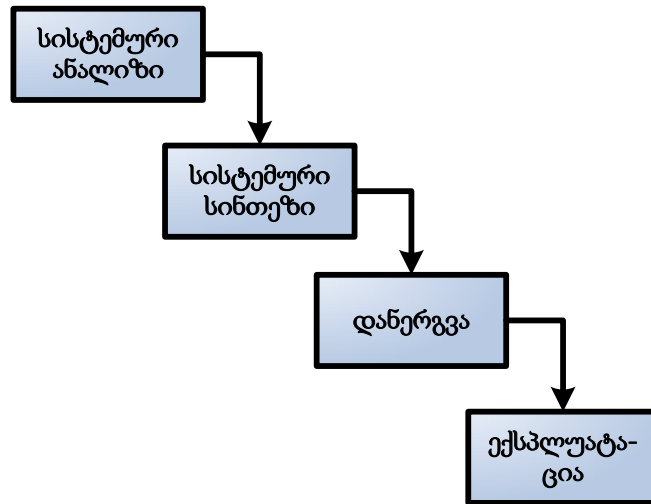
მოდელისათვის დამახასიათებელია იმ ცალკეული ამოცანების ავტომატიზაცია, რომლებიც ერთმანეთთან ინტეგრაციასა და თავსებადობას არ საჭიროებდნენ. აღნიშნულ მოდელს ხშირად, "ჩანჩქერის" სახელწოდებით მოიხსენიებენ. კასკადურ მოდელს რიგი დადებითი მახასიათებელი აქვს, ეს არის:

- ყოველ ეტაპზე დამთავრებული იმ საპროექტო დოკუმენტაციების ფორმირება, რომლებიც სისრულისა და შეთანხმების კრიტერიუმებს პასუხობს. დასკვნით ეტაპზე კი მუშავდება მომხმარებლის დოკუმენტაცია, რომელიც საინფორმაციო სისტემის სტანდარტებით გათვალისწინებულ ყველა სახის - ორგანიზაციულ, მეთოდურ, საინფორმაციო, ტექნიკურ და პროგრამულ უზრუნველყოფას მოიცავს;
- სამუშაოს ეტაპების შესრულების ლოგიკური მიმდევრობა დასრულების ვადებისა და შესაბამისი დანახარჯების დაგეგმვის საშუალებას იძლევა.

მიუხედავად ამისა, კასკადურ მოდელს საკმაოდ მრავალი ნაკლოვანება აქვს, როგორცაა:

- შედეგების დაგვიანებული მიღება;
- ნებისმიერ ეტაპზე წარმოქმნილი შეცდომები, როგორც წესი, შემდეგ ეტაპზე გამოვლინდება, რაც წინა ეტაპზე დაბრუნების აუცილებლობას იწვევს;
- სამუშაოების პარალელურად წარმართვის სიძნელე;
- ყოველი ეტაპი ინფორმაციულად საკმაოდ გადატვირთულია;
- პროექტის მართვის სირთულე;
- რისკის მაღალი დონე და ინვესტიციების არასაიმედოობა.

კასკადური მოდელების რისკის მაღალ დონეზე მიუთითებს კონსალტინგური კომპანიის The Standish Group კვლევის შედეგები, რომელთა მიხედვითაც აშშ-ში კორპორატიული საინფორმაციო სისტემის პროექტების (IT-პროექტები) 31%-ზე მეტი წარუმატებლად მთავრდება; IT-პროექტების თითქმის 53% ბიუჯეტის გადახარჯვით მთავრდება (საშუალოდ 189%, თითქმის ორჯერ); და პროექტების მხოლოდ 16,2% ეტევა ვადებსა და ბიუჯეტში.



ნახ. 7. კასკადური მოდელი

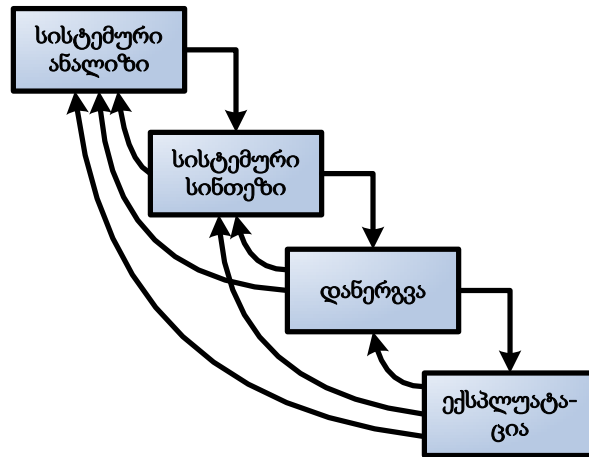
იტერაციული მოდელი

იტერაციული მოდელის (80-იანი წლები) ძირითადი თვისებაა, მას შემდეგ რაც შესრულდა მომდევნო ეტაპი, მოდელის დამუშავების პროცესის წინა ეტაპზე იტერაციული დაბრუნება. საინფორმაციო სისტემის დამუშავების პროცესში თითქმის ყოველთვის წარმოიშობა სიტუაცია, როდესაც ადრე მიღებული გადაწყვეტილების დაზუსტების ან გადახედვის მიზნით, წინა ეტაპზე დაბრუნებაა საჭირო. საინფორმაციო სისტემის დამუშავების ხანგრძლივი სასიცოცხლო ციკლი მთავრდება დანერგვის ეტაპით, რომელსაც ახალი საინფორმაციო სისტემის შექმნის სასიცოცხლო ციკლის დასაწყისი მოსდევს (ნახ. 8).

სპირალური მოდელი

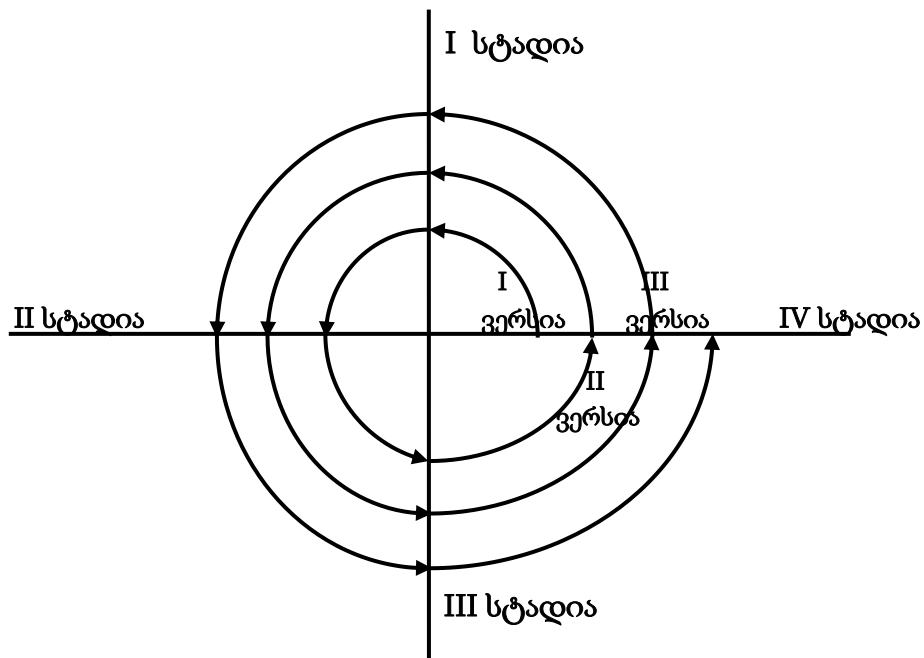
სპირალური მოდელი (90-იანი წლები), კასკადური მოდელისაგან განსხვავებით, საინფორმაციო სისტემის დამუშავების ეტაპობრივ პროცესს გვთავაზობს. საინფორმაციო სისტემის ეტაპობრივი გაფართოება, ძირითადად ეყრდნობა სასიცოცხლო ციკლის საწყის ეტაპებს, როგორცაა: ანალიზი და სინთეზი. სპირალის ყოველი ხვეული სისტემის ფრაგმენტის ან ერთ-ერთი ვერსიის შექმნას შეესაბამება, ამ დროს ზუსტდება პროექტის მიზანი და თვისებები, მისი ხარისხი განისაზღვრება და სპირალის შემდეგი ხვეულის

სამუშაო იგეგმება. ამგვარად, პროექტის დეტალების ჩაღრმავება და დაკონკრეტება ხდება (ნახ. 9).



ნახ. 8. იტერაციული მოდელი

შედეგად მიიღება პროექტის დაზუსტებული ვარიანტი. სპირალური მოდელის ძირითადი პრობლემა შემდეგ ეტაპზე გადასვლის მომენტის განსაზღვრაა. ამის გადასაწყვეტად საჭიროა შემოვიღოთ სასიცოცხლო ციკლის ყოველი ეტაპის დამუშავების დროის შეზღუდვა. გადასვლა გეგმის მიხედვით



ნახ. 9. სპირალური მოდელი

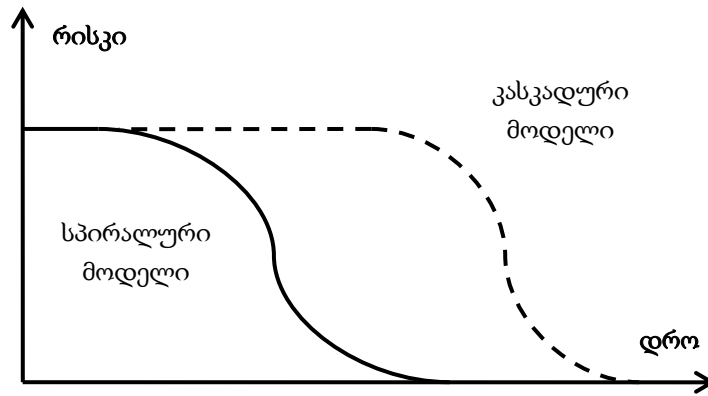
ხორციელდება, მიუხედავად იმისა, დამთავრდა თუ არა მთელ დაგეგმილ ეტაპზე მუშაობა. გეგმა წინა პროექტზე მუშაობის პროცესში დაგროვილ სტატისტიკურ მონაცემებზე დაყრდნობით და დამპროექტებლის პირადი გამოცდილების საფუძველზე დგება.

პროგრამული უზრუნველყოფის დამუშავების სპირალური მოდელი კასკადური მოდელის ნაკლოვანებების უმეტესობის გადალახვის საშუალებასა და ამის გარდა, რიგ დამატებით შესაძლებლობებს იძლევა, დამუშავების პროცესს უფრო მოქნილს ხდის.

განვიხილოთ სპირალური მოდელის ზოგიერთი უპირატესობა:

- პროექტის დამკვეთის მიერ მოთხოვნილებებში ცვლილების შეტანის შემთხვევაში, პროექტში შესაბამისი ცვლილებების განხორციელება ძალზე მარტივდება;
- სპირალური მოდელის გამოყენების შემთხვევაში, საინფორმაციო სისტემის ცალკეული ელემენტის ერთიან სისტემაში ინტეგრირება თანდათანობით, თითქმის უწყვეტად მიმდინარეობს. ვინაიდან ინტეგრაცია მცირე რაოდენობის ელემენტებიდან იწყება, ამიტომაც მისი ჩატარების დროს შედარებით მცირე პრობლემები წარმოიქმნება (ზოგიერთი შეფასებით, დამუშავების კასკადური მოდელის გამოყენების შემთხვევაში, პროექტის ბოლოს ინტეგრაციაზე მთელი დანახარჯის 40% მოდის);

რისკის დონის შემცირება. მოცემული უპირატესობა წინა პუნქტიდან გამომდინარეობს, ვინაიდან რისკი ინტეგრაციის დროს გამოვლინდება. ამიტომ, რისკის დონე პროექტის დამუშავების დასაწყისში მაქსიმალურია. დამუშავების პროცესში მოსალოდნელი რისკი მცირდება. მოცემული მტკიცებულება დამუშავების ნებისმიერი მოდელისათვისაა მართებული, მაგრამ სპირალური მოდელის გამოყენების შემთხვევაში რისკის დონის შემცირება განსაკუთრებით დიდი სიჩქარით ხდება. მე-10 ნახ-ზე, დამუშავების კასკადური და სპირალური მოდელების რისკის დროზე დამოკიდებულების შედარებითი გრაფიკია მოცემული;



ნახ. 10. დამუშავების კასკადური და სპირალური მოდელების რისკის დროზე დამოკიდებულების შედარებითი გრაფიკი

- დამუშავების სპირალური მოდელი საშუალებას იძლევა დასამუშავებელ სისტემაში შევიდეს ტაქტიკური ცვლილებები, რაც პროექტის მართვის მოქნილობას უზრუნველყოფს. მაგალითად, სისტემის ფუნქციურობის შემცირების საფუძველზე, შეიძლება დამუშავების ვადები შემცირდეს;
- კომპონენტების განმეორებითი გამოყენების გამარტივება განპირობებულია იმით, რომ პროექტის საერთო ნაწილების გამოვლენა, მაშინ როდესაც პროექტი ნაწილობრივ დამუშავებულია, შედარებით მარტივია, ვიდრე ეს პროექტის დასაწყისშია შესაძლებელი;
- დამუშავების სპირალური მოდელი საშუალებას იძლევა უფრო საიმედო და მყარი სისტემა მივიღოთ. ეს დაკავშირებულია იმასთან, რომ სისტემის განვითარებასთან ერთად, შეცდომები და სუსტი ადგილები ყოველი სტადიის შემდეგ ვლინდება და სწორდება. ამასთან, ეფექტურობის კრიტიკული პარამეტრების კორექტირება ხდება, რაც კასკადური მოდელის გამოყენების შემთხვევაში, მხოლოდ სისტემის დანერგვის დროს სრულდება;
- დამუშავების სისტემის სრულყოფა - ყოველი სტადიის ბოლოს გაკეთებული ანალიზი დამუშავების ორგანიზებაში განსახორციელებელი ცვლილებებისა და შემდეგ სტადიაზე მისი გაუმჯობესების შეფასების საშუალებას იძლევა.

დამუშავების სპირალური მოდელის ძირითადი პრობლემაა შემდეგ ეტაპზე გადასვლის მომენტის განსაზღვრა. მისი გადაწყვეტისათვის

სასიცოცხლო ციკლის ყოველ ეტაპზე აუცილებელია დროითი შეზღუდვის შემოღება. წინააღმდეგ შემთხვევაში, დამუშავების პროცესი შეიძლება უკვე დამუშავებული სისტემის უსასრულო სრულყოფის სისტემად გადაიქცეს. ამიტომ, დამუშავება გეგმით მკაცრად განსაზღვრულ ვადებში უნდა დამთავრდეს. სამუშაოთა დაგეგმვა ჩვეულებრივ წინა პროექტებში მიღებული სტატისტიკური მონაცემებისა და დამმუშავებლის პირადი გამოცდილების საფუძველზე ხდება.

I ეტაპი – სისტემური ანალიზი

ძალიან ხშირად ორგანიზაციის ხელმძღვანელებს არ შეუძლიათ სწორად განსაზღვრონ, თუ დასამუშავებელმა საინფორმაციო სისტემამ რა უნდა აკეთოს, მისი დამუშავება და ექსპლუატაცია რა დაჯდება. საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის პროექტი შეიძლება ძალიან ძვირი იყოს და ყოველთვის არ არის გარკვეული, ექნება თუ არა მას ორგანიზაციისათვის დადებითი ეფექტი, და თუ ექნება, მაშინ რაოდენობრივად როგორ შევაფასოთ იგი.

რეალური პრობლემა, რომლის გადასაწყვეტად საჭიროა საინფორმაციო სისტემა, ხარისხიანი ინფორმაციის ნაკლებობა შეიძლება იყოს, რაც ჩვენთვის საინტერესო თემატიკის ადეკვატური ანალიზის ჩატარების საშუალებას არ მოგვცემს. იმ პრობლემის განსაზღვრის ეტაპზე, რომლის გადაწყვეტა საინფორმაციო სისტემის საშუალებით უნდა განხორციელდეს, საკითხთა ჯგუფის მიახლოებითი ანალიზი ტარდება:

1. ორგანიზაციის მიზნები და ამოცანები. საინფორმაციო სისტემის მიერ დასამუშავებელი ამოცანების განსაზღვრის წინ, საჭიროა ორგანიზაციის წინაშე მდგომი მიზნები და ამოცანები, აგრეთვე მისი მუშაობის ეფექტურობის შეფასების კრიტერიუმები გაირკვეს. საინფორმაციო სისტემის მიზნები და ამოცანები, აგრეთვე მისი მუშაობის ეფექტურობის კრიტერიუმები, უშუალოდ მასთან უნდა იყოს დაკავშირებული;

2. არსებული ინფორმაციული რესურსები. ტარდება ორგანიზაციის ინფორმატიზაციის მიმდინარე – არსებული საინფორმაციო სისტემების, კვალიფიციური სპეციალისტების, ტექნიკური და პროგრამული საშუალებების, აგრეთვე, ამ რესურსების მომდევნო რამდენიმე წლის

განმავლობაში შესაძლო განვითარების დონის შეფასება. ამ გამოკვლევის დროს უნდა გამოვლინდეს უკვე არსებული ინფორმაციული რესურსები, რომელთანაც მიზანშეწონილი იქნება დასამუშავებელი საინფორმაციო სისტემის კავშირი;

3. საინფორმაციო სისტემის პოტენციურ მომხმარებელთა ჯგუფი და შესაძლო ოპონენტები. საჭიროა, განისაზღვროს საინფორმაციო სისტემის პოტენციურ მომხმარებელთა ჯგუფი ორგანიზაციის შიგნით და მის გარეთ. მიზანშეწონილია, ვიცოდეთ დასამუშავებელი საინფორმაციო სისტემის ძირითადი მოწინააღმდეგენი და პოტენციური მომხრეები;

4. საინფორმაციო სისტემის დამუშავებასა და ექსპლუატაციაზე გამოყოფილი რესურსები. საჭიროა, საინფორმაციო სისტემის დამუშავებასა და ექსპლუატაციაზე გამოყოფილი რესურსების (ფინანსური, საკადრო, ტექნიკური საშუალებანი და სხვ.) შეფასება. მოცემულ შემთხვევაში საჭიროა არა რეალური დანახარჯების შეფასება, არამედ მიახლოებითი შეფასება, ორგანიზაციის ხელმძღვანელობა საინფორმაციო სისტემის დამუშავებისათვის რა რესურსების გამოყოფას აპირებს.

ორგანიზაციის მიზნებისა და ამოცანების, არსებული ინფორმაციული რესურსების, საინფორმაციო სისტემის პოტენციურ მომხმარებელთა ჯგუფის, დამუშავებასა და ექსპლუატაციაზე გამოყოფილი რესურსების ანალიზის საფუძველზე საინფორმაციო სისტემის მიზნების ფორმულირება შეიძლება. ამასთან, მიზანშეწონილია რამდენიმე წლით ადრე მიზნების, ამოცანებისა და რესურსების ცვლილება იქნეს გათვალისწინებული. აგრეთვე, შეიძლება საინფორმაციო სისტემის ექსპლუატაციის ძირითადი მიზნების შემდეგი შესაძლო ვარიანტი იქნეს შემოთავაზებული:

ა) მომხმარებლისათვის საინტერესო სფეროში არსებული ხარისხიანი და ზუსტი ინფორმაციის შეგროვება, ანალიზი და მისთვის მოსახერხებელი ფორმით მიწოდება, საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციისათვის მოცემული რესურსებით;

ბ) მოცემულ სფეროში არსებული ინფორმაციის შეგროვება, ანალიზი და მომხმარებლისათვის მოსახერხებელი ფორმით მიწოდება, რაც საშუალებას გვაძლევს მოცემული სიზუსტით განვახორციელოთ ანალიზი.

ორგანიზაციისა და საინფორმაციო სისტემის მიზნები საჭიროა კარგად შეჯერდეს და შეთანხმდეს, აგრეთვე მათი შესაძლო ცვლილებები რამდენიმე წლით ადრე განისაზღვროს.

საინფორმაციო სისტემის მიზნების განსაზღვრის შემდეგ, საინფორმაციო სისტემის ექსპლუატაციის ეტაპზე დასამუშავებელი ძირითადი ამოცანების ჩამონათვალის მიღების განსაზღვრა ხდება. იმ მიმართულებათა სახით, რომლებზეც კონკრეტული ამოცანების სია უნდა შედგეს, შეიძლება აღვნიშნოთ:

- მომხმარებელთა თითოეული ჯგუფისათვის ინფორმაციის შეგროვება, ანალიზი და მიწოდება;
- საინფორმაციო სისტემის ფუნქციონირების ეფექტურობის კონტროლი;
- საინფორმაციო სისტემის მიმდინარე დამუშავება.

კონკრეტული საინფორმაციო სისტემისათვის უნდა მივუთითოთ (დეტალიზაციის გარეშე) კონკრეტული ამოცანა, რომელიც ამ სისტემამ მომდევნო რამდენიმე წლის განმავლობაში უნდა გადაწყვიტოს – რომელი ინფორმაცია შეგროვდეს, რა სახით და რა სიხშირით გავრცელდეს. ყველა ეს მომენტი ყოველი კონკრეტული საინფორმაციო სისტემისათვის უნიკალურია. საინფორმაციო სისტემის ინფორმაციის შეგროვების, ანალიზისა და გავრცელების ძირითადი ამოცანების სახით შეიძლება აღინიშნოს:

1. შესაგროვებელი ინფორმაციის სავარაუდო ჩამონათვალი;
2. ინფორმაციის წარმოდგენის ძირითადი ფორმები (მონაცემთა ბაზები, Web-საიტები ინტერნეტში, ცნობარები, CD-ROM-ები);
3. თითოეული სახის ინფორმაციისათვის უნდა მივუთითოთ:
 - ინფორმაციის სავარაუდო მოცულობა და მოთხოვნები მის ხარისხზე;
 - განახლების სავარაუდო პერიოდულობა;
 - ამ ინფორმაციის ანალიზის ძირითადი სახეები;
 - ის ძირითადი ფორმები, რა სახითაც ეს ინფორმაცია მომხმარებელს მიეწოდება, და მისი პერიოდულობა.

საინფორმაციო სისტემის მიზნებისა და მომდევნო რამდენიმე წლის განმავლობაში ძირითადი დასამუშავებელი ამოცანების განსაზღვრის შემდეგ, უნდა დამუშავდეს და შეთანხმდეს მისი ფუნქციონირების ეფექტურობის შეფასების კრიტერიუმები. ეს კრიტერიუმები ორგანიზაციისათვის, რომელიც ინფორმაციას თავისთვის თუ გასაყიდად აგროვებს, იქნება სხვადასხვა. ორგანიზაციის მიზნების მიხედვით, ამ კრიტერიუმების სხვადასხვა ვარიანტია შესაძლებელი:

1. საინფორმაციო სისტემის ფარგლებში შესაგროვებელი ინფორმაცია საჭიროა თვით ამ ორგანიზაციისათვის. ამ დროს საინფორმაციო სისტემის ფუნქციონირების ეფექტურობის შეფასების კრიტერიუმებს შეგროვებული ინფორმაციის რაოდენობა და ხარისხი წარმოადგენს და აგრეთვე, თუ რამდენად შეესაბამება იგი მომხმარებლის მოთხოვნებს;

2. საინფორმაციო სისტემის ფარგლებში შესაგროვებელი ინფორმაცია გასაყიდადაა გამიზნული. ამ დროს საინფორმაციო სისტემის ფუნქციონირების ეფექტურობის შეფასების კრიტერიუმი, შეგროვებული ინფორმაციის გაყიდვიდან და მასთან დაკავშირებული მომსახურებით მიღებული შემოსავალი იქნება;

3. საინფორმაციო სისტემა კონკრეტული შეკვეთით მუშავდება. ამ დროს დამმუშავებელი დამკვეთის მიერ განსაზღვრული კრიტერიუმების მიხედვით საინფორმაციო სისტემის ეფექტური ფუნქციონირებითა და ამ სამუშაოს შესრულების შედეგად მიღებული შემოსავლით არის დაინტერესებული.

შემდეგ ეტაპზე მიახლოებითი შეფასება, არსებულ და საჭირო რესურსებს შორის შედარება უნდა ჩატარდეს. რესურსებში იგულისხმება საინფორმაციო სისტემის ცალკე დამუშავებასა და ცალკე ექსპლუატაციაზე გამოყოფილი ფინანსები, კადრები, ტექნიკური და პროგრამული საშუალებანი. აგრეთვე, უნდა მოხდეს საინფორმაციო სისტემის დამუშავებასა და ექსპლუატაციაზე გამოყოფილი დროის, ასევე იმ პერიოდების შეფასება, რომლებსაც მიზანშეწონილია დაიყოს იგი. თითოეული ამ ეტაპისათვის დაწვრილებითი გეგმა დგება, ხოლო მისი დამთავრების შემდეგ – ამ ეტაპის შესრულების ეფექტურობის კონტროლი უნდა განხორციელდეს.

აუცილებლად უნდა გაირკვეს, თუ არსებული რესურსებიდან ორგანიზაციის ხელმძღვანელობა საინფორმაციო სისტემის დამუშავებასა და ექსპლუატაციაზე რის გამოყოფას აპირებს. ამ რესურსების სახით პირველ რიგში აღსანიშნავია:

- საინფორმაციო სისტემის დამუშავებისათვის გამოყოფილი დრო – ჩვეულებრივ, იგი რამდენიმე თვის ტოლია;
- საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციისათვის გამოყოფილი პერსონალი;

- საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციისათვის არსებული და/ან შესაძენი ტექნიკური და პროგრამული საშუალებანი, აგრეთვე კავშირგაბმულობის საშუალებანი;

- საინფორმაციო სისტემის დამუშავებასა და ექსპლუატაციასთან დაკავშირებული ფინანსები, რომელთა დახარჯვა საინფორმაციო სისტემის დამუშავების პროცესში სხვა ორგანიზაციისა და ექსპერტების მოსაზიდად, ინფორმაციის შესაძენად და სხვა ამოცანების გადასაწყვეტად არის საჭირო;

- სხვა რესურსები – შენობა საინფორმაციო სისტემის ექსპლუატაციის სამსახურისათვის და სხვ.

საჭიროა, აგრეთვე, ინფორმაციის შეგროვებისა და გავრცელების სისტემის დაპროექტებისას წინასწარ განისაზღვროს, თუ რა ტექნიკურ და პროგრამულ საშუალებებზე მოხდეს ორიენტირება.

ამჟამად, პერსონალური კომპიუტერები საკმაოდ მძლავრი გახდა, ამიტომ, როგორც წესი, მონაცემთა ბაზებისათვის საკმარისი იქნება ერთი ან რამდენიმე თანამედროვე კომპიუტერი, ხოლო ინფორმაციის შესატანად და მათთან სამუშაოდ საჭირო იქნება გაცილებით მეტი.

ინფორმაციის გასავრცელებლად შეიძლება გამოყენებულ იქნეს Web-საიტი ინტერნეტის ქსელში ან ორგანიზაციის შიგა ქსელი (Intranet). ამ დროს, დიდი მოცულობის ინფორმაციის გავრცელების დანახარჯები, ინფორმაციის გავრცელების სხვა მეთოდებთან შედარებით გაცილებით მცირე იქნება, ხოლო მონაცემთა გავრცელების ოპერატიულობა – გაცილებით მაღალი.

ზემოთ ჩამოთვლილი ეტაპების განხორციელების შემდეგ, საინფორმაციო სისტემის წინასწარი ანალიზის შედეგების შეთანხმება ორგანიზაციის ხელმძღვანელობასა და სხვა დაინტერესებულ პირებთან – პირველ რიგში საინფორმაციო სისტემის პოტენციურ მომხმარებლებთან უნდა მოხდეს. ანალიზის შედეგები წერილობითი სახით უნდა გაფორმდეს.

საინფორმაციო სისტემის რეალიზაციის ანალიზისა და ვარიანტების შერჩევის შედეგების შეთანხმების შემდეგ, საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის პროექტის დეტალური გეგმის შედგენა უნდა მოხდეს.

II ეტაპი – სისტემური სინთეზი

რთული საინფორმაციო სისტემების დაპროექტებას დაახლოებით 3-6 თვე, ხოლო მარტივ სისტემებს - 1-3 თვე ესაჭიროება, ამიტომ მიზანშეწონილია 6-თვიანი სამუშაო პროგრამა შედგეს. შესაძლებელია, დროის სხვა ინტერვალის გამოყენებაც, მაგრამ 6-თვიანი სამუშაო პროგრამა საერთაშორისო პროექტებში ფართოდ გამოიყენება.

თითოეული შემდეგი პროგრამის კორექტირება, წინა პროექტის შედეგების მიხედვით ხდება. ყოველი პროგრამის დასრულების შემდეგ საინფორმაციო სისტემის ფუნქციონირების ანალიზი ტარდება – ყველა დაგეგმილი სამუშაო შესრულდა თუ არა და საინფორმაციო სისტემის ფუნქციონირების ეფექტურობა დაგეგმილ მოთხოვნებს შეესაბამება თუ არა.

წინა ეტაპის ეფექტურობის ანალიზის ჩატარების შემდეგ, შემდგომი პერიოდის დეტალური და 1-2 წლის გეგმების კორექტირება უნდა მოხდეს.

თავის მხრივ, საინფორმაციო სისტემის დამუშავებისათვის ჩასატარებელი სამუშაოები შეიძლება შემდეგნაირად დავეყოთ:

სამუშაო	სამუშაოს დახასიათება
1. მარკეტინგული კვლევა	ინფორმაციის მომხმარებლის მოთხოვნილებათა ანალიზი, არსებული ინფორმაციის ანალიზი
2. საინფორმაციო სისტემის დაპროექტება	ინფორმაციის შეგროვების, ანალიზისა და გავრცელების სქემის დაპროექტება, მონაცემთა სტრუქტურა და სხვ.
3. პროგრამული უზრუნველყოფის დამუშავება	საინფორმაციო სისტემის პროგრამული უზრუნველყოფის, მათ შორის მონაცემთა ბაზების, ინტერნეტის საიტების დამუშავება, ცნობარების დაბეჭდვა
4. საინფორმაციო სისტემის დაგეგმვა, ინფორმაციის ხარისხი	ინფორმაციის შეგროვების, ანალიზისა და გავრცელების კონკრეტული ღონისძიებების დაგეგმვა. პერიოდული მარკეტინგული კვლევების დაგეგმვა, რეკლამები. თითოეული სახის შესაგროვებელი ინფორმაციის ხარისხისა და რაოდენობის დაგეგმვა.

ევროკავშირის ქვეყნებთან საერთაშორისო პროექტების ჩარჩოებში მიღებულია, რომ საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის პროექტებისათვის კონცეფციის დამუშავებაზე, დაგეგმვასა და მარკეტინგულ კვლევაზე გაწეული დანახარჯები მთელი სამუშაოს ღირებულების 10-20%-ს უნდა შეადგენდეს.

იმავე პროექტებში, რომლებიც ინფორმაციის შეგროვებასთანაა დაკავშირებული, ორ წლიანი გეგმა (მხოლოდ არა დეტალური) დგება, ხოლო დეტალური გეგმა ექვსი თვით დგება. ამგვარად, პროექტზე მუშაობის დროს შემდგომი ცვლილებებისა და კორექტივების განხორციელება ძალზე ეფექტურად მიმდინარეობს.

საინფორმაციო სისტემის ექსპლუატაციის პერიოდში თითოეული კონკრეტული პროექტისათვის ყოველი კონკრეტული ამოცანა ძალზე სპეციფიკურია.

საჭირო რესურსების განსაზღვრა

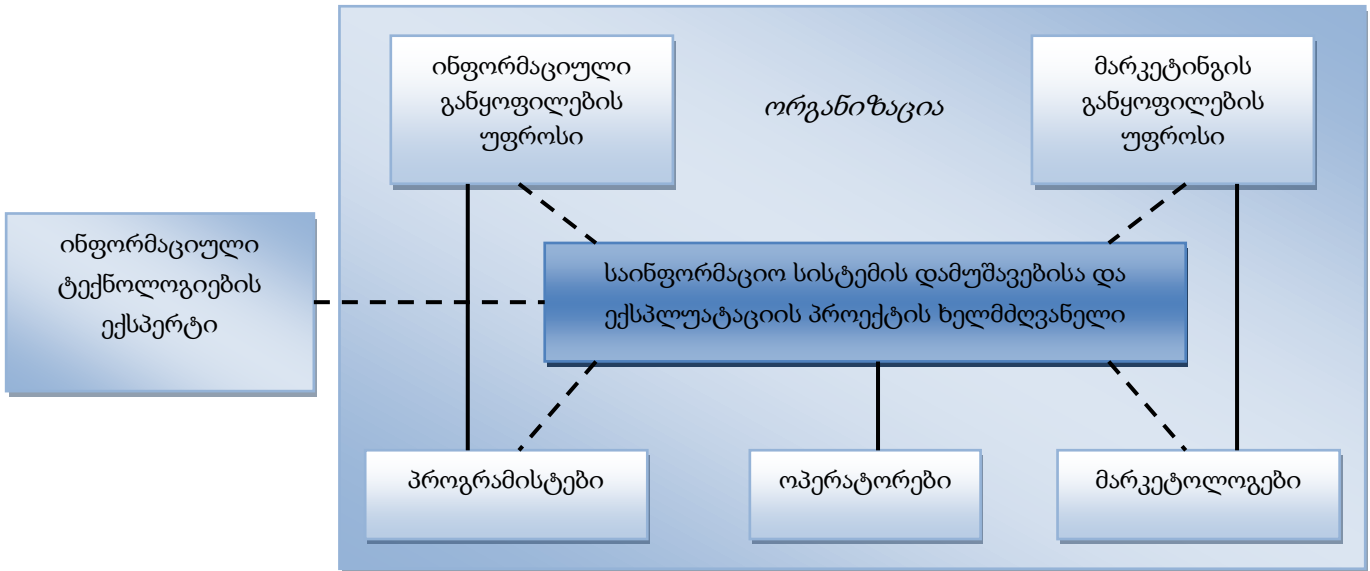
რესურსებში უნდა განვიხილოთ:

- საინფორმაციო სისტემის დამუშავებაზე გამოყოფილი დრო;
- ფინანსები;
- პერსონალი;
- ტექნიკური და პროგრამული საშუალებანი (კომპიუტერები და სხვა მოწყობილობანი, პროგრამული უზრუნველყოფა, კავშირის არხები);
- სხვა რესურსები.

საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის საკითხების შემდეგ, განიხილება რესურსებზე დანახარჯები.

ახლა კი განვიხილოთ საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის სტრუქტურული სქემა.

1. საინფორმაციო სისტემის დამუშავებასა და ექსპლუატაციასთან დაკავშირებული სტრუქტურული სქემა მე-11 ნახაზზეა მოცემული.



ნახ. 11. საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის სტრუქტურული სქემა

მოცემულ სქემაზე ჩანს, რომ ზოგიერთ თანამშრომელს რამდენიმე ხელმძღვანელი ჰყავს, რაც საპროექტო ორგანიზაციებისათვის ტიპური შემთხვევაა. კოორდინაციის ასეთი სქემა ძალზე ეფექტურია, მაგრამ თანამშრომელთა მაღალ კულტურასა და შეგნებას მოითხოვს;

2. საინფორმაციო სისტემის დამუშავებასა და ექსპლუატაციაში დასაქმებულ პერსონალთა ფუნქციური მოვალეობანი მე-2 ცხრილშია მოცემული.

ინფორმაციის კლასიფიკაცია

ინფორმაციის კლასიფიკაცია შესაძლებელია შემდეგი კრიტერიუმების მიხედვით:

- ინფორმაციის სახე;
- ინფორმაციის წარმოშობის ადგილი;
- ინფორმაციის წარმოშობის ფორმა.

საინფორმაციო სისტემის დამუშავებასა და ექსპლუატაციაში დასაქმებულ პერსონალთა ფუნქციური მოვალეობანი

№	თანამდებობა	თანამშრომელთა რაოდენობა	ფუნქცია	შენიშვნა
1.	საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის პროექტის ხელმძღვანელი	1	საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის პროექტის ხელმძღვანელობა	პროექტის ხელმძღვანელი აგრეთვე შეიძლება იყოს საინფორმაციო განყოფილების ან რომელიმე ფუნქციური განყოფილების ხელმძღვანელი, მაგალითად მარკეტინგის განყოფილების.
2.	ინფორმაციული ტექნოლოგიების ექსპერტი	1	ექსპერტი საინფორმაციო სისტემის დამუშავებისა და ექსპლუატაციის საკითხებში	საინფორმაციო სისტემის დამუშავება შესაძლებელია გარედან მოწვეული ექსპერტების დახმარებით
3.	პროგრამისტები	1-4	მონაცემთა ბაზის, ინტერნეტის საიტების, მონაცემთა ანალიზისა და ანგარიშების გენერაციის პროგრამების დამუშავება	შეიძლება მუშაობდნენ მოცემულ ორგანიზაციაში ან მოწვეულნი იყვნენ სპეციალიზებული ფირმებიდან
4.	ოპერატორები	1-...	საინფორმაციო სისტემის ექსპლუატაცია	მოცემული ორგანიზაციის თანამშრომლები

ინფორმაციის კონკრეტული წყაროსათვის შეიძლება ინფორმაციის ზოგიერთი რაოდენობრივი და ხარისხობრივი მახასიათებლები, აგრეთვე მისი მფლობელი მივუთითოთ.

სხვადასხვა დარგისათვის ინფორმაციის სახეობა სხვადასხვა იქნება, მაგრამ არის ცალკეული სახეობა, რომლებიც უმეტესობა დარგისათვის საერთოა:

- სტატისტიკური ინფორმაცია;
- საკანონმდებლო ინფორმაცია;
- სამისამართო ინფორმაცია.

ინფორმაციის წარმოდგენის ფორმა შეიძლება შემდეგი იყოს:

- ინფორმაცია ინტერნეტის ქსელში;
- მონაცემთა ბაზები ინფორმაციის ლოკალურ მატარებლებზე,

მაგალითად, CD-ზე;

- ნაბეჭდი ცნობარები;
- ინფორმაცია პრესაში;
- ინფორმაციის წარმოდგენის სხვა ფორმები.

ინფორმაციის რაოდენობრივ მახასიათებლებს შეიძლება მივაკუთვნოთ:

- ინფორმაციის მოცულობა: მონაცემთა ბაზის ზომა შეიძლება შეფასდეს ჩანაწერთა რიცხვით ან მეგაბაიტებით; ფაილებისა – აგრეთვე, მეგაბაიტებით ან კილობაიტებით; ნაბეჭდი ინფორმაციისა – ტექსტის გვერდების რაოდენობით;

- ინფორმაციის განახლების სიხშირე: ყოველდღიური; ყოველკვირეული; ყოველთვიური; 3 თვეში ერთხელ; 6 თვეში ერთხელ; წელიწადში ერთხელ; იშვიათად; საერთოდ არ ხდება ინფორმაციის განახლება;

- თითოეული სახეობის ინფორმაციის რეკვიზიტები;

- ინფორმაციის პირველად შეტანასა და მის განახლებაზე გაწეული ფულადი, დროითი და სხვა რესურსების დანახარჯები ;

ინფორმაციის ხარისხობრივ მახასიათებლებს შეიძლება მივაკუთვნოთ:

- ინფორმაციის სისრულე – თანაფარდობა არსებულ ჩანაწერთა რიცხვსა და შესაძლო რაოდენობას შორის;

- საიმედოობა – რამდენად შეესაბამება ინფორმაცია რეალობას;

- აქტუალურობა – რამდენად დროულად ხდება ინფორმაციის განახლება;

- რელევანტურობა - რამდენად შეესაბამება მიღებული ინფორმაცია მომხმარებლის მოთხოვნებს.

საინფორმაციო სისტემის პოტენციურ მომხმარებელთა კლასიფიკაცია

პოტენციურ მომხმარებელთა კლასიფიკაციისათვის შეიძლება მთელი რიგი კრიტერიუმები იქნეს გამოყენებული. ორგანიზაციების და არაფიზიკური პირების კლასიფიკაცია განვიხილოთ დაწვრილებით:

- გეოგრაფიული მდებარეობა;
- საკუთრების ფორმა;
- ორგანიზაციის ტიპი;
- ინფორმაციის შეგროვებასთან დამოკიდებულება.

ორგანიზაციის ტიპის მიხედვით შესაძლებელია სრულიად სხვადასხვანაირი კლასიფიკაცია:

- სახელმწიფო მართვის ორგანოები;
- კერძო საწარმოები;
- სააქციო საზოგადოებები;
- ანალიტიკური ინსტიტუტები;
- ორგანიზაციები, რომლებიც მხოლოდ ინფორმაციის შეგროვებითაა დაკავებული;
- სხვა ორგანიზაციები.

ძალზე საინტერესოა ორგანიზაციათა დაყოფა ინფორმაციის შეგროვებასთან დამოკიდებულების მიხედვით:

- ორგანიზაციები, რომელთათვისაც ერთადერთი მიზანი მხოლოდ ერთი სახის ინფორმაციის შეგროვება და გადამუშავებაა;

- “ინფორმაციის სპეციალიზებული წყაროები”. მათთვის შესაგროვებელი ინფორმაციის ვიწრო სპეციალიზაცია, მაღალი საიმედოობა და მისი დროული განახლებაა დამახასიათებელი;

- “ინფორმაციის არასპეციალიზებული წყაროები” ანუ ორგანიზაციები, რომლებიც სხვისი ინფორმაციის გავრცელებითა და თავისი ინფორმაციის შეგროვებით არიან დაკავებულნი. მათთვის მრავალი სახის ინფორმაციის შეგროვება და გადამუშავებაა დამახასიათებელი, მაგრამ ამასთან, ისინი ინფორმაციის ისეთივე მაღალი საიმედოობითა და დროული განახლებით არ გამოირჩევიან, როგორც “ინფორმაციის სპეციალიზებული წყაროები”;

- ორგანიზაციები, რომლებიც მრავალი სახის ინფორმაციის შეგროვებითა და გადამუშავებით არიან დაკავებულნი, მაგრამ ეს მათთვის

არის არასასურველი ქმედება. ამ ჯგუფს მიეკუთვნება უმეტესობა სახელმწიფო და საზოგადოებრივი ორგანიზაციები.

არსებული ინფორმაციის ანალიზი

ამჟამად, თითქმის მოღვაწეობის ყველა სფეროში არსებობს ინფორმაცია, რომელიც პრესაში გამოქვეყნებულია, ანგარიშის სახით, ზოგჯერ მზა მონაცემთა ბაზის სახით ან ინფორმაცია არის ინტერნეტში.

ყოველი მოძიებული ინფორმაციის წყაროს ანალიზი შეიძლება შემდეგი მახასიათებლების მიხედვით განხორციელდეს:

- რა სახის ინფორმაციას ფლობს და მისი ადგილმდებარეობა;
- რა სახით წარმოადგენს ინფორმაციას (ინტერნეტი, მონაცემთა ბაზა, ნაბეჭდი ინფორმაცია და სხვ.);
- ინფორმაციის მფლობელი ორგანიზაცია.

არსებული ინფორმაციის ანალიზი სასურველია მისი წარმოდგენის თითოეული ფორმის მიხედვით ცალკე-ცალკე და ერთად ჩატარდეს. თუ გვაქვს საწყისი მონაცემები, მაშინ ინფორმაციის წარმოდგენის თითოეული სახეობისათვის ანალიზი შეიძლება ჩატარდეს ბოლო წლებში მისი განახლების დინამიკის შესახებ და აგრეთვე, ეს მაჩვენებლები საზღვარგარეთის მაჩვენებლებს შედარდეს.

ზოგჯერ, არსებული ინფორმაციის ანალიზის გარდა, მიზანშეწონილია ჩავატაროთ იმ მონაცემების ანალიზი, რომელიც არ გვაქვს.

არსებული ინფორმაციის ანალიზი საშუალებას მოგვცემს გავაკეთოთ დასკვნები, რომელ სეგმენტში ვფლობთ საკმაოდ სრულად ინფორმაციას და რომელში პრაქტიკულად არა გვაქვს.

განსახილველ სფეროში არსებული ინფორმაციის ანალიზის შემდეგ შეიძლება ჩატარდეს სხვა ქვეყნებსა და რეგიონებში არსებულ ინფორმაციასთან მისი შედარება. ეს ანალიზი გვიჩვენებს საკმარის ინფორმაციას ვფლობთ, თუ არა, რათა განსახილველ სფეროში მიმდინარე მდგომარეობასა და მისი განვითარების პერსპექტივას შორის შედარება მოხდეს.

ინფორმაციის თითოეული სეგმენტისათვის, სხვადასხვა ინფორმაციის წყაროებიდან მათი მოპოვების სირთულის ანალიზი უნდა ჩატარდეს.

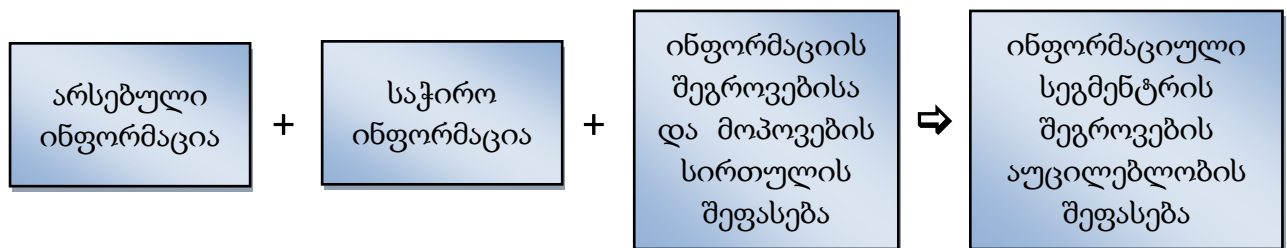
განსაკუთრებით აუცილებელი ინფორმაციის ამორჩევა

აუცილებელი ინფორმაციის ამორჩევის დროს არსებული და საჭირო ინფორმაციის, აგრეთვე მათი შეგროვებისა და მოპოვების სირთულის ანალიზი უნდა ჩატარდეს. აგრეთვე, სხვა ობიექტური თუ სუბიექტური ფაქტორები გათვალისწინებულ უნდა იქნეს: ფირმის გამოცდილება, მისი განვითარების ფაქტორები, ხელმძღვანელთა შეხედულებანი და სხვ. ეს საშუალებას მოგვცემს ცალკეული ინფორმაციული სეგმენტები, რომელთა შეგროვება და მოპოვება არარეალურია ან არამიზანშეწონილია, ამოვიღოთ განხილვიდან. აქვე, შეიძლება მიეთითოს ისეთი ინფორმაციული სეგმენტები, რომელთა არსებობა აუცილებელია.

ინფორმაციის შეგროვებას აზრი აქვს ისეთი სეგმენტებისათვის, რომლებიც მოცემული ორგანიზაციისათვის აუცილებელია, ან რომელთა გაყიდვა შესაძლებელია და გარკვეულ მოგებასაც მოგვცემს. საჭიროა, ინფორმაციის შეგროვებას ისეთი სეგმენტების მიხედვით მივცეთ უპირატესობა, რომელთათვისაც შეიძლება მისაღებ სისრულესა და აქტუალურობას მივაღწიოთ.

როგორც წესი, არ არის მიზანშეწონილი დამოუკიდებლად ისეთი ინფორმაცია შევაგროვოთ, რომელთა დუბლირება შეიძლება მონაცემთა ბაზებში ხდება – ეს ინფორმაციის შეძენაზე უფრო ძვირი დაგვიჯდება. სასურველია, საწყისი ინფორმაციის წყაროს რამდენიმე ალტერნატიული ვარიანტი განვიხილოთ.

არსებული და საჭირო ინფორმაციის, აგრეთვე მათი შეგროვებისა და მოპოვების სირთულის ანალიზი (მასზე პერსონალის დახარჯული დრო და ფინანსები), შესაძლებლობას გვაძლევს თითოეული ინფორმაციული სეგმენტის შეგროვების აუცილებლობა შევაფასოთ:



არსებულ ინფორმაციაში იგულისხმება ის ინფორმაცია, რომელიც უკვე აქვს ნებისმიერ სხვა ორგანიზაციას ან ჩვენი ორგანიზაციის სხვა ქვეგანყოფილებას. ინფორმაციის შეგროვების აუცილებლობა არსებული და

საჭირო ინფორმაციის პირდაპირპროპორციულია, იმავდროულად მისი შეგროვებისა და მოპოვების სირთულის შეფასების უკუპროპორციულია.

არსებული ინფორმაციის შეგროვების აუცილებლობა შეიძლება შემდგომში განვიხილოთ როგორც ამ ინფორმაციის შესყიდვის აუცილებლობა, ხოლო საჭირო ინფორმაციის, რომელიც თითქმის არა გვაქვს, – როგორც მისი შეგროვების აუცილებლობა.

საინფორმაციო სისტემის დაპროექტება

ყველა აუცილებელი ინფორმაციული სეგმენტისათვის, რომელიც საკუთარი ორგანიზაციის ძალებით უნდა შეგროვდეს, ინფორმაციის ხარისხობრივი და რაოდენობრივი მახასიათებლების მინიმალური და სასურველი მნიშვნელობები უნდა განისაზღვროს, როგორცაა:

1. რაოდენობრივი მახასიათებლები:

- ჩანაწერთა რაოდენობა;
- ინფორმაციის განახლების დრო;
- ინფორმაციის შეგროვებასა და მოპოვებაზე გაწეული დანახარჯები;
- სხვა მახასიათებლები.

2. ხარისხობრივი მახასიათებლები:

- ხარისხის საბოლოო შეფასება;
- ინფორმაციის სისრულე, საიმედოობა და აქტუალურობა;
- მოხმარების მოხერხებულობა;
- სხვა მახასიათებლები.

ჯერ, ინფორმაციის ხარისხობრივი მახასიათებლები უნდა შეთანხმდეს ორგანიზაციისა და საინფორმაციო სისტემის დამუშავების ხელმძღვანელებთან, განსაკუთრებით კი საპატიო მომხმარებლებთან. ეს მახასიათებლები, ასევე მომხმარებელთა ყველა იმ ჯგუფს უნდა შეუთანხმდეს, რომლებიც საინფორმაციო სისტემის დამუშავებით არიან დაინტერესებულნი.

ამის შემდეგ, საინფორმაციო ტექნოლოგიების ექსპერტმა უნდა შეაფასოს ის რაოდენობრივი მახასიათებლები, რომლებიც შეესაბამება მოსალოდნელ ხარისხობრივ მახასიათებლებს.

მთელი შესაგროვებელი ინფორმაციისათვის უნდა ჩატარდეს ანალიზი:

1. რამდენად საჭიროა ამ ინფორმაციის ელექტრონული სახით – მონაცემთა ბაზის ან ტექსტის სახით არსებობა. ამისათვის, ყველა შესაგროვებელი ინფორმაციული სეგმენტისათვის უნდა შევაფასოთ:

- რამდენად ხშირად დაგვჭირდება ანალიზისათვის, მოცემული ინფორმაცია, აგრეთვე, შემდგომი გავრცელებისათვის უნდა მოხდეს იმის შეფასება, ეს ინფორმაცია ელექტრონული თუ ტექსტური ფორმით წარმოვადგინოთ;

- მოცემულ ინფორმაციაში ძეხვის ოპერაცია ხშირად არის თუ არა განსახორციელებელი და რამდენად რთული იქნება იგი.

2. თითოეული ინფორმაციის წყაროსათვის უნდა დავადგინოთ, ორგანიზაციის მონაცემთა ბაზაში ამ ინფორმაციის შეყვანა მიზანშეწონილია თუ არა, რადგანაც:

- ეს ინფორმაცია შეიძლება დაგვჭირდეს ძალიან იშვიათად;
- ინფორმაცია ნაბეჭდი ან ხელნაწერი სახით არსებობს, რაც მონაცემთა ბაზაში მის შეტანას არსებითად ართულებს;
- ინფორმაცია ორგანიზაციის მონაცემთა ბაზაში უკვე არსებობს ან მისი შექმნა ძალზე იაფად შეიძლება;
- ინფორმაციის მონაცემთა ბაზაში შეტანით, სხვისი საავტორო უფლებები ირღვევა.

3. რამდენად რთულია ამ ინფორმაციის მონაცემთა ბაზაში შეტანა. ამისათვის, ყველა სახის ინფორმაციისა და მათი წარმოშობის ადგილის მიხედვით უნდა შეფასდეს:

- შესაგროვებელი ჩანაწერების რაოდენობა;
- ინფორმაციის ცვალებადობის სიხშირე;
- რამდენად აქვს აზრი ამ ინფორმაციის დამოუკიდებლად შეგროვებას (შესაძლებელია სხვა ორგანიზაცია ან ამ ორგანიზაციის სხვა ქვეგანაყოფი აპირებს იმავე ინფორმაციის შეგროვებას);
- ინფორმაციის სტრუქტურის სირთულე (ჩანაწერში ველების რაოდენობა).

4. სხვა შენიშვნები:

- მაღალხარისხიანი მონაცემთა ბაზის დამოუკიდებლად შექმნა ძალზე რთულია, განსაკუთრებით ხანგრძლივი პერიოდის მანძილზე;
- ძალზე ხშირად, საჭირო ინფორმაციის დიდი ნაწილი მონაცემთა ბაზის სახით უკვე არსებობს და სხვა ფირმების მიერ იყიდება;
- ინფორმაციული ინდუსტრია ძალზე სწრაფად ვითარდება და იმის დიდი ალბათობა არსებობს, რომ მოცემული ინფორმაცია მონაცემთა ბაზის სახით მალე გამოვა;

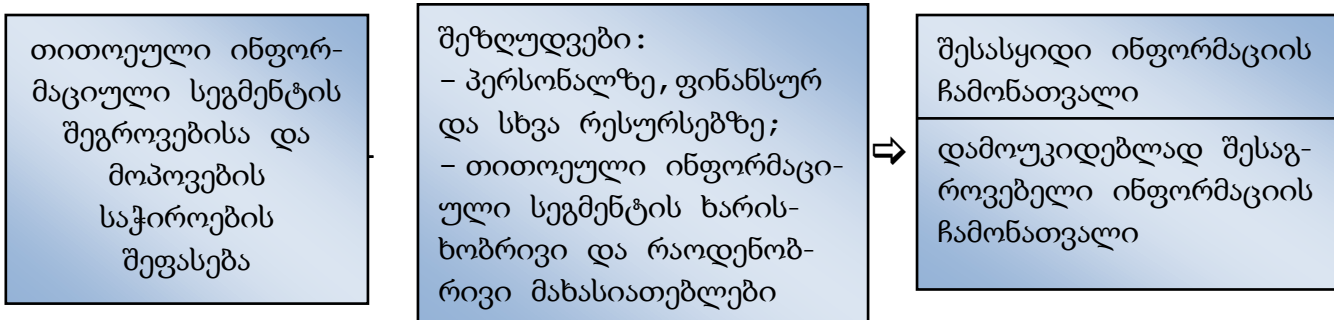
- დიდი მოცულობის ინფორმაციის შეგროვებასა და დამუშავებაზე გაწეული დანახარჯები შეიძლება ბევრად დიდი იყოს; ამიტომ, ზოგჯერ მარტივი, იაფი ტექნიკური და პროგრამული საშუალებებით შესაძლებელია გადავწყვიტოთ მცირე, მაგრამ ორგანიზაციისათვის ძალზე საჭირო ამოცანები.

ზემოთ ჩამოთვლილი მიზეზების გამო, შეიძლება გაკეთდეს შემდეგი დასკვნა:

- თუ ორგანიზაციისათვის საჭირო ინფორმაციული სეგმენტები უკვე შეგროვილია სხვა ორგანიზაციების მიერ, მაშინ, როგორც წესი, უფრო მარტივია მისი შეძენა ან გაცვლა ბარტერის წესით. დამოუკიდებლად, ამ ინფორმაციის შეგროვებაზე შეიძლება დაახლოებით 2-ჯერ მეტი სახსრები იყოს საჭირო;

- დამოუკიდებლად შეიძლება შეგროვდეს ინფორმაციული სეგმენტის მხოლოდ მცირე ნაწილი, რომლის შეძენა ამჟამად ან უახლოეს მომავალში შეუძლებელი იქნება. ეს ინფორმაცია მოცემული ორგანიზაციისათვის სტრატეგიული მნიშვნელობის უნდა იყოს.

განვიხილოთ არსებული მონაცემთა ბაზისა და საკუთარი ძალებით შესაგროვებელი ინფორმაციის წყაროს ამორჩევის პროცედურა. არსებული და საჭირო ინფორმაციის, აგრეთვე მათი შეგროვების სირთულის ანალიზი საშუალებას გვაძლევს თითოეული ინფორმაციული სეგმენტის შეგროვების საჭიროება შევაფასოთ. თუ გვაქვს შესაგროვებელი ინფორმაციისათვის გამოყოფილი ხარჯები და თითოეული ინფორმაციული სეგმენტისათვის მათი ხარისხობრივი და რაოდენობრივი მახასიათებლების მიმართ მოთხოვნები, მაშინ შეიძლება მივიღოთ იმ ინფორმაციის ჩამონათვალი, რომლის შესყიდვაც და რომელთა შეგროვებაც დამოუკიდებლად უნდა მოხდეს.



ინფორმაციის შეგროვების დაპროექტებისას არაფორმალური პარამეტრების მნიშვნელობა ძალიან დიდია. ამიტომ, შესაგროვებელი ინფორმაციის ოპტიმიზაციის განხილული ალგორითმი პირველ რიგში იმისათვისაა განკუთვნილი, რომ:

- თითოეული შესაგროვებელი ინფორმაციული სეგმენტისა და წყაროს ამორჩევის პროცესი არსებითად ობიექტური გავხადოთ;

- თითოეული შესაგროვებელი ინფორმაციული სეგმენტისა და წყაროსათვის სხვადასხვა ალტერნატიული ვარიანტი განვიხილოთ;

- არსებითად ზუსტად მოხდეს ინფორმაციის შეგროვებასა და გავრცელებაზე გასაწევი დანახარჯების დაგეგმვა;

- შესაგროვებელი ინფორმაციისაგან, მოცემული კრიტერიუმების გათვალისწინებით, მაქსიმალური ეფექტი მივიღოთ;

- ამჟამად გავრცელებული შეცდომა გავითვალისწინოთ – როცა ერთხელ მხოლოდ თანამშრომელთა ენთუზიაზმზე საკმაოდ ხარისხიანი ინფორმაციის შეგროვება მოხდება, მაგრამ მისი განახლება ამის შემდეგ ვერ ხერხდება – აღარ რჩება არც ძალა, აღარც საშუალებები და აღარც ენთუზიაზმი.

თითქმის ყოველთვის, უკვე შეგროვებული ინფორმაციის მონაცემთა ბაზის ყიდვა უფრო იაფი ჯდება, ვიდრე მისი დამოუკიდებლად შეგროვება. ინფორმაციის დამოუკიდებლად შეგროვება მხოლოდ მაშინ შეიძლება განხორციელდეს, როდესაც აღნიშნული ინფორმაცია მოცემული ორგანიზაციისათვის ძალზე საჭიროა, და ისიც მხოლოდ მაშინ, როდესაც შეგროვილი ინფორმაციის მოპოვება შეუძლებელია, ან მისი ხარისხი აბსოლუტურად არადადამაკმაყოფილებელია.

არსებული მონაცემთა ბაზის შესყიდვა იმ ინფორმაციული სეგმენტებისათვის უნდა მოხდეს, რომელთა ხარისხობრივი და რაოდენობრივი მახასიათებლები მოცემული ორგანიზაციის მოთხოვნებს პასუხობს და უკეთესია, ვიდრე ის ხელახლა შეგროვების შემდეგ იქნებოდა.

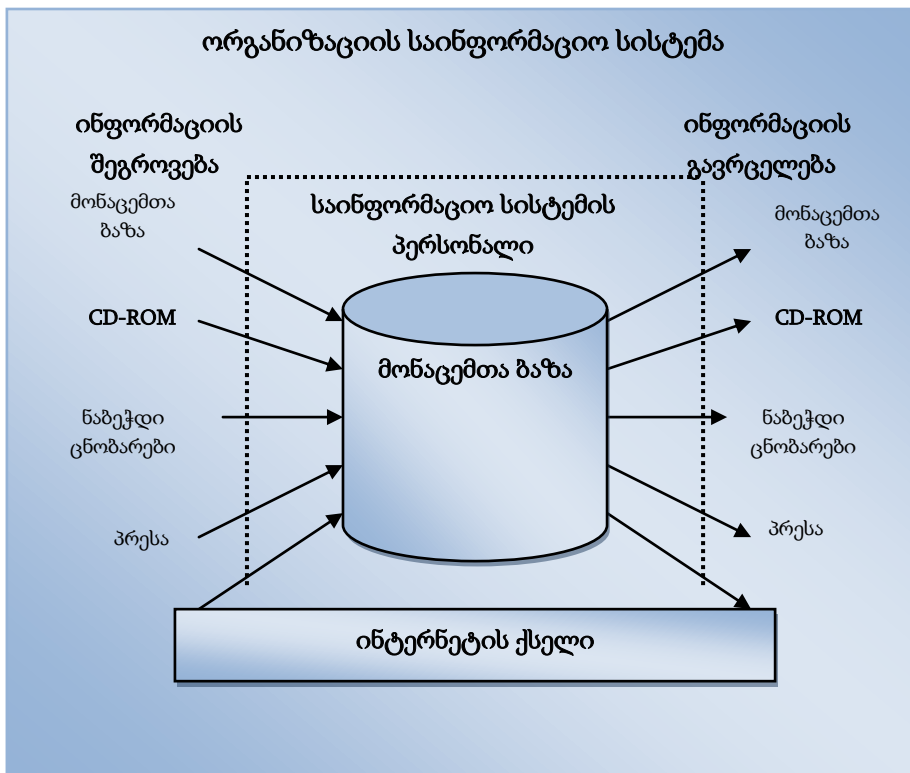
შესასყიდი და შესაგროვებელი ინფორმაციის ხარისხობრივი და რაოდენობრივი მახასიათებლები იმაზე მაღალი არ უნდა იყოს, რასაც მოცემული ორგანიზაცია მოითხოვს. მიზანშეწონილია, შესასყიდი ინფორმაციის ისეთი ჩამონათვალი, მისი ისეთი სისრულე და განახლების ინტერვალი იქნეს ამორჩეული, რომელიც ამ ორგანიზაციისათვის იქნება მისაღები და მის რეალურ მოთხოვნებს ძალიან არ აღემატება.

არსებული ძალებისა და საშუალებებისათვის შეიძლება შესაგროვებელი ინფორმაციის სხვადასხვა ალტერნატიული ვარიანტები გავითვალისწინოთ. სეგმენტების ასარჩევად, მთელი სეგმენტის საბოლოო ხარისხობრივი და რაოდენობრივი მახასიათებლები გამოვიყენოთ:

- ხარისხობრივი მახასიათებლები: ინფორმაციის სისრულე, საიმედოობა, აქტუალურობა, გამოყენების სიმარტივე;

- რაოდენობრივი მახასიათებლები: დანახარჯები ინფორმაციის პირველად შეგროვებასა და განახლებაზე.

მას შემდეგ, რაც დამუშავდება ყველა წინა ეტაპი, მათ შორის შესაგროვებელი ინფორმაციის წყაროები დაზუსტდება, შეიძლება ინფორმაციის შეგროვების სქემა შედგეს. რადგანაც, ხშირად ინფორმაციის შეგროვებითა და გავრცელებით ერთი და იგივე პიროვნებები არიან დაკავებულნი, ამიტომ მიზანშეწონილია, ინფორმაციის გავრცელების სქემაც აქვე გაკეთდეს. ინფორმაციის შეგროვების, ანალიზისა და გავრცელების სქემა შეიძლება მე-12 ნახ-ზე მოცემული სახით წარმოვადგინოთ:



ნახ. 12. ინფორმაციის შეგროვების, ანალიზისა და გავრცელების სქემა

ყოველი ინფორმაციული სეგმენტისათვის ინფორმაციის წყაროდ მიზანშეწონილია ყველაზე ხარისხიანი და იაფი ინფორმაციის წყარო იქნეს

გამოყენებული; ამასთან, ამ მონაცემების საინფორმაციო სისტემაში შესატანად ყველაზე დაბალი რესურსები უნდა დაიხარჯოს, ანუ პირველ რიგში, ინფორმაციის ისეთი წყარო უნდა შეირჩეს, რომელთათვისაც შეფარდება ფასი/ხარისხი მაქსიმალური იქნება. თუ ინფორმაციის ერთი წყარო არ ყოფნის, მაშინ ინფორმაციის შემდეგი წყარო დაემატება. ინფორმაციის წყაროდ განიხილება როგორც კონკრეტული ორგანიზაცია, ისე მონაცემთა სხვადასხვა ბაზები, ინფორმაცია ინტერნეტში და სხვ.

ინფორმაციის გავრცელების საშუალებად ინტერნეტის ქსელის გამოყენება ძალზე პერსპექტიულია. განსაკუთრებით მიზანშეწონილია საჭირო ინფორმაცია პერიოდულად ნაბეჭდი ცნობარების სახით გამოვუშვათ.

თითოეული სეგმენტისათვის, ინფორმაციის რაოდენობრივი მახასიათებლების ჩატარებულმა მიახლოებითმა ანალიზმა საშუალება უნდა მოგვცეს გადავწყვიტოთ, ზოგიერთი ინფორმაციული სეგმენტის შეგროვებასა და გავრცელებაზე, რომელი ტექნიკური და პროგრამული საშუალებები შეიძლება დაგვჭირდეს. ამისათვის, საჭიროა შესაგროვებელი ინფორმაციის მოცულობა და მოთხოვნები მათი განახლების სიხშირის მიმართ შეფასდეს.

საინფორმაციო სისტემის დაპროექტებისას თავიდანვე უნდა გადაწყდეს, შეგროვებული ინფორმაციის შენახვა რა სახით მოხდება. თანამედროვე პირობებში ეს შეიძლება მოხდეს: ტექსტის ან სხვა სახის ფაილების სახით; მონაცემთა ბაზაში; CD-ROM-ზე; html-გვერდის სახით ინტერნეტ-ქსელის საიტებზე, ან ორგანიზაციის ლოკალურ ქსელში; ინფორმაციის ანალიზისათვის, როგორც წესი, დაბეჭდილი სახით.

თანამედროვე პროგრამული საშუალებების გამოყენებით შეგვიძლია ინფორმაცია წარმოდგენის ერთი ფორმიდან მეორეში გარდავქმნათ, მაგრამ ყველაფერი ეს საინფორმაციო სისტემის დაპროექტებისას უნდა იქნეს გათვალისწინებული. ასევე, საინფორმაციო სისტემის დაპროექტებისას მონაცემების არსებული სტრუქტურის ცვლილებები ან ახლის დამატება უნდა იქნეს გათვალისწინებული.

მონაცემთა ბაზებს, ინტერნეტის საიტებსა და ცნობარებს შორის ინფორმაციის მიმოცვლის სქემის დაპროექტებისათვის შეიძლება შემდეგი რეკომენდაციები იქნეს შემოთავაზებული:

- უმეტესობა არატექსტური ინფორმაცია მიზანშეწონილია მონაცემთა ბაზებში, ხოლო ტექსტური ინფორმაცია – ფაილების სახით და საიტებზე იქნეს შენახული;

- მთელი ინფორმაცია მონაცემთა ბაზიდან საიტებზეც უნდა იყოს ხელმისაწვდომი;

- მონაცემთა ბაზაში შენახული ინფორმაციის ჩასწორება მხოლოდ მონაცემთა ბაზის მართვის სისტემიდან უნდა ხდებოდეს, მაგრამ შორს მყოფი მომხმარებლებიდან ინფორმაციაში ცვლილებების შეგროვება ელექტრონული ფოსტის გამოყენებითაა უფრო მოსახერხებელი;

- ნაბეჭდი ცნობარებისათვის ინფორმაციის გენერირება შეიძლება განხორციელდეს მონაცემთა ბაზისა და ინტერნეტ-საიტების ინფორმაციის ბაზაზე (MS Visual Basic-ზე შეიძლება დამუშავდეს სპეციალური პროგრამა, რომელიც მოახდენს ტექსტის გენერაციას MS Word-ის ფორმატში).

პროგრამული უზრუნველყოფის დამუშავება

მონაცემთა ბაზის დაპროექტების თანამედროვე საშუალებანი საკმაოდ კარგად არის განვითარებული. საინფორმაციო სისტემის დაპროექტებისას მონაცემთა ბაზის დაპროექტება ყველაზე რთულ ამოცანას არ წარმოადგენს. ამიტომ, მხოლოდ მონაცემთა ბაზის დამუშავების რამდენიმე თავისებურება აღვნიშნოთ.

ასეთი საინფორმაციო სისტემების ექსპლუატაციის ანალიზმა გვიჩვენა, რომ ინფორმაციის წარმოშობის თითოეული ადგილისათვის მომხმარებელს უნდა მიეცეს საშუალება ნახოს:

- მთელი არსებული ინფორმაცია (ცალ-ცალკე ინფორმაციის თითოეული სახეობისათვის);

- განსახილველი ტერიტორიის სტატისტიკური მონაცემები (ფართობი, მოსახლეობა და სხვ.);

- ინფორმაციის სხვა წყაროებზე მიმართვები (პირველ რიგში, ინტერნეტში), ინფორმაციით მოცემულ ტერიტორიებზე და შესაგროვებელი ინფორმაციის თითოეული სახისათვის;

- იმ დოკუმენტებზე მიმართვა (Word-ის ტექსტები), რომელიც ეხება წარმოშობის ადგილს.

აქედან გამომდინარე, მონაცემთა ბაზაში კონკრეტული საგნის მიხედვით შენახული, თითოეული ინფორმაციის ტიპის სტრუქტურირება სამი საერთო მახასიათებლის მიხედვით ჩავატაროთ:

- ინფორმაციის სახე (ქვესახეობებად დაყოფით);
- ინფორმაციის წარმოშობის ადგილი (ტერიტორიის უფრო დეტალურად დაყოფით);
- დრო, რომელსაც მიეკუთვნება ინფორმაცია.

ორგანიზაციის საინფორმაციო სისტემის მარკეტინგული ინფორმაციის შეგროვების ჩარჩოებში მიზანშეწონილია, ინტერნეტის ყველა მომხმარებლისათვის ორგანიზაციის სარეკლამო გვერდები და Price-გვერდები ხელმისაწვდომი იყოს, ხოლო შეგროვებული ინფორმაცია ბაზრისა და საქონლის შესახებ ორგანიზაციის მხოლოდ ლოკალური ქსელის შიგა საიტებზე განთავსდეს.

საიტების დამუშავებამდე, შემდეგი სამუშაოები უნდა ჩატარდეს:

1. პოტენციურ მომხმარებელთა რაოდენობის შეფასება. საინფორმაციო სისტემის მომხმარებელთა თითოეული ჯგუფისათვის უნდა ჩატარდეს შეფასება, მოცემულ საიტებს ვინ და რამდენად ხშირად მიმართავს;

2. მომხმარებელთა საშუალო და მაქსიმალური რაოდენობის შეფასება. ეს განსაკუთრებით ტექნიკური და პროგრამული უზრუნველყოფის არჩევის დროსაა მნიშვნელოვანი;

3. უნდა განისაზღვროს, თუ რითი მოხდეს პოტენციურ მომხმარებელთა მიზიდვა საიტებზე. შეიძლება შემდეგი შევთავაზოთ:

- უნიკალური ინფორმაცია;
- ინფორმაციის მაღალი ხარისხი;
- მოხერხებული მუშაობა ინფორმაციასთან;
- მომხმარებელთა მოცემული ჯგუფისათვის, უახლესი და საინტერესო ინფორმაცია;
- მისი მსგავსი (კონკურენტი) საიტებისაგან სრულიად განსხვავებული ინფორმაცია.

ინფორმაცია საიტზე მომხმარებლისათვის მისაღები და გასაგები ფორმით უნდა იყოს წარმოდგენილი. ამისათვის, უზრუნველვეყოთ:

1. საჩვენებელი ფორმით ინფორმაციის გამოტანა;
2. დეტალიზაციის სხვადასხვა დონით ინფორმაციის გამოტანა;

3. ინფორმაციათა თანაკვეთა შემდეგი შესაძლებლობებით: მათი სახეობისა და ქვესახეობების მიხედვით; იმ ტერიტორიის მიხედვით, რომელსაც მიეკუთვნება მოცემული ინფორმაცია; დროის დიაპაზონის მიხედვით, რომელსაც მიეკუთვნება მოცემული ინფორმაცია.

ინფორმაციის თითოეული სახეობისათვის მომხმარებელს საშუალება უნდა მიეცეს ინფორმაციის ძებნა მისთვის ცნობილი ველების მიხედვით განახორციელოს. აგრეთვე, აუცილებელია გააკეთოს მიმართვები: მოცემული სახეობის ინფორმაციის ანალიზის შედეგებზე; საინფორმაციო სისტემაში არსებულ იმავე თემატიკის სხვა ინფორმაციაზე; იმავე თემატიკის ინფორმაციის სხვა წყაროებზე.

ამის შემდეგ, უნდა დამუშავდეს საინფორმაციო სისტემის ჩარჩოებში შესაგროვებელი ინფორმაციის ანალიზის პროგრამული უზრუნველყოფა, რომელიც ძალზე სპეციფიკურია და თითოეული კონკრეტული შემთხვევისათვის განსაკუთრებულ კონკრეტულ მიდგომას მოითხოვს.

საინფორმაციო სისტემის პროგრამული უზრუნველყოფის დამუშავების პროცესში ანალიზის ელექტრონულ ფორმებთან ერთად, ანგარიშების ნაბეჭდი პროდუქციისა და ინფორმაციის სხვა მატარებლების (CD-ROM) სახით გამოცემისა და გავრცელების საკითხები უნდა იქნეს გათვალისწინებული.

III სტადია – დანერგვა

საინფორმაციო სისტემის დანერგვის სტადიაზე დამკვეთთა მიერ დამუშავებული საპროექტო დოკუმენტაციის შესწავლა და საინფორმაციო სისტემის ექსპლუატაციისათვის მათი ეტაპობრივი მომზადება ხორციელდება. ამ სტადიაზე, სამუშაოთა შესრულების პროცესში ცალკეული და სისტემური შეცდომებისა და დაუმთავრებელი სამუშაოს გამოვლენა - გასწორება ხდება.

დანერგვა შეიძლება შემდეგი მეთოდების გამოყენებით:

- მიმდევრობითი მეთოდი, როდესაც ქვესისტემების დანერგვა ერთმანეთის მიმდევრობით ხდება და როდესაც ერთ ამოცანას შემდეგი მოსდევს;

- პარალელური მეთოდი, როდესაც ყველა ქვესისტემაში ყველა ამოცანის დანერგვა ერთდროულად მიმდინარეობს;

- შერეული მეთოდი, როდესაც დამპროექტებლების მიერ რამდენიმე ქვესისტემის დანერგვა პირველი მეთოდით ხდება, ხოლო გამოცდილების დაგროვების შემდეგ დანარჩენების დანერგვა მეორე მეთოდით ხორციელდება.

პირველი მეთოდის ნაკლი ის არის, რომ იზრდება დანერგვის დრო, რაც თავის მხრივ პროექტის ღირებულების ზრდას იწვევს. მეორე მეთოდის გამოყენების დროს კი, მართალია, დანერგვის დრო მცირდება, მაგრამ შეცდომების გაპარვის შესაძლებლობა იზრდება, ამიტომ ყველაზე ხშირად, საინფორმაციო სისტემის დანერგვის შერეული მეთოდი გამოიყენება.

პროექტის დანერგვის პროცესი სამი ეტაპისაგან შედგება:

- ობიექტის დანერგვისათვის მომზადება;
- საცდელი დანერგვა;
- პროექტის საწარმო ექსპლუატაციაში ჩაბარება.

პირველ ეტაპზე – ობიექტის დანერგვისათვის მომზადება და შემდეგი ოპერაციების განხორციელება:

- ობიექტის ორგანიზაციული სტრუქტურის ცვლილება;
- ინფორმაციის დამუშავების, სისტემის ექსპლუატაციისა და შესაბამისი საპროექტო დოკუმენტაციის სფეროში შესაბამისი კვალიფიკაციის კადრების შერჩევა;

- შესაბამისი კომპიუტერული ტექნიკის განსათავსებლად, შენობის მომზადება;

- შესაბამისი კომპიუტერული და სხვა საჭირო ტექნიკის შესყიდვა და მონტაჟი;

- ობიექტის ცალკეულ სტრუქტურებში პირველადი ინფორმაციის შეგროვებისა და რეგისტრაციის, აგრეთვე მათი კავშირგაბმულობის არხებით გადამცემი მოწყობილობების დადგმა;

- კავშირგაბმულობის არხების მონტაჟი; ახალი დოკუმენტებისა და კლასიფიკატორების დამუშავება;

- ნორმატიულ-საცნობარო ინფორმაციული ბაზის ფაილების შექმნა.

მეორე ეტაპზე – საცდელი დანერგვა, რამდენიმე ქვესისტემის რამდენიმე ამოცანის პროექტის დანერგვა უნდა განხორციელდეს. საცდელი დანერგვის პროცესში შემდეგი სამუშაოები სრულდება:

- საწყისი ოპერატიული მონაცემების იმ ამოცანებისათვის მომზადება, რომელთათვის საცდელი ექსპლუატაცია ხორციელდება;

- საწყისი მონაცემების კომპიუტერში შეტანა და მათზე ყველა შესაძლო ოპერაციის ჩატარება;

- შესაძლო შეცდომების გამოვლენის მიზნით, საბოლოო მონაცემების ანალიზის ჩატარება.

შეცდომების გამოვლენის შემდეგ შეცდომის მიზეზი და წყარო უნდა მოიძებნოს და პროგრამებში, ინფორმაციის დამუშავების ტექნოლოგიაში, ტექნიკური საშუალებების მუშაობაში, საწყის ოპერატიულ მონაცემებსა და ნორმატიულ-საცნობარო ინფორმაციული ბაზის ფაილებში შესაბამისი კორექტივები განხორციელდეს. ამის გარდა, შეიძლება ოპერატორების არაკვალიფიციური მუშაობა გამოვლინდეს, რისთვისაც კადრების მომზადების გაუმჯობესების სხვადასხვა ღონისძიებანი უნდა გატარდეს.

შეცდომათა აღმოფხვრის ეტაპის დამთავრების შემდეგ უნდა მივიღოთ “აქტი საცდელი დანერგვის ჩატარების შესახებ”, რომლის შემდეგაც მომდევნო ეტაპის განხორციელება იწყება.

მესამე ეტაპზე - პროექტის საწარმო ექსპლუატაციაში ჩაბარება, შემდეგი სამუშაოები სრულდება:

- დახარჯული დროის, ფინანსური სახსრებისა და შესრულებული სამუშაოს მოცულობის მიხედვით, ხელშეკრულებით გათვალისწინებული სამუშაოს შესრულების რეალურ მდგომარეობასთან შეესაბამისობის შემოწმება;

- საპროექტო დოკუმენტაციის, სახელმწიფო სტანდარტებით გათვალისწინებულ მოთხოვნებთან შეესაბამისობის შემოწმება;

- თითოეული ამოცანისა და ქვესისტემის მიხედვით, მონაცემთა დამუშავების ტექნოლოგიური პროცესის შემოწმება;

- ინფორმაციული ბაზის ფუნქციონირების ხარისხის შემოწმება;

- შეკითხვებზე პასუხების ოპერატიულობისა და სისრულის შემოწმება;

- ლოკალური და სისტემური შეცდომების გამოვლენა და მათი გასწორება.

ამის გარდა, მიმდები კომისია, პროექტის სამეცნიერო-ტექნიკურ დონეს და პროექტის გაფართოების შესაძლებლობებს, ახალი კომპონენტების დამატების გზით განსაზღვრავს. ამ ეტაპის ბოლოს, ლოკალური და სისტემური შეცდომების გასწორება ხდება და უნდა შედგეს “პროექტის საწარმო ექსპლუატაციაში მიღების აქტი”, რითაც მოცემული სტადია მთავრდება.

IV სტადია – ექსპლუატაცია და პროექტის მომსახურება

საინფორმაციო სისტემის ექსპლუატაციისა და პროექტის მომსახურების სტადიაზე საკითხის გადაწყვეტა უნდა მოხდეს, თუ ვისი ძალებით (დამკვეთის თუ პროექტის დამმუშავებლის პერსონალის მეშვეობით) განხორციელდება საინფორმაციო სისტემის ექსპლუატაცია, და თუ არჩეული იქნება მეორე ვარიანტი, მაშინ მათ შორის უნდა გაფორმდეს პროექტის მომსახურების შესახებ შესაბამისი ხელშეკრულება.

მოცემული სტადია შემდეგ ეტაპებად იყოფა:

- პროექტის ექსპლუატაცია;
- პროექტის მომსახურება და მოდერნიზაცია.

საინფორმაციო სისტემის ექსპლუატაციის პირველ ეტაპზე რეალურად იწყება სისტემის გაშვება. ეს ეტაპი ძალზე პასუხსაგებია. მისი დაწყებისას პირველ რიგში, საკუთარი ინფორმაციის პირველადი შეგროვება და მონაცემთა ბაზებში შეტანა უნდა განხორციელდეს, რომლის დროსაც შეიძლება პერსონალის გამოუცდელობის გამო, გარკვეული პრობლემები წარმოიშვას.

ამავე დროს, მონაცემთა ბაზაში შესატანი მთელი ინფორმაციის მომზადება უნდა მოხდეს, განხორციელდეს სხვა ორგანიზაციების მონაცემთა ბაზების, ნაბეჭდი ცნობარებისა და სხვა სახის ინფორმაციის შესყიდვა. პირველადი ინფორმაციის შეგროვება შეიძლება საკმაოდ რთული და შრომატევადი იყოს.

პროექტის ექსპლუატაციის ეტაპის დროს, სისტემის მუშაობის პროცესში რაიმე უწყესივრობის აღმოჩენისას შესწორებების შეტანა, მათი რეგისტრაცია, სისტემის ტექნიკურ-ეკონომიკური მახასიათებლებისადმი თვალყურის დევნება და სამუშაოს ხარისხის შესახებ სტატისტიკის შეგროვება ხორციელდება.

პროექტის ექსპლუატაციის დროს ინფორმაციის განახლების, მისი ანალიზისა და გავრცელების პროცესი განსაკუთრებით შრომატევადია. ამ სამუშაოს არსებული რესურსების (სამუშაო ძალის, ფინანსური და სხვა რესურსები) დიდი ნაწილი მიაქვს, ამიტომ ამ სამუშაოს სულ მცირე ოპტიმიზაციამ შეიძლება საკმაოდ არსებითი ეკონომია მოგვცეს.

საინფორმაციო სისტემის ექსპლუატაციის შეწყვეტის გარეშე უნდა შეფასდეს ექსპლუატაციის პერიოდის მიმდინარე შედეგები. მიზანშეწონილია, არსებულ და საჭირო ინფორმაციაში განხორციელებული, ორგანიზაციაში მის ძირითად მომხმარებელთა, პარტნიორთა და კონკურენტებში მომხდარი ცვლილებების ანალიზი.

მიღებული შედეგების მიხედვით უნდა მოხდეს შემდეგი ეტაპის გეგმის კორექტირება, ხოლო პარალელურად - პროგრამული უზრუნველყოფის - მონაცემთა ბაზის, ინტერნეტ-საიტების და პროგრამული უზრუნველყოფის გადამუშავება.

პროექტის მომსახურებისა და მოდერნიზაციის ეტაპზე შეგროვილი სტატისტიკური ინფორმაციის ანალიზი ტარდება. ანალიზის შედეგები საშუალებას მოგვცემს:

- გაკეთდეს დასკვნა მთელი პროექტის ან მისი ნაწილის მოდერნიზაციის შესახებ;
- განისაზღვროს შესწორებების, ვადებისა და ღირებულების მოცულობა.

თუ გამოვლინდა, რომ მოცემული პროექტი უკვე მორალურად მოძველდა, მაშინ სპეციალური კომისიის მიერ მოცემული პროექტის უტილიზაციის ჩატარების ან ახალი პროექტის დამუშავების შესახებ მიიღება გადაწყვეტილება.

კორპორაციული საინფორმაციო სისტემები

საინფორმაციო სისტემების სპეციფიკა

რა თქმა უნდა, კონკრეტული დარგის გამოყენების მიხედვით შესაძლებელია საინფორმაციო სისტემები თავისი ფუნქციების, არქიტექტურის, რეალიზაციის მიხედვით ერთმანეთისაგან სრულიად განსხვავდებოდეს, მაგრამ შეიძლება, უკიდურეს შემთხვევაში გამოიყოს ორი თვისება, რომელიც ყველასთვის საერთოა. ჯერ ერთი, ნებისმიერი საინფორმაციო სისტემა ინფორმაციის შეგროვების, შენახვისა და დამუშავებისათვისაა განკუთვნილი, ამიტომაც, ნებისმიერი საინფორმაციო სისტემის საფუძველში დევს გარემო მონაცემთა შენახვისა და წვდომისათვის. ეს გარემო შენახვის უსაფრთხოების ხარისხსა და წვდომის ეფექტურობას უნდა უზრუნველყოფდეს და რომელიც საინფორმაციო სისტემის გამოყენების სფეროს უნდა შეესატყვისებოდეს. შევნიშნოთ, რომ გამომთვლელ პროგრამულ სისტემებში ასეთი გარემოს არსებობა აუცილებელი არ არის. პროგრამისადმი ძირითადი მოთხოვნა, რომელიც რიცხვით გამოთვლებს ასრულებს, მისი სწრაფქმედებაა. საჭიროა, პროგრამამ დათქმულ დროში საკმაოდ ზუსტი შედეგები მოგვცეს. სერიოზული გამოსათვლელი ამოცანის ამოხსნისას თვით სუპერკომპიუტერებშიც კი ეს დრო შეიძლება კვირაობით, ზოგჯერ თვეობითაც კი გაიზომოს. ამიტომაც, პროგრამისტ-გამომთვლელელები ყოველთვის ძალიან სკეპტიკურად უყურებენ გარემოებისაში მონაცემთა შენახვას, ამჯობინებენ ისე მოახდინონ პროგრამის ორგანიზება, რომ რაც შეიძლება დიდი დროის განმავლობაში დამუშავებაში მყოფი მონაცემები კომპიუტერის ძირითად მეხსიერებაში იმყოფებოდეს. ჩვეულებრივ, გარე მეხსიერება გამოთვლების შედეგების პერიოდულად შენახვისათვის გამოიყენება, რათა კომპიუტერის მწყობრიდან გამოსვლის შემთხვევაში, შენახული საკონტროლო წერტილიდან მუშაობის გაგრძელება შესაძლებელი იყოს.

მეორე მხრივ, საინფორმაციო სისტემები საბოლოო მომხმარებელზეა ორიენტირებული. ასეთი მომხმარებელი კომპიუტერული სამყაროდან შესაძლოა ძალიან შორს იყოს. მათთვის ტერმინალი, პერსონალური კომპიუტერი ან სამუშაო სადგური მხოლოდ მათი პროფესიონალური საქმიანობის ხელსაწყოა წარმოდგენს. ამიტომაც, საინფორმაციო სისტემა

აუცილებლად უნდა ფლობდეს მარტივ, კომფორტულ, იოლად ასათვისებელ ინტერფეისს, რომელმაც საბოლოო მომხმარებელს თავისი სამუშაოსთვის საჭირო ყველა ფუნქცია უნდა წარუდგინოს: მაგრამ ამავე დროს, არ მისცეს საშუალება ნებისმიერი ზედმეტი მოქმედება შეასრულოს. ზოგჯერ ეს ინტერფეისი შეიძლება იყოს გრაფიკული მენიუთი, ღილაკებით, მინიშნებებით და ა.შ. ამჟამად, გრაფიკული ინტერფეისი ძალიან გავრცელებულია და ინფორმაციული მოწყობილობების შემქმნელების უმეტესობა გრაფიკული ინტერფეისის შექმნაზეა ორიენტირებული. მეორე მხრივ, საკმაოდ უცნაურია ის ფაქტი, რომ ბევრ მომხმარებელს გრაფიკული ტერმინალები არ უყვარს და საინფორმაციო სისტემასთან წვდომისათვის თანამედროვე, მაგრამ ტრადიციულ ალფავიტურ-ციფრულ ტერმინალს ანიჭებს უპირატესობას. ეს ნამდვილად ოდნავ უცნაურად ჩანს, რადგან დასავლეთში, სადაც პრაქტიკულად ნებისმიერი სალარო აპარატი სინამდვილეში პერსონალურ კომპიუტერს წარმოადგენს, შეუძლებელია ერთი ალფაბეტურ-ციფრული მონიტორი მაინც ნახოთ.

შევნიშნოთ, რომ აუცილებელი არაა გამომთვლელ პროგრამულ სისტემებს განვითარებული ინტერფეისი ჰქონდეს. რა თქმა უნდა, ეს პროგრამული პროდუქციის მოთხოვნილებაზეა დამოკიდებული. თუ სისტემა გასაყიდადაა განკუთვნილი, მაშინ მარკეტინგული თვალსაზრისით მაინც მას კარგი ინტერფეისი უნდა ჰქონდეს. მაგრამ, როგორც წესი, სერიოზული გამომთვლელი პროგრამები თითქმის უნიკალურია. გამოთვლები ან პროგრამის შემქმნელებისაგან, ან იმავე წრის ხალხის მიერ იწარმოება. მათთვის გაცილებით სწრაფქმედი გამოთვლებია მნიშვნელოვანი, ვიდრე პროგრამის გაშვების კომფორტულობა. განვითარებული ინტერფეისის მოზღვავება კომპიუტერული რესურსების საკმაოდ დანახარჯებს მოიაზრებს. როგორც კომპიუტერული სამყაროს პროფესიონალებს, მათ შეუძლიათ კომპიუტერთან მუშაობის დროს ზოგიერთ ხარვეზს გაუმკლავდნენ.

საინფორმაციო სისტემის ამოცანები

კონკრეტული ამოცანები, რომლებიც საინფორმაციო სისტემის მეშვეობით უნდა გადაიჭრას, იმ გამოყენებით სფეროზეა დამოკიდებული, რომლისთვისაც ეს სისტემაა განკუთვნილი. ინფორმაციული დანართების

გამოყენების სფეროები სხვადასხვაგვარია: საბანკო საქმე, დაზღვევა, მედიცინა, ტრანსპორტი, განათლება და ა.შ. ძნელია საქმიანობის ისეთი სფერო მოიძებნოს, სადაც დღესდღეობით საინფორმაციო სისტემის გამოყენების გარეშე არსებობაა შესაძლებელი. მეორე მხრივ, ნათელია მაგალითად, ბანკების საინფორმაციო სისტემების მიერ გადაჭრილი კონკრეტული ამოცანები, იმ ამოცანებისგან განსხვავდება, რომელთა გადაწყვეტა სამედიცინო საინფორმაციო სისტემებითაა შესაძლებელი.

შეიძლება გამოიყოს იმ ამოცანათა გარკვეული რაოდენობა, რომლებიც გამოყენებითი სფეროს სპეციფიკაზე არ არის დამოკიდებული. ბუნებრივია, ასეთი დავალებები საინფორმაციო სისტემების საერთო თვისებებთანაა დაკავშირებულნი. უპირველესად ჩანს, რომ ის შემადგენელი ინფორმაციაა უფრო მნიშვნელოვანი, რომელიც დიდხანს გროვდება და რომლის დანაკარგის შევსება შეუძლებელია.

რა თქმა უნდა, საიმედოობის ხარისხი და ინფორმაციის შენახვის ხანგრძლივობა საინფორმაციო სისტემის მიმართ უმეტეს შემთხვევაში კორპორაციის მოთხოვნილებების მიხედვით ყალიბდება. მაგალითად, შეგვიძლია წარმოვიდგინოთ მცირე სავაჭრო კომპანია სწრაფი ბრუნვით, რომლის ინფორმაციულ სასაწყობო სისტემაში საკმარისია ინახებოდეს ინფორმაცია საქონელზე, რომელიც საწყობშია, და მომხმარებელთა შეუსრულებელ დაკვეთებზე. მაგრამ ვინ იცის, მომავალში საწყობის მთლიანი ისტორია, დაწყებული კომპანიის დაარსების მომენტიდან, იქნება თუ არა საჭირო.

მომდევნო ამოცანა, რომელიც საინფორმაციო სისტემების უმეტესობამ უნდა შეასრულოს – სხვადასხვა სტრუქტურის მონაცემთა შენახვაა. ძნელია წარმოვიდგინო ასე თუ ისე განვითარებული საინფორმაციო სისტემა, რომელიც მხოლოდ ერთ მონაცემთა ერთგვაროვან ფაილთან მუშაობს. უფრო მეტიც, საინფორმაციო სისტემის მიმართ გონივრული მოთხოვნაა, რომ მას განვითარება შეეძლოს. შესაძლოა გაჩნდეს ახალი ფუნქციები, რომელთა შესასრულებლადაც ახალი სტრუქტურის დამატებითი მონაცემები იყოს საჭირო. ამასთან, ადრე დაგროვილი მთელი ინფორმაცია შენახული უნდა იყოს. თეორიულად, ამ ამოცანის გადაჭრა შეიძლება გარე მეხსიერების იმ რამდენიმე ფაილის გამოყენებით, რომელიც მონაცემებს ფიქსირებული სტრუქტურით ინახავს. იმის მიხედვით, თუ ორგანიზაცია რა ხერხით მართავს ფაილებს, ეს სტრუქტურა შეიძლება იყოს ფაილის სტრუქტურული

ჩანაწერი ან სპეციალურად ამ საინფორმაციო სისტემისათვის დაწერილი ცალკეული ბიბლიოთეკური ფუნქციით იყოს გამაგრებული.

ამ მიდგომის გამოყენებისას საინფორმაციო სისტემა ორგანიზაციის მონაცემთა საცავის დეტალებით გადაიტვირთება. სამომხმარებლო ინტერფეისის დონის ფუნქციის შესრულებისას ინფორმაციულ სისტემას მოცემული კრიტერიუმის მიხედვით, ინფორმაციის ფაილებიდან ამორჩევა თავად უხდება. ფაილების მართვის ზოგიერთი სისტემა ფაილების მარტივი კრიტერიუმით ამორჩევის შესაძლებლობას იძლევა, მაგალითად, ჩანაწერის გასაღების მნიშვნელობის მიხედვით. მაგრამ, ჯერ ერთი, ამორჩევის ასეთი შესაძლებლობები ყოველთვის შეზღუდულია და დიდი ალბათობით შესაძლებელია, რომ ამორჩევის ფუნქციის ნაწილის გამოტანა თვით საინფორმაციო სისტემის კოდში გახდება საჭირო. მეორე, რომ სხვადასხვა სტრუქტურის მონაცემთა რამდენიმე ფაილის არსებობა არაცხადად მოიაზრებს, რომ საინფორმაციო სისტემის ზოგიერთი ფუნქციის შესრულების დროს რამდენიმე ფაილიდან ინფორმაციის შეთანხმებული (მოცემული კრიტერიუმით) ამორჩევის საჭიროება შეიქმნება. ასეთი შესაძლებლობები ფაილური სისტემებით არასდროს არ მყარდება.

ცნობილია რეალურად ფუნქციონირებადი საინფორმაციო სისტემების მაგალითები, რომლებშიც მონაცემთა შენახვა ფაილების სახით იგეგმებოდა. ამგვარი სისტემების განვითარების შედეგად, მათში გამოიკვეთა ცალკეული კომპონენტები, რომლებიც მონაცემთა ბაზების მართვის სისტემის (მბმს) პრიმიტიულ ნაირსახეობას წარმოადგენს. თვითნაკეთი მბმს, საინფორმაციო სისტემის მთავარი ნაკლია. თავიდან ჩანს, რომ თითქოს ყველაფერი ძალიან მარტივია: შესაძლო შეკითხვების ნაკრები საინფორმაციო სისტემის დაპროექტების დროს ხდება ცნობილი; ყოველი ტიპის შეკითხვისათვის შესაძლოა მისი შესრულების ეფექტური ხერხი მოიძებნოს. ამის შემდეგ, პროგრამისტის მარტივი სამუშაო რჩება და სპეციალიზებული მბმს მზადაა. მაგრამ აღმოჩნდება, რომ ყველა შესაძლო შეკითხვის გათვალისწინება დაპროექტების დროს ვერ მოხერხდა. მბმს-ის შემქმნელი მასში ახალ ფუნქციებს მუდმივად ამატებს, სანამ არ გადაწყვეტს შექმნას შეკითხვათა საერთო ენა, რომელზეც შესაძლოა შესაბამისი საინფორმაციო სისტემის მონაცემთა ბაზისადმი ნებისმიერი შეკითხვის ფორმულირება მოხდეს. რამდენიმე ხნის შემდეგ კორპორაციაში თუ მიიღებენ გადაწყვეტილებას, რომ შექმნან კიდევ ერთი საინფორმაციო სისტემა, რომლის მონაცემთა

შენახვის სტრუქტურა, რასაკვირველია, განსხვავდება იმათგან, რომლებიც არსებობდა პირველი საინფორმაციო სისტემის მონაცემთა ბაზაში; მაშ, კიდევ ერთი სპეციალიზებული მბმს უნდა შეიქმნას? არა, პასუხობენ მმართველები. ჩვენ უკვე გვაქვს ერთი. მოდით ვცადოთ მისი გამოყენება. მას მივყავართ იქამდე, რომ შემქმნელს უწევს გააკეთოს საერთო დანიშნულების მარტივი მბმს, რომელსაც შეუძლია მონაცემთა ბაზისგან მიიღოს ინფორმაცია მისი ფაილების სტრუქტურის შესახებ, აგრეთვე, შეასრულოს ამ მონაცემთა ბაზისადმი წარმოქმნილი ნებისმიერი კითხვა. შედეგად, თუ მაინც მიიღწევა დამუშავებული მბმს ქმედითუნარიანობა, ეს მხოლოდ კიდევ ერთი ველოსიპედის გამოგონება იქნება, რადგანაც ასეთი დონის მბმს უამრავი არსებობს. იგი იაფია და შემქმნელებისგანაც იღებს მხარდაჭერას.

აქამდე, ჩვენ საინფორმაციო სისტემის იმ ფუნქციებზე ვსაუბრობდით, რომლებიც გარე საცავიდან მონაცემთა მიწოდებას საჭიროებს, მაგალითად, როცა აკეთებენ მოხსენებას. მაგრამ საიდან ჩნდება მონაცემები გარე საცავში? როგორ ხდება შენახული ინფორმაციის შესაბამისობა საგნის გარემოს მდგომარეობასთან? რა თქმა უნდა, ამისათვის საინფორმაციო სისტემის ის დამატებითი ფუნქციები უნდა არსებობდეს, რომლებიც მონაცემთა შეტანას, განახლებას და წაშლას უზრუნველყოფს. ამ ფუნქციათა არსებობა მბმს-სადმი მოთხოვნების დონეს მნიშვნელოვნად ზრდის.

თუ ვისაუბრებთ ჯგუფურ ანუ კორპორაციულ ინფორმაციულ სისტემებზე, მაშინ მათი არსებობა სისტემასთან მუშაობას რამდენიმე სამუშაო ადგილიდან გულისხმობს. ზოგიერთი მომხმარებელი მონაცემთა ბაზის შემადგენლობას ცვლის (შეაქვს, ანახლებს, შლის მონაცემებს), ზოგი მონაცემთა ბაზების პასუხთან დაკავშირებულ ოპერაციებს ასრულებს, სხვები კი ერთსაც და მეორესაც აკეთებენ. მთელი პრობლემა იმაში მდგომარეობს, რომ ასეთი კოლექტიური სამუშაო შეთანხმებულად უნდა შესრულდეს და სასურველია, მოქმედებათა შეთანხმება ავტომატურად ხდებოდეს. მოქმედებათა შეთანხმებაში მოვიაზრებთ, რომ ოპერატორი, რომელიც ანგარიშების ფორმირებას ახდენს, იმ მონაცემების გამოყენებას ვერ შეძლებს, რომლის ფორმირებაც სხვა ოპერატორმა დაიწყო, მაგრამ ჯერ არ დაუსრულებია. ოპერატორს მონაცემთა ფორმირებისას, იმ მონაცემებზე ოპერაციის ჩატარება არ შეუძლია, რომელსაც ამ დროს იყენებს ოპერატორი, რომელმაც ანგარიშების ფორმირება დაიწყო, მაგრამ ჯერ არ დაუსრულებია. ოპერატორს, რომელსაც სურს მონაცემები განაახლოს ან წაშალოს, ვერ

შეასრულებს ოპერაციას მანამ, სანამ არ დასრულდება ანალოგიური ოპერაცია იმავე მონაცემებზე, რომლებიც უფრო ადრე დაიწყო, მაგრამ არ დაუსრულებია სხვა ოპერატორს. მოქმედებათა შეთანხმებულობის დროს ყველა შედეგი, რომლებიც საინფორმაციო სისტემისაგან მიიღება, მონაცემთა ბაზის შეთანხმებული მდგომარეობის შესატყვისი იქნება, ე. ი. იქნება სანდო და არაურთიერთსაწინააღმდეგო.

ამგვარმა მსჯელობებმა კლასიკური ტრანზაქციის გაგება მოიტანა. საინფორმაციო სისტემის მონაცემთა ბაზის მთლიანობის ქვეშ იგულისხმება მისი ისეთი მდგომარეობა, რომელიც გამოყენებითი სფეროების მოთხოვნებს ან უფრო სწორედ, შეესაბამება გამოყენებითი სფეროს იმ მოდელის მოთხოვნებს, რომლის საფუძველზეც საინფორმაციო სისტემის დაპროექტება ხდება. მაშინ კლასიკურ ტრანზაქციას მონაცემთა ბაზის ცვლილებისა და/ან მონაცემთა ბაზიდან ამორჩევის იმ ოპერაციათა მიმდევრობას უწოდებენ, რომელსაც მბმს, როგორც ატომარულ ქმედებას აღიქვამს. ეს ნიშნავს, რომ ტრანზაქციის წარმატებით დასრულების შემთხვევაში, მბმს მონაცემთა ბაზაში ყველა ცვლილების ოპერაცია, რომლებიც ტრანზაქციის შესრულების პროცესში განხორციელდა, შედეგის არსებობის გარანტიას იძლევა. ტრანზაქციის წარმატებით დასრულების პირობაა, რომ მონაცემთა ბაზა მთლიანობის მდგომარეობაში იმყოფება. თუ ეს პირობა არ სრულდება, მაშინ მბმს მთელი ტრანზაქციის უკუსვლას იძლევა, მონაცემთა ბაზაში განხორციელებული ყველა იმ ცვლილების შედეგების ლიკვიდირებას ახდენს, რომლებიც ტრანზაქციის დროს შესრულდა. ამით ადვილია დავინახოთ, რომ მონაცემთა ბაზა ნებისმიერი ტრანზაქციის დაწყების მდგომარეობაში იქნება და ნებისმიერი ტრანზაქციის წარმატებით დასრულების შემდეგაც, მთლიანობის მდგომარეობაში დარჩება.

ყველა განვითარებულ მბმს აქვს ტრანზაქციის გაგება. თუ საინფორმაციო სისტემის ბაზა ასეთი კლასის მბმს არის, მაშინ პარალელურად მომუშავე მომხმარებელთა მოქმედებების შეთანხმებულობის გარანტიისათვის, საკმარისია სისტემის დაპროექტებისას საინფორმაციო სისტემის ოპერაციები მბმს-ის ტრანზაქციებთან სწორად იყოს დაკავშირებული.

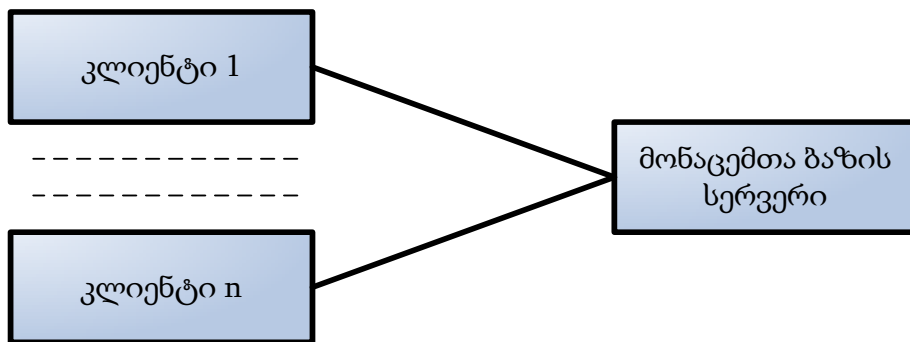
ტრანზაქციებთან დაკავშირებული კიდევ ერთი პატარა შენიშვნა. მბმს-ს ძალიან იოლად შეუძლია ტრანზაქციის დამუშავება, თუ მიმდევრობით

შეასრულებს. ეს საკმარისია, რათა "პარალელურად" მომუშავე ოპერატორების მოქმედებათა შეთანხმებულობა უზრუნველყოფილ იქნეს. ამ შემთხვევაში, რეალური პარალელურობა, რა თქმა უნდა, არ იქნება. მბმს ყველა მომხმარებელს საერთო რიგში ჩააყენებს და მათ რიგრიგობით გაუშვებს, მაშინაც კი, თუ მონაცემების მიხედვით საერთოდ ერთმანეთთან კონფლიქტში არ შედიან. განვითარებული მბმს ასე არ მუშაობს და ცდილობს შეკითხვები და მონაცემთა ბაზის შეცვლის ოპერატორები, რომლებიც სხვადასხვა ტრანზაქციებიდან მოდის, ერთმანეთში მაქსიმალურად აურიოს, მხოლოდ იმ პირობით, რომ მთელი ნაკრები ტრანზაქციის შესრულების საბოლოო შედეგი, მათი მიმდევრობით შესრულების შედეგის ეკვივალენტური იქნება. მონაცემთა ბაზების სამყაროში მბმს-ს ასეთ პოლიტიკას, ტრანზაქციების შერევის სრული სერიალიზაციის პოლიტიკას უწოდებენ. ამკარაა, რომ ტრანზაქციის სრული სერიალიზაცია საინფორმაციო სისტემის ოპერატორების პარალელური მუშაობის დროს მოქმედებათა შეთანხმებულობის მისაღწევად საკმარისი. მაგრამ, ტრანზაქციის სრული სერიალიზაცია მოქმედებათა აუცილებელი შეთანხმებულობისათვის ყოველთვის არაა საჭირო. არსებობს უფრო სუსტი სერიალიზაციის მოდელი, რომელიც უფრო დიდ პარალელურობას უშვებს და უფრო მცირე დანახარჯებს იწვევს.

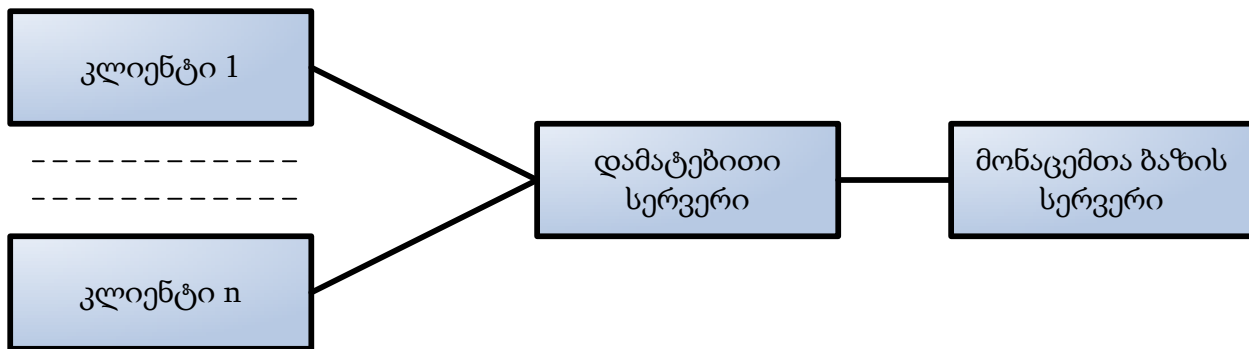
კორპორაციულ საინფორმაციო სისტემებში ჩვეულებრივ, აუცილებელი მონაცემების საერთო ბაზაში გადანაწილება ხშირად ხდება. მაგალითად, გონივრულია ინფორმაციის რაღაც ნაწილის იმ სამუშაო ადგილთან ახლოს შენახვა, სადაც ყველაზე ხშირად გამოიყენება. ამ მიზეზით, საინფორმაციო სისტემის აგებისას საჭიროა მონაცემთა განაწილებული ბაზის შეთანხმებული მართვის ამოცანის გადაჭრა. ერთგვაროვანი ბაზების შემთხვევაში, ეს პრობლემა მბმს-ის საშუალებით გადაიჭრება.

საინფორმაციო სისტემების ორგანიზების ტრადიციულ მეთოდს ორრგოლიანი არქიტექტურა "კლიენტი-სერვერი" წარმოადგენს (ნახ. 13). ამ შემთხვევაში, მთელი საინფორმაციო სისტემის გამოყენებითი ნაწილი სისტემის სამუშაო სადგურებზე სრულდება, სერვერის მხარეს კი ბაზებთან წვდომის განხორციელება ხდება. თუ გამოყენებითი ნაწილის ლოგიკა საკმაოდ რთულია, მაშინ ასეთი მიდგომა "მსუქანი" კლიენტის პრობლემას აჩენს. ყოველ სამუშაო სადგურს უნდა ჰქონდეს რესურსების საკმარისი

ნაკრები, რომ იმ მონაცემების გამოყენებითი დამუშავება შეძლოს, რომლებიც მომხმარებლისაგან და/ან მონაცემთა ბაზისაგან მიიღება. იმისათვის, რომ კლიენტი იყოს "სუსტი", სისტემის საერთო ეფექტურობის ასამაღლებლად, უფრო და უფრო ხშირად სამრგოლიან არქიტექტურას "კლიენტი-სერვერი" იყენებენ (ნახ. 14). ამ არქიტექტურაში კლიენტის სისტემის ნაწილისა და მონაცემთა ბაზის სერვერის გარდა, საშუალოდ დამატებითი სერვერი იქმნება. კლიენტის მხარეს სრულდება მხოლოდ ინტერფეისული მოქმედებები და ინფორმაციის დამუშავების მთელი ლოგიკა დამატებით სერვერში ხორციელდება.



ნახ. 13. ორრგოლიანი არქიტექტურა "კლიენტ-სერვერი"



ნახ. 14. სამრგოლიანი არქიტექტურა "კლიენტ-სერვერი"

დაბოლოს, კიდევ ამოცანათა ერთი კლასი, რომელიც საინფორმაციო სისტემას შესაბამისი მიზნების განხორციელებას, შექმნას კომფორტული და სამომხმარებლო ინტერფეისი, უწყობს ხელს. ასე თუ ისე, ადვილია ინტერფეისის ფუნქციონალური კომპონენტები გაარკვიოთ, მაგალითად, თუ როგორი სახით უნდა წარმოგვიდგენს ფორმები და რა სახით უნდა გაიცეს ანგარიშები. მაგრამ მომხმარებლისათვის ნამდვილად კომფორტული და საინტერესო ინტერფეისის შექმნა, ეს უკვე ინტერფეისის დიზაინერის

ამოცანაა. უნდა გავითვალისწინოთ, რომ ინტერფეისის ერგონომიულობით ამოცანის ფორმულიზაცია არ ხდება. მის გადაწყვეტაში ინტერფეისის ავტომატური დამუშავების რომელიმე ხერხი ვერ დაგვეხმარება. სრული ინტერფეისის შექმნა შემოქმედებითი დავალებაა, რომლის შექმნის პროცესში გავითვალისწინებულ უნდა იქნეს ესთეტიკური მოთხოვნები და კომფორტის საშუალებები, აგრეთვე მხედველობაში უნდა იქნეს მიღებული იმ კონკრეტული გარემოს თავისებურებანი, სადაც საინფორმაციო სისტემა გამოიყენება.

საინფორმაციო სისტემის აგების პრობლემები

ყველაზე მთავარია დაპროექტების პრობლემა. არ შეიძლება ტექნიკური დამუშავების დაწყება, თუ გულდასმით დამუშავებული პროექტი არ არსებობს. თუ უფრო შესამჩნევი პრობლემების გადაჭრას დავიწყებთ და პოტენციურად არსებულ პრობლემებს არ მივაქცევთ ყურადღებას, მაშინ ასეთი სისტემა მუდმივად დამუშავებისა და გადაკეთების სტადიაში იქნება.

დაპროექტების პირველი ეტაპი კორპორაციის მოთხოვნების ანალიზი უნდა იყოს. ამისათვის, ექსპერტული გამოკითხვების შედეგად უნდა იქნეს გამოვლენილი კორპორაციის ყველა აქტუალური და პოტენციური მოთხოვნები, რომლებიც დასაპროექტებელი საინფორმაციო სისტემის მიერ უნდა დაკმაყოფილდეს, შესწავლილ იქნეს კორპორაციის შიგნით არსებული მონაცემთა ნაკადი, შეფასდეს დასამუშავებელი ინფორმაციის მოცულობა. ეს სტადია, როგორც წესი, არაფორმალურ ხასიათს ატარებს, მაგრამ სასურველია მიღებული ინფორმაციის შენახვა, რადგან ის სისტემის დოკუმენტაციაში უნდა შევიდეს. არსებობს მაღალი დონის CASE-საშუალებები, რომლებიც მიღებულ მონაცემებს პროექტის საერთო რეპოზიტორში ათავსებს და დაპროექტების შემდეგ სტადიებში მათი გამოყენების საშუალებას იძლევა.

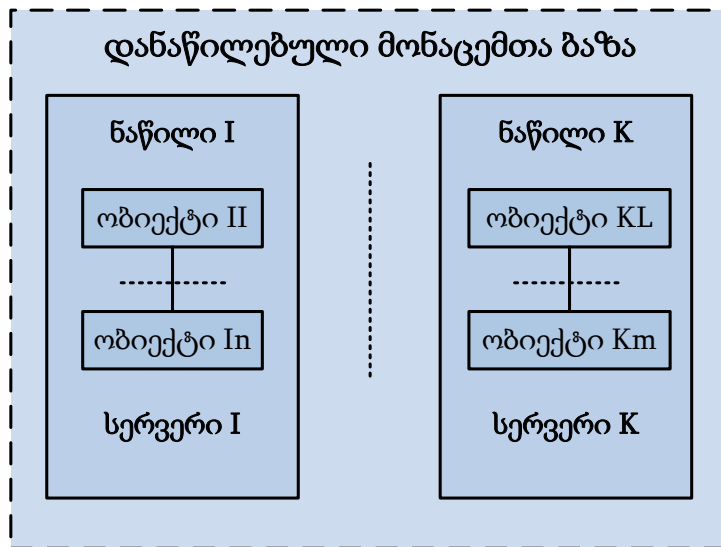
დაპროექტების შემდგომი სტადიაა მონაცემთა ბაზების კონცეპტუალური სქემის შემუშავება, რომელიც საინფორმაციო სისტემის საფუძველში იქნება განთავსებული. თავიდან საჭიროა აირჩეს ნოტაციების სისტემა, რომელშიც კონცეპტუალური სქემა იქნება მოცემული. ასეთი ნოტაცია ძალიან ბევრი არსებობს და მიუხედავად იმისა, რომ ყველა დიაგრამული სახისაა (ER-მოდელის სტილში), ისინი ერთმანეთისაგან

განსხვავდებიან. ზოგიერთი CASE-საშუალებების გამოყენების შემთხვევაშიც დამპროექტებელს შეუძლია აარჩიოს რამდენიმე ნოტაცია, რომელიც ძალზე მნიშვნელოვანი არჩევანია. მონაცემთა ბაზის კონცეპტუალური წარმოდგენა უნდა ინახებოდეს, როგორც საინფორმაციო სისტემის დოკუმენტაციის ნაწილი და შეიძლება მისი თანმხლები დოკუმენტაციისა და განვითარების დროს იქნეს გამოყენებული.

შემდეგ, დიდი ალბათობით, რომ საინფორმაციო სისტემის საფუძველში რელაციურ მონაცემთა ბაზა იქნება გამოყენებული, მიუხედავად ობიექტზე ორიენტირებული (Object Store, Objectivity, O2, Jasmin და ა.შ.) და ობიექტ-რელაციური (Illustra, UniSQL) მბმს-ის დიდი მომხიბვლელობისა, უახლეს მომავალში კარგად დაწყობილ, განვითარებულ, თანმხლები დოკუმენტაციის იმ სისტემებთან მოგვიწევს მუშაობა, რომლებიც SQL სტანდარტს ინარჩუნებს (მაგალითად, Oracle, Informix, CA-Openingres, Sybase, DB2). ეს იმიტომ, რომ უნდა გავიდეს დრო, რათა ეს სისტემები დალაგდეს, შეიძინოს აუცილებელი უსაფრთხოება, დაფუძნებული იყოს რაღაც სტანდარტებზე და ა.შ.

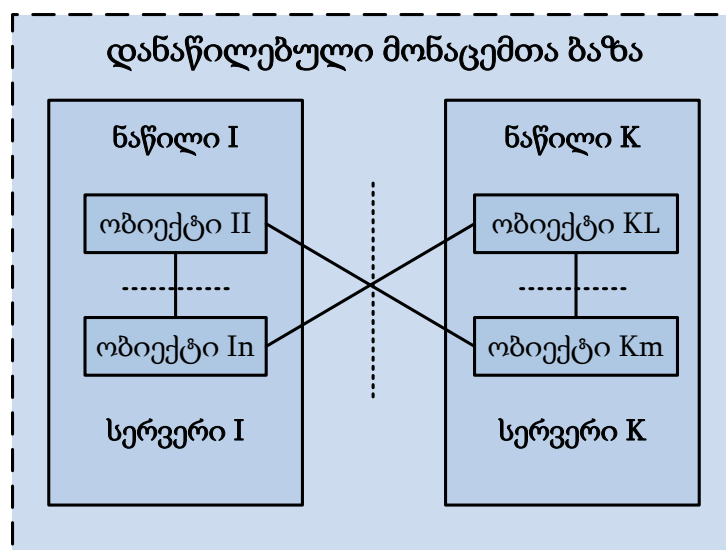
ამიტომაც შესაძლებელია დაპროექტების შემდეგ სტადიაზე შეიქნას არსებული კონცეპტუალური სქემის საფუძველზე SQL ენის ტერმინებში რელაციური მონაცემთა ბაზის განსაზღვრული სქემის ნაკრები. ამ სტადიაზევე აუცილებელია განისაზღვროს, რომელი ცხრილი იქნება რეალურად და რომელი წარმოსახვით (view) შესანახი.

იმის შემდეგ, რაც მთლიანი მონაცემთა ბაზის რელაციური სქემა განისაზღვრება, აუცილებელია აგრეთვე, არქიტექტურული სისტემაც განისაზღვროს. საკუთრივ, ძალიან მნიშვნელოვანია გადაწყდეს, რომელი მონაცემთა ბაზა იქნება გამოყენებული – ცენტრალიზებული თუ განაწილებული (სხვა სიტყვებით რომ ვთქვათ, მონაცემთა ბაზის მხოლოდ ერთი სერვერი იქნება გამოყენებული თუ რამდენიმე).



ნახ. 15. მონაცემთა ბაზა ლოგიკური ავტონომიური ნაწილებით

დეკომპოზიციის ყველაზე მარტივი შემთხვევა მაშინაა, როდესაც მონაცემთა ბაზის წარმოქმნილი ნაწილები ლოგიკურად ავტონომიურია. კონცეპტუალური სქემის ტერმინებში ეს ნიშნავს, რომ დაყოფილ ნაწილებს შორის პირდაპირი ან ტრანზიტული კავშირები არსებობს. რელიაციური სქემის ტერმინებში: არც ერთ ნაწილში არ არსებობს ცხრილი, რომელიც მიმართავს სხვა ნაწილში არსებულ ცხრილს (ნახ. 15 და 16), თუ განაწილებული მონაცემთა ბაზის კომპონენტების ლოგიკური ავტონომიურობის მოთხოვნები შესრულებულია და მაშინ მომავალი თითოეული კომპონენტისათვის დაპროექტება შეიძლება დამოუკიდებლად გაწარმოთ.



ნახ. 16. მონაცემთა ბაზა ლოგიკური არავტონომიური ნაწილებით

დაპროექტების შემდეგი სტადია განაწილებული მონაცემთა ბაზის, რელიაციური სქემის ტრიგერებითა და შენახული პროცედურებით შევსებაში მდგომარეობს, ნაწილების მთლიანობის საერთო შეზღუდვების განსაზღვრებით. განვიხილოთ თითოეული ამ კომპონენტის სქემები ცალ-ცალკე.

SQL ენა შესაძლებლობას იძლევა განისაზღვროს მთლიანობის შეზღუდვა, რომელიც მონაცემთა ბაზის საერთო მდგომარეობას ეხება და ნებისმიერი რაოდენობის ცხრილებზე მიმართვებს მოიცავს.

ტრიგერი ფაქტობრივად, პარამეტრების გარეშე შენახული პროცედურაა, რომელიც მონაცემთა ბაზის შეცვლისა და რაიმე შემთხვევის დროს სერვერის მიერ ავტომატურად გამოძახების ოპერატორს ან ოპერატორებს მოიცავს. არსებობს ტრიგერის მექანიზმის გამოყენების რამდენიმე პრაქტიკული მხარე. უპირველესად, ტრიგერი იძლევა მონაცემთა ბაზის ლოგიკური მთლიანობის საშუალებას იმ შემთხვევაში, როცა სამომხმარებლო ტრანზაქცია ამ მთლიანობას არღვევს.

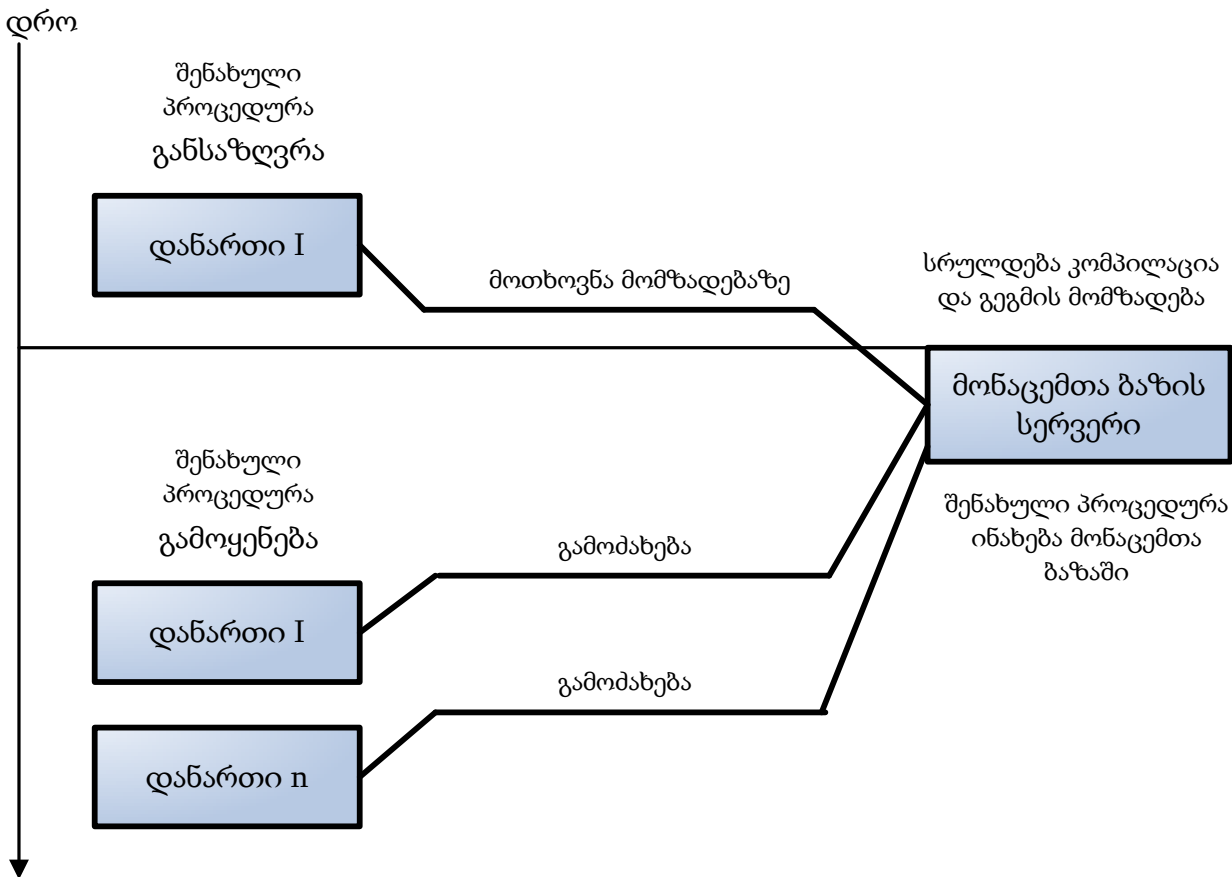
მეორე ველი, სადაც ტრიგერი გამოიყენება, მომხმარებელთა ქმედების ავტომატური მონიტორინგია.

ორი შენიშვნა. ერთი, რომ დღესდღეობით გავრცელებულ SQL სტანდარტში ტრიგერის მექანიზმი მბმს-ებში, სადაც ამ მექანიზმს იყენებენ, არ არის სპეციფირებული, შესაბამისი ენობრივი საშუალებები და მათი სემანტიკა განსხვავებულია. მეორე, ისევე როგორც წარმოდგენის შექმნისა და ასევე ორგანული მთლიანობის შემთხვევაში, ტრიგერების განსაზღვრის დროს აუცილებელია გათვალისწინებულ იქნეს მონაცემთა ბაზის განაწილებული თვისება და სერვერის შესაძლებლობა დაუკავშირდეს უცხო ცხრილებს.

დაბოლოს, მონაცემთა ბაზაში შესაძლებელია ინახებოდეს „შენახული პროცედურები“. საინტერესოა, რომ SQL სტანდარტში საერთოდ არ გვხვდება ტერმინი “შენახული პროცედურა”. სტანდარტში სპეციფიცირებულია სერვერთან ურთიერთობის გამოყენებითი პროგრამის მონაცემთა ბაზის ორი პროცედურა. პირველი, უფრო ხშირად გამოყენებადი ხერხია, SQL ენის ოპერატორების ჩართვა, დაპროგრამების ტრადიციულ ენაზე ჩაწერილ პროგრამაში. მეორე ხერხი დაფუძნებულია სპეციფირებულ სტანდარტზე “SQL მოდულთა ენაზე”. ამ ენის გამოყენებით შესაძლებელია განისაზღვროს

მოდული, რომელიც რამდენიმე პროცედურას შეიცავს, თითოეული მათგანი კი ზოგიერთ SQL პარამეტრიან ოპერატორს შეესაბამება.

მოდულების ენაზე განსაზღვრულმა SQL პროცედურების იდეამ, შენახული პროცედურების მექანიზმის მრავალ რეალიზაციაში ჩართვა გამოიწვია. შენახული პროცედურა არის სახელდებული, პარამეტრიზებული კონსტრუქცია, SQL ენაზე ან რომელიმე მის გაფართოებაზე განსაზღვრული, გამოყენებით პროგრამაში ჩაშენებული და სერვერის მხარეს კომპილირებული პროცედურის ტექსტის პრეკომპილატორით დამუშავების დროს მიღებული კოდი. შენახული პროცედურის შესრულებადი ან ინტერპრეტირებული კოდი ინახება მონაცემთა ბაზაში, თვითონ კი ნებისმიერი გამოყენებითი პროგრამის საშუალებით ავტორიზებული მომხმარებლის (ის მომხმარებელი, რომელმაც მიიღო ამ პროცედურის შესრულების პრივილეგია) მიერ, ფაქტობრივი პარამეტრების მითითებით შეიძლება იქნეს გამოძახებული (ნახ. 17). როგორც წესი, მზმს განვითარება SQL მოდულთა ენას ემყარება, მაგრამ შენახული პროცედურების ენა ჩვეულებრივ უფრო ფართოა და არასტანდარტული. ამიტომაც, შენახულ



ნახ. 17. შენახული პროცედურის მომზადება და გამოყენება

პროცედურებს ფრთხილად უნდა მოვეპყრათ, რათა კონკრეტულ მწარმოებელზე მთლიანად არ გავხდეთ დამოკიდებული.

ამით საინფორმაციო სისტემის მონაცემთა ბაზის ლოგიკური დაპროექტება დამთავრდა და შეგვიძლია მომდევნო სტადიებზე გადავიდეთ. უმთავრესად დარჩა ორი სტადია: მონაცემთა ბაზის ფიზიკური დაპროექტება; ინტერფეისების დაპროექტება-დამუშავება და გამოყენებითი სისტემის დამამუშავებელი ნაწილი. ეს ორი სტადია შესაძლებელია პარალელურად შესრულდეს.

ფიზიკური დაპროექტება მოიცავს ორ ბიჯს, რომელთაგან პირველი, როგორც წესი, ამორჩეული სერვერის SQL-ორიენტირებადი პროდუქტის თავისებურებებზე არაა დამოკიდებული, ხოლო მეორე დამოკიდებულია და ამასთან, კრიტიკულად. ამ სტადიის პირველ ნაბიჯზე საჭირო ინდექსების ნაკრები განისაზღვრება. მეორე ნაბიჯი კი გარე მეხსიერების იმ არეების განსაზღვრას მოიცავს, რომლებშიც მონაცემთა ბაზის ფრაგმენტები იქნება შენახული.

საინფორმაციო სისტემის მონაცემთა ბაზის ფიზიკური დაპროექტების პარალელურად შეიძლება ინტერფეისის დაპროექტება და დამუშავება. ამ მომენტისათვის უკვე ნათელი უნდა იყოს, თუ რა მოთხოვნებია ინტერფეისის მიმართ და რა ფუნქციები უნდა შეასრულოს სისტემამ. ასე რომ, ამ სტადიის ძირითად პრობლემას ისეთი ინსტრუმენტული საშუალებების ამორჩევა წარმოადგენს, რომლებიც საკმაოდ სწრაფი ეფექტური რეალიზაციის საშუალებას მოგვცემს.

ყველა შემთხვევაში, ინტერფეისის დაპროექტების პროცესში სასურველია ნახევარფაბრიკატების გამოყენება. მაგრამ, აქ არჩევანის დიდი საშუალება არსებობს: ბაზური ბიბლიოთეკები, რომლებიც ფანჯრულ სისტემას იყენებს (მაგალითად, X Toolkit Intrinsic X ფანჯრული სისტემა), დაპროექტების უფრო მაღალი დონის გრაფიკული სამომხმარებლო ინტერფეისის უნივერსალური ინსტრუმენტული საშუალებები (მაგალითად, Motif ან Tcl/Tk), მეოთხე თაობის ენები და დაპროგრამების სისტემები (მაგალითად, Power Builder, Jam და სხვ.). არჩევანი უნდა გაკეთდეს შემქმნელის გემოვნებითა და ჩვევების მიხედვით, აგრეთვე, პროექტის საერთო ორიენტაციით.

რაც შეეხება საინფორმაციო სისტემის დასამუშავებელ ნაწილებს, ყველაფერი იმაზეა დამოკიდებული, თუ რას უნდა ასრულებდნენ ისინი.

არსებობს საინფორმაციო სისტემების უამრავი მაგალითი, რომლებშიც მონაცემთა დამუშავება მათი შეტანის და გამოტანის დროს, მათი ფორმატის შეცვლაში, ანგარიშების ფორმირებაში და ა.შ. მდგომარეობს. ფაქტობრივად, შესაძლებელია ასეთი ფუნქციების დაპროგრამება არ მოხდეს, რადგანაც ნებისმიერ UGL-ს შესაბამისი აღწერილობით მათი ავტომატურად გენერირება შეუძლია; მაგრამ თუ საჭიროა მონაცემთა უფრო რთული დამუშავება, მაშინ ყველა შემთხვევაში საჭირო იქნება დაპროგრამება და უკვე აქ აღარაა მნიშვნელოვანი თუ რა ენაზე მოხდება ეს. კერძოდ, თუ თქვენ ინტერფეისის დასამუშავებლად იყენებთ Delphi-ის, მაშინ დაპროგრამებისას გონივრულია გამოვიყენოთ Object Pascal. სხვა შემთხვევებში, შესაძლოა გამოვიყენოთ UGL-ის პროცედურული ნაწილი ან C/C++.

მოთხოვნები ტექნიკური საშუალებების მიმართ

ნათელია, რომ ტექნიკური საშუალებების მიმართ მოთხოვნები მთლიანად საინფორმაციო სისტემის მიმართ მოთხოვნების მიხედვით ყალიბდება. რა ინფორმაციული საშუალებანიც არ უნდა სჭირდებოდეს კორპორაციის თანამშრომლებს, საბოლოო გადაწყვეტილებას იღებს მისი მმართველობა, რომელიც აპარატული გარემოს შესახებ საბოლოო წარმოდგენას უკეთებს კორექტირებას. ჩვენი აზრით, საინფორმაციო სისტემის ადგილის თაობაზე კორპორაციაში მმართველის 4 სავარაუდო პოზიცია არსებობს: პესიმისტური, პესიმისტურ-ოპტიმისტური, ოპტიმისტურ-პესიმისტური და ოპტიმისტური.

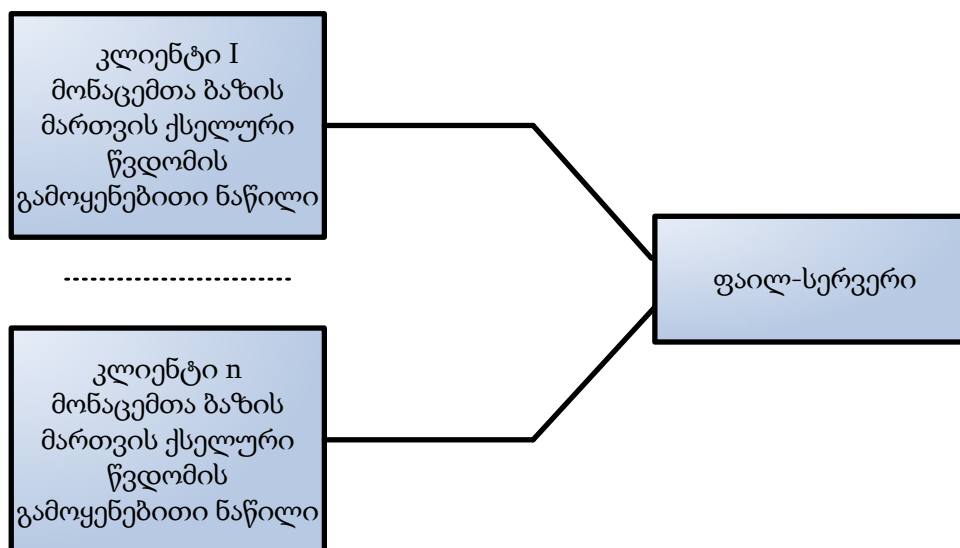
რა თქმა უნდა, მოყვანილი კლასიფიკაცია რამდენადმე უტრირებულია, ცხოვრებაში ყველაფერი უფრო რთულადაა; კერძოდ, მმართველებზე ტექნიკური სპეციალისტების ზეგავლენა არ უნდა დაგვავიწყდეს. ამრიგად, კორპორაციის საინფორმაციო სისტემის ასაგებად ტექნიკური საშუალებების შერჩევა მარტივი ამოცანა არ არის, იგი ტექნიკურ, პოლიტიკურ და ემოციურ ასპექტებს მოიცავს.

ინფორმაციული დანართების არქიტექტურის ზოგადი კლასიფიკაცია

ფაილ-სერვერული დანართები

ფაილ-სერვერულ არქიტექტურაზე დაყრდნობით, გამოყენებითი პროგრამული უზრუნველყოფის (და უმეტესობა სისტემური პროგრამული უზრუნველყოფის), რომლებიც ქსელის თითოეულ პერსონალურ კომპიუტერზე მუშაობენ, ავტონომიურობის შენარჩუნება ხდება. ფაქტობრივად, საინფორმაციო სისტემის კომპონენტები, რომლებიც სხვადასხვა პერსონალურ კომპიუტერზე სრულდება, მხოლოდ იმ საერთო ფაილთა საწყობთან ურთიერთობს, რომელიც ფაილ-სერვერზე ინახება. ფაილ-სერვერი პერსონალური კომპიუტერის მიერ განაწილებული ყველა დისკური მეხსიერების გაფართოების კომპლექსს წარმოადგენს (ნახ. 18).

რა თქმა უნდა, მისი მთავარი ღირსება ორგანიზაციის სიმარტივეა. გრაფიკული სამომხმარებლო ინტერფეისის დასამუშავებლად არსებობს მოხერხებული და განვითარებული საშუალებები, მონაცემთა ბაზის სისტემებისა და/ან მზმს-ის დამუშავების მარტივი საშუალებანი. მაგრამ ბევრგან ეს სიმარტივე მოჩვენებითია.



ნახ. 18. საინფორმაციო სისტემის კლასიკური წარმოდგენა არქიტექტურაში
“ფაილ-სერვერი”

ჯერ ერთი, საჭიროა საინფორმაციო სისტემის მონაცემთა ბაზასთან მუშაობა. შესაბამისად, ეს მონაცემთა ბაზა უნდა დაპროექტდეს ძალზე

ხარისხიანად. რა თქმა უნდა, მონაცემთა ბაზის დაპროექტების სირთულე, საგნის გარემოს მოდელირების სირთულითაა გამოწვეული.

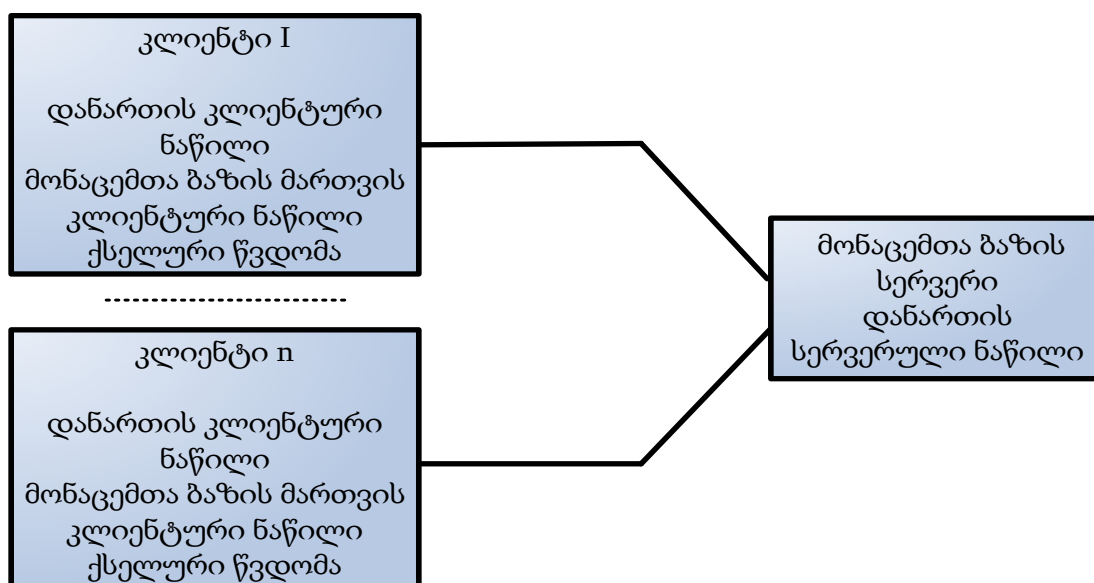
მეორე მხრივ, აუცილებელი მოთხოვნაა საინფორმაციო სისტემის მონაცემთა ბაზის მიმართ მისი მთლიანობის შენარჩუნება და ინფორმაციის შენახვის უსაფრთხოების გარანტია. მინიმალური პირობები, რომელთა შესრულებისას შესაძლებელია ამ მოთხოვნების დაკმაყოფილება, არის:

- ტრანზაქციული მართვის არსებობა;
- ზედმეტი მონაცემების შენახვა;
- ორგანული მთლიანობის ფორმულირების შესაძლებლობა და მათი შესრულების შემოწმება.

პრინციპში, ფაილ-სერვერული ორგანიზაცია, როგორც მე-18 ნახ-ზეა ნაჩვენები, ხსენებული პირობების შესრულებას არ გამორიცხავს; მაგალითად, სისტემა, რომელიც ასრულებს ამ პირობებს, მაგრამ ფაილ-სერვერულ არქიტექტურაზეა დაფუძნებული, შესაძლოა იყოს წარსულში პოპულარული “მონაცემთა ბაზის სერვერი” Informix SE.

კლიენტ-სერვერული დანართები

კლიენტ-სერვერულ დანართში მოიაზრება საინფორმაციო სისტემა, რომელიც სერვერულ მონაცემთა ბაზის გამოყენებაზეა დაფუძნებული. საინფორმაციო სისტემის ზოგადი სახე არქიტექტურაში „კლიენტი-სერვერი“ მე-19 ნახ-ზეა ნაჩვენები.



ნახ. 19. საინფორმაციო სისტემის საერთო წარმოდგენა არქიტექტურაში “კლიენტი-სერვერი”

1. კლიენტის მხარეს დანართის კოდი სრულდება, რომელშიც აუცილებლად შედის საბოლოო მომხმარებელთან დამაკავშირებელი ინტერფეისი, ანგარიშების ჩასატარებელი დანართისათვის დამახასიათებელი სხვა ფუნქციების შესასრულებელი კომპონენტები.

2. დანართის კლიენტური მხარე ურთიერთობს მონაცემთა ბაზების მართვის პროგრამული უზრუნველყოფის კლიენტურ მხარესთან, რომელიც ფაქტობრივად დანართისათვის მზის ინდივიდუალური წარმომადგენელია.

შევნიშნოთ, რომ დანართის კლიენტური ნაწილსა და მონაცემთა ბაზის სერვერის კლიენტურ ნაწილს შორის ინტერფეისი, SQL ენის გამოყენებაზეა დაფუძნებული. ამიტომ, ისეთი ფუნქციები, როგორცაა, მაგალითად, ფორმების წინასწარი დამუშავება, მონაცემთა ბაზისადმი შეკითხვისათვის ან დანართის კოდში საბოლოო ანგარიშების ფორმირებისათვისაა განკუთვნილი.

ბოლოს, მონაცემთა ბაზის კლიენტური ნაწილი, ქსელური წვდომის საშუალებების გამოყენებით, გადასცემს მას SQL ენის ოპერატორის ტექსტს და მონაცემთა ბაზის სერვერს მიმართავს.

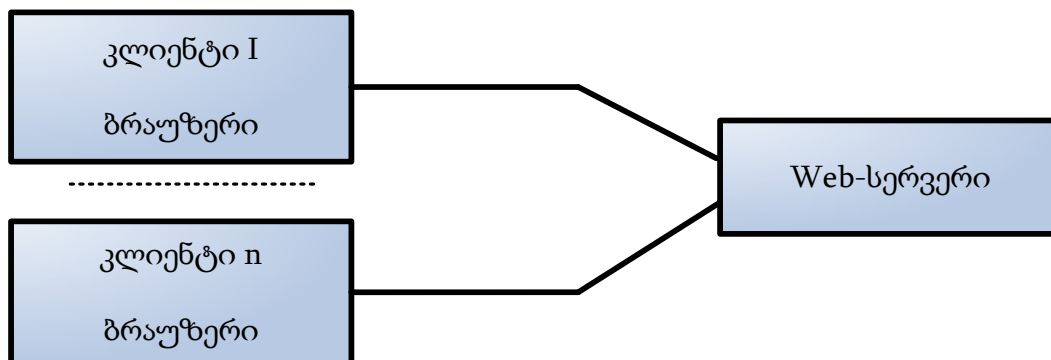
ამ არქიტექტურაზე დაფუძნებულ საინფორმაციო სისტემის დაპროექტებისას, უფრო დიდი ყურადღება უნდა მიექცეს საერთო გადაწყვეტილების სისწორეს. პილოტური ვერსიის ტექნიკური საშუალებანი შეიძლება მინიმალური იყოს. პილოტური ვერსიის შექმნის შემდეგ, საჭიროა დამატებითი კვლევითი სამუშაო ჩატარდეს, რათა სისტემის პრობლემური ნაწილები გაირკვეს და მხოლოდ ამის შემდეგ შეიძლება შეირჩეს სერვერის აპარატურა, რომელიც შემდეგ პრაქტიკაში გამოიყენება.

Intranet – დანართები

ფართო პრაქტიკაში მსოფლიოს Internet ქსელის გაჩენამ და ჩართვამ ძალაუვნებურად, კორპორაციული საინფორმაციო სისტემების შექმნის ტექნოლოგიაზე მოახდინა გავლენა, რამაც Intranet ქსელის სახელწოდებით ცნობილი ახალი მიმართულების შექმნა გამოიწვია. ინფორმაციული Intranet-სისტემა კორპორაციული სისტემაა, რომელშიც Internet-ის მეთოდები და საშუალებები გამოიყენება. ასეთი სისტემა შეიძლება იყოს ლოკალური, Internet-ის სამყაროსაგან იზოლირებული, ან ვირტუალურ კორპორაციულ Internet ქვექსელზე დამოკიდებული. ამ შემთხვევაში,

განსაკუთრებით არასანქცირებული შეღწევისაგან ინფორმაციის დაცვაა მნიშვნელოვანი.

ზოგადად, Intranet-სისტემაში ყველა შესაძლო Internet მომსახურება გამოიყენება და უფრო მეტი ყურადღება ექცევა ჰიპერმედიულ მომსახურებას WWW (World Wide Web-მსოფლიო ობობას ქსელი). ჩანს, რომ ამისათვის არსებობს ორი ძირითადი მიზეზი. ერთი, რომ HTML ჰიპერმედიული ენის გამოყენებისას შესაძლოა უფრო მარტივად შეიქმნას ინფორმაციის გამოყენების მოსახერხებელი სტრუქტურა, რომელსაც მომავალში ერთ-ერთი მზა Web-სერვერი მოემსახურება. მეორე, რომ გამოყენებისათვის რამდენიმე მზა კლიენტური ნაწილის არსებობა ბრაუზერებს საკუთარი ინტერფეისის შექმნის აუცილებლობისაგან ათავისუფლებს, აწვდის რა მომხმარებლებს საჭირო და განვითარებულ ინფორმაციასთან წვდომის საშუალებებს. რიგ შემთხვევებში, კორპორაციის საინფორმაციო სისტემის ასეთი ორგანიზაცია (ნახ. 20) საკმარისია, რათა კომპანიის მოთხოვნები დაკმაყოფილდეს.



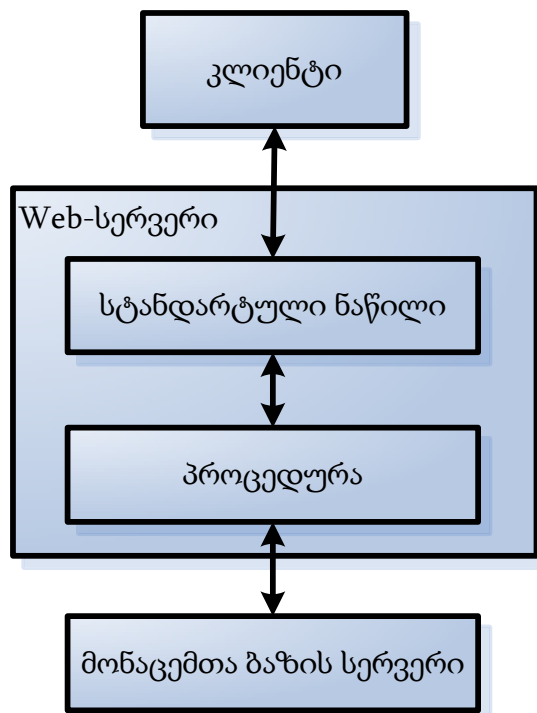
ნახ. 20. Intranet-სისტემის WWW საშუალებების გამოყენებით მარტივი ორგანიზაცია

რაც შეეხება დანართის ლოგიკას, Web-ტექნოლოგიების გამოყენებისას მათი რეალიზება Web-სერვერის მხარეს არის შესაძლებელი. ამისათვის, ორი მიდგომა შეიძლება იქნეს გამოყენებული – CGI CGI (Common Gateway Interface) და API (Application Programming Interface). ორივე მიდგომა HTML ენაში დაფუძნებულია არსებულ სპეციალურ კონსტრუქციაზე, რომელიც კლიენტ-ბრაუზერს ატყობინებს, რომ მან Web-სერვერს სპეციალური შეტყობინება უნდა გაუგზავნოს, რომლის მიღებისთანავე სერვერმა შესაბამისი გარე პროცედურა უნდა გამოიძახოს, მისი შედეგები მიიღოს და HTTP-ის სტანდარტულ ფორმატში კლიენტს დაუბრუნოს (ნახ. 21).



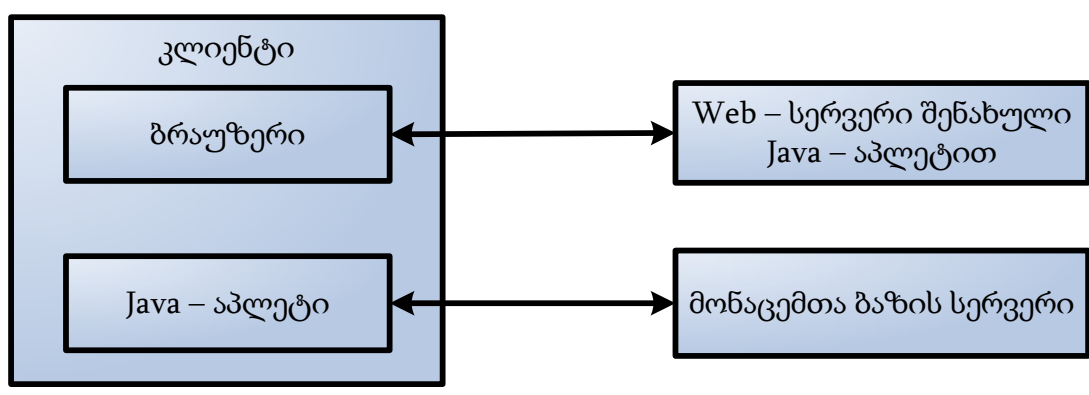
ნახ. 21. Web-სერვერის გარე პროცედურის გამოძახება

ანალოგიური ტექნიკა Intranet-სისტემებში მონაცემთა ბაზასთან უნიფიცირებული წვდომისათვის ფართოდ გამოიყენება. HTML ენა ჰიპერტექსტურ დოკუმენტებში ფორმის ჩასმის საშუალებას იძლევა. როდესაც ბრაუზერი ფორმას აწყდება, მომხმარებელს სთავაზობს შეავსოს იგი; შემდეგ, სერვერს უგზავნის შეტყობინებას, რომელიც შეტანილ პარამეტრებს შეიცავს. როგორც წესი, სერვერის ზოგიერთი გარე პროცედურა ფორმას მიეწერება. კლიენტისგან შეტყობინების მიღებისას, ამ გარე პროცედურას სერვერი გამოიძახებს მომხმარებლის პარამეტრების გადაცემით. ნათელია, რომ ასეთ გარე პროცედურას, შეუძლია Web-სერვერსა და მონაცემთა ბაზის სერვერს შორის დამაკავშირებელი როლი შეასრულოს. ამ შემთხვევაში წარმოიქმნება საინფორმაციო სისტემის კონფიგურაცია, რომელიც 22-ე ნახ-ზე სქემატურადაა გამოსახული.



ნახ. 22. Intranet-სისტემაში მონაცემთა ბაზებთან წვდომა.

Intranet-ზე წარმოდგენა არასრული იქნება, თუ Java ენის შესაძლებლობას არ აღვნიშნავთ. Java - ეს ინტერპრეტირებადი, C++ ენის საფუძველზე შექმნილი, ობიექტზე ორიენტირებული დაპროგრამების ენაა. Java-პროგრამის კომპილაციის შედეგად მიღებული მობილური კოდები (აპლეტები) შესაძლებელია HTML-დოკუმენტთან იყოს დაკავშირებული. ამ შემთხვევაში, ისინი დოკუმენტთან ერთად კლიენტის მხარეს ხვდებიან და შეიძლება ან ავტომატურად, ან აშკარა მითების საფუძველზე შესრულდეს. აპლეტი, შესაძლოა მონაცემთა ბაზის სერვერთან იყოს დამაკავშირებელი. Intranet-სისტემის ორგანიზაციის სქემა მონაცემთა ბაზებზე წვდომის ასეთი ტექნიკის გამოყენების შემთხვევაში, 23-ე ნახ-ზეა ნაჩვენები.



ნახ. 23. Intranet-სისტემის კლიენტის მხარეზე მონაცემთა ბაზის წვდომა

ჩვენი აზრით, Intranet-ი საინფორმაციო სისტემების დამუშავებისა და გამოყენების საკმაოდ მოსახერხებელი და უძლიერესი საშუალებაა. მის ერთადერთ ნაკლად შეიძლება ჩაითვალოს მექანიზმის მუდმივი ცვლილება და ბუნებრივი სტანდარტების არარსებობა. მეორე მხრივ, თუ საინფორმაციო სისტემა ტექნოლოგიის მიმდინარე დონის მიხედვით იქნება შექმნილი და კორპორაციის მოთხოვნების შესაბამისი აღმოჩნდება, მაშინ არავინ იქნება ვალდებული უფრო სრულყოფილი მექანიზმების გამოჩენის შემდეგ სისტემაში რაიმე შეცვალოს.

მონაცემთა საცავები (Data Warehousing), მონაცემების დამუშავების ოპერატიული და ანალიტიკური სისტემები

საჭირო მონაცემების უზრუნველყოფის თვალსაზრისით, განასხვავებენ ოპერატიულ და ანალიტიკურ ინფორმაციულ დანართებს. საქმე ეხება ე.წ. OLAP-სისტემებს, ანუ ანალიტიკურ სისტემებს, რომლებიც დინამიკურად შექმნილი ანალიზის, მონაცემთა მოდელირების და/ან პროგნოზირების საფუძველზე, მონაწილეობენ ბიზნეს-გადაწყვეტილების მიღებაში.

- ინფორმაციის ძირითად წყაროს, რომელიც ოპერატიულ მონაცემთა ბაზაში ხვდება, კორპორაციის საქმიანობა წარმოადგენს. მონაცემთა ანალიზის წარმოებისათვის ინფორმაციის გარე წყაროების ჩართვაა საჭირო. ამით, მონაცემთა საცავმა უნდა მოიცვას როგორც შიგა კორპორაციული მონაცემები, ის გარე მონაცემები, რომლებიც მთლიანობაში ბაზარს ახასიათებს.

- ოპერატიული დამუშავებისათვის ახალი მონაცემებია საჭირო. როგორც წესი, მონაცემთა ოპერატიულ ბაზაში ინფორმაცია კორპორაციის საქმიანობასა და ბაზრის მდგომარეობაზე მხოლოდ რამდენიმე თვის განმავლობაში ინახება; მაშინ როდესაც მონაცემთა საცავში ზუსტი ანალიზისა და პროგნოზისთვის ინფორმაცია რამდენიმე წლის განმავლობაში უნდა ინახებოდეს. შედეგად, ანალიტიკურ მონაცემთა ბაზას, დაახლოებით ოპერატიულზე უფრო დიდი მოცულობა აქვს.

- ბევრ მსხვილ კორპორაციაში ერთდროულად რამდენიმე ოპერატიული საინფორმაციო სისტემა არსებობს, საკუთარი მონაცემთა ბაზებით. ოპერატიულ მონაცემთა ბაზა შეიძლება შეიცავდეს სემანტიკურად ეკვივალენტურ ინფორმაციას, რომელიც მისი მიღების დროის სხვადასხვა მაჩვენებლით სხვადასხვა ფორმატშია წარმოდგენილი. კორპორაციის მონაცემთა საცავი ერთდროულად ოპერატიულ მონაცემთა ბაზებისაგან წარმოდგენილ ყველა მონაცემს უნდა შეიცავდეს. ეს ინფორმაცია ოპერატიული მონაცემთა ბაზების მიმდინარე შემცველობას მაქსიმალურად სრულად უნდა ესადაგებოდეს და იყოს შეთანხმებული. აქედან, გამომდინარეობს იმ მონაცემთა საცავის კომპონენტის არსებობის

აუცილებლობა, რომელიც ინფორმაციას ოპერატიულ მონაცემთა ბაზისაგან იღებს და “ასუფთავებს” ამ ინფორმაციას.

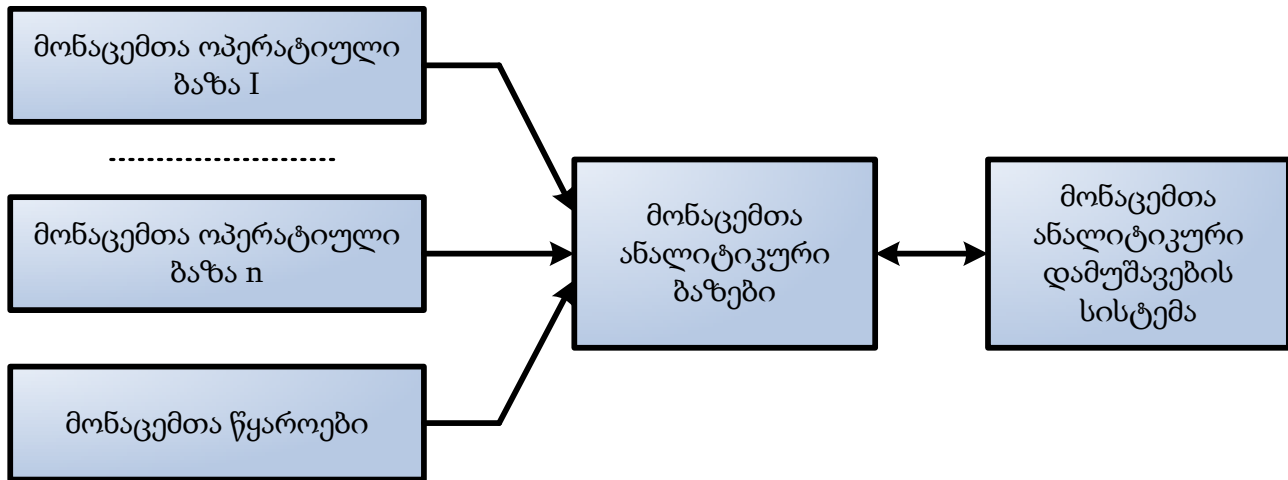
- ოპერატიული საინფორმაციო სისტემების დაპროექტება და დამუშავება კონკრეტული ამოცანების გადაწყვეტისათვის ხდება. ჩვეულებრივ, ოპერატიულ მონაცემთა ბაზის მიმართ შეკითხვათა ნაკრები უკვე სისტემის დაპროექტების ეტაპზე ხდება ცნობილი. ინფორმაციის ამორჩევა მონაცემთა ბაზებისაგან ხშირად და მცირე ულუფების სახით ხდება. ამიტომაც, ოპერატიულ მონაცემთა ბაზების დაპროექტებისას საჭიროცაა და აუცილებელიც წინასწარ ცნობილი შეკითხვათა ნაკრები იქნეს გათვალისწინებული. ანალიტიკური მონაცემთა ბაზისადმი შეკითხვების ნაკრების წინასწარმეტყველება შეუძლებელია. მონაცემთა საცავები იმისათვის არსებობს, რომ ანალიტიკოსების მოულოდნელ შეკითხვებს უპასუხოს. მხოლოდ იმის გათვლაა შესაძლებელი, რომ შეკითხვები ძალიან ხშირად არ შემოვა და ინფორმაციის დიდ მოცულობას არ მოითხოვს. ანალიტიკური მონაცემთა ბაზების ზომები შეკითხვების წარმოქმნას იწვევს თავისი აგრეგატებით (ჯამი, მინიმალური, მაქსიმალური, საშუალო მნიშვნელობა და სხვ.).

- ოპერატიულ მონაცემთა ბაზები, თავისი ბუნებით ცვალებადია. ეს გათვალისწინებულია მზმს-ში; კერძოდ, გავრცელებულ ინდექსაციის მექანიზმს წარმოადგენს B-ხეები, რომელთა მოდიფიკაცია სწრაფად სრულდება, ხოლო სტრიქონები ცხრილებში მოუწესრიგებლად ინახება. ანალიტიკურ მონაცემთა ბაზები მხოლოდ მაშინ იცვლება, როდესაც მათში ოპერატიული ან გარე ინფორმაცია იტვირთება. შედეგად, უფრო გონივრულია სხვა, უფრო სწრაფი მასიური ამორჩევის ოპერაციის ინდექსაციის მეთოდები იქნეს გამოყენებული, ინფორმაციული მასივების მოწესრიგებას დაუჭიროს მხარი, აგრეგირებული ფუნქციების წინასწარ გამოთვლილი მნიშვნელობები შეინახოს და ა.შ.

- თუ ოპერატიული საინფორმაციო სისტემებისათვის ინფორმაციის დაცვა ცხრილების დონეზეა (SQL – ორიენტირებული მონაცემთა ბაზების წესების მიხედვით) საკმარისი, ანალიტიკურ მონაცემთა ბაზების ინფორმაცია კორპორაციისათვის იმდენად კრიტიკულია, რომ მისი დაცვისათვის უფრო ფრთხილი მეთოდების (მაგალითად, რელაციურ მონაცემთა ბაზების გამოყენებისას ცხრილების ინდივიდუალური

სტრუქტურებისათვის და/ან სვეტებისათვის წვდომის ინდივიდუალური პრივილეგიების დაყენება) გამოყენებაა საჭირო.

ამ შენიშვნების გათვალისწინებით, მონაცემთა საცავისა და მონაცემთა ანალიტიკური დამუშავების სისტემების საერთო არქიტექტურა შესაძლოა ისე გამოიყურებოდეს, როგორც ეს 24-ე ნახ-ზეა ნაჩვენები.



ნახ. 24. ანალიტიკური საინფორმაციო სისტემის არქიტექტურის სქემატური წარმოდგენა

1993 წ. მონაცემთა ბაზების ორგანიზაციის რელიაციური მიდგომის დამაარსებელმა ედვარ კოდმა, მონაცემთა დინამიკური ანალიტიკური დამუშავების სისტემების მოთხოვნილების საფუძველზე, სისტემებისადმი ჩამოაყალიბა 12 ძირითადი მოთხოვნა, რომლებსაც ანალიტიკურ მონაცემთა ბაზები უჭერს მხარს. მოვიყვანოთ ამ მოთხოვნების შინაარსი, რათა მონაცემთა ანალიტიკური დამუშავების სისტემით დამპროექტებლისა და დამმუშავებლის თვალთახედვა წარმოვადგინოთ.

- მონაცემთა მრავალგანზომილებიანი კონცეპტუალური წარმოდგენა. ეს მოთხოვნა იმ მიზეზით ჩნდება, რომ ბიზნეს-მომხმარებელი ჩვეულებრივ, საკუთარი კორპორაციის ისტორიას და საქმიანობას, როგორც მრავალგანზომილებიანს (მაგალითად, პირველი განზომილება - დრო, მეორე - დამკვეთი, მესამე - წარმოებული პროდუქცია და ა.შ) წარმოადგენს. OLAP-მოდელებმა ამ წარმოდგენას მხარი უნდა დაუჭირონ და რა თქმა უნდა, ანალიტიკურ მონაცემთა ბაზის შესაძლებლობებს უნდა დაეყრდნონ.

- გამჭვირვალობა. ბიზნეს-მომხმარებლისათვის მნიშვნელოვანი არ უნდა იყოს, კონკრეტულად მონაცემთა დინამიკური ანალიზის საშუალებები სადაა განთავსებული. OLAP-სისტემის შემუშავებისას საჭიროა გახსნილი

სისტემების მიდგომა შენარჩუნდეს, რაც საგნის ანალიზის საშუალებების კორპორაციული ქსელის ნებისმიერ კვანძში განთავსების საშუალებას იძლევა.

- შეღწევადობა. ლოგიკური სქემა, რომელთანაც OLAP-სისტემა მუშაობს, მონაცემთა სხვადასხვაგვარ ფიზიკურ საცავებში უნდა აისახოს. მონაცემებთან წვდომისას, მათი მთლიანი და შეთანხმებული წარმოდგენა უნდა შენარჩუნდეს.

- ანგარიშების წარმოების შეთანხმებული ეფექტურობა. ამ ეფექტურობის დეგრადირება, გაზომვების რიცხვის გაზრდის შემთხვევაში არ უნდა მოხდეს.

- არქიტექტურა „კლიენტ-სერვერი“. OLAP-სისტემის სერვერული კომპონენტი საკმაოდ განვითარებული უნდა იყოს, რომ დამატებით “ინტერპრეტირებად” დაპროგრამებაზე გაწეული მინიმალური ძალისხმევითა და დანახარჯებით სხვადასხვა კლიენტს მასთან მიერთება შეეძლოს.

- თანდაყოლილი მრავალგანზომილება. სამუშაოს სტრუქტურული და ოპერაციული შესაძლებლობები მონაცემთა თითოეული განზომილებისათვის ეკვივალენტური უნდა იყოს. ყველა განზომილებისათვის მხოლოდ ერთი ლოგიკური სტრუქტურა უნდა არსებობდეს. ერთი განზომილებისათვის გამოყენებული ნებისმიერი ფუნქცია, სხვა ნებისმიერი განზომილებისათვისაც უნდა იყოს გამოყენებადი.

- დინამიკურად დაყოფილი მატრიცების მართვა. OLAP-სისტემის სერვერს დაყოფილი მატრიცების ეფექტურად შენახვა და დამუშავება უნდა შეეძლოს. წვდომის ფიზიკური მეთოდები სხვადასხვაგვარი უნდა იყოს.

- მრავალმომხმარებლიანი რეჟიმის მხარდაჭერა. OLAP-სისტემამ მონაცემებთან მრავალმომხმარებლიანი წვდომა (ამორჩევა და შეცვლა), მონაცემების მთლიანობა და უსაფრთხოება უნდა უზრუნველყოს.

- განზომილებებს შორის შეუზღუდავი ოპერაციები. მონაცემთა მრავალგანზომილებიანი ანალიზის ჩატარებისას ყველა განზომილება ერთნაირად იქმნება და მუშავდება. OLAP-სისტემას შესაბამისი გამოთვლების ჩატარება განზომილებებს შორის უნდა შეეძლოს.

- მონაცემების ინტუიციური მანიპულირება. მანიპულაციები, როგორც ანალიზის გზის ან დეტალიზაციის დონის ცვლილება, OLAP-მოდელის

ელემენტებზე პირდაპირი ზემოქმედებით, მენიუს ან სხვა დამხმარე საშუალებების გამოყენების მოთხოვნის გარეშე უნდა შესრულდეს.

- ანგარიშების მოხერხებული სისტემა. ბიზნეს-მომხმარებელმა უნდა შეძლოს მონაცემებით მანიპულირება, ანალიზი და/ან სინთეზირება, აგრეთვე მათი ისე დათვალიერება, როგორც ის მოინდომებს.

- გაზომვებისა და აგრეგაციის დონეების შეუზღუდავი რაოდენობა. OLAP-სერვერს თითოეული ანალიტიკური მოდელისათვის უნდა შეეძლოს არანაკლებ 15 განზომილებას დაუჭიროს მხარი. თითოეული განზომილები-სათვის მომხმარებლებისაგან განსაზღვრული აგრეგატების შეუზღუდავი რაოდენობა უნდა იქნეს დაშვებული.

ინტეგრირებული განაწილებული დანართები

არ არსებობს არანაირი პრობლემა, თუ დასაწყისიდანვე ინფორმაციული დანართების დაპროექტება და დამუშავება ღია სისტემების მიდგომების სახით ხდება: ყველა კომპონენტი მობილური და ინტეროპერაბელურია, სისტემის საერთო ფუნქციონირება კონკრეტული კომპონენტების ადგილმდებარეობაზე არაა დამოკიდებული, სისტემას თანმხლებადობისა და განვითარების კარგი შესაძლებლობები აქვს. სამწუხაროდ, პრაქტიკაში ეს იდეალი ძნელად მისაღწევია. დამოუკიდებელი და სხვადასხვაგვარად ორგანიზებული ინფორმაციულ-გამოთვლითი რესურსების ინტეგრაციის მოთხოვნილებები სხვადასხვა მიზეზით ჩნდება. ვერც ერთი, ნამდვილად სერიოზული დანაწილებული საინფორმაციო სისტემა ინტეგრაციის ტექნოლოგიის გამოყენების გარეშე ფონს ვერ გადის. საბედნიეროდ, დღეს არსებობს ამ პრობლემის გადაჭრის გზა, რომელიც გახსნილი სისტემების მართვაში დევს – მიდგომა, რომელიც მსხვილი საერთაშორისო კონსორციუმის OMG (Object Management Group) მიერ იყო შემოთავაზებული.

არაერთგვაროვანი ინფორმაციული რესურსების ინტეგრაციის პრობლემის გადაწყვეტა არაერთგვაროვან მონაცემთა ბაზის ინტეგრაციის ცდით დაიწყო. არაერთგვაროვან მონაცემთა ბაზის და მულტიმონაცემთა ბაზის ინტეგრირებული ან ფედერაციული სისტემების მიმართულება იმ მონაცემთა ბაზების სისტემის კომპლექსირების აუცილებლობით გაჩნდა,

რომლებიც სხვადასხვა მოდელებზე დაფუძნებულია და სხვადასხვა მბმს-ის საშუალებით იმართება.

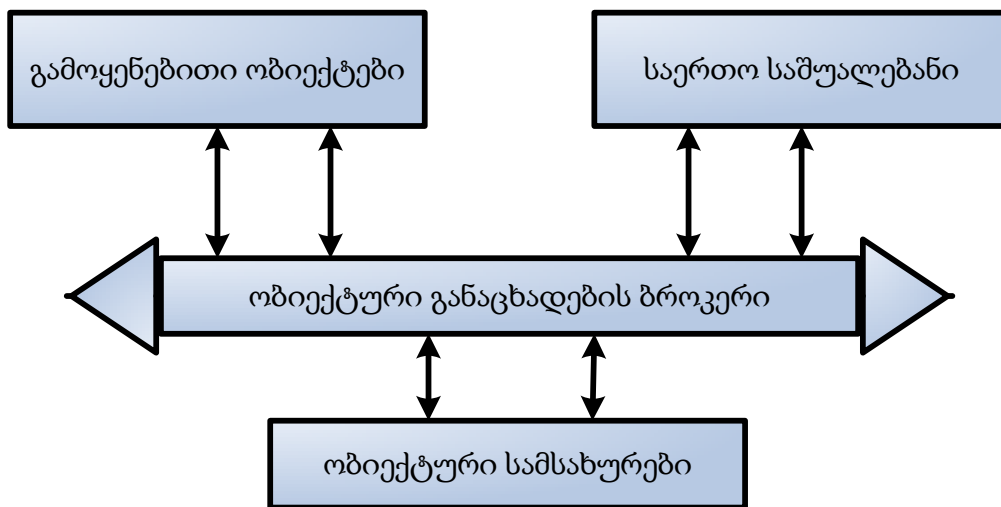
არაერთგვაროვან მონაცემთა ბაზის ინტეგრაციის ძირითადი ამოცანაა გლობალური სქემის მონაცემთა ბაზის ინტეგრალური სისტემის მომხმარებლებისათვის წარმოდგენა. მონაცემთა ზოგიერთი მოდელით წარმოდგენილი და გლობალური დონის ოპერატორების მონაცემთა ბაზის მანიპულირებით ოპერატორების ავტომატურად გარდაქმნა, შესაბამისი ლოკალური მბმს მიერაა აღქმული. თეორიულად, გარდაქმნის ეს პრობლემა გადაწყვეტილია, რეალიზაციაც არსებობს.

როგორც წესი, გამოთვლით ქსელში განლაგებული არაერთგვაროვან მონაცემთა ბაზების ინტეგრირება ხდება. ეს რეალიზაციას უფრო ართულებს. ინტეგრირებისას, საკუთარი პრობლემების გარდა, აუცილებელია ყველა იმ პრობლემის გადაჭრა, რომლებიც განაწილებულ მბმს თან სდევს: გლობალური ტრანზაქციების მართვა, შეკითხვების ქსელური ოპტიმიზაცია და ა.შ. ძალზე რთულია ეფექტურობის მიღწევა. როგორც წესი, ინტეგრირებული და მულტი-მონაცემთა ბაზების გარე წარმოდგენისათვის, მონაცემების რელაციური მოდელი გამოიყენება. ბოლო დროს, სულ უფრო ხშირად გამოიყენება ობიექტზე ორიენტირებული მოდელები, მაგრამ პრაქტიკაში ძირითადი მაინც რელაციური მოდელებია. ამიტომ, ლოკალურ რელაციური მბმს-ის ინტეგრირებულ სისტემაში ჩართვა გაცილებით მარტივი და ეფექტურია, ვიდრე მონაცემთა სხვა მოდელებზე დაფუძნებულ მბმს-ში ჩართვა.

არაერთგვაროვან მონაცემთა ბაზის ინტეგრაციის სისტემის მთავარი ნაკლი ის არის, რომ მასში გამოყენებითი სისტემების კომპონენტების “ქცევითი” ასპექტები არ არის გათვალისწინებული.

ადვილი შესამჩნევია, რომ ინტეგრაციული სისტემის არსებობის დროსაც კი, ზემოთ აღნიშნული მრავალი პრობლემა ვერ გადაიჭრება. ამ ტექნოლოგიის შექმნაში ყველაზე არსებითი წვლილი შეიტანა საერთაშორისო კონსორციუმმა OMG, რომელმაც გამოუშვა რიგი დოკუმენტები, სადაც განაწილებული საინფორმაციო სისტემების მხარდამჭერი არქიტექტურისა და ინსტრუმენტული საშუალებების სპეციფიკირება ხდება.

ბაზურ დოკუმენტში დანაწილებული საინფორმაციო სისტემის OMA (Object Management Architecture) არქიტექტურის ეტალონური მოდელის სპეციფიცირება ხდება (ნახ. 25).



ნახ. 25. OMA ეტალონური მოდელი

OMA-ს არქიტექტურასთან შეთანხმებული გამოყენებითი საინფორმაციო სისტემა წარმოადგენს კლასებისა და ობიექტების ეგზემპლარების ერთობლიობას, რომლებიც ORB (Object Request Broker) ობიექტური განაცხადების ბროკერის მეშვეობით ურთიერთობენ. ORB, საერთო საშუალებანი (Common Facilities) და ობიექტური სამსახურები (Object services) შუალედური პროგრამული უზრუნველყოფის (middleware) კატეგორიას განეკუთვნება და ერთად უნდა მიეწოდოს.

ობიექტური სამსახურები წარმოადგენს მომსახურების (ინტერფეისების და ობიექტების) ნაკრებს, რომელიც იმ საბაზო ფუნქციების შესრულებას უზრუნველყოფს, რაც გამოყენებითი ობიექტებისა და "საერთო საშუალებების" კატეგორიის ობიექტების რეალიზაციისათვის არის საჭირო (მაგალითად, ობიექტთა სახელდების სპეციფიცირებული სამსახური, ობიექტთა ხანგრძლივი შენახვის სამსახური, ტრანზაქციების მართვის სამსახური და ა.შ.). საერთო საშუალებანი მოიცავს კლასებისა და ობიექტთა ეგზემპლარების ნაკრებს, რომლებსაც სხვადასხვა გამოყენებით სფეროში სასარგებლო ფუნქციები უჭერს მხარს (მაგალითად, მომხმარებლის ინტერფეისის მხარდამჭერი საშუალებანი, ინფორმაციის მართვის საშუალებანი და ა.შ.).

OMA-ს საფუძველში OM (Cove Object Model) ბაზური ობიექტური მოდელი დევს, რომელშიც ისეთი გაგება სპეციფირდება, როგორცაა ობიექტი, ოპერაცია, ტიპი, ქვეტიპიზაცია, შთამომავლობა, ინტერფეისი. აგრეთვე, სხვადასხვა ობიექტურ სამსახურში COM-ის შეთანხმებული გაფართოების ხერხებია განსაზღვრული.

არაერთგვაროვანი ინფორმაციული რესურსების ინტეგრაციის პრობლემა კორპორაციისათვის აქტუალურია და ამ პრობლემის გადაჭრის ტექნოლოგიები არსებობს. დაბოლოს, ზემოთ მოყვანილი მეთოდებისა და ტექნოლოგიების კლასიფიკაცია ორთოგონალური არ არის. რეალურ ცხოვრებაში საჭიროა ტექნოლოგიური და არქიტექტურული გადაწყვეტის ოპტიმალური შერწყმის გამოყენება.

ფაილ-სერვერული დანართების დაპროექტების, დამუშავებისა, თანმხლები დოკუმენტების საშუალებანი და მეთოდოლოგია

ფაილ-სერვერული დანართების დამუშავების ტრადიციული საშუალებანი და მეთოდოლოგიები

უნდა აღვნიშნოთ, რომ, მართალია, ფაილ-სერვერული დანართების დამუშავებისათვის მთელი რიგი ინსტრუმენტული საშუალებები არსებობს, მაგრამ საერთო მიღებული მეთოდოლოგიები არ არსებობს. უფრო სწორად, როცა გამოიყენება მეთოდოლოგია, მაშინ იგი ისეთივეა, როგორც კლიენტ-სერვერულ დანართებში. ჩვეულებრივ კი, ფაილ-სერვერული დანართების დაპროექტება და დამუშავება რაიმე სტანდარტული მეთოდის გამოყენების გარეშე, „ადგილზე“ ხდება.

დაპროგრამების სისტემები და ბიბლიოთეკები

მესამე თავის 3GL დაპროგრამების სისტემები თანამედროვე ინსტრუმენტული საშუალებების წინაპრებს წარმოადგენს და შესაძლოა, ინფორმაციული დანართების დამუშავების დროს, დიალოგის რეალიზაციისა და მონაცემთა ბაზების წვდომისათვის სათანადო

ჩაშენებული ან ბიბლიოთეკური საშუალებების არსებობის შემთხვევაში იყოს გამოყენებული.

პერსონალური კომპიუტერისათვის სისტემურმა დაპროგრამებამ განვითარების საკმაოდ დიდი გზა გაიარა. შესაძლოა გამოკვეთილი იქნეს სამი მკაფიო ენობრივი ხაზი, რომლებიც ერთმანეთს ამდიდრებს და ერთმანეთზე დიდ ზეგავლენას ახდენს. ესენია – C, Pascal, Basic.

აღვნიშნოთ დაპროგრამების სისტემების განვითარების გზაზე არსებული ძირითადი დაბრკოლებები:

- ერთეული უტილიტების დაპროგრამების სისტემებიდან ინტეგრირებული დიალოგური დაპროგრამების გარემოზე გადასვლა (მაგ.: ფირმა Borland-ის Turbo-პროდუქტები);

- იმ ინსტრუმენტული ნაკრებების განვითარება, რომლებიც დაპროგრამების სისტემების გამაფართოებელი საშუალებებია, კერძოდ, დიალოგის ველში (სხვადასხვა სახის Tool Box);

- ობიექტზე ორიენტირებული დიალექტების გაჩენა C და Pascal ენების სახით; მიუხედავად იმისა, რომ Pascal უფრო მკაცრ და კორექტულ ენას წარმოადგენს, C++ ფენომენს სტანდარტის არსებობაში უფრო დიდი მნიშვნელობა აქვს;

- ჩაშენებულდიალოგიანი ოპერაციული გარემო Windows-ის გაჩენა და პირველი Windows-დანართი SDK (Software Development Keet) მეშვეობით;

- ობიექტზე ორიენტირებული იმ ბიბლიოთეკის შექმნა, რომელიც DOS და Windows გარემოს დიალოგურ რეჟიმში მუშაობს (Turbo Vision, Object Windows და MFC);

- დაპროგრამების სისტემები, რომლებიც Windows-ისათვის დანართების შექმნას აადვილებს;

- ჩაშენების და OLE2-ის ობიექტებთან დაკავშირების მექანიზმის განვითარება;

- ვიზუალური დაპროგრამების სისტემებზე გადასვლა (Visual C++, Delphi, Visual Basic), რომლებიც ინფორმაციული დანართების შექმნაზეა ორიენტირებული.

დიალოგური რეჟიმის განვითარება თვით დაპროგრამების განვითარებასთან ერთად გრძელდებოდა და ბუნებრივად იყო მათთან ინტეგრირებული. მონაცემთა ბაზებთან წვდომის ბიბლიოთეკები კი, თავისი გზით ვითარდებოდა. რელაციური მბმს წვდომის ბიბლიოთეკების

უმეტესობა 3GL დონის დაპროგრამების ენებიდან xBase (Clipper, FoxPro, dbase) ჯგუფს წარმოადგენს. დაპროგრამების ენებიდან ყველაზე ხშირად C-ენა გამოიყენება. ასევე ისეთი ბიბლიოთეკის არსებობა უნდა აღინიშნოს, როგორც CodeBase და DB Tools.

CodeBase ბიბლიოთეკაში მონაცემთა იგივე ფორმატები, ინდექსები და memo-ფაილები გამოიყენება, როგორც dbase IV მბმს-ში. ბიბლიოთეკაში არსებულ Code Base ფუნქციებს შეუძლია არა მხოლოდ xBase ჯგუფის მბმს ყველა სტანდარტული ფუნქცია შეასრულოს, არამედ ფილტრების დაყენების, დამოკიდებულების მიწოდების და მბმს-ში ყველა დასაშვები გამოსახულებების გამოთვლის შესაძლებლობას იძლევიან.

ბიბლიოთეკა DBTools C/C++ ენებისათვის მრავალპლატფორმიან ბიბლიოთეკას წარმოადგენს და xBase, Informix, Oracle და სხვა ჯგუფის მბმს-ის მხარდასაჭერადაა გათვლილი.

პერსონალურ კომპიუტერებზე მბმს საფუძველზე დანართის დამუშავების საშუალებები და მეთოდები

დანართები, რომლებიც დაპროგრამების ინსტრუმენტული საშუალებების გამოყენებითაა შექმნილი და პერსონალურ კომპიუტერებში მონაცემთა ბაზების გამოყენებასთანაა დაკავშირებული, მნიშვნელოვანი ნაწილი ფაილ-სერვერულ დანართებს უკავიათ. თუ მხოლოდ “რელაციურ” მბმს განვიხილავთ, მაშინ xBase-პროდუქტების ჯგუფი, ცალკეული და ჯგუფური ინფორმაციული დანართების დამუშავებაში ლიდერს წარმოადგენს. შემდეგ ადგილს მბმს Paradox იკავებს, ამის შემდეგ მოდის დანართები, რომლებიც ჩანაწერების მართვის Clarion სისტემის გამოყენებაზეა ბაზირებული. ცალკეა ისეთი პაკეტები, როგორც MS Access და Lotus Approach, რომლებმაც საშუალება მოგვცა ახლებურად დაგვენახა პერსონალური მბმს შესაძლებლობები და როგორც დანართების დამუშავების პროფესიული საშუალებანი, დღემდე არ არის შეფასებული.

ფაილ-სერვერული დანართების დამუშავების ახალი საშუალება

რაც მეტი დრო გადის, მით უფრო მეტი ფაილ-სერვერული დანართების უფრო განვითარებულ კლიენტ-სერვერულ დანართებთან დაახლოება ხდება. ბოლო დროს გაჩნდა მთელი პროგრამული

პროდუქტების სერია, რომელიც ერთდროულად პერსონალური კომპიუტერების დანართების დამუშავების მარტივი და ამავე დროს, “კლიენტ-სერვერ” ტექნოლოგიაში დანართების დამუშავების უფრო რთული საშუალებაა. ეს კარგიცაა, რადგან შესაძლებელია ეტაპობრივად შეიცვალოს ტექნოლოგია.

თანამედროვე საშუალებების საერთო დახასიათება

პერსონალური კომპიუტერების მბმს-ის ინდუსტრიაში სისტემების ნორმალიზაციის (Rightsizing) ტენდენცია გამოიკვეთა. ბოლო დროს, ამ სფეროში ორი შემხვედრი პროცესი მიმდინარეობდა: 1. მონაცემთა ბაზების სერვერების გამსხვილება - მონაცემთა ბაზების სერვერების Informix, Oracle და ა.შ., თავიდან მუშა ჯგუფისათვის ახალი ვერსიების, შემდეგ კი ცალკეული პერსონალური კომპიუტერებისათვის გამარტივებული ვერსიების გაჩენა; 2. პერსონალური კომპიუტერებისათვის მბმს გამსხვილება – ახალი “პერსონალური” მბმს და მათთან დაკავშირებული ინსტრუმენტული საშუალებები „ნამდვილად რელიაციური” მბმს-ის ანუ მონაცემთა ბაზების სერვერების, კლიენტ-სერვერებისა და დაპროგრამების UGL და RAD სწრაფი დამუშავების ინსტრუმენტული საშუალებების დანართების მხარეს ვითარდებოდა.

პერსონალური კომპიუტერების ახალი მბმს და შესაბამის ფაილ-სერვერული დანართების დამუშავების ინსტრუმენტულ საშუალებებს, შემდეგი ჩამოთვლილი საერთო თვისებები აქვს:

- დანართების დაპროგრამების ვიზუალური ხასიათი, განსაკუთრებით სამომხმარებლო დიალოგური გრაფიკული ინტერფეისის შექმნის ნაწილში. ეს რთული პროცედურების რეალიზების დამხმარე დიალოგური ობიექტებისა და drag-and-drop მექანიზმის მხარდაჭერი საშუალება და ოსტატების სიმრავლეა;

- დანართების დიალოგის მოვლენების თანახმად, მართვა და მონაცემთა ბაზების წვდომის უზრუნველყოფა შესაძლებლობას იძლევა მოქნილი მომხმარებლის ინტერფეისი აიგოს და მონაცემთა ბაზის მიმართვის მთლიანობა შეინარჩუნოს;

- SQL (Standard Query Language) ენის ფაილ-სერვერული დანართების სტრუქტურირებული შეკითხვების ჩამუშავებული მხარდაჭერა, რომლებიც

კლიენტ-სერვერული დანართის დონემდე უნდა შეიქმნას, მასშტაბურობის შესაძლებლობას იძლევა;

- კლიენტ-სერვერული დანართების მონაცემთა ბაზების სერვერებთან პირდაპირ ან ODBC ინტერფეისით წვდომის რეალიზაციის ხარჯზე, მონაცემთა ბაზებთან ღია ურთიერთობისათვის აგების შესაძლებლობა არსებობს;

- დანართის დამუშავების ობიექტზე ორიენტირებული ენის გამოყენება შთამომავლობის მექანიზმის ფართოდ გამოყენების შესაძლებლობას იძლევა და ამით, ადრე შექმნილი პროგრამული კომპონენტები შეიძლება იქნეს გამოყენებული;

- კომპონენტზე ორიენტირებული დაპროგრამების მხარდაჭერა მზა გარე ვიზუალური ობიექტების გამოყენების ხარჯზე, დანართის გაფართოების შესაძლებლობას იძლევა;

- “ნამდვილად რელაციური” მონაცემთა ბაზა ფაილების გაერთიანებული ნაკრებია, იგივე შეიცავს ცხრილებს, ინდექსებს და ა.შ. ამარტივებს მონაცემთა ბაზის და დამატების თანხლებას ამავდროულად, მონაცემების მთლიანობის შენარჩუნების საფუძველს წარმოადგენს;

- საინფორმაციო სისტემისათვის იწარმოება საერთო მონაცემთა ლექსიკონი (data dictionary), რომელიც მოიცავს მონაცემთა ბაზების სტრუქტურის აღწერას, ველების ტიპებს, მთლიანობის შეზღუდვის შენარჩუნების წესებს და ა.შ.;

- მონაცემთა ბაზის მთლიანობის შენარჩუნება (მონაცემები, მიმართვები და ტრანზაქციები) უსაფრთხოებისა და მონაცემთა შენახვის შესაბამისი დონით, დანართების შექმნის შესაძლებლობას იძლევა;

- შესაქმნელი დანართის პროექტის აღწერილობის მონაცემთა ბაზაში შენახვა RAD და CASE-სისტემების სწრაფი დამუშავების ინსტრუმენტული საშუალებების რეპოზიტორის გარდაქმნა;

- დამუშავების სერვერული პროცედურების (ტრიგერი და შენახული პროცედურები) შესაძლებლობები მასშტაბირებული დანართის საფუძველს ქმნის, რაც კლიენტ-სერვერულ არქიტექტურაზე გადასვლისას, გამოყენებითი ლოგიკა კლიენტსა და სერვერს შორის მოქნილად გადანაწილების საშუალებას იძლევა.

შედარებით ახალი ინსტრუმენტული საშუალებების რამდენიმე მაგალითი უფრო დაწვრილებით განვიხილოთ: MS Access, Visual FoxPro და CA-Visual Objects.

ახალი მიდგომის მაგალითები

MS Access პაკეტი

Microsoft Access – პერსონალური კომპიუტერებისათვის Windows გარემოში სამუშაოდ შექმნილი პირველი მბმს, და ფაილ-სერვერული დანართების დასამუშავებელი ახალი ინსტრუმენტული სისტემების მრავალ თვისებას ატარებს. ეს სისტემა როგორც საბოლოო მომხმარებელზე, ისე პროფესიონალ პროგრამისტებზეა ორიენტირებული და როგორც ერთის, ასევე მეორისათვის მონაცემთა ბაზების დამუშავებასა და წვდომას, გრაფიკული ინტერფეისით ინფორმაციული დანართების სწრაფ შექმნას აიოლებს. სისტემა ოპერაციული სისტემების შემდეგ ვერსიებში მუშაობს: Windows XP და Windows Vista, Windows 7. Microsoft Access პაკეტი Microsoft Office პროგრამული ოჯახის შემადგენელ ნაწილს წარმოადგენს. ყველა ოჯახი დაფუძნებულია Intellisense ინტელექტუალურ ტექნოლოგიაზე, რომელიც „გრძნობს“ რა უნდა მომხმარებელს და საჭირო შედეგს იძლევა, რუტინულ ოპერაციებს ავტომატურად ასრულებს და აიოლებს რთულ ამოცანებს.

Microsoft Access-ში მონაცემებთან უშუალო მუშაობის დროს MS Word-ის გამოყენებაცაა შესაძლებელი. Microsoft Access-ში კავშირის გრაფიკული კონსტრუქტორისა (Graphical System Relationships Builder) და გრაფიკული შეკითხვების (Graphical query) სამსახურები არსებობს. ეს საშუალებები არა მარტო მონაცემთა ბაზებს ქმნის, არამედ მის კონსტრუქციას თვალნათლივ ახდენს, რაც Microsoft Access-ს CASE-ტექნოლოგიებთან აახლოებს. კავშირის გრაფიკული კონსტრუქტორი მონაცემთა ბაზის კონსტრუქციის ინტუიციურად მოხდენის საშუალებას იძლევა, ხოლო გრაფიკული შეკითხვის ფუნქცია ძალზე რთული შეკითხვების შექმნასაც კი ამარტივებს. Microsoft Access-ში აგრეთვე არსებობს ფუნქციები და ტექნოლოგიები, რომლებიც სიჩქარეს ზრდის და საბოლოო საშუალებების მოხმარებას ამარტივებს. მათ მიეკუთვნება:

- ტექნოლოგია Rushmore;
- სწრაფი დახარისხება (Quick Sort);
- განსაკუთრებით ხშირად შესრულებადი შეკითხვების საშუალებანი (Top Value queries).

Microsoft Access-ში ინფორმაციის სხვა დანართებთან ერთად, გამოყენების რიგი საშუალებები არსებობს; Office Links OLE 2.0 ტექნოლოგიის გამოყენებით, ერთი პროგრამიდან მეორეში ინფორმაციის გადაცემის შესაძლებლობას იძლევა. Analyze It და Publish It დილაკებით მომხმარებელს შეუძლია ანალიზისათვის მონაცემები Excel-ში ან Word-ში გადაიტანოს.

Microsoft Access ფაილების ფორმატების დიდ ნაწილთან შეუძლია მუშაობა, ვინაიდან Microsoft Access-ი Btrieve, dBase PLUS და dBase, Microsoft FoxPro, Paradox, Microsoft SQL Server, SYBASE SQL SERVER სისტემების სრულ მხარდაჭერას უზრუნველყოფს. აგრეთვე, შესაძლებელია ODBC დრაივერების სხვა მონაცემთა ბაზებთან წვდომისათვის გამოყენება.

Microsoft Access შემქნელებისათვის მძლავრი ინსტრუმენტია და მისი უნივერსალური გარემო Microsoft Visual Basic დონეზე დაპროგრამების შესაძლებლობას იძლევა.

მენიუს კონსტრუქტორი (Menu Builder) ისეთ გრაფიკულ ინსტრუმენტებს აწვდის, რომლითაც მენიუს შექმნა დაპროგრამების გარეშე შესაძლებელი. დამუშავების სიჩქარე Microsoft Access შესაძლოა ორი ცალკეული პაკეტის საშუალებით გაიზარდოს: Microsoft Access Solutions Pack და Microsoft Access Developer's Toolkit. აღსანიშნავია, რომ Microsoft Access Solutions Pack ბიზნესის ინფორმაციული უზრუნველყოფისათვის ოთხ მზა უნივერსალურ დანართს მოიცავს:

- Sales Manager – დამკვეთებთან კონტაქტისა და მათი საქმიანი შესაძლებლობების შესახებ ინფორმაციის შენახვას, თვალყურის დევნასა და მოძებნას ამარტივებს;

- Asset Tracker – აღრიცხვისა და აქტივების მართვის დროს ეხმარება;
- Registration desu – შედეგების რეგისტრაციის რთულ, მაგრამ აუცილებელ შრომას ამარტივებს;

- Service Desk – მომსახურების ხარისხს ამაღლებს და მომსახურების შესახებ მიღებული შეკვეთების დამუშავებაში, რეგისტრაციიდან დამუშავების დამთავრებამდე და შემოწმებამდე ეხმარება.

Microsoft Access DDeveloper's Toolkit-ი Microsoft Access-ისათვის დანართების შესაქმნელ საჭირო ინსტრუმენტებს შეიცავს, ისეთებს როგორცაა: ცნობების კომპილატორი, Microsoft Access შესრულებადი ვერსია; Microsoft Graph, Setup Wizard, დოკუმენტაცია და ობიექტების შექმნის პროგრამების მაგალითები, რომლებიც უზრუნველყოფს მონაცემებთან წვდომას; ოსტატები და OLE 2.0 მართვის დილაკები; Microsoft Access (Microsoft Access Language Reference) ცნობარი და სახელმძღვანელო გამოცდილი მომხმარებლისათვის (Advanced Topics).

სისტემა Visual FoxPro

Visual FoxPro-ში მრავალი ახალი ნიშან-თვისებაა, როგორცაა: ობიექტზე ორიენტირებული ენა, აქტიური ლექსიკონი, მონაცემთა ბაზის სერვერებთან მიმართვის ჩაშენებული საშუალებები და ა.შ.

Visual FoxPro უკვე Microsoft სხვა პროდუქტებისაგან განცალკევებით აღარ დგას. თვით პროდუქტის ინტერფეისი და დანართები, რომლებიც მის ბაზაზე მუშავდება, პროგრამის პროდუქტების კომპლექსში მიღებულ Microsoft Office სტანდარტებს შეესატყვისება. უფრო მეტიც, Visual FoxPro-ს Microsoft Office-ის სხვა დანართებთან OLE Automation-ის საშუალებით მთლიანად ინტეგრირება ხდება. Visual FoxPro-ში დაწერილი პროგრამა, Microsoft Word, Microsoft Excel და სხვა დანართებთან სრულყოფილად შეძლებს ურთიერთობას. როგორც ადრე, მონაცემების დინამიკური გაცვლა DDE არის შენარჩუნებული.

პაკეტი Visual FoxPro სრულყოფილი 32-თანრიგიანი დანართია, რომელიც Windows XP-თან მუშაობს. ტექნიკის მიმართ მოთხოვნებით, Visual FoxPro ჰგავს Microsoft Access.

დაპროგრამების გარემო CA-VISUAL OBJECTS

Computer Associates ფირმის მიერ დამუშავებული ახალი პროდუქტი CA-VISUAL OBJECTS, ფართოდ გავრცელებული დაპროგრამების სისტემის - Clipper შთამომავალს წარმოადგენს. იგი, Windows გარემოში მომხმარებლის გრაფიკული ინტერფეისით ინფორმაციული დანართების შექმნაზეა ორიენტირებული.

სინტაქსის თვალსაზრისით, VISUAL OBJECTS თითქმის 90%-ით ემთხვევა CA-Clipper-ს. მაგრამ Clipper-ის ხაზის უბრალო გაგრძელებას არ წარმოადგენს. თუმცა, Visual Objects-ში მუშაობის დროს შესაძლებელია Clipper გამოვიყენოთ და დანართის ახალ სისტემაში ადრე დამუშავებული კოდი ჩავრთოთ.

დანართები, მოდულები და კომპონენტები რეპოზიტორში - ერთგვარ გლობალურ მონაცემთა ბაზაში, ინახება. რეპოზიტორს კიდევ ერთი მნიშვნელოვანი ფუნქცია აქვს – დანართის ყველა ნაწილი უმცირესი დონის კომპონენტებიც კი მართოს.

Visual Objects-თან ერთად, მომხმარებელი იღებს ბიბლიოთეკების დიდ არჩევანს, რომლებიც IDE-შია ინტეგრირებული. ცენტრალური ადგილი System Library ბიბლიოთეკას უჭირავს, რომელიც ყველა სისტემურ საშუალებას მოიცავს.

Visual Objects-თან მუშაობის დროს გრაფიკულ ინტერფეისში დაპროგრამების 2 ვარიანტი არსებობს – ტერმინალის ემულაციის პროგრამების ან Common View ბიბლიოთეკის გამოყენება.

CA-VISUAL OBJECTS შესაძლებლობას იძლევა ვიზუალური დაპროგრამების საშუალებების მძლავრი ნაკრები გამოვიყენოთ. Visual Objects-ის გამოყენების სირატულე ისაა, რომ დასამსხვრევია ტრადიციული შეხედულებები და ჩვევები, რომლებიც პროცედურული დაპროგრამებიდან ობიექტზე ორიენტირებულ დაპროგრამებაზე გადასვლისას წარმოიქმნება. მალე გაჩნდება პროგრამული სისტემების დიდი რაოდენობა, რომლებიც Visual Objects ბაზაზე იქნება შექმნილი და კლიენტ-სერვერული არქიტექტურის საინფორმაციო სისტემების ყველა ღირსების რეალიზებას მოახდენს.

ფაილ-სერვერული დანართების კლიენტ-სერვერულ გარემოში გადატანა

ფაილ-სერვერული დანართების კლიენტ-სერვერულ არქიტექტურაში ინტეგრირებისა და ადაპტაციის რამდენიმე მიდგომა არსებობს:

- მონაცემთა ბაზების სერვერთან წვდომის ბიბლიოთეკების გამოყენება;
- მონაცემთა ბაზების სერვერთან ODBC ღია პროტოკოლით კავშირი;

- ფაილ-სერვერული დანართების გამსხვილება.

ამ საშუალებების შესაძლებლობათა შეფასების მიზნით, ისინი განვიხილოთ შედარებით დაწვრილებით.

მონაცემთა ბაზებთან წვდომის ბიბლიოთეკები

დანართების სერვერთან წვდომის ბიბლიოთეკების გამოყენება Clipper ან Clavon დაპროგრამების სისტემებზე აგებული ფაილ-სერვერული დანართების ადაპტაციისათვის არის მოსახერხებელი.

ინტერფეისი ფუნქციების ბიბლიოთეკის სახითაა რეალიზებული, რომელიც Clipper ენაზე შედგენილ გამოყენებით პროგრამებში გამოსაყენებლადაა ხელმისაწვდომი და ყველა საჭირო ოპერაციას ასრულებს. Oracle მონაცემთა ბაზაში ფუნქციები, Clipper და C ენებზეა დაწერილი.

ამ ბიბლიოთეკის ფუნქციების საშუალებით სისტემა Oracle-ის მონაცემთა ბაზის ცხრილებზე შესაძლებელია შემდეგი ოპერაციები შესრულდეს: შევუერთდეთ სისტემა Oracle-ს, მონაცემთა ბაზაში ახალი სტრიქონი ჩავამატოთ; არსებული სტრიქონი წავშალოთ; არსებული სტრიქონის ველის შემცველობის მოდიფიკაცია ჩავატაროთ; შევასრულოთ სტრიქონის ძებნა და სხვ.

აგრეთვე, SQL ენის, რომელიც არა მხოლოდ Oracle სისტემაში, არამედ სხვა მრავალ განვითარებულ მბმს-ში მონაცემების დამუშავების ძირითად ენას წარმოადგენს, პირდაპირი გამოყენების შესაძლებლობა არსებობს.

ბიბლიოთეკის ფუნქციების საშუალებით შეიძლება Oracle-ის ყველა ტიპის ცხრილი დამუშავდეს, მათ შორის ვირტუალური, კლასტერიზებული, ინდექსიანი და უინდექსო ცხრილები. ამასთან, უზრუნველყოფილი იქნება Oracle-ის მონაცემთა ფორმატების Clipper-ის მონაცემთა ფორმატებად გარდაქმნა და პირიქით.

ბიბლიოთეკის შემადგენლობაში შემდეგი ფუნქციები შედის:

INIT () – Oracle-სთან მიერთება;

OPEN () – ცხრილის გახსნა;

INSERT () – სტრიქონის დამატება;

UPDATE () – სტრიქონის კორექტირება;

DELETE () – სტრიქონის წაშლა;

SELECT () – სტრიქონის ძებნა;

NEXT () – მომდევნო სტრიქონის კითხვა;

SET () – განსაზღვრული ნომრით სტრიქონის კითხვა;
 SKIP () – სტრიქონის გამოტოვება;
 FILTER () – პირობის მიხედვით, სტრიქონის ამორჩევა;
 GETIND () – ინდექსის შენახვა;
 SETIND () – ინდექსის დაყენება;
 COMMIT () – ტრანზაქციის დასასრული;
 SAVE () – საკონტროლო წერტილის დაყენება;
 ROLL () – ტრანზაქციის უკუსვლა;
 LOCK () – ცხრილის ბლოკირება;
 SQL () – SQL ოპერატორის შესრულება;
 CLOSE () – ცხრილის დახურვა;
 STOP () – Oracle-დან გათიშვა.

ეს ფუნქციები ასევე Clipper-ს ხელს არ უშლის მონაცემთა მართვის საკუთარი საშუალებები გამოიყენოს, რაც შესაძლებლობას ქმნის სხვადასხვაგვარი მონაცემთა ბაზები ერთობლივად დამუშავდეს. ასე რომ, მონაცემთა ბაზა Oracle ცენტრალურად შეიძლება ჩაითვალოს, ხოლო მონაცემთა ბაზა Clipper მომხმარებლის პერსონალურ კომპიუტერზე შეიძლება ლოკალურად გამოიყენონ.

პროტოკოლი ODBC და მისი რეალიზაცია

გამოყენებითი დაპროგრამების ODBC API ინტერფეისი, SQL ენის საფუძველზე მონაცემთა ბაზების როგორც რელიაციურ, ისე არარელიაციურ (ISAM) მონაცემთა წყაროებთან წვდომის საერთო მეთოდებს წარმოადგენს.

ANSI SQL ყველაზე თანამედროვე სტანდარტი ინტერფეისის სპეციფიკაციას (CLI – Call - Level Interface) შეიცავს გამოძახების დონეზე, რომელსაც ODBC მონაცემთა ბაზების მართვის ბევრ სისტემაში მონაცემთა წვდომისა და მუშაობის უზრუნველსაყოფად ეფუძნება.

ODBC API ინტერფეისი Windows-ისათვის რეალიზებულია, როგორც DLL-ფუნქციების ფენებად დაყოფილი ნაკრები. ODBC DLL-ის დინამიკური ბიბლიოთეკა არის ODBC დრაივერების მართვის ძირითადი ბიბლიოთეკა, რომელიც სპეციალიზებული დრაივერების გამოძახების ფუნქციებს მოიცავს. ყოველი დრაივერი თავისი CLI დონის შესაბამისია და ორიდან ერთ კატეგორიას მიეკუთვნება: ერთდონიანი ან მრავალდონიანი დრაივერები.

ერთდონიანი დრაივერი მონაცეთა იმ წყაროებთან სამუშაოდ არის განკუთვნილი, რომელთა დამუშავება ANSI SQL-ის გამოყენებითაა შეუძლებელი. ჩვეულებრივ, ეს მონაცემთა ლოკალური ბაზებია, როგორცაა dBase, Paradox, Fox Pro და Excel.

მრავალდონიანი დრაივერები სერვერებს SQL-წინადადების დასამუშავებლად იყენებენ და კლიენტ-სერვერულ გარემოში სამუშაოდაა განკუთვნილი.

ODBC API საშუალებით მონაცემების მოთხოვნის პროცედურის 4 მნიშვნელოვანი ეტაპი არსებობს:

1. შეერთების დამყარება – (handle) ODBC შორის იმ მიმთითებლების განთავსებაში გამოიხატება, რომლებიც ODBC დრაივერებისა და ბიბლიოთეკებისათვის ოპერატიულ მეხსიერებას გამოყოფენ. შემდეგ, კავშირის მიმთითებლებისათვის ხდება მეხსიერების გამოყოფა და ხორციელდება შეერთება.

2. SQL ოპერატორის შესრულება. გამოიყოფა ოპერატორის მიმთითებელი, ლოკალური ცვლადები SQL-გამოსახულების სვეტებს უკავშირდება და გამოსახულება დამუშავებისათვის მთავარ ODBC-დრაივერს მიეწოდება.

3. მონაცემების ამოღება. ამოღების წინ ინფორმაცია შედეგობრივ ნაკრებზე ბრუნდება, კერძოდ სვეტების რიცხვი ნაკრებში. ამ რიცხვიდან გამომდინარე, შედეგობრივი ნაკრები ჩანაწერების ბუფერში თავსდება, მისი დათვალიერების ციკლი სრულდება და თითოეული სვეტის შემადგენლობა შესაბამის ლოკალურ ცვლადში თავსდება. ეს ნაბიჯი აუცილებელი არაა, თუ გამოიყენება სვეტების დაკავშირება ლოკალურ ცვლადებთან.

4. რესურსების გათავისუფლება. იმის შემდეგ, რაც მონაცემები მიღებულია, რესურსები ოპერატორის შეერთებისა და გარემოს მიმთითებლების გათავისუფლების ფუნქციის გამოძახების გზით თავისუფლდება. ოპერატორის მიმთითებლები და შეერთება შეიძლება დამუშავების პროცესში იქნეს გამოყენებული.

ODBC ტექნოლოგია, მონაცემთა წყაროებისგან დამოუკიდებელი, მუშავდებოდა როგორც საერთო მონაცემებთან წვდომის საშუალება. ტექნოლოგიის გამოყენებას ასევე დანართების გადატანის უნარი, სხვადასხვა მონაცემთა ბაზების გარემოში უნდა უზრუნველყო, თავად დანართების გადამუშავების საჭიროების გარეშე. ამ მხრივ, ODBC

ტექნოლოგია უკვე საწარმო სტანდარტი გახდა, მას თითქმის ყველა მბმს შემქმნელი და დამუშავების საშუალებანი უჭერს მხარს.

დანართის გამსხვილება (Upsizing)

პაკეტი Microsoft Access Upsizing Tools სამაგიდო მონაცემთა ბაზებიდან კლიენტ-სერვერულ არქიტექტურაზე გადასვლის პროცესის ავტომატიზების შესაძლებლობას იძლევა, ამ შემთხვევაში – Microsoft Access-იდან Microsoft SQL Server-ზე. ეს პროდუქტი Access ისეთ მოდულებს უმატებს, როგორცაა Upsizing Wizard და SQL Server Browser. Access მონაცემთა ბაზა (ფაილი გაფართოებით .MDB) მონაცემთა ცხრილებს და ინდექსებს, აგრეთვე მონაცემების მართვის კომპონენტებს (შეკითხვები, მაკროსები, მოდულები, ფორმები და ანგარიშები) შეიცავს.

Microsoft Access Upsizing Tools-ის გამოყენება მომხმარებლისათვის მონაცემების გამჭვირვალედ განთავსების კონცეფციას ეფუძნება: SQL Server-ის მონაცემთა ბაზაში ინფორმაციის გადატანას Access MDB დანარჩენი ნაწილის თავდაპირველი სახით დატოვების დროს. კლიენტ-სერვერულ არქიტექტურაში Access-ს კლიენტის სახით მუშაობა შეუძლია, ამიტომაც პროგრამისტები, რომლებსაც ამ ინსტრუმენტის საშუალებით გადააქვთ დანართები არ უნდა უარყონ Access-ი. მოთხოვნები, ანგარიშები და სხვა ამგვარი ობიექტები შეუცვლელი სახით რჩება, რადგანაც Upsizing Wizard მათ გადაამისამართებს ექსპორტირებად ცხრილებზე, ამავდროულად ლოკალურ ცხრილებს შეუცვლის სახელებს.

მოდულები Microsoft Wizard – არის ოპერაციების ავტომატიზაციისათვის ინსტრუმენტები, რომლებიც მომხმარებლის მოქმედებებს საჭირო მიმართულებით მიმართავს. Access Upsizing Wizard პრიმიტიულ პროდუქტს არ წარმოადგენს, რადგანაც მომხმარებელს ნორმალური მუშაობის უზრუნველსაყოფად შესაძლოა სისტემის კონფიგურაციის ხელით გაკეთება დასჭირდეს, მაგრამ მიუხედავად ამისა, Microsoft Access Upsizing Tools-ის შექმნა გამართლებული იქნება მათთვის, ვისთვისაც Access-იდან მონაცემების გადატანა SQL Server-ში სჭირდება.

ეს პროგრამა ამ პროცესის ბევრ მძიმე მომენტს უკეთებს ავტომატიზებას, ხოლო დოკუმენტაცია კლიენტ-სერვერულის ტიპის დანართებში კლიენტის სახით Microsoft Access გამოყენების ოპტიმიზაციისათვის სპეციალურ განყოფილებას შეიცავს.

კლიენტ-სერვერული დანართების დაპროექტების საშუალებები და მეთოდები. დამუშავება და თანმხლები დოკუმენტები

კლიენტ-სერვერული დანართების შექმნის ბაზური საშუალებები არქიტექტურაში “კლიენტ-სერვერი”

მონაცემთა ბაზების სისტემებისათვის არქიტექტურა “კლიენტი-სერვერი” უმთავრესად იმიტომაა საინტერესო და აქტუალური, რომ ლოკალურ ქსელში მონაცემთა ბაზებთან კოლექტიური წვდომის პრობლემის მარტივ და შედარებით იაფ გადაწყვეტას უზრუნველყოფს. არქიტექტურის “კლიენტი-სერვერი” რეალური გავრცელება განვითარების ღია სისტემების კონცეფციის პრაქტიკაში ფართოდ დანერგვის ხარჯზე გახდა შესაძლებელი.

წაშლის პროცედურების გამოძახება

სისტემის მობილურობის პრობლემის საერთო გადაწყვეტა, რომელიც “კლიენტი-სერვერი” არქიტექტურაზეა დაფუძნებული, პროგრამულ პაკეტებზე საყრდენს წარმოადგენს და RPC (Remote Procedure Call) პროცედურის დაშორებული გამოძახების პროტოკოლების რეალიზებას ახდენს. სერვისზე მიმართვის ასეთი საშუალებების გამოყენებისას დაშორებულ კვანძში გამოიყურება, როგორც ჩვეულებრივი გამოძახების პროცედურა. RPC საშუალებანი, რომლებშიც ჩვეულებრივ ლოკალური ქსელის აპარატურის სპეციფიკისა და ქსელურ პროტოკოლებზე მთელი ინფორმაცია ინახება, გამოძახება ქსელური ურთიერთქმედების მიმდევრობაში გადაყავს.

RPC პროგრამების დაცილებული პროცედურების გამოძახების დროს კლიენტის მონაცემების ფორმატის გარდაქმნას შუალედურ მანქანურ-დამოუკიდებელ ფორმატებში ახდენენ და შემდეგ სერვერის მონაცემების ფორმატებში გარდაქმნიან. საპასუხო პარამეტრების გადაცემისას ანალოგიური გარდაქმნები სრულდება.

დაშორებული პროცედურების გამოძახების მექანიზმის მთავარი იდეები, რომლებიც არქიტექტურის “კლიენტი-სერვერი” ტექნოლოგიურ საფუძველს წარმოადგენს, შემდეგია:

- ბევრ შემთხვევაში, პროცესების ურთიერთობა აშკარად გამოხატულ ასიმეტრიულ ხასიათს ატარებს. ერთ-ერთი პროცესი (“კლიენტი”) მეორე პროცესს (“სერვერი”) რაიმე სერვისის შესრულებას სთხოვს და თავის მუშაობას აღარ აგრძელებს მანამ, სანამ ეს სერვისი არ იქნება შესრულებული. ნათელია, რომ სემანტიკურად, ურთიერთქმედების ასეთი რეჟიმი გამოძახების პროცედურის ეკვივალენტურია.

- ოპერაციული სისტემა UNIX თავისი იდეალოგიით თავიდანვე ნამდვილი ქსელური ოპერატიული სისტემა იყო. გადატანის თვისება შესაძლებლობას იძლევა ძალზე მარტივად შეიქმნას “ერთგვაროვანი ოპერაციული” ქსელი, რომელიც სხვადასხვაგვარ კომპიუტერებს მოიცავს. მაგრამ სხვადასხვა არქიტექტურის კომპიუტერებში მონაცემთა სხვადასხვაგვარი წარმოდგენის პრობლემა რჩება. ცუდია, როდესაც მონაცემთა სხვადასხვაგვარად წარმოდგენის პრობლემის გადაჭრა მომხმარებელს ევალება. ამიტომაც, RPC-ის მეორე იდეაა პროცესების ზეგავლენით მონაცემთა ფორმატების ავტომატური გარდაქმნა, რომელიც სხვადასხვაგვარ კომპიუტერებზე სრულდება.

RPC პროტოკოლი და მისი რეალიზაცია

პირველად, RPC პაკეტი Sun Microsystems კომპანიის მიერ 1984 წ. მისი პროდუქტის NFS (Network File System – ქსელური ფაილური სისტემა) ჩარჩოებში იყო რეალიზებული. პაკეტი გულდასმით იმასთან იყო სპეციფიცირებული, რომ მომხმარებლის ინტერფეისი და მისი ფუნქციები გამოყენებული ტრანსპორტირების მექანიზმზე არ ყოფილიყო დამოკიდებული.

შევნიშნოთ, რომ დღეს Sun პაკეტის ორ ვარიანტს ავრცელებს – უფასოს (Public Domain), რომელიც პროგრამული ბუდეების გამოყენებაზეა დაფუძნებული, და კომერციულს, რომელიც ნაკადის მექანიზმზეა (ინტერფეისი TLI-ზე) ბაზირებული. ორივე შემთხვევაში, პაკეტი რეალიზდება როგორც ბიბლიოთეკური ფუნქციების ნაკრები.

პროტოკოლი XDR

კონკრეტული მანქანური წარმოდგენისაგან დამოუკიდებლობა ცალკე სპეციფირებული პროტოკოლით XDR (External Data Representation – მონაცემების გარე წარმოდგენა) არის უზრუნველყოფილი. ეს პროტოკოლი განსაზღვრავს მონაცემების წარმოდგენის სტანდარტულ მეთოდს, რომელიც ისეთ მანქანაზე დამოუკიდებულ თვისებებს ფარავს, როგორცაა: ბაიტების მიმდევრობა სიტყვაში, სტრუქტურის საწყისი მისამართის გათანაბრების მოთხოვნა, მონაცემების სტანდარტული ტიპების წარმოდგენა და ა.შ. არსებითად, XDR რეალიზდება როგორც დამოუკიდებელი პაკეტი, რომელიც არა მხოლოდ RPC-ში, არამედ სხვა პროდუქტებშიც გამოიყენება.

TCP/IP პროტოკოლების სტეკი, როგორც RPC-ის საფუძველი

TCP/IP (Transmission Control Protocol/Internet Protocol) იმ პროტოკოლების ოჯახს წარმოადგენს, რომელთა მთავარი დანიშნულება სხვადასხვა ტექნოლოგიებზე დაფუძნებული კომპიუტერული ქსელების ურთიერთთან სასარგებლო არსებობის შესაძლებლობის უზრუნველყოფაა.

1969 წ. აშშ-ს თავდაცვის სამინისტროს პერსპექტიული კვლევის პროექტების სააგენტომ (DARPA – Department of Defense Advanced Research Project Agency) მხარი დაუჭირა და დააფინანსა პროექტი, რომელიც სხვადასხვა ტექნოლოგიების ქსელების კავშირის ზოგადი საფუძვლების ძიებას ეძღვნებოდა. ამ პროექტის შესრულების შედეგად, ერთიანი ვირტუალური ქსელი იყო მიღებული, რომელმაც Internet-ის სახელწოდება მიიღო.

Internet-ში დამოუკიდებელი ქსელების (ანუ დომენების) დასაკავშირებლად გამოიყენება არხის ჩამკეტების ერთობლიობა. ყოველი ინდივიდუალური ქსელის კვანძის იდენტიფიცირება ხდება უნიკალური მისამართით, რომელსაც Internet-ში მისამართი ჰქვია.

კადრების ფორმატებში განსხვავების პრობლემის გადასაჭრელად მონაცემთა პაკეტის უნივერსალური ფორმატი, სახელად IP-დეიტაგრამა (Internet Protocol Datagram) იყო განსაზღვრული და რომელიც სათაურისა და მონაცემების ულუფისაგან შედგება; ამიტომაც, იგი ჩვეულებრივი ქსელური კადრის მსგავსია.

კვანძები, რომლებიც Internet-ში ურთიერთობენ, ინდივიდუალური ზემოქმედების მიზნით ერთმანეთში ფიზიკურ შეერთებას არ განიცდიან და ამიტომაც, დეიტაგრამების დამუშავება რაღაც განსაზღვრული წესით არ ხდება. პირიქით, თითოეული დეიტაგრამა სხვებისაგან დამოუკიდებლად მუშავდება, რაც მთელი ლოგიკურად დაკავშირებული კვანძებისათვის რესურსების ეფექტურად განაწილების საშუალებას იძლევა. მაგრამ ეს, სამწუხაროდ ნიშნავს, რომ Internet-ის მიერ წარმოდგენილი სერვისი, უსაფრთხო არაა, რადგანაც პაკეტების საჭირო რიგით მიწოდების, დეიტაგრამების არდაკარგვის ან მათი დუბლირების არარსებობის გარანტიას არ იძლევა.

ამ პრობლემას გადაჭრის TCP პროტოკოლი, რომელიც შეტყობინებების უსაფრთხო მიწოდებას, უზრუნველყოფს დეიტაგრამების მიწოდების და საჭიროების შემთხვევაში, მათი მეორედ გადაცემის დადასტურების ხარჯზე.

ცხრილი 3

პროტოკოლის დასახელება	პროტოკოლის აღწერა
TCP	(Transmission Control Protocol) გადაცემის მართვის პროტოკოლი
UDP	(User Datagram Protocol) მომხმარებელთა დეიტაგრამების პროტოკოლი
ARD	(Address Resalution Protocol) მისამართების ნებართვის მიცემის პროტოკოლი
RARD	(Reverse Address Resalution Protocol) მისამართის უკუნებართვის გაცემის პროტოკოლი
IP	(Internet Protocol) Internet პროტოკოლი
ICMP	(Internet Control Message Protocol) Internet შეტყობინების მართვის პროტოკოლი
FTP	(File Transfer Protocol) ფაილთა გადაგზავნის პროტოკოლი
TFTD	(Trivial File Transfer Protocol) ფაილთა გადაგზავნის მარტივი პროტოკოლი

IP-დეიტაგრამების გამოყენებაზე დაფუძნებულ პროტოკოლების მთელ ოჯახს, TCP/IP პროტოკოლი ჰქვია. ამ ოჯახის ყველაზე საბაზისო და მნიშვნელოვანი პროტოკოლებია IP და TCP (ყველა ეს პროტოკოლი მე-3

ცხრილშია მოცემული). OC UNIX კომუნიკაციური საშუალებების უმეტესი ნაწილი TCP/IP სტეკის პროტოკოლების გამოყენებაზეა დაფუძნებული.

დღესდღეობით, UNIX System V Release 4 რეალიზაციის ყველაზე სტანდარტული პროტოკოლია TCP/IP, რომელიც როგორც მოდულების სპეციალიზებული ნაკადების ნაკრები პლუს დამატებითი კომპონენტი TLI (Transport Level Interface) არის რეალიზებული. TLI გამოყენებით პროგრამასა და სატრანსპორტო მექანიზმს შორის ინტერფეისს წარმოადგენს. დანართს, რომელიც TLI ინტერფეისს იყენებს, შესაძლებლობა ეძლევა გამოიყენოს TCP/IP.

მონაცემთა ბაზების სერვერები

ტერმინი „მონაცემთა ბაზის სერვერი“ ჩვეულებრივ, მთელი იმ მზმს-ის აღსანიშნავად გამოიყენება, რომელიც არქიტექტურაზე „კლიენტი-სერვერი“ არის დაფუძნებული და რომელიც როგორც სერვერის, ისე კლიენტის ნაწილს მოიცავს. ასეთი სისტემები მონაცემთა ბაზების შენახვისა და წვდომის უზრუნველსაყოფად არის განკუთვნილი.

თუმცა, ჩვეულებრივ, მონაცემთა ერთი ბაზა მთლიანად ქსელის ერთ კვანძში ინახება და ერთი სერვერი უჭერს მხარს. მონაცემთა ბაზის სერვერი განაწილებულ მონაცემთა ბაზებთან მარტივ და იაფ მიახლოებას წარმოადგენს, ვინაიდან მონაცემთა საერთო ბაზა ლოკალური ქსელის ყველა მომხმარებლისათვისაა მისაწვდომი.

მონაცემთა ბაზის სერვერის ცნება

გამოყენებითი პროგრამიდან ან მომხმარებლის მონაცემთა ბაზებთან წვდომა სისტემის კლიენტური ნაწილთან მიმართვით ხორციელდება. კლიენტურ და სერვერულ ნაწილებს შორის ძირითად ინტერფეისად, მონაცემთა ბაზების SQL ენა გამოდის.

მონაცემთა ბაზების სერვერებს, რომელთა ინტერფეისი SQL ენაზეა სრულად დაფუძნებული, თავისი უპირატესობა და ნაკლიც აქვს. აშკარა უპირატესობაა ინტერფეისის სტანდარტულობა. ზღვარში, რომელიც მთლიანად მიღწევადი არაა, ნებისმიერ, SQL-ორიენტირებული მზმს-ის კლიენტურ ნაწილს ნებისმიერ SQL-სერვერთან შეეძლო ემუშავა, მიუხედავად იმისა, თუ ვინ შექმნა ის.

ნაკლიც საკმაოდ აშკარაა. სისტემის კლიენტურ და სერვერულ ნაწილებს შორის ასეთი მაღალი დონის ინტერფეისის არსებობისას კლიენტის მხარეს ძალიან ცოტა მბმს-ის პროგრამები მუშაობს. ეს ნორმალურიცაა, თუ კლიენტის მხარეს მცირეგაბარიტიანი სამუშაო სადგური გამოიყენება. მაგრამ, თუ კლიენტის კომპიუტერი საკმაოდ დიდი სიმძლავრისაა, მაშინ ხშირად ჩნდება სურვილი მას მონაცემთა ბაზების მართვის უფრო მეტი ფუნქციები დავაკისროთ, ამით განვტვირთავთ სერვერს, რომელიც მთელი სისტემის „ვიწრო“ ნაწილს წარმოადგენს.

მონაცემთა ბაზის სერვერის ბაზური არქიტექტურა

მონაცემთა ბაზების ტიპური სერვერი შემდეგი ფუნქციების შესრულებაზე აგებს პასუხს:

- ლოგიკურად შეთანხმებული ფაილების ნაკრების მხარდაჭერა;
- მონაცემებით მანიპულირების ენის უზრუნველყოფა;
- ინფორმაციის სხვადასხვა სახის ავარიული შეცდომების შემდეგ აღდგენა;
- რამდენიმე მომხმარებლის რეალურად პარალელური მუშაობის ორგანიზება.

მონაცემთა ბაზების ენები

მონაცემთა ბაზებთან სამუშაოდ სპეციალური ენები გამოიყენება, რომლებსაც ერთობლივად, მონაცემთა ბაზების ენებს უწოდებენ. ყველაზე ხშირად, ორი ენა გამოიყოფა – მონაცემთა ბაზის სქემის განმსაზღვრელი SDL (Schema Definition Language) და მონაცემებით მანიპულირების DML (Data Manipulation Language). SDL უმთავრესად, მონაცემთა ბაზის ლოგიკური სტრუქტურის, ე.ი. მონაცემთა ბაზის იმ სტრუქტურის განსაზღვრას ემსახურებოდა, რომელსაც მომხმარებლები წარმოადგენს. DML მონაცემების მანიპულირების ოპერატორებს მოიცავდა, ე.ი. ოპერატორებს, რომლებიც მონაცემთა ბაზაში მონაცემების შეტანის, წაშლის, მოდიფიცირების ან არსებული მონაცემების არჩევის საშუალებას იძლეოდა.

თანამედროვე მბმს-ში ჩვეულებრივ გამოიყენება ერთიანი ინტეგრირებული ენა, რომელიც მონაცემთა ბაზებთან მუშაობის ყველა საჭირო საშუალება მოიცავს, მისი შექმნიდან დაწყებული და მონაცემთა ბაზებთან

სამომხმარებლო საბაზო ინტერფეისის უზრუნველყოფით. დღესდღეობით, ყველაზე გავრცელებულ რელაციურ მბმს-ის სტანდარტულ ენას SQL (Structured ან Standard Query Language) ენა წარმოადგენს.

უპირველესად, SQL ენა SDL და DML საშუალებებს აერთიანებს, ე.ი. რელაციურ მონაცემთა ბაზის სქემის განსაზღვრის და მონაცემებით მანიპულირების საშუალებას იძლევა. ამასთან, მონაცემთა ბაზის ობიექტების სახელდება ენობრივ დონეზე იმ სახით ხდება, რომ SQL ენის კომპილატორი ობიექტების სახელებს გარდაქმნის თავის შინაგან იდენტიფიკატორებზე სპეციალურად არსებული მომსახურე ცხრილ-კატალოგების საფუძველზე.

SQL ენა მონაცემთა ბაზის შეზღუდული მთლიანობის დასადგენად, სპეციალურ საშუალებებს შეიცავს. სპეციალურ ოპერატორებს შეუძლია განსაზღვროს ე. წ. მონაცემთა ბაზების წარმოდგენა, რომელიც ფაქტობრივად მონაცემთა ბაზაში შენახულ მოთხოვნებს წარმოადგენს. დაბოლოს, მონაცემთა ბაზების ობიექტებთან წვდომის ავტორიზაცია, SQL სპეციალური ოპერატორების ნაკრების საფუძველზე წარმოებს.

SQL ენა - SQL-სერვერის საბაზო ინტერფეისი

მონაცემთა ბაზების ორგანიზაციისადმი რელაციური მიდგომის ძირითადი უპირატესობა ისაა, რომ მომხმარებელს საშუალება ეძლევა ეფექტურად იმუშაოს ცხრილებთან და ამასთან, კომპიუტერის გარე მესხიერებაში მონაცემთა რეალური ორგანიზაციის ცოდნა არ მოეთხოვება.

მომხმარებელს შეუძლია არანავიგაციური მანერით იმუშაოს, ანუ მონაცემთა ბაზიდან ინფორმაციის ამოსარჩევად მიუთითოს მხოლოდ მისთვის საინტერესო ცხრილები და ის პირობები, რომელთაც ამორჩეული მონაცემები უნდა აკმაყოფილებდეს. რელაციური სისტემების განსაკუთრებით მნიშვნელოვან თავისებურებას წარმოადგენს ის, რომ მონაცემთა ბაზის ცხრილებისადმი ნებისმიერ კითხვაზე პასუხი ისევ ცხრილია, რომელიც შეიძლება მონაცემთა ბაზაში იქნეს შენახული და ან რომლის მიმართაც შეიძლება ახალი შეკითხვა დაისვას.

რელაციური მბმს-ის მიმართ საბაზო მოთხოვნაა მძლავრი და ამასთან, იმ მარტივი ენის არსებობა, რომელიც მომხმარებლის ნებისმიერ აუცილებელ ოპერაციას შეასრულებს. ბოლო წლების განმავლობაში ასეთ ენად

მიღებულია მონაცემთა ბაზების რელაციური ენა SQL – Structured Query Language (ხშირად მას Standard Query Language-ად მოიხსენიებენ).

მონაცემთა ბაზებთან ურთიერთქმედებისათვის SQL ენა XX საუკუნის 70-იანი წლების შუა პერიოდში გამოჩნდა და ექსპერიმენტული რელაციური მბმს-ის System R პროექტის ჩარჩოებში იყო დამუშავებული. SQL ენა მომხმარებლის მიერ რელაციურ მონაცემთა ბაზების მიმართ მოხერხებული და გასაგები შეკითხვის ფორმულირებაზე იყო ორიენტირებული. მაგრამ, სინამდვილეში ის იყო მონაცემთა ბაზის სრული ენა, რომელიც შეკითხვების ფორმულირებისა და მონაცემთა ბაზის მანიპულირების გარდა, შეიცავდა მონაცემთა ბაზების სქემების განსაზღვრებებსა და მანიპულირებას; მთლიანობისა და ტრიგერების განსაზღვრებებს; მონაცემთა ბაზების წარმოდგენებს და სხვ.

ამჟამად, SQL ენა პრაქტიკულად ყველა კომერციულ რელაციურ მბმს-შია რეალიზებული, ყველა ფირმა ცდილობს დაამოწმოს, რომ მისი სისტემები SQL სტანდარტს შეესაბამება.

მოდულთა ენა, თუ ჩაშენებული SQL?

SQL სტანდარტში დაპროგრამების ტრადიციულ ენაზე დაწერილი გამოყენებითი პროგრამიდან მონაცემთა ბაზებთან ურთიერთობის ორი ხერხია განსაზღვრული.

პირველი ხერხი მდგომარეობს იმაში, რომ SQL ყველა ოპერატორი, რომელთანაც მოცემულ გამოყენებით პროგრამას შეუძლია მუშაობა, ერთ მოდულშია შეკრებილი და გაფორმებულია როგორც ამ მოდულის პროცედურები. ამისათვის, SQL-ში სპეციალური ქვეენა – მოდულთა ენა არსებობს. მონაცემთა ბაზასთან ურთიერთობის ასეთი ხერხის დროს, გამოყენებითი პროგრამა SQL მოდულის პროცედურების გამოძახებებს შეიცავს მათთვის ფაქტობრივი პარამეტრების გადაცემით და საპასუხო პარამეტრების მიღებით.

მეორე ხერხი, ე.წ. ჩაშენებული SQL-ის გამოყენებაში მდგომარეობს, როდესაც სპეციალური სინტაქსის მეშვეობით დაპროგრამების ტრადიციულ ენაზე დაწერილ პროგრამაში SQL ოპერატორები ჩაშენდება. ამ შემთხვევაში, გამოყენებითი პროგრამის თვალსაზრისით SQL ოპერატორი „ადგილის მიხედვით“ სრულდება. SQL ოპერატორების აშკარა პარამეტრიზაცია არ არსებობს, მაგრამ SQL ჩაშენებულ ოპერატორებში შესაძლოა ძირითადი

პროგრამის ცვლადების სახელები იქნეს გამოყენებული და კავშირის დამყარება გამოყენებით პროგრამასა და მბმს-ს შორის ამის ხარჯზე ხდება.

კონცეპტუალურად, ეს ორი ხერხი ეკვივალენტურია. უფრო მეტიც, სტანდარტში ჩაშენებული SQL პროგრამით SQL არაცხადი მოდულის წარმოშობის წესები დგინდება. მაგრამ უმეტეს რეალიზაციებში SQL მოდულში არსებული ჩაშენებული SQL ოპერატორები სრულიად სხვადასხვაგვარად მუშავდება. SQL მოდულის კომპილაცია ჩვეულებრივ გამოყენებითი პროგრამიდან ცალკე ხდება, რის შედეგადაც ე. წ. შენახული პროცედურების ნაკრები ჩნდება, ე.ი. SQL მოდულის გამოყენების შემთხვევაში SQL ოპერატორების კომპილაცია ერთხელ ხდება და შემდეგ შესაბამისი პროცედურების გამოყენებითი პროგრამიდან გამოძახება მრავალჯერაა შესაძლებელი.

ამისგან განსხვავებით, გამოყენებით პროგრამაში ჩაშენებული SQL ოპერატორებისათვის, კომპილაცია ყოველთვის ჩვეულებრივ, მათი გამოყენების დროს ხდება.

რა თქმა უნდა, მომხმარებელი არ არის ვალდებული მბმს-თან მუშაობის ხერხების ტექნიკურ განსხვავებას იცნობდეს. არსებობს ისეთი სისტემებიც, რომლებიც SQL ჩაშენებული ოპერატორების ერთჯერად კომპილაციას აწარმოებს და კომპილირებულ კოდს ინახავს.

რელაციურ მონაცემთა ბაზების დაპროექტების კლასიკური მიდგომა

რელაციურ მონაცემთა ბაზების დაპროექტებისას ორი უმთავრესი პრობლემის გადაწყვეტა ხდება:

1. როგორ გამოვსახოთ საგნობრივი სფეროს ობიექტები მონაცემთა მოდელის აბსტრაქტულ ობიექტებში ისე, რომ ეს ასახვა საგნის გარემოს სემანტიკასთან წინააღმდეგობაში არ მოდიოდეს და შეძლებისდაგვარად უკეთესი იყოს? ხშირად ამ პრობლემას, მონაცემთა ბაზების ლოგიკური დაპროექტების პრობლემას უწოდებენ.

2. როგორ უზრუნველვყოთ მონაცემთა ბაზისადმი შეკითხვების ეფექტური შესრულება, ე.ი. მხედველობაში რომ გვქონდეს კონკრეტული მბმს თავისებურებები, გარე მეხსიერებაში მონაცემთა განთავსება, როგორი

დამატებითი სტრუქტურის შექმნა მოვითხოვთ და ა.შ. ამ პრობლემას, მონაცემთა ბაზების ფიზიკური დაპროექტების პრობლემას უწოდებენ.

რელაციურ მონაცემთა ბაზების შემთხვევაში, ფიზიკური დაპროექტების ნაწილში რაღაც საერთო რეცეპტების წარმოდგენა რთულია. აქ ძალიან ბევრი რამ, გამოყენებულ მბმს-ზეა დამოკიდებული.

აგრეთვე, ძალიან მნიშვნელოვანი ასპექტია ორგანული მთლიანობის განსაზღვრა. ორგანული მთლიანობის დასადგენად ძნელია გამოიყენო რაღაც საერთო მიდგომა. ამ შეზღუდვებს ზოგადი სახე აქვს და მისი ფორმულირება უფრო ხელოვნების სფეროს განეკუთვნება, ვიდრე საინჟინრო ოსტატობას.

ჩავთვალოთ, რომ რელაციურ მონაცემთა ბაზის დაპროექტების კლასიკური პრობლემა იმის დასაბუთებულ გადაწყვეტაში მდგომარეობს, თუ როგორი ურთიერთდამოკიდებულებისაგან უნდა შედგებოდეს მონაცემთა ბაზა და რა ატრიბუტები უნდა ჰქონდეს ამ დამოკიდებულებებს.

ფუნქციური და სხვა დამოკიდებულებები

დაპროექტების პროცესი წარმოადგენს დამოკიდებულებების სქემების ნორმალიზაციის პროცესს, სადაც თითოეულ მომდევნო ნორმალურ ფორმას უკეთესი თვისებები აქვს, ვიდრე წინას.

თითოეულ ნორმალურ ფორმას ზოგიერთი განსაზღვრული შეზღუდვების ნაკრები შეესაბამება და დამოკიდებულება რომელიმე ნორმალურ ფორმაშია, თუ მისთვის დამახასიათებელი შეზღუდვების ნაკრებს აკმაყოფილებს. შეზღუდვების ნაკრების მაგალითს, პირველი ნორმალური ფორმის შეზღუდვა წარმოადგენს – დამოკიდებულების ყველა ატრიბუტის მნიშვნელობა ატომურია. რადგანაც პირველი ნორმალური ფორმის მოთხოვნა მონაცემთა კლასიკური რელაციური მოდელის საბაზო მოთხოვნაა, შეიძლება ჩაითვალოს, რომ დამოკიდებულების საბოლოო ნაკრები ამ მოთხოვნას უკვე შეესაბამება.

რელაციურ მონაცემთა ბაზების თეორიაში ჩვეულებრივ ნორმალური ფორმების შემდეგ მიმდევრობას გამოყოფენ: პირველი ნორმალური ფორმა (1NF); მეორე ნორმალური ფორმა (2NF); მესამე ნორმალური ფორმა (3NF), ბოის-კოდის ნორმალური ფორმა (BCNF); მეოთხე ნორმალური ფორმა (4NF); მეხუთე ნორმალური ფორმა, ან პროექცია-შეერთების ნორმალური ფორმა (5NF ან PJ/NF).

ნორმალური ფორმების ძირითადი თვისებებია:

- ყოველი მომდევნო ნორმალური ფორმა რაღაც აზრით, წინაზე უკეთესია;

- მომდევნო ნორმალურ ფორმაზე გადასვლისას წინა ნორმალური თვისებები შენარჩუნებულია.

დაპროექტების პროცესის საფუძველში დევს ნორმალიზაციის მეთოდი, დამოკიდებულებების დეკომპოზიცია, წინა ნორმალურ ფორმაში მყოფი ორი ან უფრო მეტი დამოკიდებულებები, რომლებიც მომდევნო ნორმალური ფორმის მოთხოვნებს აკმაყოფილებს.

პრაქტიკაში განსაკუთრებით მნიშვნელოვანი ნორმალური ფორმების დამოკიდებულებები რელაციური მონაცემთა ბაზების თეორიაში ფუნქციური დამოკიდებულებების ცნებას ეფუძნება.

მონაცემთა ბაზების კონცეპტუალური მოდელები და სქემები

რელაციური მბმს-ის ფართო გავრცელება და სხვადასხვა დანართებში მათი გამოყენება აჩვენებს, რომ მონაცემთა რელაციური მოდელი საგანთა სფეროს მოდელირებისათვისაა საკმარისი. მაგრამ დამოკიდებულებათა ტერმინებში რელაციურ მონაცემთა ბაზების დაპროექტება ხშირად, საკმაოდ რთული და არაკომფორტული პროცესია.

ამასთან, მონაცემების რელაციური მოდელის შეზღუდვა შემდეგ ასპექტებში ჩნდება:

- მოდელი მონაცემების აზრის საჩვენებლად საკმარის საშუალებებს არ იძლევა. რეალური საგნობრივი სფეროს სემანტიკა, დამპროექტებლის გონებაში წარმოდგენილი მოდელის საშუალებებისაგან დამოუკიდებელი უნდა იყოს. კერძოდ, ეს შეზღუდვათა მთლიანობის წარმოდგენის პრობლემას ეხება;

- ბევრი დანართისათვის რთულია ბრტყელი ცხრილების საფუძველზე საგნობრივი გარემოს მოდელირება. რიგ შემთხვევებში, დაპროექტების პირველივე სტადიაზე, ერთი ცხრილის სახით საგნობრივი გარემოს აღწერისათვის დაპროექტების ავტორს დიდი ძალისხმევა ესაჭიროება;

- მიუხედავად იმისა, რომ დაპროექტების მთელი პროცესი დამოკიდებულებების აღრიცხვის საფუძველზე მიმდინარეობს, რელაციური მოდელი ამ დამოკიდებულებების აღწერისათვის რაიმე საშუალებებს არ იძლევა. მონაცემების რელაციურ მოდელს არსისა და კავშირების გასაყოფად

რაიმე აპარატი არა აქვს, ამის მიუხედავად, დაპროექტების პროცესი დანართისათვის ზოგიერთი არსებითი საგნობრივი გარემოს ობიექტების („არსი“) გამოყოფით და ამ არსებს შორის კავშირის გამოვლენით იწყება.

მონაცემთა ბაზების დაპროექტების ავტორის მოთხოვნებმა საგნობრივი გარემოს მოდელირების უფრო მოხერხებულ და მძლავრ საშუალებებზე მონაცემების სემანტიკური მოდელის მიმართულება წარმოშვა. ამასთან, ნებისმიერ მონაცემთა განვითარებული სემანტიკური მოდელი, ისევე როგორც რელაციური მოდელი, სტრუქტურულ, მანიპულაციურ და მთელ ნაწილს მოიცავს. სემანტიკური მოდელის მთავარი დანიშნულებაა მონაცემების სემანტიკის გამოხატვის უზრუნველყოფა.

ყველაზე ხშირად, პრაქტიკაში მონაცემთა ბაზების დაპროექტების პირველ სტადიაზე სემანტიკური მოდელირება გამოიყენება. ამავე დროს, სემანტიკური მოდელის ტერმინებში იქმნება მონაცემთა ბაზების კონცეპტუალური სქემა, რომელიც შემდგომში რელაციურ სქემად ხელით გადაკეთდება. ეს პროცესი იმ მეთოდის გამოყენებით სრულდება, რომელშიც ამ გარდაქმნის ყველა ეტაპები საკმაოდ მკაფიოდაა აღწერილი.

კონცეპტუალური სქემის რეალიზება რელაციურ მონაცემთა ბაზაში ავტომატიზებული კომპილაციით უფრო ნაკლები სიხშირით ხდება. ამასთან, ორი მიდგომაა ცნობილი: კომპილატორისათვის, როგორც საწყისი ინფორმაციის კონცეპტუალური სქემის ცხადი წარმოდგენის საფუძველზე და დაპროექტების ინტეგრირებული სისტემების აგება კონცეპტუალური სქემის ავტომატიზებული შექმნით, საგნობრივი გარემოს ექსპერტების ინტერვიუს საფუძველზე. როგორც ერთ, ისე მეორე შემთხვევაში შედეგად მესამე ნორმალურ ფორმაში მონაცემთა ბაზების რელაციური სქემა წარმოიქმნება.

ბოლოს, მესამე შესაძლებლობა. ესაა სემანტურ მოდელში მონაცემთა ბაზებთან მუშაობა, ანუ მბმს, რომლებიც მონაცემების სემანტურ მოდელებს ეფუძნება. ამავე დროს, ისევ განიხილება ორი ვარიანტი: სემანტიკური მოდელის მონაცემების საფუძველზე რელაციურ მონაცემთა მოდელში კონსტრუქციის ავტომატური ასახვით სამომხმარებლო ინტერფეისის უზრუნველყოფა და მბმს პირდაპირი რეალიზაცია, ნებისმიერ სემანტიკურ მონაცემთა მოდელზე დაფუძნებული.

საინფორმაციო სისტემების დაპროექტების CASE-სისტემები

თანამედროვე CASE-საშუალებები საინფორმაციო სისტემების დაპროექტების მრავალრიცხოვანი ტექნოლოგიის ფართო სპექტრს მოიცავს: უბრალო ანალიზის საშუალებიდან და დოკუმენტირებიდან ავტომატიზაციის ფართომასშტაბიან საშუალებამდე.

დამუშავების ყველაზე შრომატევადი ეტაპებია ანალიზისა და დაპროექტების ეტაპები. ამ დროს CASE-საშუალებები მიღებულ ტექნიკურ გადაწყვეტილებათა და პროექტის დოკუმენტაციის მომზადების ხარისხს უზრუნველყოფს. ამასთან, დიდ როლს ასრულებს ინფორმაციის ვიზუალური წარმოდგენის მეთოდები. ეს გულისხმობს სტრუქტურების ან სხვა დიაგრამების დროის რეალურ მასშტაბში აგებას, მრავალნაირ ფერთა პალიტრის გამოყენებას, სინტაქსური წესების შემოწმებას. საგნობრივი გარემოს მოდელირების გრაფიკული საშუალება დამმუშავებლებს ეხმარება არსებული საინფორმაციო სისტემის თვალსაჩინოდ შესწავლისას, მათი დასახული მიზნებისა და არსებული შეზღუდვების გათვალისწინებით გადაკეთებისას.

CASE-სისტემების დანიშნულება და სახესხვაობა

CASE-საშუალებებს მიაკუთვნებენ ნებისმიერ პროგრამულ საშუალებას, რომელიც პროგრამული უზრუნველყოფის სასიცოცხლო ციკლის ავტომატიზებას ახდენს და შემდეგი ძირითადი თავისებურებანი ახასიათებს:

- საინფორმაციო სისტემების აღწერისა და დოკუმენტირებისათვის მძლავრი გრაფიკული საშუალებანი;
- CASE-საშუალებების ცალკეული კომპონენტების ინტეგრაცია, რომელიც საინფორმაციო სისტემების დამუშავების პროცესების მართვას უზრუნველყოფს;
- საპროექტო მეტამონაცემების სპეციალურად ორგანიზებული საცავის (რეპოზიტორი) გამოყენება.

ინტეგრირებული CASE-საშუალება შემდეგ კომპონენტებს მოიცავს:

- რეპოზიტორი, CASE-საშუალების საფუძველია. მან უნდა უზრუნველყოს პროექტისა და მისი ცალკეული კომპონენტების შენახვა, ჯგუფური დაპროექტების შემთხვევაში ცალკეული დამმუშავებლისაგან შემოსული ინფორმაციის სინქრონიზაცია, სრული და ურთიერთსაწინააღმდეგო მეტამონაცემების კონტროლი;

- ანალიზისა და დაპროექტების გრაფიკული საშუალებანი, რომლებიც იერარქიულად დაკავშირებული დიაგრამების (DFD, ER-დიაგრამა და სხვ.) შექმნასა და რედაქტირებას უზრუნველყოფს;

- დანართების დამუშავების საშუალებანი, რომლებიც აგრეთვე მოიცავს 4GL ენებს და კოდის გენერატორებს;

- კონფიგურაციული მართვის საშუალებანი;
- დოკუმენტირების საშუალებანი;
- ტესტირების საშუალებანი;
- პროექტის მართვის საშუალებანი;
- რეინჟინერინგის საშუალებანი.

ყველა თანამედროვე CASE-საშუალების კლასიფიკაცია, ტიპებისა და კატეგორიების მიხედვით უნდა მოხდეს. ტიპების მიხედვით კლასიფიკაცია, სასიცოცხლო ციკლის პროცესების დროს CASE-საშუალებების ფუნქციურ ორიენტაციას ასახავს. კატეგორიების მიხედვით კლასიფიკაცია ინტეგრირების დონეს შესრულებული ფუნქციების მიხედვით განსაზღვრავს და მცირე ავტონომიური ამოცანების (tools) გადასაჭრელად ცალკეულ ლოკალურ საშუალებებს მოიცავს, ნაწილობრივ ინტეგრირებული საშუალებების ნაკრები, რომელიც საინფორმაციო სისტემის სასიცოცხლო ციკლის უმრავლეს ეტაპს (toolkit) მოიცავს და მთლიანად ინტეგრირებული საშუალებები, რომლებიც საინფორმაციო სისტემის მთელ სასიცოცხლო ციკლს უჭერს მხარს და საერთო რეპოზიტორითაა დაკავშირებული. ამის გარდა, CASE-საშუალებების კლასიფიკაცია შემდეგი თვისებების მიხედვითაცაა შესაძლებელი:

- გამოყენებული მეთოდოლოგიების, სისტემებისა და მონაცემთა ბაზის მოდელების მიხედვით;

- მბმს-თან ინტეგრირების დონის მიხედვით;
- მიწვდომილი პლატფორმის მიხედვით.

ტიპების მიხედვით კლასიფიკაცია ძირითადად CASE-საშუალებების კომპონენტურ შემადგენლობას ემთხვევა და შემდეგ ძირითად ტიპებს მოიცავს:

- ანალიზის საშუალებანი (Upper CASE), მოდელების საგნობრივი გარემოს ასაგებად და ანალიზისათვის განკუთვნილი (Design/IDEF (Meta Software), BPwin (Logic Works));

- ანალიზისა და დაპროექტების საშუალებანი (Middle CASE), საპროექტო სპეციფიკაციების შესაქმნელი დაპროექტებისა და გამოყენების მხარდამჭერი ყველაზე გავრცელებული მეთოდოლოგიები (Vantage Team Builder (Cayenne), Designer/2000 (ORACLE), Silverrun (CSA), PRO-IV (McDonnell Douglas), CASE.-ანალიტიკოსი (მაკროპროექტი)). ასეთი საშუალებების გამოსავალია სისტემების კომპონენტების სპეციფიკაცია და ინტერფეისები, სისტემების არქიტექტურა, მონაცემთა ალგორითმები და სტრუქტურები;

- მონაცემთა ბაზების დაპროექტების საშუალებანი, რომლებიც მონაცემთა მოდელირებასა და მონაცემთა ბაზების სქემების გენერაციას უზრუნველყოფს. მათ განეკუთვნება ERwin (Logic Works), S-Designor (SDP) და DataBase Designer (ORACLE). მონაცემთა ბაზების დაპროექტების საშუალება აგრეთვე CASE-საშუალებების შემადგენლობაში Vantage Team Builder, PRO-IV, Silverrun-სა და Designer/2000 არის;

- დანართების დამუშავების საშუალებს მიეკუთვნება 4GL (Uniface (Compuware), JAM (JYACC), PowerBuilder (Sybase), Developer/2000 (ORACLE), New Era (Informix), SQL Windows (Gupla), Delpni (Borland) და სხვ.) და კოდების გენერატორები, რომლებიც Vantage Team Builder შემადგენლობაში, PRO-IV და ნაწილობრივ Silverrun-ში შედის.

Silverrun

CASE-საშუალება Silverrun (ამერიკული ფირმა computer ayatems Advisers Inc. (CSA)) ბიზნეს-კლასის საინფორმაციო სისტემების ანალიზისა და დაპროექტებისათვის გამოიყენება და ძირითადად, სასიცოცხლო ციკლის სპირალურ მოდელზეა ორიენტირებული; იმ ნებისმიერი მეთოდოლოგიის მხარდასაჭერად გამოიყენება, რომელიც ფუნქციურ და ინფორმაციულ მოდელთა განცალკევებულ შექმნას (მონაცემთა ნაკადის დიაგრამები და დიაგრამები „არსი-კავშირი“) ეფუძნება.

კონკრეტულ მეთოდოლოგიაზე დაფუძნება მოდელების საჭირო გრაფიკული ნოტაციების და პროექტის თავისებურებების შემოწმების წესების ერთობლიობის ამორჩევითაა უზრუნველყოფილი. სისტემაში ყველაზე გავრცელებული მეთოდოლოგიებისათვის მზა ელემენტები არსებობს: DATARUN, Gane/Sarson, Yourdon/DeMarco, Merise, Ward/Mellor, Information Engineering. თითოეულ ცნებას, რომელიც პროექტებში შედის, საკუთარი აღწერილობის დამატების შესაძლებლობა აქვს. Silverrun არქიტექტურა აუცილებლობის შემთხვევაში დამუშავების გარემოს გაფართოების შესაძლებლობას იძლევა.

Silverrun-ს მოდულური სტრუქტურა აქვს და შედგება ოთხი მოდული-საგან, რომელთაგან თითოეული დამოუკიდებელ პროდუქტს წარმოადგენს, მისი შექმნა და გამოყენება სხვა მოდულების გარეშეც შესაძლებელია.

მონაცემების ნაკადის ბიზნეს-პროცესების მოდელების შექმნის მოდული (BPM – Business Process Modeler) საკვლევი ორგანიზაციის ფუნქციონირების ან დასამუშავებელი საინფორმაციო სისტემის დიაგრამის ფორმით მოდელირების შესაძლებლობას იძლევა. BPM მოდულში დიდი სირთულის მოდელებთან მუშაობის შესაძლებლობაა უზრუნველყოფილი: ავტომატური გადანომვრა, პროცესების ხესთან მუშაობა, მოდელების ნაწილების კოლექტიური დამუშავებისათვის განცალკევება და მიერთება. შესაძლოა დიაგრამები სხვადასხვა წინასწარგანსაზღვრული ნოტაციებით, Yourdon/DeMarco და GANE/Sarson-ის ჩათვლით იყოს გამოსახული. ასევე არსებობს საკუთარი ნოტაციების შექმნის შესაძლებლობა.

ERX (Entity Relationship expert) - მონაცემების კონცეპტუალური მოდელირების მოდული უზრუნველყოფს მონაცემთა მოდელების “არსი-კავშირი” აგებას, რომელიც კონკრეტულ რეალიზაციას არ უკავშირდება. ამ მოდულს აქვს ჩაშენებული ექსპერტული სისტემა, რომელიც მონაცემთა ურთიერთკავშირის შესახებ კითხვებზე პასუხის გაცემით, კორექტული მონაცემების ნორმალურიზებული მოდელის შექმნის შესაძლებლობას იძლევა. შესაძლებელია მონაცემთა სტრუქტურის აღწერით მონაცემთა მოდელის ავტომატური აგება. ატრიბუტების ფუნქციური დამოკიდებულების ანალიზი მესამე ნორმალური ფორმის მოთხოვნებთან მოდელის შესაბამისობის შემოწმებისა და მათი შესრულების შესაძლებლობას იძლევა. შემოწმებული მოდელი, RDM მოდულში გადაეცემა.

RDM (Relational Data Modeler) - რელაციური მოდელირების მოდული რელაციურ მონაცემთა ბაზაში რეალიზებისათვის განკუთვნილ დეტალიზებული მოდელის “არსი-კავშირი” შექმნის შესაძლებლობას იძლევა. ამ მოდელში მონაცემთა ბაზების აგებასთან დაკავშირებული ყველა კონსტრუქციის დოკუმენტირება ხდება: ინდექსები, ტრიგერები, შენახული პროცედურები და ა.შ. ეს მოდული რელაციურ მონაცემთა ბაზების დაპროექტებასა და სრულ დოკუმენტირებას უზრუნველყოფს.

WRM (Workgroup Repository Manager) – სამუშაო ჯგუფის რეპოზიტორიის მენეჯერი გამოიყენება, როგორც მონაცემების ლექსიკონი ყველა მოდელის საერთო ინფორმაციის შესანახად, აგრეთვე, დაპროექტების ერთიან გარემოში Silverrun მოდულების ინტეგრაციას უზრუნველყოფს.

Silverrun-ის ნაკლს სხვადასხვა მოდელების კომპონენტებს შორის, მყარი ურთიერთკონტროლის არარსებობა წარმოადგენს. მაგრამ უნდა აღინიშნოს, რომ ამ ნაკლს არსებითი მნიშვნელობა მხოლოდ პროგრამული უზრუნველყოფის სასიცოცხლო ციკლის კასკადური მოდელის გამოყენების დროს აქვს.

მონაცემთა ბაზების სქემის ავტომატური გენერაციისათვის Silverrun-ს ყველაზე გავრცელებულ ბმბს-თან აქვს ხიდები: Oracle, Informix, DB2, Ingres, Progress, SQL Server, SQLBase, Sybase. დანართების დამუშავების საშუალებებში მონაცემების გადასაცემად, 4GL ენებთან არსებობს ხიდები: JAM, PowerBuilder, SQL Windows, Uniface, NewEra, Delphi.

ყველა ეს ხიდი იძლევა შესაძლებლობას Silverrun-ში შესაბამისი მბმს-ის კატალოგიდან ან 4GL ენებიდან RDM ინფორმაცია ჩაიტვირთოს. ეს უკვე ექსპლუატაციაში მყოფი მონაცემთა ბაზების და გამოყენებითი სისტემების დოკუმენტირების, პროექტის შეცვლის ან ახალ პლატფორმაზე გადატანის საშუალებას იძლევა. ხიდის გამოყენებისას Silverrun მიზნობრივი სისტემისათვის თავის შინაგან რეპოზიტორს სპეციფიკური ატრიბუტებით აფართოებს. ამ ატრიბუტების მნიშვნელობის განსაზღვრის შემდეგ, დანართების გენერატორს ისინი გარემოს დამუშავების შიგა კატალოგში გადააქვს ან SQL ენაზე კოდის გენერაციის დროს გამოიყენებს. ამგვარად, შესაძლებელია კონკრეტული მბმს ყველა შესაძლებლობის გამოყენებით მონაცემთა ბაზების ბირთვი ტრიგერებით, შენახული პროცედურებით, მიმართვის მთლიანობის შეზღუდვით მთლიანად განისაზღვროს.

Silverrun სისტემაში დაპროექტების ავტომატიზაციის სხვა საშუალებებთან მონაცემების გაცვლისათვის, ანალიზის სპეციალური პროცედურების შესაქმნელად და პროექტის სპეციფიკაციების შესამოწმებლად, სხვადასხვა სტანდარტების შესაბამისად სპეციალური ანგარიშის შესადგენად, გარე ფაილებში საპროექტო ინფორმაციის გადასაცემად სამი ხერხი არსებობს:

1. ანგარიშების სისტემა. შესაძლებელია, რეპოზიტორიით ანგარიშების შემადგენლობის განსაზღვრის შემდეგ, ანგარიში ტექსტურ ფაილში გადაიციოს. ეს ფაილი შემდგომში შეიძლება ტექსტურ რედაქტორში ჩაიტვირთოს ან სხვა ანგარიშში იქნეს ჩართული;

2. სისტემა ექსპორტი/იმპორტი. ფაილთა სტრუქტურაზე უფრო სრულყოფილი კონტროლისათვის, სისტემაში ექსპორტი/იმპორტი შესაძლებელია არა მხოლოდ ექსპორტული ფაილის შინაარსის, არამედ ჩანაწერთა გამყოფების, ჩანაწერებში ველების, ტექსტური ველების დასაწყისისა და დასასრულის მარკერების განსაზღვრა. მითითებული სტრუქტურების ფაილების არა მხოლოდ ფორმირება, არამედ მათი რეპოზიტორში ჩატვირთვაცაა შესაძლებელი. ეს სხვადასხვა სისტემებთან (სხვა CASE-საშუალებებთან, მზმს-თან, ტექსტურ რედაქტორებთან და ელექტრონულ ცხრილებთან) მონაცემების გაცვლის შესაძლებლობას იძლევა;

3. ODBC-დრაივერის საშუალებით გარე ფაილებში რეპოზიტორის შენახვა. ყველაზე გავრცელებული მონაცემთა ბაზის მართვის სისტემებიდან, რეპოზიტორის მონაცემებთან წვდომის უზრუნველსაყოფად, მთელი საპროექტო ინფორმაციის უშუალოდ ამ მზმს-ის ფორმატში შენახვაა.

ჯგუფურ მუშაობას Silverrun სისტემაში ორი ხერხით ახორციელებენ:

1. სტანდარტულ ერთმომხმარებლიან ვერსიაში მოდელების კონტროლირებადი დაყოფისა და შეერთების მექანიზმი მოიპოვება. მოდელის ნაწილებად გაყოფის შემდეგ, შესაძლებელია ის რამდენიმე დამპროექტებელს დაურიგდეს. დეტალური დამუშავების შემდეგ, მოდელი მთლიან სპეციფიკაციაში ერთიანდება;

2. Silverrun-ის ქსელური ვერსია მოდელებთან, რომლებიც ქსელურ რეპოზიტორში მზმს Oracle, Sybase და Informix ბაზაზე ინახება, ერთობლივი ჯგუფური მუშაობის შესაძლებლობას იძლევა. ამასთან, რამდენიმე

პროგრამისტს ერთსა და იმავე მოდელთან მუშაობა შეუძლია, რადგანაც ობიექტების ბლოკირება მოდელის ცალკეული ელემენტების დონეზე ხდება.

Silverrun-ის ფუნქციონირების გარემოს MS Windows და Macintosh წარმოადგენს – მათ შორის საპროექტო მონაცემების გაცვლის შესაძლებლობით.

JAM

დანართების დამუშავების საშუალება JAM (JYA CC's APPLication Manager) – ფირმა JYACC (USA) პროდუქციაა.

JAM-ის ძირითად ნიშანს RAD მეთოდოლოგიასთან მისი შესაბამისობა წარმოადგენს, რადგანაც ის დანართების დამუშავების ციკლის რეალიზებას სწრაფად ახერხებს, რომელიც დანართის პროტოტიპის წინა ნაბიჯზე გამოვლენილი და მომხმარებლისთვის წარდგენილი მოთხოვნების გათვალისწინებით, შემდგომი ვერსიის ფორმირებაში მდგომარეობს.

JAM-ს მოდულური სტრუქტურა აქვს და შემდეგი კომპონენტებისაგან შედგება:

- სისტემის ბირთვი;
- JAM/DB – მზმს-სთან ინტერფეისის სპეციალიზებული მოდული (JAM/DBi-Oracle, JAM/DBi-Informix, JAM/DBi-ODBC და სხვ.);
- JAM/RW – ანგარიშთა გენერატორის მოდული;
- JAM/CASEi – CASE-საშუალებებისათვის ინტერფეისის სპეციალიზებული მოდული (JAM/CASE-TeamWork, JAM/CASE-Innovator და სხვ.);
- JAM/TPi – ტრანზაქციის მენეჯერებისათვის ინტერფეისის სპეციალიზებული მოდული (მაგ: JAM/TPi-Server TUXEDO და სხვ.);
- Xterm – X-ტერმინალის სპეციალიზირებული ემულგატორი.

სისტემის ბირთვი დასრულებული პროდუქტია და დანართების დასამუშავებლად მისი დამოუკიდებელი გამოყენება შესაძლებელია. ყველა დანარჩენი მოდული დამატებითია და მათი დამოუკიდებლად გამოყენება შეუძლებელია.

სისტემის ბირთვი შემდეგ ძირითად კომპონენტებს მოიცავს:

- ეკრანის რედაქტორი. მის შემადგენლობაში შედის: ეკრანის დამუშავების გარემო ობიექტების ვიზუალური რეპოზიტორი, საკუთარი

მზმს JAM-JOB, ტრანზაქციების მენეჯერი, გამართვის საშუალებანი, სტილის რედაქტორი;

- მენიუს რედაქტორი;
- დამხმარე უტილიტების ნაკრები;
- დანართის საწარმოო ვერსიის შექმნის საშუალებანი.

JAM დამუშავების გამოყენების დროს დანართის გარე ინტერფეისის დამუშავება ვიზუალურ დაპროექტებას წარმოადგენს და ეკრანის ფორმის შექმნით, მათზე ინტერფეისული კონსტრუქციების განთავსებითა და ინფორმაციის შეტანა-გამოტანის ეკრანული ველების განსაზღვრით მთავრდება. JAM-ში ინტერფეისის დაპროექტება ეკრანის რედაქტორის საშუალებით წარმოებს. JAM-ში დამუშავებულ დანართებს მრავალფანჯრიანი ინტერფეისი აქვს. ცალკეული ეკრანის დამუშავება, მასზე ცალკეული ინტერფეისული ელემენტების განთავსებაში მდგომარეობს. მენიუს რედაქტორი სისტემის მენიუს დამუშავებისა და გამართვის შესაძლებლობას იძლევა. დანართის ამა თუ იმ ობიექტის თითოეული კონკრეტული მენიუს დანიშნულება ეკრანის რედაქტორში განისაზღვრება.

JAM ბირთვში ერთმომხმარებლიანი რელაციური მზმს JDB ჩაშენებული. JDB-ს ძირითადი დანიშნულება დანართების პროტოტიპების შექმნაა, იმ შემთხვევებში, როცა მუშაობა არსებულ მზმს-თან შეუძლებელი ან არაგონივრულია. JDB-ში რელაციური მზმს-ის შესაძლებლობათა აუცილებელი მინიმუმია რეალიზებული, ინდექსების, შენახული პროცედურების, ტრიგერებისა და წარმოდგენების გამოკლებით. JDB-ს მეშვეობით მიზნობრივი მონაცემთა ბაზის, იდენტური მონაცემთა ბაზისა და დანართის მნიშვნელოვანი ნაწილის აგებაა შესაძლებელი.

JAM უტილიტები 3 ჯგუფს მოიცავს:

1. ეკრანული JAM ფაილების ტექსტურში კონვერტირები. JAM ეკრანებს საკუთარი ფორმატის ორმაგი ფაილების სახით ინახავს. რიგ შემთხვევებში, ეკრანების ტექსტური აღწერაა საჭირო;

2. ინფორმაციის შეტანა/გამოტანის მოწყობილობათა კონფიგურაცია. JAM და მისი მეშვეობით შექმნილი დანართები, უშუალოდ შეტანა/გამოტანის მოწყობილობებთან არ მუშაობს. ამის ნაცვლად, JAM შეტანა/გამოტანის ლოგიკურ მოწყობილობებს მიმართავს. ლოგიკური მოწყობილობების ფიზიკურში ასახვა კონფიგურაციის საშუალებებით წარმოებს;

3. ეკრანის ბიბლიოთეკების მომსახურება;

ერთ-ერთი JAM დამატებითი მოდულია ანგარიშთა გენერატორი, რომელიც ანგარიშში გამოტანილი მონაცემების განსაზღვრის, გამოტანილი ინფორმაციის დაჯგუფების, გამოტანის დაფორმატების და სხვათა შესაძლებლობას იძლევა.

JAM-ის გამოყენებით დამუშავებული დანართები ე.წ. შემსრულებელ სისტემებს არ საჭიროებს და შესაძლებელია შემსრულებელი მოდულის სახით იყოს დამუშავებული. ამისათვის, პროგრამისტს C-ენის კომპილატორი და კავშირების რედაქტორი უნდა ჰქონდეს. JAM-ის შემადგენლობაში საწარმო ვერსიის შესაქმნელად, აწყობის ფაილი (makefile), დანართების რიგი მოდულების საწყისი ტექსტები და აუცილებელი ბიბლიოთეკები შედის.

JAM დაპროგრამების ჩაშენებულ JPL (JAM Procedural Language) ენას შეიცავს, რომლის მეშვეობით საჭიროებისამებრ სპეციფიკური ქმედებების რეალიზაციის მოდულის დაწერაა შესაძლებელი. მოცემული ენა ინტერპრეტირებადია, რაც მის გამართვას ამარტივებს. ვიზუალურად შექმნილი დანართის გარემოსა და ამგვარ მოდულებს შორის არსებობს ინფორმაციის გაცვლის შესაძლებლობა. ამას გარდა, JAM-ში რომელიმე ენაზე დაწერილი გარე მოდულების მიერთების შესაძლებლობაა რეალიზებული.

JAM-ის მეშვეობით დამუშავებული დანართის საწარმოო ვერსია შემდეგ კომპონენტებს მოიცავს:

- დანართის ინტერპრეტატორის შესრულებადი მოდული. ამ მოდულში შესაძლოა პროგრამისტის მიერ მე-3 თაობის ენებზე დაწერილი ფუნქციები იყოს ჩაშენებული;
- ეკრანები, რომლებიც თვით დანართს ქმნიან. შესაძლოა იგი ეკრანების ბიბლიოთეკების შემადგენლობაში ცალკეული ფაილების სახით იყოს წარმოდგენილი, ან ინტერპრეტატორის ტანში იყოს ჩაშენებული;
- გარე JPL-მოდულები. შესაძლოა წარმოდგენილი იყოს ტექსტური ფაილების ან კომპილირებამდე სახით, ამავე დროს კომპილირებამდე გარე JBL-მოდულები შეიძლება როგორც ცალკეული ფაილების სახით, ისე ეკრანების ბიბლიოთეკების შემადგენლობაში იყოს წარმოდგენილი;

- დანართების კონფიგურაციის ფაილები – კლავიატურის და ტერმინალის კონფიგურაციის ფაილები, სისტემური შეტყობინების ფაილი, საერთო კონფიგურაციის ფაილი.

მბმს-თან პირდაპირი ურთიერთქმედების რეალიზებას JAM/DBi (Data Base interface) მოდული ახდენს. JAM-ში ურთიერთქმედების რეალიზაციის საშუალებები 2 კლასად იყოფა: ხელით და ავტომატური.

დანართის ხელით დამუშავებისას, პროგრამისტი SQL-ში მოთხოვნებს თვითონ წერს, რომლებშიც როგორც შეკითხვების შესრულების შედეგების წყაროები, ასევე მიღების ადრესატებიც, როგორც გარე ასევე შიგა დონის ვიზუალურად დაპროექტებული ინტერფეისული ელემენტები, საბოლოო მომხმარებლისათვის შეიძლება უხილავი ცვლადები იყოს.

JAM ტრანზაქციის მენეჯერის მიერ რეალიზებული ავტომატური რეჟიმი მონაცემთა ბაზებთან ტიპური და უფრო გავრცელებული სახეობის ოპერაციებისათვის არის განხორციელებადი, ე. წ. QBE (Query By Example) შეკითხვები ნიმუშის მიხედვით, მონაცემთა ბაზის ცხრილებს შორის საკმაოდ რთული ურთიერთკავშირის გათვალისწინებით და ეკრანული ველების შეტანა/გამოტანის ატრიბუტების ავტომატური მართვით.

JAM შესაძლებლობას იძლევა დანართები 20-ზე მეტ მბმს-თან მუშაობისათვის შეიქმნას.

JAM-ის განმასხვავებელი ნიშანია სხვადასხვა პლატფორმებს შორის დანართების გადატანის მაღალი დონე. გადატანის დროს ძირითადი მოთხოვნაა დანართისათვის შეტანა/გამოტანის ფიზიკურ მოწყობილობასა და მათ ლოგიკურ წარმოდგენას შორის მხოლოდ შესაბამისობის განსაზღვრა.

მბმს-სთან SQL-ის ურთიერთქმედების საშუალებად გამოყენება მბმს შორის გადატანის უზრუნველყოფის წინაპირობას ქმნის. თვით მონაცემთა ბაზის სტრუქტურის გადატანის შემთხვევაში, ზოგჯერ დანართი არავითარ მოდიფიკაციას არ საჭიროებს, გამონაკლისია მუშაობის სენსის ინიციალიზაცია.

სისტემაზე დატვირთვის გაზრდისა და გადასაწყვეტი ამოცანების სირთულის დროს ტრანზაქციის მენეჯერების მეშვეობით არქიტექტურის სამრგოლიანი მოდელი “კლიენტი-სერვერი” გამოიყენება. JAM/TPi-Client და JAM/TPi-Server კომპონენტებით შესაძლებელია საკმაოდ მარტივად გადასვლა სამრგოლიან მოდელზე. ამ დროს გადამწყვეტ როლს ასრულებს მოდული JAM/TPi-Server, რადგანაც სამრგოლიანი მოდელის დანერგვის

ძირითადი სიძნელეა ტრანზაქციის მენეჯერების სერვისში დანართის ლოგიკის რეალიზაცია.

JAM/CASEi ინტერფეისი მბმს-ის ინტერფეისის მსგავსია და ინფორმაციის გაცვლას JAM ობიექტების რეპოზიტორსა და CASE-საშუალებების რეპოზიტორს შორის ახორციელებს, როგორც პირდაპირ მონაცემთა ბაზიდან JAM რეპოზიტორში მონაცემთა ბაზის სტრუქტურის იმპორტირება ხდება. განსხვავება ისაა, რომ CASE-ის ინტერფეისის შემთხვევაში ეს გაცვლა ორ მიმართულებიანია. JAM/CASEi მოდულის გარდა, აგრეთვე არსებობს JAM/CASEi Developer's Kit მოდული, რომლის მეშვეობითაც შესაძლებელია კონკრეტული CASE-საშუალებისათვის ინტერფეისი დამოუკიდებლად დამუშავდეს, თუ მისთვის მზა JAM/CASEi მოდული არ არსებობს.

ხიდი Silverrun-RDM ↔ JAM Silverrun-სა და JAM-ის CASE-საშუალებებს შორის ურთიერთკავშირის რეალიზაციას ახდენს. მოცემულ პროგრამულ პროდუქტს მუშაობის 2 რეჟიმი აქვს:

1. პირდაპირი რეჟიმი (Silverrun-RDM → JAM) განკუთვნილია CASE-ლექსიკონის ობიექტებისა და JAM რეპოზიტორის ელემენტების შექმნისათვის Silverrun-RDM-ში სქემის დონეზე წარმოსადგენად. ამ რეჟიმში ხიდი საშუალებას იძლევა მოხდეს ეკრანებისა და JAM რეპოზიტორის ელემენტების გენერაცია. ხიდი, ცხრილებს და RDM რელაციური სქემების დამოკიდებულებას JAM ობიექტების შესაბამისი ტიპებით მიმდევრობად გარდაქმნის. დანართის ეკრანების პროტოტიპირებისათვის Silverrun-RDM-ში ინტერფეისის მონაცემთა მოდელების აგების მეთოდისა კვესქემის მექანიზმის გამოყენებას გულისხმობს. თითოეული RDM კვესქემის აღწერისას, ხიდი JAM-ის ეკრანული ფორმის გენერირებას ახდენს.

2. უკურეჟიმი (JAM → Silverrun-RDM) განკუთვნილია CASE-ლექსიკონის ობიექტების მოდიფიკაციის Silverrun-RDM რელაციურ მოდელში გადასატანად.

რეინჟინერინგის რეჟიმი JAM ეკრანების ყველა თვისების Silverrun სქემაში RDM-იდან ადრე იმპორტირებულის გადატანის შესაძლებლობას იძლევა. ამ ეტაპზე მონაცემთა ბაზის მთლიანობის კონტროლისათვის სქემის შეცვლა ცხრილების და ცხრილების ველების დამატების ან წაშლის სახით არ დაიშვება.

JAM ბირთვს კონფიგურაციული მართვის საშუალებებთან ჩაშენებული ინტერფეისი (PVCS Windows-ის პლატფორმაზე და SCCS UNIX-ის პლატფორმაზე) აქვს. ამ სისტემების მართვის საშუალებით, ეკრანის ბიბლიოთეკები და/ან რეპოზიტორი გადაიცემა. ამ სისტემების არარსებობის შემთხვევაში, JAM ჯგუფური დამუშავების დამხმარე ფუნქციების ნაწილის რეალიზებას თვითონ ახდენს.

ობიექტზე ორიენტირებული CASE-საშუალებები (Rational Rose)

Rational Rose – ფირმა Rational Software Corporation CASE-საშუალება – პროგრამული უზრუნველყოფის ანალიზის ეტაპების და პროექტების ავტომატიზაციისათვის, აგრეთვე საპროექტო დოკუმენტაციის სხვადასხვა ენაზე კოდებისა და გამოშვების გენერაციისათვის არის განკუთვნილი. Rational Rose ობიექტზე ორიენტირებული ანალიზისა და დაპროექტების იმ სინთეზ-მეთოდოლოგიას იყენებს, რომელიც ამ სფეროში სამი წამყვანი სპეციალისტის: ბუჩის, რამბოსა და ჯეკობსონის შრომებს ეფუძნება. მათ მიერ დამუშავებული UML (Unified Modeling Language) ობიექტების მოდელირების უნივერსალური ნოტაცია ობიექტზე ორიენტირებული ანალიზისა და დაპროექტების სფეროში შესაძლოა სტანდარტად ჩაითვალოს.

Rational Rose-ის კონკურენტული ვარიანტი იმ ენის მიხედვით განისაზღვრება, რომლითაც პროგრამის კოდების გენერირება (C++, Smalltalk, PowerBuilder, Ada, SQL Windows და ObjectPro) ხდება. ძირითადი ვარიანტი - Rational Rose/C++ – შესაძლებლობას იძლევა პროექტის დოკუმენტაცია დიაგრამებისა და სპეციფიკაციების სახით დამუშავდეს, აგრეთვე პროგრამული კოდის გენერირება C++-ზე მოხდეს. ამის გარდა, Rational Rose პროგრამების რეინჟინერინგის საშუალებებს შეიცავს, რომლებიც ახალ პროექტებში პროგრამული კომპონენტების ხელახალ გამოყენებას უზრუნველყოფს.

Rational Rose მუშაობის საფუძველშია სხვადასხვა სახის დიაგრამებისა და სპეციფიკაციების აგება დევს, რომლებიც მოდელის ლოგიკურ და ფიზიკურ სტრუქტურას, მის სტატიკურ და დინამიკურ ასპექტებს

განსაზღვრავს. მათ რიცხვში შედის კლასების, მდგომარეობების, სცენარების, მოდულების, პროცესების დიაგრამები.

Rational Rose შემადგენლობაში შეიძლება 6 ძირითადი სტრუქტურული კომპონენტი გამოვყოთ: რეპოზიტორი, მომხმარებლის გრაფიკული ინტერფეისი, პროექტის დათვალიერების საშუალება, პროექტის კონტროლის საშუალება, სტატისტიკის შეგროვების საშუალება და დოკუმენტების გენერატორი. მათ ემატებათ კოდების გენერატორი და C++-ისათვის ანალიზატორი, რომლებიც რეინჟინერინგს – პროგრამების საწყისი ტექსტის საშუალებით პროექტის მოდელის აღდგენას უზრუნველყოფს.

რეპოზიტორი ობიექტზე ორიენტირებულ მონაცემთა ბაზას წარმოადგენს. დათვალიერების საშუალებები პროექტში "ნავიგაციას" უზრუნველყოფს, მათ შორის კლასების და ქვესისტემების იერარქიებით გადანაცვლებას, ერთი სახის დიაგრამიდან მეორეზე გადართვას და ა. შ. სტატისტიკის კონტროლისა და შეგროვების საშუალებებით შეცდომის პოვნა და აღმოფხვრა პროექტის განვითარების დონის მიხედვით, მისი აღწერის დასრულებამდე ხდება. რეპოზიტორში არსებული ინფორმაციის საფუძველზე მოხსენებათა გენერატორი გამომავალი დოკუმენტების ტექსტებს უკეთებს ფორმირებას.

CASE-საშუალება Rational Rose-ით პროექტის დამუშავების დროს შემდეგი დოკუმენტების ფორმირება ხდება:

- კლასების დიაგრამები;
- მდგომარეობათა დიაგრამები;
- სცენარების დიაგრამები;
- მოდულთა დიაგრამები;
- პროცესთა დიაგრამები;
- კლასების, ობიექტების, ატრიბუტებისა და ოპერაციების სპეციფიკაციები;
- პროგრამების ტექსტების მზა სახეობები;
- დასამუშავებელი პროგრამული სისტემის მოდელი.

ჯგუფური მუშაობის ორგანიზებისათვის Rational Rose-ში მოდელის მართვად ქვემოდელებად დაყოფაა შესაძლებელი. თითოეული მათგანი დამოუკიდებლად დისკზე ინახება ან მოდელში იტვირთება. ქვემოდელის სახით შესაძლოა კლასების კატეგორია ან ქვესისტემა იქნეს წარმოდგენილი.

მონაცემთა ბაზების ფიზიკური დაპროექტება

მონაცემთა ბაზების ფიზიკური დაპროექტების პრობლემების ძირითადი ნაწილი გამოყენებული მონაცემთა ბაზის სერვერის თავისებურებებზეა დამოკიდებული. კერძოდ, ეს მონაცემთა ბაზების სხვადასხვა ნაწილების დისკურ მეხსიერებაში განაწილების დაგეგმარებას ეხება: ცხრილებს, ინდექსებს, BLOB-ს, ჟურნალებს და ა. შ. შესაბამისი რეკომენდაციები ჩვეულებრივ, გამოყენებადი სისტემის ადმინისტრატორის სახელმძღვანელოში ინახება.

შესაძლებელია გამოიკვეთოს ზოგიერთი საერთო მოსაზრება, რომელიც სერვერის დეტალების რეალიზაციისაგან დამოუკიდებლად მოიაზრება. უპირველესად, ეს ინდექსებს ეხება. გასაგებია, რომ მონაცემთა ბაზის ცხრილებში რაც უფრო მეტი ინდექსი არსებობს, მით უფრო ძნელი იქნება მონაცემთა ამორჩევის მოთხოვნის შესრულება და უფრო ნელა მოხდება მონაცემთა ბაზის მოდიფიკაციის ოპერაციების შესრულება.

უმრავლეს სისტემაში ინდექსი ცხრილში განსაზღვრული თითოეული პირველადი, შესაძლო და გარე გასაღებისათვის ავტომატურად იქმნება. გასაღების განსაზღვრის აუცილებლობა საგნის გარემოს სემანტიკიდან გამომდინარეობს, ხოლო მბმს-ს გასაღებების მხარდასაჭერად და გამოსაყენებლად შესაბამისი ინდექსები ესაჭიროება. რაც შეეხება დამატებით ინდექსებს, რომლებიც მოთხოვნათა უფრო ეფექტურად შესასრულებლად შეტანილი, მათთან ძალზე ფრთხილი მოპყრობაა საჭირო. მნიშვნელოვან მოთხოვნათა ნაკრების წინასწარი გულდასმითი ანალიზია აუცილებელი. ასევე, დიდი, შევსებული ცხრილისათვის ახალი ინდექსის შექმნაა გასათვალისწინებელი – ეს სერიოზული, ძვირი ოპერაციაა. იაფად და უსაფრთხოდ შეიძლება ახალი ცხრილი ინდექსების ნაკრებით შეიქმნას და არსებულ შევსებულ ცხრილებს დაემატოს სვეტები. დიდი შევსებული ცხრილის ან მისი ცალკეული სვეტის განადგურება ძვირი დაგვიჯდება და უსაფრთხო არაა.

Intranet-დანართის დაპროექტების, დამუშავების, თანმხლები დოკუმენტაციის საშუალებები და მეთოდოლოგია

Intranet-დანართი კორპორაციული სისტემაა, რომლის ორგანიზები-სათვის Internet მექანიზმი გამოიყენება. Intranet-სისტემა შესაძლოა კომპიუ-ტერთა ლოკალურ ქსელზე, საკუთარ კორპორაციულ გლობალურ ქსელზე ან Intranet-ის ვირტუალურ კორპორაციულ ქვექსელზე იყოს დაფუძნებული. Intranet-სისტემის რამდენიმე ტიპს განასხვავებენ, თითოეული მათგანის რეალიზაციისათვის კი სხვადასხვა საშუალება გამოიყენება:

1. კომუნიკაციური Intranet-სისტემები უმთავრესად კორპორაციის ტერიტორიულად დაშორებული ქვესისტემების დასაკავშირებლად არის განკუთვნილი და ამით მრავალრიცხოვანი გამოყოფილი კავშირის ხაზის აუცილებლობა მცირდება. ამგვარი ტიპის სისტემების რეალიზების დროს სისტემის ეფექტურობას, სტანდარტებთან შესაბამისობასა და მართვის უნარს განსაკუთრებული ყურადღება უნდა მიექცეს.

2. ინტეგრირებადი Intranet-სისტემები სხვადასხვა კომუნიკაციური და დამუშავებაში მყოფი კორპორაციული ქვესისტემების ინტეგრაციას ემსახურება. ამ თვალსაზრისით, Intranet-სისტემის ღირსებას სისტემებთან წვდომის საერთო ინტერფეისის მხარდაჭერა და ინფორმაციის ჰიპერტექსტური წარმოდგენის ხარჯზე მათ შორის კავშირის დამყარება წარმოადგენს.

3. თუ Intranet-სისტემისაგან ინფორმაციის დიდ მოცულობასთან ფართო წვდომა მოითხოვება, კერძოდ, მულტიმედიურთან, მაშინ განსაკუთრებული ყურადღება მონაცემთა ბაზების ბაზური სერვერის არჩევას უნდა მიექცეს. სერვერის შესაძლებლობები ძალიან დიდი მოცულობის მონაცემების მართვის ნაწილში და მონაცემთა რთული ტიპების მხარდაჭერაში უნდა იქნეს გათვალისწინებული.

4. Intranet-სისტემები მომხმარებლისათვის წვდომის გამარტივებული პროცედურით ჩვეულებრივ ელექტრონული ხელმოწერის მექანიზმს ეფუძნება. ასეთი სისტემები გარე სამყაროსაგან განსაკუთრებით საიმედოდ უნდა იყოს დაცული.

რა თქმა უნდა, ზოგადად, ინფორმაციულ Intranet-სისტემას შეუძლია თითოეული ჩამოთვლილი ტიპის თვისებები ჰქონდეს და ამ შემთხვევაში, მისი დამუშავებისას ყველა მოთხოვნის გათვალისწინებაა საჭირო.

CASE-ტექნოლოგიების ძირითადი ცნებები და მათი კლასიფიკაცია

ტერმინი CASE (Computer Aided System/Software Engineering) საკმაოდ ფართო გაგებით გამოიყენება. დასაწყისში CASE ტერმინის ცნება მხოლოდ და მხოლოდ პროგრამული უზრუნველყოფის ავტომატიზაციის ზოგიერთი საკითხებით შემოიფარგლებოდა. ამჟამად, მან შეიძინა ახალი აზრი, რომელიც რთული ეკონომიკური საინფორმაციო სისტემების დამუშავების მთელ პროცესს მოიცავს. თავდაპირველად, CASE-ტექნოლოგიები დაპროექტების სტრუქტურული მეთოდოლოგიის გამოყენების დროს შეზღუდვების გადალახვის მიზნით (გაგების სირთულის, შრომის მაღალი მოცულობისა და გამოყენების ძვირად ღირებულების, დაპროექტების სპეციფიკაციებში ცვლილებების შეტანის სიმძნელის გამო და სხვ.), მისი ავტომატიზაციისა და მხარდამჭერი საშუალებების ინტეგრაციის ხარჯზე ვითარდებოდა. ამგვარად, CASE-ტექნოლოგიები არ შეიძლება დამოუკიდებელ სისტემად ჩაითვალოს, როგორც მინიმუმ, მხოლოდ გამოყენების მაღალ ეფექტურობას, ხოლო ზოგიერთ შემთხვევაში, შესაბამისი მეთოდოლოგიის გამოყენების პრინციპულ შესაძლებლობას უზრუნველყოფს. უმეტესობა, არსებული CASE-სისტემები პროგრამული უზრუნველყოფის დაპროექტების ავტომატიზაციაზეა ორიენტირებული და სტრუქტურაზე ან ობიექტზე ორიენტირებული დაპროექტებისა და დაპროგრამების მეთოდოლოგიაზეა დაფუძნებული. სპეციფიკაციებს სისტემური მოთხოვნების აღწერის, სისტემის მოდელს შორის კავშირის, სისტემის ქცევის დინამიკისა და პროგრამული საშუალებებისათვის დიაგრამებისა და ტექსტების სახით იყენებს. ბოლო დროს ისეთი CASE-სისტემები გამოჩნდა, რომლებიც ძირითად ყურადღებას სპეციფიკაციებისა და ტექნიკურ საშუალებათა მოდელირების პრობლემებს უთმობს.

CASE-სისტემების გამოყენებაზე განსაკუთრებული მოთხოვნილება დამუშავების საწყის ეტაპზე, კერძოდ საინფორმაციო სისტემებისადმი მოთხოვნილებების ანალიზისა და სპეციფიკაციების ეტაპზე შეიმჩნევა. ეს იმით აიხსნება, რომ დაპროექტების საწყის ეტაპზე დაშვებული შეცდომის

გამოსწორება უფრო მეტ ხარჯებს გამოიწვევს, ვიდრე დაპროექტების შემდგომ ეტაპებზე დაშვებული შეცდომა.

CASE-ტექნოლოგიების გამოჩენას წინ უსწრებდა დაპროგრამების მეთოდოლოგიის სფეროში წარმოებული კვლევები. ორიგინალური დაპროექტების ტრადიციულ ტექნოლოგიებთან შედარებით, CASE-ტექნოლოგიების უპირატესობა შემდეგია:

- კონტროლისა და გენერაციის ავტომატიზაციის საშუალებების ხარჯზე დამუშავებული პროგრამული დანართის ხარისხი უმჯობესდება;

- დამუშავებული სისტემის კომპონენტების განმეორებითი გამოყენება ხდება შესაძლებელი;

- სისტემის დამუშავების დროის შემცირება ხდება შესაძლებელი, რაც დაპროექტების ადრეულ სტადიაში მომავალი სისტემის პროტოტიპის მიღებისა და მისი შეფასების საშუალებას გვაძლევს;

- დამპროექტებლები პროექტის დოკუმენტების შედგენის მოსაბეზრებელი სამუშაოსაგან თავისუფლდებიან, რადგანაც ამ დროს მასში ჩადგმული დოკუმენტატორი გამოიყენება;

- საინფორმაციო სისტემის კოლექტიური დამუშავება რეალური დროის რეჟიმში ხდება შესაძლებელი.

CASE-ტექნოლოგიები მეთოდოლოგიაში გულისხმობს მეთოდებს, რომლის მეშვეობითაც ინსტრუმენტული გარემოს მხარდაჭერით გრაფიკული ნოტაციის საფუძველზე დიაგრამები იგება.

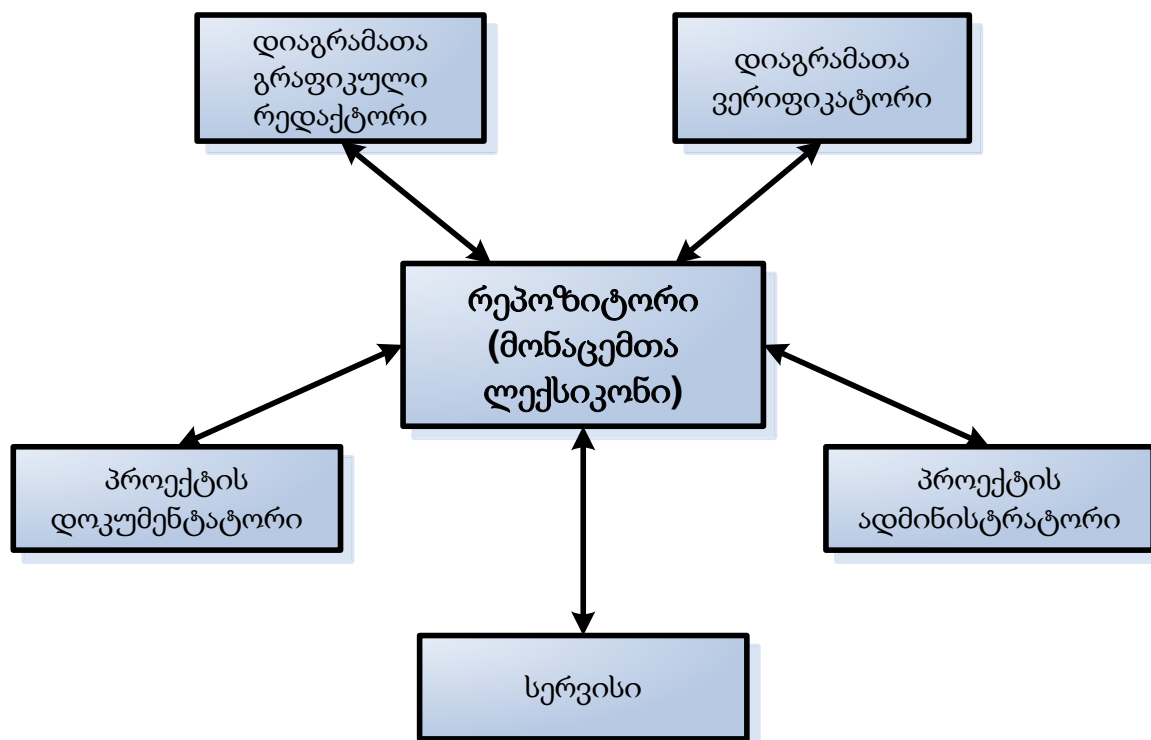
მეთოდოლოგია პროექტის რეალიზაციის ბიჯებსა და ეტაპებს, აგრეთვე იმ მეთოდების გამოყენების წესებს განსაზღვრავს, რომელთა მეშვეობითაც მუშავდება პროექტი.

მეთოდი არის საინფორმაციო სისტემების აღწერის გენერაციის პროცედურა ან ტექნიკა (მაგალითად, მონაცემთა ნაკადისა და სტრუქტურის დაპროექტება).

ნოტაცია, სისტემის სტრუქტურის, მონაცემთა ელემენტების, დიაგრამების სპეციალური გრაფიკული სიმბოლოების საშუალებით დამუშავების ეტაპების ასახვა, აგრეთვე სისტემის პროექტის ფორმალური და ჩვეულებრივი ენით აღწერაა.

CASE-ის ინსტრუმენტული საშუალებანი – საინფორმაციო სისტემების ანალიზისა და დაპროექტების ერთი ან რამდენიმე მხარდამჭერი მეთოდოლოგიის სპეციალური პროგრამებია.

განვიხილოთ CASE-საშუალებათა არქიტექტურა, რომელიც 26-ე სურათზეა მოცემული:



ნახ. 26. CASE-საშუალებათა არქიტექტურა

სისტემის ბირთვს პროექტის მონაცემთა ბაზა – რეპოზიტორი (მონაცემთა ლექსიკონი) წარმოადგენს. ის მონაცემთა სპეციალური ბაზაა, რომელიც დროის ნებისმიერ მომენტში დასაპროექტებელი საინფორმაციო სისტემის მდგომარეობის წარმოსადგენად არის განკუთვნილი. ყველა დიაგრამის ობიექტები მონაცემთა ლექსიკონის საერთო ინფორმაციის საფუძველზე, სინქრონულია.

რეპოზიტორი მოიცავს ინფორმაციას დასაპროექტებელი საინფორმაციო სისტემის ობიექტებისა და მათ შორის კავშირის შესახებ, ამავდროულად, მასთან ყველა ქვესისტემის მონაცემების მიმოცვლა ხდება. რეპოზიტორში შემდეგი ობიექტების აღწერა ინახება:

- დამპროექტებლებისა და სისტემის სხვადასხვა კომპონენტებისადმი მათი მიმართვის უფლებების;
- ორგანიზაციული სტრუქტურის;
- დიაგრამების;
- დიაგრამათა კომპონენტების;

- დიაგრამებს შორის კავშირის;
- მონაცემთა სტრუქტურის;
- პროგრამული მოდულების;
- პროცედურების;
- მოდულთა ბიბლიოთეკისა და სხვათა შესახებ.

დიაგრამათა გრაფიკული რედაქტორის დანიშნულებაა დასაპროექტებელი საინფორმაციო სისტემის მოცემულ ნოტაციაში გრაფიკული სახით გამოსახვა. ის საშუალებას იძლევა შემდეგი სამუშაო შესრულდეს:

- შეიქმნას დიაგრამათა ელემენტები და მათ შორის კავშირები;
- მომზადდეს დიაგრამათა ელემენტების აღწერა;
- მომზადდეს დიაგრამათა ელემენტებს შორის კავშირების აღწერა;
- განხორციელდეს დიაგრამათა ელემენტების, მათ შორის კავშირებისა და აღწერათა რედაქტირება.

დიაგრამათა ვერიფიკატორის დანიშნულებაა საინფორმაციო სისტემის დაპროექტების მოცემულ მეთოდოლოგიაში დიაგრამათა აგების სისწორის კონტროლი. იგი შემდეგ ფუნქციებს ასრულებს:

- დიაგრამათა აგების სისწორის მონიტორინგი;
- შეცდომათა დიაგნოსტიკა და შეტყობინებათა გაცემა;
- არასწორი ელემენტების გამოყოფა დიაგრამაზე.

პროექტის დოკუმენტატორი საშუალებას გვაძლევს სხვადასხვა ანგარიშების სახით მივიღოთ ინფორმაცია პროექტის მდგომარეობის შესახებ. ანგარიშები შეიძლება რამდენიმე ნიშნის მიხედვით აიგოს, მაგალითად დროის, ავტორის, დიაგრამათა ელემენტების, მთლიანად დიაგრამის ან პროექტის მიხედვით.

პროექტის ადმინისტრატორი არის ინსტრუმენტები, რომლებიც შემდეგი ადმინისტრაციული ფუნქციების შესასრულებლადაა განკუთვნილი:

- პროექტის ინიციალიზაცია;
- პროექტის საწყისი პარამეტრების დადგენა;
- პროექტის ცალკეულ ელემენტებთან მიმართვის უფლების დადგენა და შეცვლა;

- პროექტის შესრულების მონიტორინგი.

სერვისი არის რეპოზიტორის მომსახურების სისტემური უტილიტების ნაკრები. მოცემული უტილიტები მონაცემთა არქივირების, მონაცემთა აღდგენისა და ახალი რეპოზიტორის შექმნის ფუნქციებს ასრულებს.

თანამედროვე CASE-სისტემების კლასიფიკაცია შემდეგი თვისებების მიხედვით ხდება:

1. დაპროექტების მეთოდოლოგიის მიხედვით: სტრუქტურაზე ორიენტირებული, ობიექტზე ორიენტირებული და კომპლექსურად ორიენტირებული;

2. დიაგრამების აგების გრაფიკული ნოტაციის მიხედვით: ფიქსირებული ნოტაციით, ცალკეული ნოტაციით და ყველაზე უფრო გავრცელებული ნოტაციით;

3. ინტეგრირების ხარისხის მიხედვით: tools (ცალკეული ლოკალური საშუალებანი), toolkit (არაინტეგრირებული საშუალებათა ნაკრები, რომლებიც საინფორმაციო სისტემების დამუშავების უმეტესობა ეტაპს მოიცავს) და workbench (სრულად ინტეგრირებული საშუალებანი, რომლებიც საპროექტო მონაცემთა საერთო ბაზასთან – რეპოზიტორთან არის დაკავშირებული);

4. გამოთვლითი ტექნიკის ტიპისა და არქიტექტურის მიხედვით: პერსონალურ კომპიუტერებზე ორიენტირებული, ლოკალურ გამოთვლით ქსელზე ორიენტირებული, გლობალურ გამოთვლით ქსელზე ორიენტირებული და შერეული ტიპის;

5. პროექტის კოლექტიური დამუშავების რეჟიმის მიხედვით: არ ხდება პროექტის კოლექტიური დამუშავება, პროექტის დამუშავება დროის რეალურ რეჟიმზე და ქვეპროექტების გაერთიანების რეჟიმზეა ორიენტირებული;

6. ოპერაციული სისტემის ტიპის მიხედვით: მუშაობს WINDOWS-ის, UNIX-ის და სხვადასხვა ოპერაციული სისტემის მართვის ქვეშ.

თანამედროვე CASE-სისტემები დაპროექტებისა და დაპროგრამების მხარდამჭერ სხვადასხვა ტექნოლოგიების ფართო სფეროს მოიცავს: საინფორმაციო სისტემების ანალიზისა და დოკუმენტირების მარტივი საშუალებებიდან ავტომატიზაციის იმ ფართომამტაბიან საშუალებებამდე, რომლებიც საინფორმაციო სისტემების მთელ სასიცოცხლო ციკლს ფარავს.

დიდი მნიშვნელობა აქვს აგრეთვე, ისეთ CASE-სისტემებს, რომლებიც მონაცემთა ბაზისა და მომხმარებელთა ინტერფეისების დაპროექტებასა და

გენერაციაზეა ორიენტირებულნი. მონაცემთა ბაზების ინტერფეისის გენერაციასა და მონაცემთა სხვადასხვა კონცეპტუალურ სქემებსა და მოდელებს შორის გარდაქმნის (კონვერტირების) შესაძლებლობები სხვა ოპერაციულ გარემოში გადასვლის დროს გამოყენებითი სისტემის მობილურობას ზრდის.

ანალიზისა და დაპროექტების ეტაპი საინფორმაციო სისტემების დამუშავების ყველაზე შრომატევადი ეტაპია, ამიტომაც CASE-სისტემები, როგორც წესი, მიღებული საპროექტო გადაწყვეტილებებისა და დოკუმენტების მომზადების ხარისხისადმი თვალყურისდევნების ავტომატიზაციისათვის არის განკუთვნილი.

კონკრეტული გამოყენებისათვის CASE-სისტემების შერჩევის სტრატეგია როგორც თვითონ პროექტის მიზნებსა და მოთხოვნილებაზე, ისე დაპროექტების პროცესში მოზიდული სპეციალისტების კვალიფიკაციაზე დამოკიდებულია. როგორც წესი, უმეტესობა შემთხვევაში ერთი გარემო ვერ უზრუნველყოფს პროექტის ყველა მოთხოვნა.

საინფორმაციო სისტემების სტრუქტურაზე ორიენტირებული დაპროექტება

სტრუქტურაზე ორიენტირებული CASE-ტექნოლოგიების ძირითად იდეას საინფორმაციო სისტემების სტრუქტურული ანალიზისა და დაპროექტების იდეა წარმოადგენს. იგი შემდეგში მდგომარეობს:

1. მთელი სისტემის რამდენიმე იერარქიულად ურთიერთდამოკიდებული სიმრავლის ფუნქციებად დეკომპოზიცია;
2. მთელი ინფორმაციის გრაფიკული ნოტაციის სახით წარმოდგენა. როდესაც სისტემა წარმოდგენილია გრაფიკული სახით, მაშინ მისი გაგება ბევრად მარტივია.

სტრუქტურული ანალიზისა და დაპროექტების ინსტრუმენტული საშუალებების სახით, შემდეგი დიაგრამები გამოიყენება:

- BFD (Business Function Diagram) – ბიზნეს-ფუნქციის დიაგრამა (ფუნქციური სპეციფიკაცია);
- DFD (Data Flow Diagram) – მონაცემთა ნაკადის დიაგრამა;
- STD (State Transition Diagram) – მდგომარეობათა გადასვლის დიაგრამა (ჯვარედინ მიმართვათა მატრიცა);

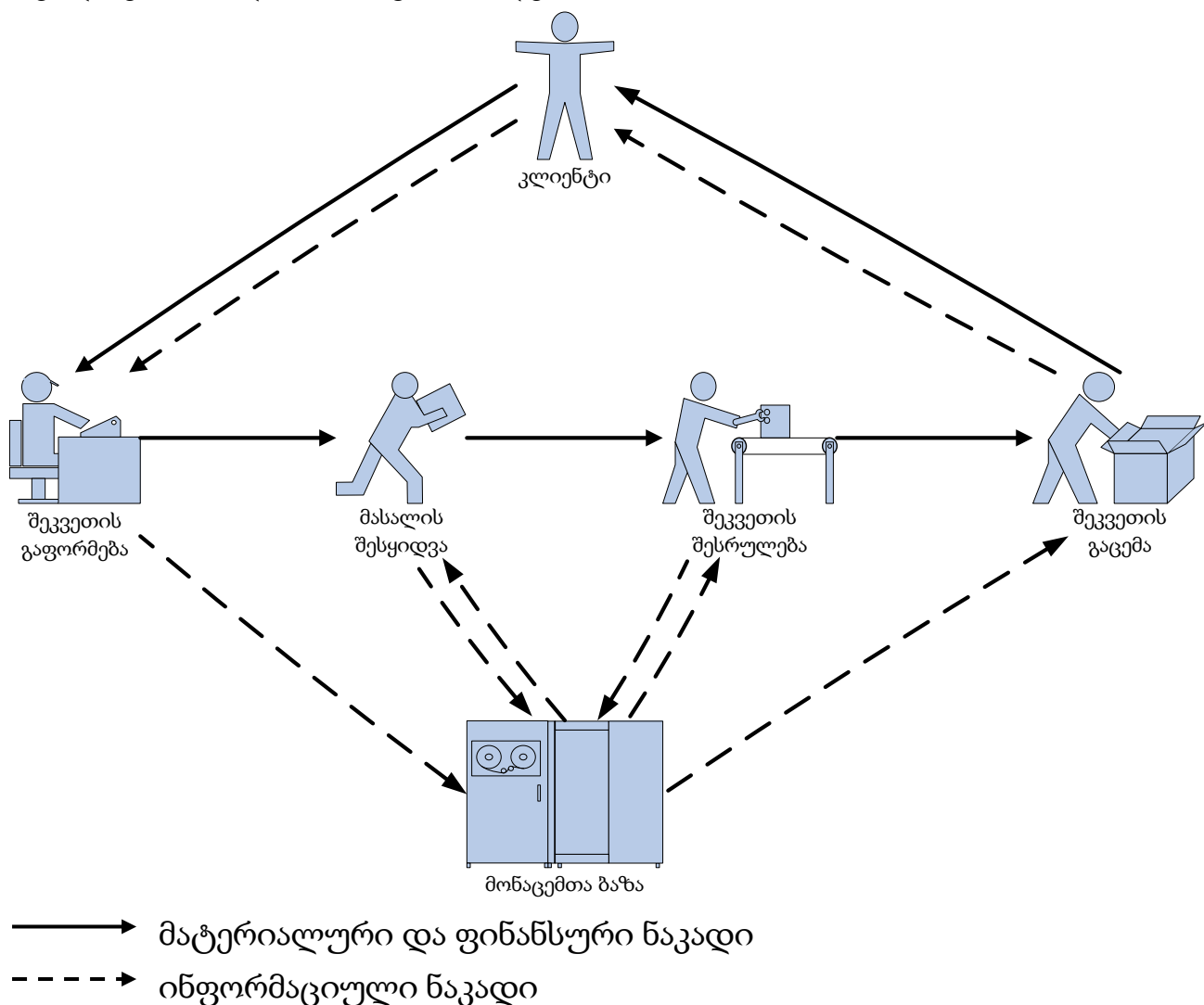
- ERD (Entity Relationship Diagram) – საგნობრივი არის მონაცემთა ER-მოდელი (ინფორმაციულ-ლოგიკური მოდელი “არსი-კავშირი”);

- SSD (System Structure Diagram) – პროგრამული დანართის სტრუქტურის დიაგრამა.

მაგალითი. საწარმოში შეკვეთის მიღებისა და მისი შესრულების ინფორმაციული ნაკადების დიაგრამა.

ბიზნეს-პროცესი არის ურთიერთდაკავშირებული ოპერაციების (სამუშაოს) ერთობლიობა მზა პროდუქციის დასამზადებლად ან რესურსის მოთხოვნის საფუძველზე მომსახურების შესასრულებლად.

ბიზნეს-პროცესის მართვის მიზანია მომხმარებლის (კლიენტის) ხარისხიანი მომსახურების განხორციელება. ამასთან, ბიზნეს-პროცესის მართვის დროს ყველა მატერიალური, ფინანსური და ინფორმაციული ნაკადი ერთობლიობაში განიხილება.



შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის Work Flow Diagram შაბლონის მეშვეობითაა შესრულებული, რომელიც დიაგრამების შესაქმნელად ვექტორულ გრაფიკას იყენებს.

ფუნქციური სპეციფიკაციების დიაგრამა

ფუნქციური სპეციფიკაციების დიაგრამა (BFD) საშუალებას გვაძლევს წარმოვადგინოთ საინფორმაციო სისტემის საერთო სტრუქტურა, რომელიც საჭირო შედეგების მიღების პროცესში სხვადასხვა ამოცანების ურთიერთკავშირს ასახავს. BFD-ს ძირითად ობიექტებს წარმოადგენს:

- ფუნქცია – ეკონომიკური ამოცანის ამოხსნისათვის განკუთვნილი საინფორმაციო სისტემის რაიმე მოქმედება;
- ფუნქციის დეკომპოზიცია – ფუნქციის დაყოფა რამდენიმე ქვეფუნქციებად.

ფუნქციის იერარქიის დიაგრამების ობიექტების გამოსახვა მე-5 ცხრილშია მოცემული, შემდეგ ნოტაციებში:

- იორდანის (Yourdon);
- ჰეინე-სარსონის (Gane-Sarson);
- SADT (Structure Analysis and Design Technique);
- SAG (Software AG).

ცხრილი 5

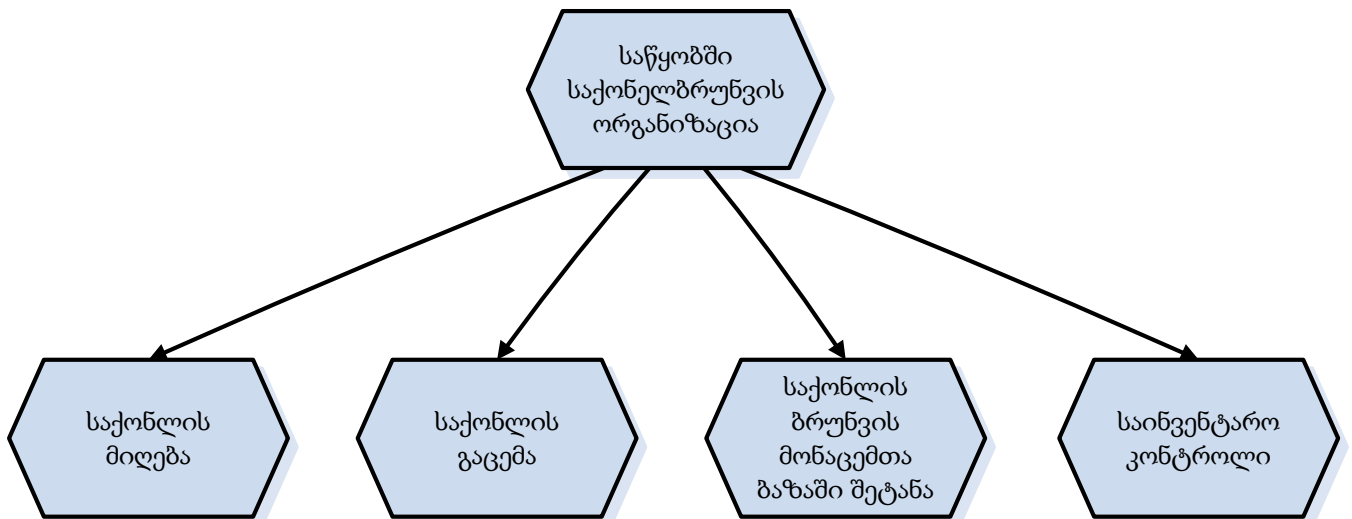
ფუნქციის იერარქიის დიაგრამების ობიექტების გამოსახვა

ობიექტი	იორდანის	ჰეინე-სარსონის	SADT	SAG
ფუნქცია				
ფუნქციის დეკომპოზიცია			ხორციელდება დეტალიზაციის იერარქიულად ურთიერთდაკავშირებული დონეების საშუალებით	

ფუნქციური სპეციფიკაციების დიაგრამების (BFD) შესრულება შემდეგი ამოცანების გადაწყვეტაზე დაიყვანება:

- ძირითადი ფუნქციის გამოსახვა;
- ძირითადი ფუნქციის დეკომპოზიცია ქვეფუნქციებად;
- ქვეფუნქციების შემდგომი დეკომპოზიცია დეტალიზაციის აუცილებელ ხარისხამდე;
- აგებული დიაგრამების სისწორის კონტროლი.

მაგალითი. საწყობში საქონლის ანალიტიკური აღრიცხვის ამოცანისათვის ფუნქციური სპეციფიკაციების დიაგრამა (BFD) ანუ ბიზნეს-ფუნქციის დიაგრამა SAG ნოტაციაში.



შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის Basic Flowchart შაბლონის საშუალებითაა შესრულებული.

მონაცემთა ნაკადის დიაგრამა

მონაცემთა ნაკადის დიაგრამა (DFD), როგორც წესი, მონაცემთა დამუშავების რაიმე ტექნოლოგიაზე მყარად არის ორიენტირებული და დამუშავების მოცემული ტექნოლოგიის ფარგლებში ერთი ფუნქციიდან მეორემდე ინფორმაციის გადაცემას ასახავს. მონაცემთა ნაკადის დიაგრამის კვანძებში (მართკუთხედებით) პროცედურები აისახება, ხოლო კვანძებს

შორის ისრებით მონაცემთა ნაკადია (ისრებზე მოცემული უნდა იყოს გადასაცემი ინფორმაციის სახელი/ინფორმაციის ერთეული) ნაჩვენები.

მონაცემთა ნაკადის დიაგრამა სისტემის მიმართ მის გარეთ მყოფ იმ მონაცემთა წყაროს და ადრესატებს გვიჩვენებს, რომლებიც ინფორმაციას სისტემისაგან იღებს, აგრეთვე მონაცემთა იმ საცავს წარმოადგენს, რომელზეც მოცემული სისტემა ახდენს მიმართვას. სისტემის თითოეული ლოგიკური ფუნქციის (ბიზნეს-ფუნქცია) აღწერა მონაცემთა ნაკადის დიაგრამით ხდება. ამასთან, მონაცემთა ნაკადის ეს დიაგრამა შეიძლება ფუნქციის იერარქიულ დეტალიზაციას მის ქვეფუნქციებად ახდენდეს.

მონაცემთა ნაკადი არის ერთი პროცესიდან მეორეზე ინფორმაციის გადაცემის მექანიზმი; სქემებზე ნაჩვენებია ისრებით, რომლებიც ინფორმაციის მოძრაობის მიმართულებას უჩვენებს.

პროცესი – მისი ფუნქციაა შემოსული ინფორმაცია საბოლოო ინფორმაციად გარდაქმნას. პროცესის სახელი აუცილებლად ზმნას განუსაზღვრელ ფორმით, შემდგომი დამატებით უნდა შეიცავდეს (მაგალითად, “დაიხატოს ფორმა”).

მონაცემთა საცავი საშუალებას იძლევა მონაცემთა ნაკადის დიაგრამის გარკვეულ უბნებზე პროცესებს შორის შევინახოთ მონაცემები. აუცილებელი არ არის, რომ საცავი ინფორმაციის მაგნიტური მატარებელი იყოს (მაგალითად, საქალაქი). მონაცემთა საცავის სახელი მისი და მისი შინაარსის იდენტიფიცირებას უნდა ახდენდეს.

გარე არსი (მონაცემთა წყარო/მიმღები) არის რაიმე ობიექტი სისტემის გარეთ.

კონტექსტური დიაგრამა – სისტემის დეკომპოზიციის ყველაზე მაღლა მდგომი პროცესი, რომელიც სისტემაზე საერთო წარმოდგენას იძლევა. კონტექსტურ დიაგრამაში არის ერთი პროცესი, რომელთანაც გარე არსია დაკავშირებული.

შემდეგ, კონტექსტური დიაგრამის იმ სისტემაში მიმდინარე ძირითად პროცესებად დეკომპოზიცია ხდება. შეიძლება ყოველი ძირითადი პროცესის დეკომპოზიცია უფრო მცირე პროცესებად მოხდეს. მონაცემთა ნაკადის იერარქიული დიაგრამების აგების დროს უფრო დაბალი დონის ყოველი პროცესი, უფრო მაღალი დონის პროცესს უნდა მივუთანადოთ. ჩვეულებრივ, ამისთვის, გამოიყენება კვანძების მემკვიდრეობის მექანიზმი.

მონაცემთა ნაკადის იერარქიულად ურთიერთდაკავშირებული დიაგრამების აგების მიზანი ის არის, რომ სისტემის მიმართ მოთხოვნები მისი დეტალიზაციის თითოეულ დონეზე ნათლად იყოს წარმოდგენილი. ამისათვის, შემდეგი რეკომენდაციები უნდა გავითვალისწინოთ:

- თითოეულ დონეზე წარმოვადგინოთ არა უმეტეს 3-6 პროცესი;
- დიაგრამა დეტალიზაციის მოცემულ მომენტში არაარსებითი მომენტებით არ გადავტვიროთ;
- პროცესებისა და ნაკადების დეკომპოზიცია ვაწარმოოთ პარალელურად;
- მონაცემთა ნაკადის დიაგრამის ყველა ობიექტისათვის გასაგები, ობიექტის არსის გამომხატველი სახელი ავარჩიოთ;
- ფუნქციურად იდენტური პროცესები ერთხელ აღწეროთ (სხვა ადგილებში კი მათზე მიმართვა მოვახდინოთ);
- მონაცემთა ნაკადის დიაგრამები იმ პროცესებისათვის გამოვიყენოთ, რომელთა აღწერა შეიძლება განხორციელდეს მათი საშუალებით.

განვსაზღვროთ მონაცემთა ნაკადის დიაგრამათა ძირითადი ობიექტები და მათი გრაფიკული გამოსახვა სხვადასხვა ნოტაციებში.

მონაცემთა ნაკადის დიაგრამების ობიექტების გამოსახვა მე-6 ცხრილშია მოცემული.

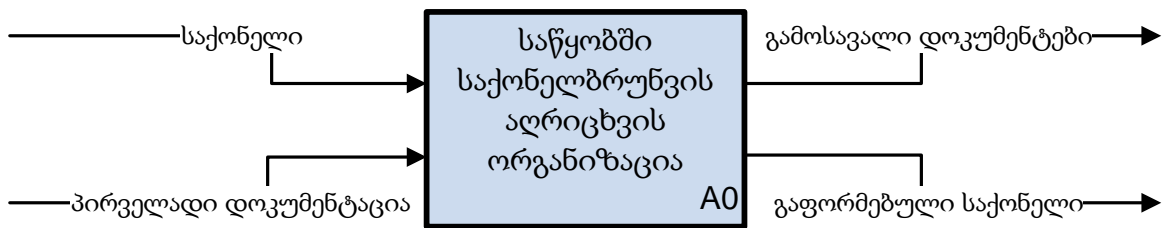
ცხრილი 6

მონაცემთა ნაკადის დიაგრამების ობიექტების გამოსახვა

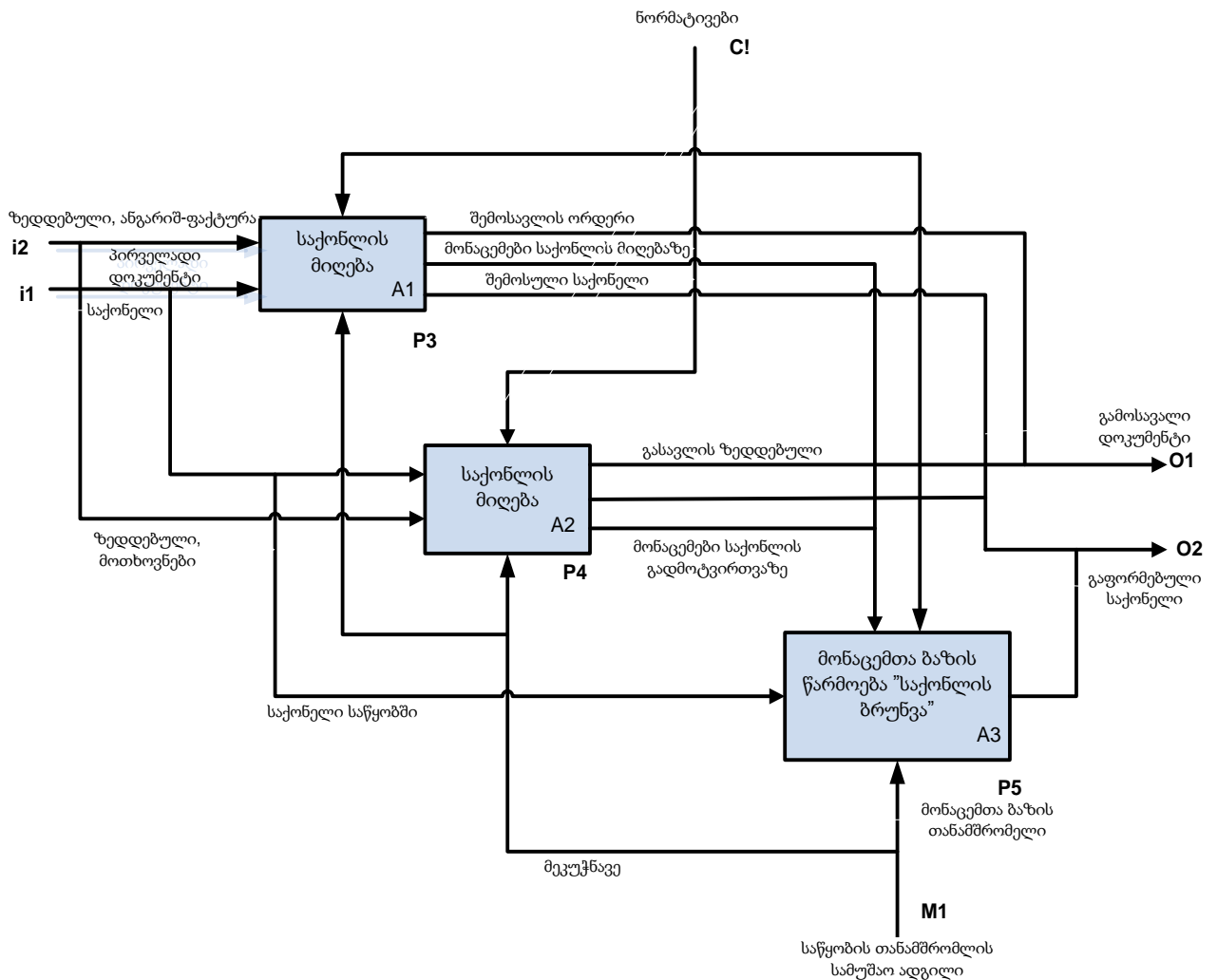
ობიექტი	იორდანის	ჰეინე-სარსონის	SADT	SAG
პროცესი				
მონაცემთა ნაკადი				
მონაცემთა საცავი			არა აქვს	
ინფორმაციის წყარო/მიმღები			ტექსტური ჭდე	

არსი	არა აქვს	არა აქვს	არა აქვს	სახელი ატრიბუტი
წაკითხვა/ჩაწერა	არა აქვს	არა აქვს	არა აქვს	წაკითხვა წაკითხვა/ჩაწერა ჩაწერა
ნაკადის დაჯგუფება				უნდა გაკეთდეს დამატებითი პროცესი
ნაკადის დაყოფა				არა აქვს
გამოუყენებელი კვანძი (სქემაზე არის, მაგრამ სისტემაში არ არის აღწერილი)		არა აქვს		
”წინაპარი” კვანძები (მემკვიდრეობითი კვანძები)			ICOM-ჭდეები	ავტომატური მემკვიდრეობა არ ხდება

მაგალითი. საწყობში საქონლის ანალიტიკური აღრიცხვის ამოცანისათვის მონაცემთა ნაკადის კონტექსტური დიაგრამა (DFD) SADT ნოტაციაში.



მაგალითი. საწყობში საქონლის ანალიტიკური აღრიცხვის ამოცანისათვის მონაცემთა ნაკადის პირველი დონის დიაგრამა (DFD) SADT ნოტაციაში.



შენიშვნა. ზემოთ მოყვანილ მაგალითებში დიაგრამები Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის Basic Flowchart შაბლონის საშუალებით არის შესრულებული.

მდგომარეობათა გადასვლის დიაგრამა

მდგომარეობათა გადასვლის დიაგრამა (STD) – სისტემის ქცევის მოდელირებას ახდენს დროში, რომელიც მომხდარ მოვლენებზეა დამოკიდებული. ასეთი დიაგრამები საშუალებას იძლევა სისტემაში მომხდარი მმართველი პროცესების დეკომპოზიცია და მმართველ ნაკადებს შორის არსებული დამოკიდებულების აღწერა განხორციელდეს. მდგომარეობათა გადასვლის დიაგრამის საშუალებით სისტემის წინა და მიმდინარე მდგომარეობიდან გამომდინარე, სისტემის შემდგომი ფუნქციონირების მოდელირება შეიძლება.

მოდელირების პროცესში სისტემა დროის მოცემულ მომენტში შესაძლო მდგომარეობებიდან მხოლოდ ერთ-ერთ მდგომარეობაში იმყოფება. დროის განმავლობაში მან შეიძლება შეიცვალოს მდგომარეობა და მომდევნო მდგომარეობაში გადავიდეს. სხვა მდგომარეობაში გადასვლისათვის, მას რაიმე განსაკუთრებული პირობა – გადასვლის პირობა ესაჭიროება. იგი შეიძლება ინფორმაციული (რაიმე ინფორმაციის წარმოშობა) ან დროითი იყოს. მდგომარეობათა გადასვლის დიაგრამის სიმბოლოები სხვადასხვა ნოტაციაში, მე-7 ცხრილშია მოცემული.

მდგომარეობათა გადასვლის დიაგრამის ძირითადი ობიექტები განვსაზღვროთ:

მდგომარეობა – განიხილება როგორც რაიმე თვისების მდგრადი მნიშვნელობა დროის რაიმე მონაკვეთის განმავლობაში. მოცემულ მდგომარეობაში ყოფნის დროს, აუცილებელია მისი წინა მდგომარეობის ცოდნა, რათა მომდევნო მდგომარეობაში გადასვლის პირობა განისაზღვროს.

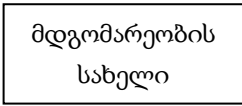
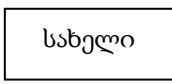



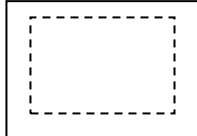
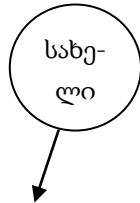
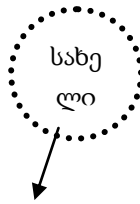
საწყისი მდგომარეობა არის მდგომარეობათა გადასვლის დიაგრამის ძირითადი კვანძი, რომელიც სასტარტო წერტილს საწყისი სისტემური გადასვლისათვის წარმოადგენს. მდგომარეობათა გადასვლის დიაგრამას მხოლოდ ერთი საწყისი მდგომარეობა აქვს, მაგრამ მას შეიძლება მრავალი საბოლოო მდგომარეობა ჰქონდეს.

გადასვლა – სისტემის გადანაცვლებას ერთი მდგომარეობიდან სხვა მდგომარეობაში განსაზღვრავს. ამასთან, გადასვლის სახელი არის ქმედება, რამაც ეს გადასვლა გამოიწვია. გადასვლა შეიძლება რაღაც მოქმედებით იყოს გამოწვეული.

ტრიგერი ლოგიკური გამოსახულებაა, რომელიც მაკრობრძანებების საშუალებით არის დაწერილი და მოცემულ მდგომარეობაში გადასვლის პირობას უჩვენებს.

გადასვლის პირობა არის ქმედება, რომელიც გადასვლას იწვევს და მისი იდენტიფიკაცია გადასვლის სახელში ხდება.

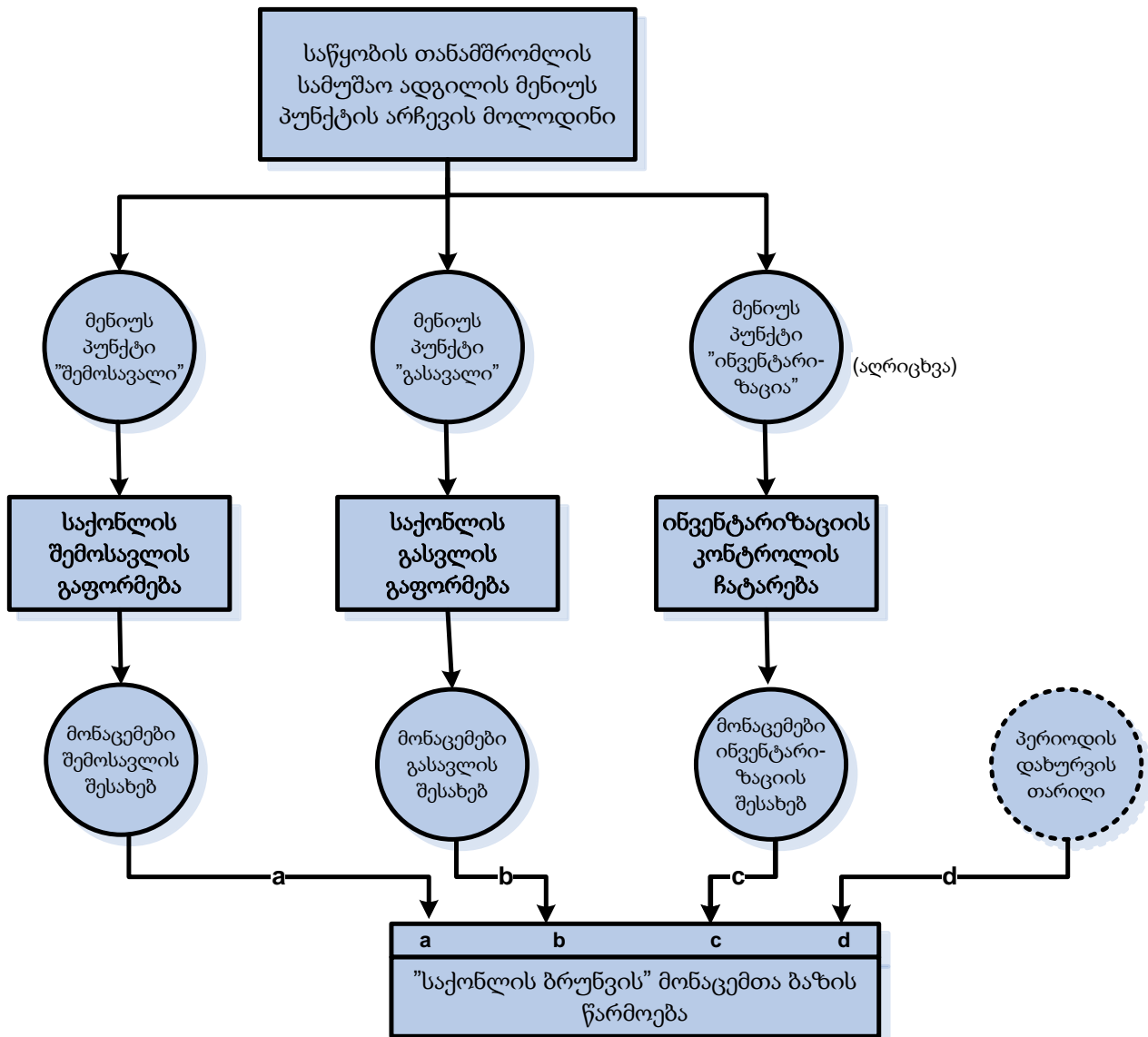
მდგომარეობათა გადასვლის დიაგრამების ობიექტების გამოსახვა

ობიექტი	იორდანის	ჰეინე-სარსონის	SADT	SAG
მდგომარეობა (processing step)				არა აქვს
საწყისი მდგომარეობა				არა აქვს
გადასვლა	<p>გადასვლის პირობა</p> <hr/> <p>გადასვლის მოქმედება</p>	<p>გადასვლის პირობა</p> <hr/> <p>გადასვლის მოქმედება</p>	<p>ა) </p> <p>ბ) </p> <p>ა) - პირობა მონაცემების მიხედვით ბ) - პირობა დროის მიხედვით</p>	არა აქვს

მდგომარეობათა გადასვლის დიაგრამის (STD) აგების დროს სასურველია ქვემოთ მოყვანილი წესები გავითვალისწინოთ:

- მდგომარეობათა გადასვლის დიაგრამის აგება მისი დეტალიზაციის უმაღლესი დონიდან იქნეს დაწყებული;
- რაც შეიძლება მარტივი, 4-6 მდგომარეობის მომცველი დიაგრამა აიგოს;
- მდგომარეობათა დასახელებების, ქმედებებისა და მოქმედებების იგივე წესები იქნეს გამოყენებული, რომელიც პროცესებისა და ნაკადების დასახელებებშია.

მაგალითი. საწყობში საქონლის ანალიტიკური აღრიცხვის ამოცანისათვის მდგომარეობათა გადასვლის დიაგრამის (STD) SADT ნოტაციაში აგება.



შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის TQM Diagram შაბლონის საშუალებითაა შესრულებული.

მდგომარეობათა გადასვლის დიაგრამის (STD) აგების ორი ხერხი გამოიყენება:

- პირველი ხერხი იმაში მდგომარეობს, რომ ჯერ სისტემის შესაძლო მდგომარეობები უნდა გაირკვეს და შემდეგ ერთი მდგომარეობიდან მეორეში გადასვლის გზები უნდა გამოვლინდეს;

- მეორე ხერხის დროს კი ჯერ საწყისი მდგომარეობა აიგება, ხოლო შემდეგ მომდევნო მდგომარეობაში გადასვლა ხორციელდება და ა. შ. (მიმდევრობითი გადასვლა).

ინფორმაციულ-ლოგიკური მოდელი ”არსი-კავშირი” (ER-დიაგრამა)

ერთ-ერთი ყველაზე პოპულარული მონაცემთა სემანტიკური მოდელი არის მოდელი „არსი-კავშირი“, რომელსაც ხშირად ER-მოდელს უწოდებენ, ინგლისური სიტყვების Entity (არსი) და Relation (კავშირი) პირველი ასოების მიხედვით.

მონაცემთა ბაზების (ძირითადად რელაციური) დაპროექტების უმეტესობა თანამედროვე საშუალებანი ER-მოდელის ნაირსახეობას იყენებს. საგნობრივი არის მოდელირება შედარებით მცირე რაოდენობის კომპონენტების მქონე გრაფიკული დიაგრამების გამოყენებას ეფუძნება. მონაცემთა ბაზის კონცეპტუალური სქემის თვალნათლივ წარმოდგენის მიზნით, ER-მოდელებმა CASE-საშუალებებში ფართო გავრცელება პოვა.

მონაცემთა სტრუქტურის მოდელირებისათვის ER-დიაგრამები (დიაგრამები „არსი-კავშირი“) გამოიყენება, რომლებიც არსებს შორის კავშირს უფრო ცხადი ფორმით წარმოადგენს. ER-დიაგრამებმა CASE-სისტემებში, რომლებიც რელაციურ მონაცემთა ბაზის ავტომატიზებულ დაპროექტებას უჭერს მხარს, ფართო გავრცელება პოვა. განსაკუთრებით ის დიაგრამებია გავრცელებული, რომლებსაც პოპულარული CASE-სისტემები იყენებს (კერძოდ, ERwin, Design/IDEF, Power Designer).

ინფორმაციულ-ლოგიკური მოდელი ”არსი-კავშირი” (ER-დიაგრამები) ისეთი მონაცემთა ბაზის დამუშავებაზეა ორიენტირებული, რომლის სტრუქტურა კონკრეტულ ინფორმაციულ მოთხოვნილებებზე არ არის დამოკიდებული და საშუალებას იძლევა მომხმარებლის ნებისმიერი მოთხოვნა დააკმაყოფილოს.

ERD-დიაგრამა ”არსი-კავშირი” წარმოადგენს მრავალი ობიექტის, მათი თვისებებისა და მათ შორის არსებული ურთიერთკავშირების ნაკრებს, რომლებიც შემდეგში დასაპროექტებელი სისტემის ფუნქციების მიერ გამოიყენება.

ER-დიაგრამების ძირითადი ცნებებია არსი, კავშირი და ატრიბუტი.

არსი – საერთო თვისებების (ატრიბუტების) რეალური და აბსტრაქტული ობიექტების სიმრავლეს წარმოადგენს. იგი რეალური ან ვირტუალური ობიექტია, რომელსაც განსახილველი საგნობრივი არის არსებითი მნიშვნელობა აქვს და რომელზეც ინფორმაცია შენახვას ექვემდებარება. თითოეულ არსს შემდეგი თვისებები უნდა ჰქონდეს:

- უნიკალური იდენტიფიკატორი;
- შეიცავდეს ერთ ან რამდენიმე ატრიბუტს, რომელიც ან ეკუთვნის არსს, ან სხვა არსებთან კავშირით, მემკვიდრეობით აქვს მიღებული;
- ატრიბუტების ერთობლიობას შეიცავს, რომლებიც არსის თითოეული ეგზემპლარის იდენტიფიცირებას ცალსახად ახდენს.

ნებისმიერ არსს სხვა არსებთან ნებისმიერი რაოდენობის კავშირები შეიძლება ჰქონდეს.

კავშირი (დამოკიდებულება) ორი არსის შეერთებაა, რომლის დროსაც, როგორც წესი, ერთი არსის თითოეული ეგზემპლარი, რომელსაც მშობლიური არსი ეწოდება, მეორე არსის ნებისმიერი (მათ შორის, ნულოვანი) რაოდენობის ეგზემპლარებთან ასოცირდება, რომელსაც არსი-შთამომავალი ეწოდება, ხოლო არსი-შთამომავალის თითოეული ეგზემპლარი არსი-მშობელის ზუსტად ერთ ეგზემპლართან ასოცირდება.

ატრიბუტი არსის მახასიათებელია, რომელიც განსახილველი საგნობრივი არისათვისაა მნიშვნელოვანი.

დამოუკიდებელი არსი – დამოუკიდებელ მონაცემებს წარმოადგენს, რომლებიც ყოველთვის არის სისტემაში.

დამოკიდებელი არსი – წარმოადგენს მონაცემებს, რომლებიც სხვა არსზეა არის დამოკიდებული.

ERD-ს ობიექტები სხვადასხვა მეთოდოლოგიებში, მე-8 ცხრილშია მოცემული.

ER-დიაგრამების აგება შემდეგ ეტაპებზე დაიყვანება:

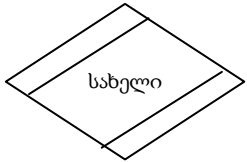
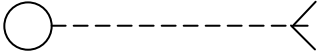
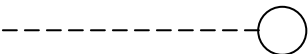
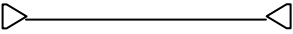
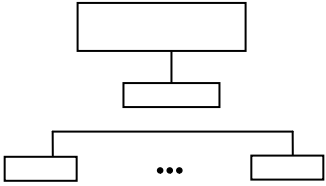
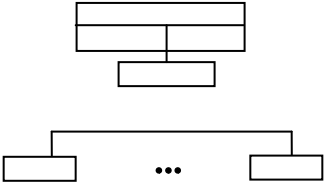
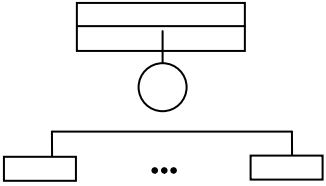
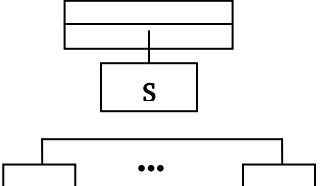
- უნდა მოხდეს ყველა არსის, მათი ატრიბუტებისა და აგრეთვე, პირველადი გასაღების იდენტიფიკაცია;

ERD-ს ობიექტები სხვადასხვა მეთოდოლოგებში

ობიექტი	ჩენა	ბერკერი	SADT	SAG
დამოუკიდებელი არსი				
დამოკიდებელი არსი				
დამოკიდებულება (კავშირი)				
დამოკიდებულების სიმძლავრე 1:1				
1:N				
N:N				
იერარქიული დამოკიდებულების მხარდაჭერის შესაძლებლობა	-	-	-	დიახ

163

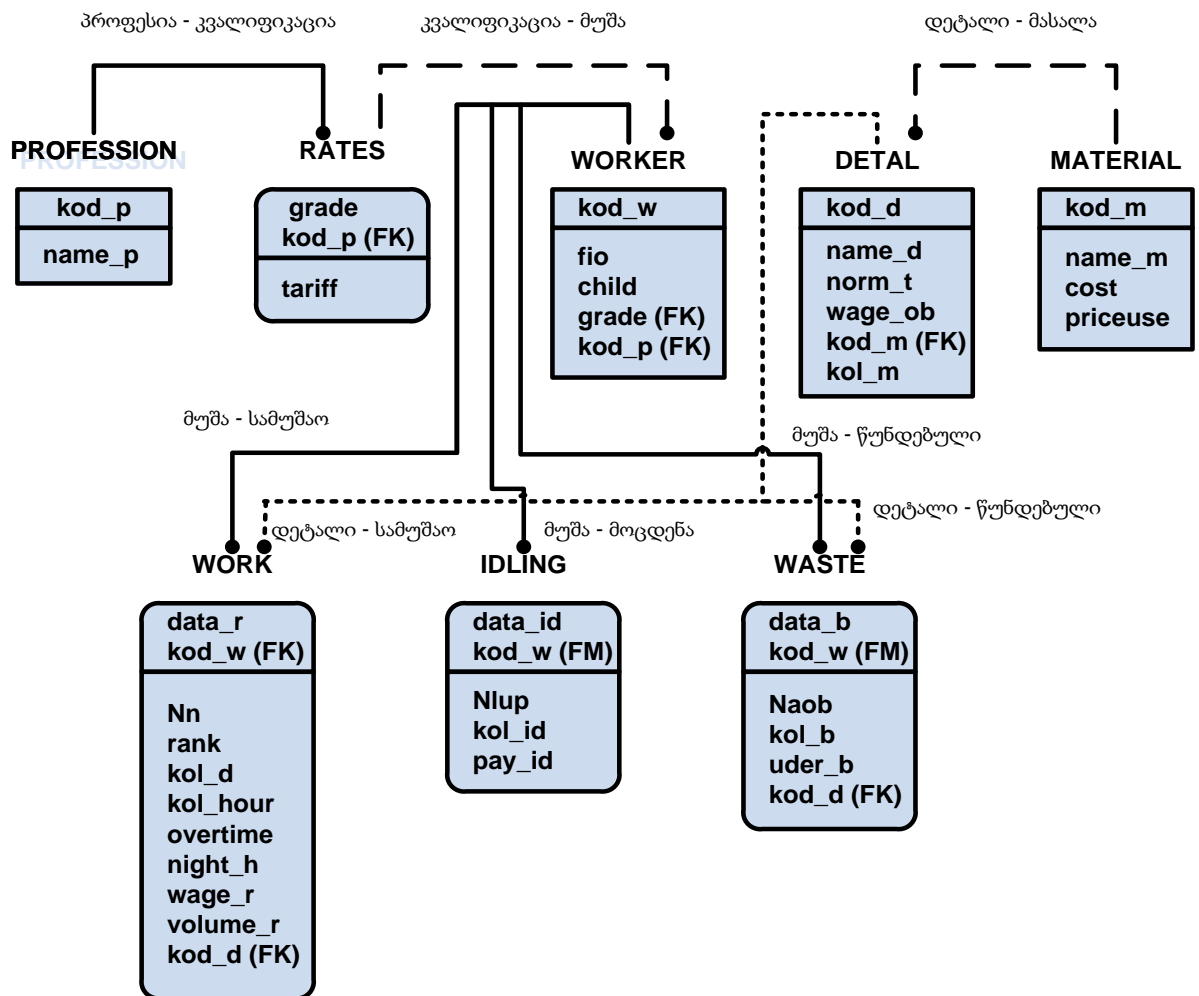
მე-8 ცხრილის გაგრძელება

ობიექტი	ჩენა	ბერკერი	SADT	SAG
არააუცილებელი დამოკიდებულება (შეზღუდვა)				
არსი კატეგორიზაცია				
საგასაღებო ატრიბუტების არსებობა	-	-	PK – პირველადი FK – მემკვიდრეობითი	i
სქემის გენერაცია	-	დიახ	დიახ	დიახ

- უნდა მოხდეს არსებს შორის დამოკიდებულების იდენტიფიკაცია და უნდა მიეთითოს ამ დამოკიდებულების სიმძლავრე;

- თუ მეორე ეტაპზე გამოვლინდა N:N დამოკიდებულება, ასეთი დამოკიდებულება არასპეციფიკურია და საჭიროა მათი გარდაქმნა 1:N ან 1:1 სიმძლავრის დამოკიდებულებად. როგორც წესი, ეს ახალი არსის დამატების საშუალებით კეთდება.

მაგალითი. შრომის აღრიცხვისა და ხელფასის ამოცანისათვის ER-დიაგრამა "არსი-კავშირი" SADT ნოტაციაში.



- wage_ob** – თვითღირებულება;
- cost** – ღირებულება;
- priceuse** – ფასი;
- rank** – რანგი (თანრიგი);
- overtime** – ნორმის ზემოთ დრო;
- wage_r** – ხელფასი;
- pay_id** – დასაქვითი თანხა.

შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის TQM Diagram შაბლონის საშუალებით არის შესრულებული.

პროგრამული დანართის სტრუქტურის დიაგრამა

პროგრამული დანართის სტრუქტურის დიაგრამა (SSD) ფუნქციისა და იმ პროგრამული მოდულების ურთიერთკავშირს იძლევა, რომლებიც მათ რეალიზებას ახდენს (მენიუ, ფორმები, ანგარიშები და სხვ.).

პროგრამული დანართის სტრუქტურის დიაგრამა (SSD) იმ პროგრამული მოდულების ურთიერთკავშირს წარმოადგენს, რომლებიც საინფორმაციო სისტემის რეალიზებას ახდენს. SSD წინა დიაგრამებში (BFD, DFD, STD, ERD) გამოხატულ სისტემურ მოთხოვნებსა და საინფორმაციო სისტემის რეალიზაციას შორის ხიდის ფუნქციას ასრულებს.

SSD-ს ობიექტები სხვადასხვა მეთოდოლოგიებში, მე-9 ცხრილშია მოცემული.

პროგრამული დანართის სტრუქტურის დიაგრამის (SSD) აგების ეტაპებია:

- ბიზნეს-ფუნქციის დიაგრამაში უნდა გამოიყოს ფუნქცია, რომლის რეალიზება პროგრამული სახით მოხდება;

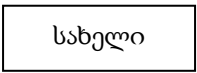
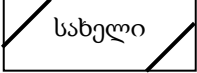
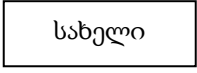
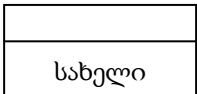

- ავილოთ მონაცემთა ნაკადის დიაგრამა (DFD) გამოყოფილი ფუნქციებისა და ქვეფუნქციებისათვის, შემაჯავალი და გამომავალი მონაცემების ნაკადის მიხედვით, მისი ანალიზი განხორციელდეს;

- ER-დიაგრამის არსთა ატრიბუტების სიის საშუალებით განისაზღვროს მონაცემთა ნაკადის სტრუქტურა;

- მდგომარეობათა გადასვლის დიაგრამაზე განისაზღვროს მდგომარეობა, მისი გამომწვევი გადასვლები და ქმედებები, რომლებიც ბიზნეს-ფუნქციის რეალიზებას ახდენს;

- თითოეული მდგომარეობის პროგრამული რეალიზაცია სხვა ენაზე დაწერილი CASE-სისტემის ბიბლიოთეკური მოდულის ან მოდულის სახით განხორციელდეს;

SSD-ს ძირითადი ობიექტები სხვადასხვა მეთოდოლოგიებში

ობიექტი	კონსტატინის	ჯექსონის	SADT	SAG
მოდელი		არა აქვს	_____	 სისტემური ფუნქცია
ბიბლიოთეკის მოდული		არა აქვს	_____	
მოდულის გამოძახება	→	არა აქვს	→	→
კავშირი მონაცემების მიხედვით	 →	_____	_____	→ + შესაბამისი სექციის აღწერა
ბიზნეს- ფუნქციასთან კავშირი	→	_____	ICOM-ჭდეების მეშვეობით მემკვიდრეობითობის მექანიზმის გამოყენებით	მემკვიდრეობი- თობის საშუალებით

- თითოეული გამოყოფილი ფუნქციისათვის დაიხაზოს სისტემური სტრუქტურის დიაგრამა და შემდგომ ბიზნეს-ფუნქციის დიაგრამიდან გამომდინარე, მათი გაერთიანება მოხდეს;

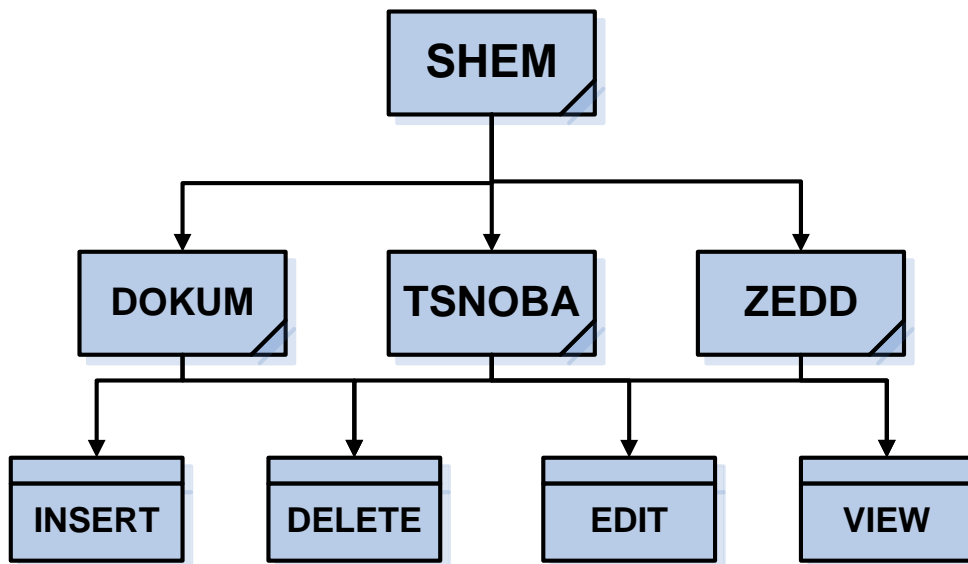
- თუ CASE-საშუალებებით შესაძლებელია აგებული სისტემური სტრუქტურული დიაგრამის გაკონტროლება და თუ მასში შეცდომები არ აღმოჩნდა, მაშინ სისტემური სტრუქტურული დიაგრამის საფუძველზე, პროგრამული ინტერფეისის მაკეტის შედგენაზე უნდა გადავიდეთ;

- თითოეული მოდულისათვის არსებული ბიბლიოთეკიდან ინტერფეისის შაბლონების შერჩევა აუცილებლად უნდა მოხდეს ან ახალი შაბლონის კონსტრუირება უნდა განხორციელდეს, ან პროგრამული მოდული არსებულ პროგრამულ ენაზე დაიწეროს.

შემდეგ, ინტერფეისის გათვალისწინებით უნდა განისაზღვროს სისტემური სტრუქტურული დიაგრამის ყველა ელემენტი და ER-მოდელის ცხრილებთან მათი კავშირი დადგინდეს. ყოველივე ზემოთ აღნიშნულის

შემდეგ, ხორციელდება მოდულების გენერაცია. გენერაცია ორი სახისაა: სრული და არასრული. გენერაციის შედეგს იმ პროგრამული დანართების მოდული წარმოადგენს, რომელიც საინფორმაციო სისტემის რეალიზაციას ახდენს.

მაგალითი. საწყობში საქონლის მიღების ანალიტიკური აღრიცხვის ამოცანისათვის პროგრამული დანართის სტრუქტურის დიაგრამა (SSD) SAG ნოტაციაში.



შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის Audit Diagram შაბლონის საშუალებით არის შესრულებული.

საინფორმაციო სისტემების ობიექტზე ორიენტირებული დაპროექტება

საინფორმაციო სისტემების სტრუქტურული დეკომპოზიცია ობიექტზე ორიენტირებული მიდგომის საფუძველზე, სტრუქტურაზე ორიენტირებული მიდგომისაგან შექმნილ სიტუაციიდან გამომდინარე სისტემის დინამიკური ქცევის გამოსახვის უკეთესი საშუალებებით განსხვავდება. ამ დროს, პრობლემური სფეროს მოდელი, როგორც დროში ურთიერთდაკავშირებული ობიექტების ერთობლიობა განიხილება. მაშინ ინფორმაციის დამუშავების კონკრეტული პროცესი მიმდევრობით ურთიერთმოქმედი ობიექტების სახით ფორმირდება. მონაცემთა დამუშავების ერთი ოპერაცია შეიძლება განხილულ იქნეს, როგორც ობიექტების ერთი ურთიერთქმედების შედეგი.

ობიექტზე ორიენტირებული მოდელების სისტემა საინფორმაციო სისტემის ფუნქციური წარმოდგენიდან ობიექტების დინამიკური ურთიერთქმედების მოდელის მიმართულებით მიმდევრობით მიემართება, რის საფუძველზეც კონკრეტულ პროგრამულ და ტექნიკურ გარემოში ობიექტთა კლასების გენერირება შეიძლება განხორციელდეს.

ამჟამად, ობიექტზე ორიენტირებული მოდელირების დროს ფართოდ გამოიყენება მოდელირების უნიფიცირებული ენა UML (Unified Modeling Language), რომელიც მოწინავე კომპიუტერული ფირმების OMG (Object Management Group) მიერაა დამუშავებული და ფაქტობრივად ობიექტზე ორიენტირებული ტექნოლოგიებისათვის სტანდარტს წარმოადგენს. UML ენა CASE-ტექნოლოგიების ფარგლებში მრავალი პროგრამული უზრუნველყოფის მწარმოებელი ფირმის მიერ არის რეალიზებული, მაგალითად, Rational Rose (Rational), Natural Engineering Workbench (Software AG), ARIS Toolset (IDS prof. Scheer) და სხვ.

ობიექტზე ორიენტირებული მოდელების სისტემა UML ნოტაციების შესაბამისად, შემდეგ დიაგრამებს მოიცავს:

1. გამოყენების პრეცედენტების დიაგრამა (Use-case diagram), რომელიც საინფორმაციო სისტემის ფუნქციურობას შესრულებადი ტრანზაქციის მიმდევრობის ერთობლიობის სახით ასახავს;

2. ობიექტთა კლასის დიაგრამა (Class diagram), რომელიც ურთიერთ-დაკავშირებული ობიექტთა კლასების ერთობლიობის სტრუქტურას ER-დიაგრამის ანალოგიურად ასახავს;

3. მდგომარეობის დიაგრამა (Statechart diagram), რომელთაგან თითოეული ერთი კლასის ობიექტებისა და მათთან დაკავშირებულ ქმედებათა მდგომარეობის დინამიკას გამოხატავს;

4. ობიექტთა ურთიერთქმედების დიაგრამა (Interaction diagram), თითოეული მათგანი გამოყენების ერთი პრეცედენტის ფარგლებში, ობიექტთა დინამიკური ურთიერთქმედებას გამოხატავს;

5. მოლვაწობის დიაგრამა (Activity diagram), რომელიც გამოყენების ურთიერთდაკავშირებულ პრეცედენტებში სამუშაოს ნაკადს გამოსახავს (მათი დეკომპოზიცია შეიძლება უფრო დეტალურ დიაგრამებად განხორციელდეს);

6. პაკეტების დიაგრამა (Package diagram), რომელიც ობიექტთა განაწილებას ფუნქციური და უზრუნველყოფის ქვესისტემებად გამოსახავს (მათი დეკომპოზიცია შეიძლება უფრო დეტალურ დიაგრამებად განხორციელდეს);

7. კომპონენტთა დიაგრამა (Component diagram), რომელიც პროგრამული კოდის ფიზიკურ მოდულებს ასახავს;

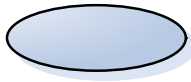
8. განთავსების დიაგრამა (Deployment diagram), რომელიც ობიექტთა განაწილებას გამოთვლითი ქსელის კვანძებში ასახავს.

გამოყენების პრეცედენტის დიაგრამა

გამოყენების პრეცედენტების დიაგრამა ძირითად ბიზნეს-პროცესებს ავლენს, როგორც ტრანზაქციის მიმდევრობებს, რომლებიც მთლიანად უნდა შესრულდეს, როდესაც ქმედების განკერძოებული ქვესიმრავლის შესრულებას მთელი მიმდევრობის შესრულების გარეშე არა აქვს მნიშვნელობა. გამოყენების პრეცედენტის ინიცირება საინფორმაციო სისტემის იმ გარე მომხმარებელთა გარემოდან ხდება, რომლებსაც *აქტიორი* ეწოდება. მოდელირების ამ ეტაპზე, პროცესების რეალიზაციის მექანიზმის გახსნა არ ხდება. წარმოდგენილ არსს შემდეგი გრაფიკული აღნიშვნები აქვს:



აქტიორი – პროცესის გარე მომხმარებელი



გამოყენების პრეცედენტი (ბიზნეს-პროცესი)

აქტიორი გამოყენების პრეცედენტის შესრულებას ახდენს და მისგან შედეგს ღებულობს. ურთიერთქმედება აქტიორსა და გამოყენების პრეცედენტს შორის ქმედების შედეგად ხორციელდება და აღინიშნება ისრით, რომელზეც ქმედება ან ინფორმაციის სახეობაა მითითებული. ერთი აქტიორი შეიძლება რამდენიმე გამოყენების პრეცედენტში მონაწილეობდეს, ხოლო ერთ გამოყენების პრეცედენტში შეიძლება რამდენიმე აქტიორი მონაწილეობდეს.

გამოყენების პრეცედენტის რეალიზებაში შეიძლება ქმედებათა რამდენიმე ნაკადი იქნეს გამოყოფილი:

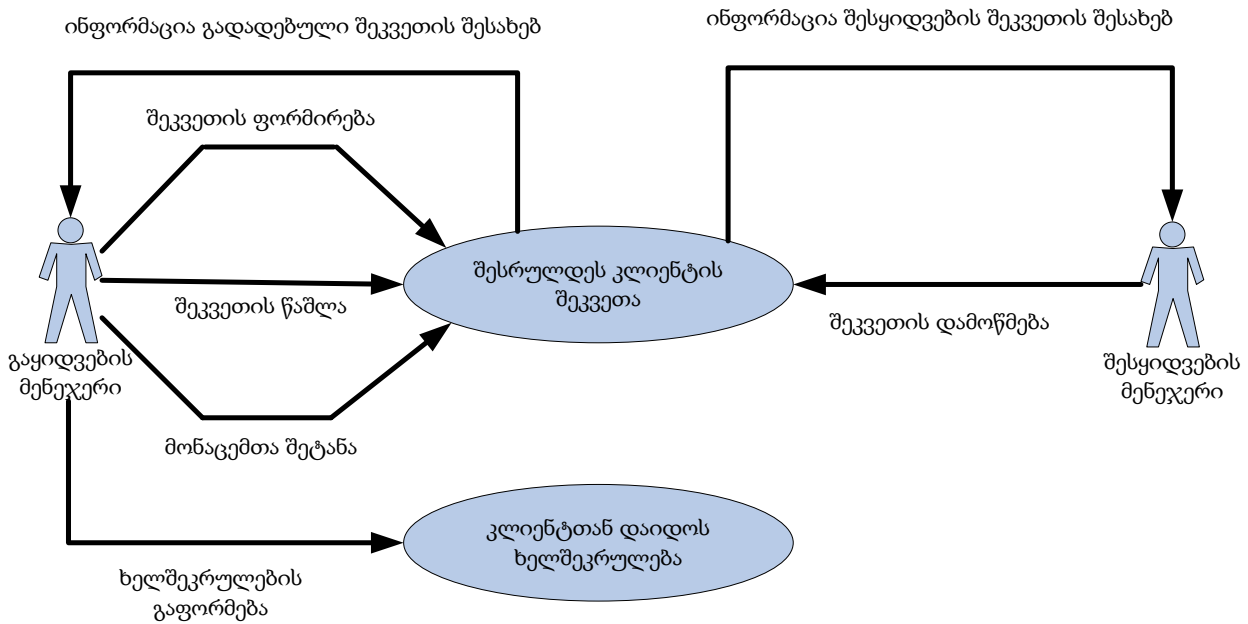
- ქმედებათა ძირითადი ნაკადი, რომელსაც საჭირო შედეგამდე უმოკლესი გზით მივყავართ;

- ქმედებათა ალტერნატიული ნაკადები.

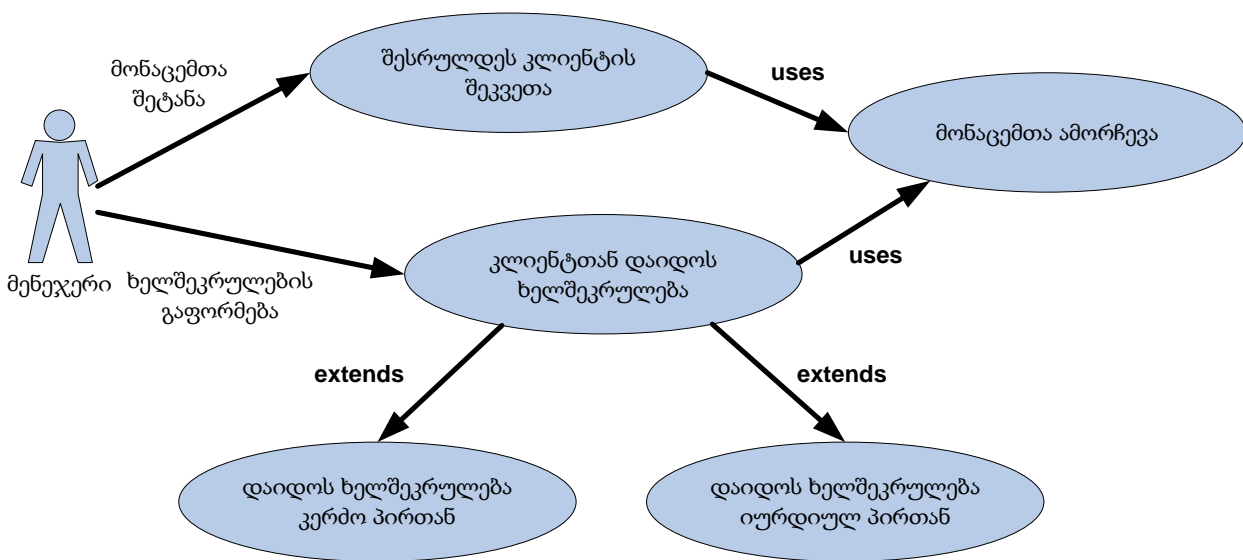
ქმედებათა ძირითადი და ალტერნატიული ნაკადები გამოყენების პრეცედენტის მოდელში არაფორმალური ტექსტური კომენტარის სახით აღიწერება.

რამდენიმე გამოყენების პრეცედენტს, გამოყოფილი დამოუკიდებელი გამოყენების პრეცედენტის სახით შეიძლება ის საერთო ნაწილი ჰქონდეს, რომელთანაც გამოყენების დამოკიდებულება (uses) მყარდება. მეორე მხრივ, ზოგიერთი გამოყენების პრეცედენტი შეიძლება დეტალებით იქნეს გაფართოებული. ამ შემთხვევაში იქმნება დამატებითი გამოყენების პრეცედენტი, რომელთანაც გაფართოების თანაფარდობა (extends) მყარდება.

მაგალითი. კლიენტის შეკვეთის შესრულების ამოცანისათვის გამოყენების პრეცედენტების დიაგრამა:



გამოყენებისა და გაფართოების დამოკიდებულების მაგალითი:

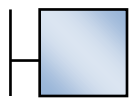


შენიშვნა. ზემოთ მოყვანილ მაგალითებში დიაგრამები Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Block Diagram კატეგორიის Basic Diagram შაბლონის საშუალებითაა შესრულებული.

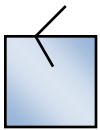
ობიექტთა კლასის დიაგრამა

ობიექტთა კლასის დიაგრამა (Class diagram), ობიექტთა კლასის სტატიკურ სტრუქტურას ასახავს. ეს დიაგრამა პრობლემური სფეროს შიგა სტრუქტურას, ობიექტთა კლასის იერარქიას, ობიექტთა სტატიკურ კავშირებს განიხილავს.

ობიექტთა კლასს შეიძლება ქცევის სხვადასხვანაირი სტერეოტიპები ჰქონდეს: ობიექტი-არსი, მმართველი ობიექტები, ინტერფეისული ობიექტები:



ინტერფეისული ობიექტი (Interface Object) – აქტიური ობიექტი, ის აქტიური ობიექტია, რომელიც საინფორმაციო სისტემის მომხმარებელთან ურთიერთობის ფორმას (ეკრანული ფორმა, მენიუ, საბრძანებო სტრიქონი, დილაკი) განსაზღვრავს



მმართველი ობიექტი (Control Object) – აქტიური ობიექტი, რომელიც ფუნქციის შესრულების კოორდინაციას ახორციელებს



არსი (Entity Object) – პასიური ობიექტი, რომელზეც პროცესის დამუშავების ოპერაცია ხორციელდება

შენიშვნა. მართკუთხედებში ზედა ნაწილში უნდა მიეთითოს ობიექტთა კლასის სახელი, შუა ნაწილში – ატრიბუტების სახელი, ხოლო ქვედა ნაწილში – მეთოდების სახელი.

ობიექტთა კლასის დიაგრამაში გამოსახული ობიექტები ერთმანეთს იმ სტატიკურ დამოკიდებულებით უკავშირდებიან, რომლებიც კონკრეტული ბიზნეს-პროცესისაგან დამოუკიდებლად ობიექტებს შორის მუდმივ კავშირს ასახავს. სტატიკურ დამოკიდებულებას მიეკუთვნება ობიექტების განზოგადება, აგრეგაცია, ასოციაცია:

0..1 * ასოციაციის დამოკიდებულება 0..1:1; 0..1:M; M:N (მასზე შეიძლება იყოს გაკეთებული წარწერა)
0..1 – კავშირის არააუცილებლობა;
* – მრავალჯერადობა.

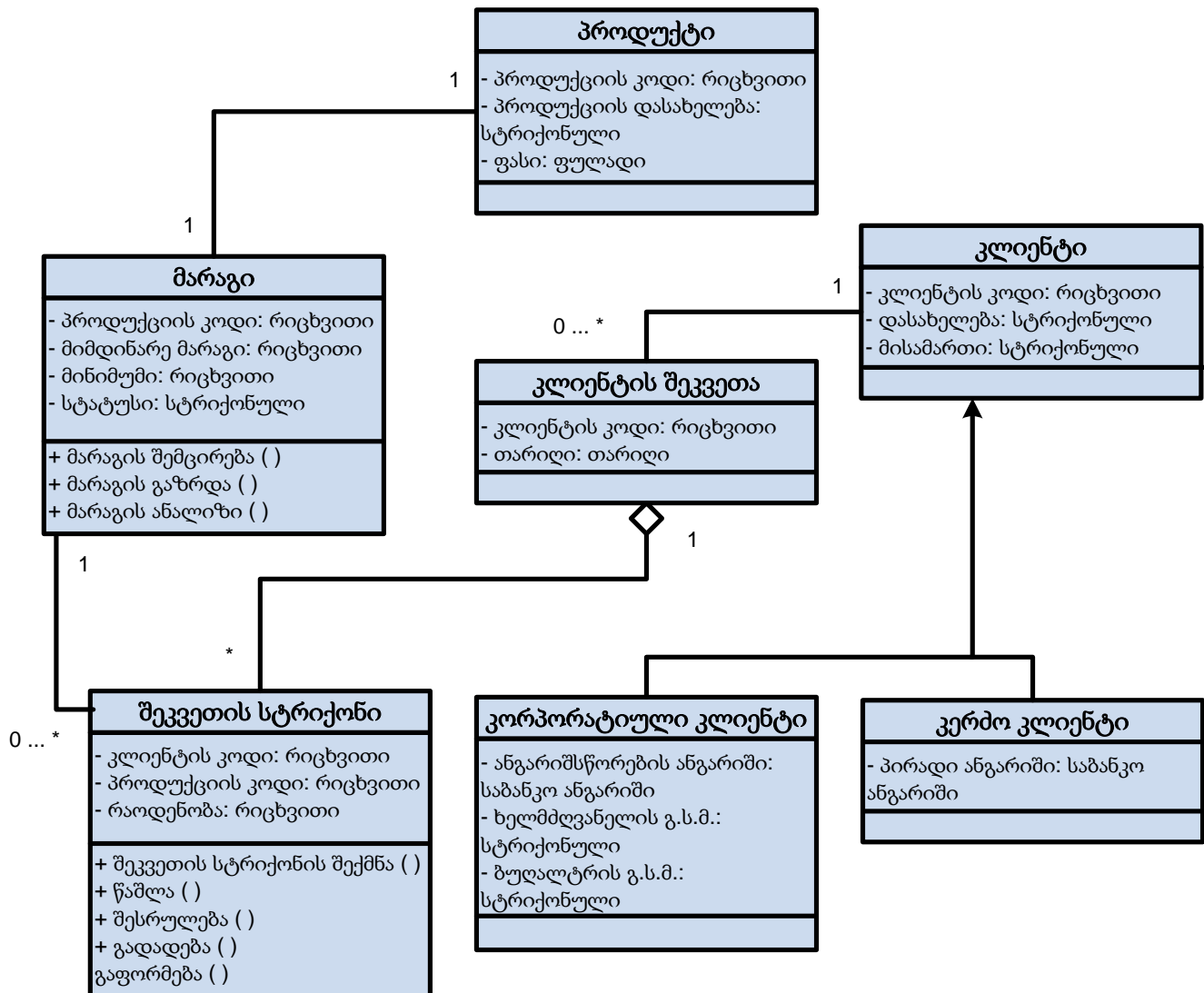
→ განზოგადების დამოკიდებულება (მემკვიდრეობითობა)

◊ აგრეგაციის დამოკიდებულება (მთელი – ნაწილი)

ქვემოთ, სტატიკური დამოკიდებულების მაგალითია მოყვანილი.

მართკუთხედის ზედა ნაწილში ობიექტთა კლასების, შუა ნაწილში ატრიბუტთა, ხოლო ქვედა ნაწილში მეთოდების დასახელებებია მოცემული.

მაგალითი. კლიენტის შეკვეთის შესრულების ამოცანისათვის ობიექტთა კლასის დიაგრამის ფრაგმენტი.



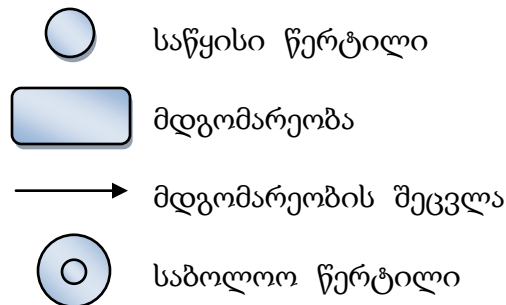
შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Flowchart კატეგორიის SDL Diagram შაბლონის საშუალებით არის შესრულებული.

მდგომარეობის დიაგრამა

მდგომარეობის დიაგრამა (Statechart diagram) ასახავს ერთი კლასის ობიექტების დინამიკაში ქცევას, ობიექტის მდგომარეობის ქმედებებთან კავშირს და განსაზღვრავს:

- როგორ ტიპურ მდგომარეობებს გაივლის ობიექტი;
- რომელი ქმედებები იწვევს ობიექტის მდგომარეობის ცვლილებებს;
- რა მოქმედებებს ასრულებს ობიექტი, როდესაც ის ღებულობს შეტყობინებას მდგომარეობის შეცვლის შესახებ;
- როგორ იქმნება და ნადგურდება ობიექტი (დიაგრამის საწყისი და საბოლოო წერტილები).

მდგომარეობის დიაგრამაში შემდეგი აღნიშვნები გამოიყენება:



საწყისი წერტილი განსაზღვრავს ქმედებას, რომელიც ობიექტის საწყისი მდგომარეობის დროს წარმოიქმნება. ობიექტის სხვა მდგომარეობიდან საწყის წერტილზე გადასვლა არ შეიძლება.

საბოლოო წერტილი ობიექტის არსებობის დასასრულს განსაზღვრავს. საბოლოო წერტილიდან ობიექტის სხვა მდგომარეობაში გადასვლა არ ხდება.

მდგომარეობა წარმოადგენს სიტუაციას, რომლის დროსაც ობიექტი სტაციონარულ მდგომარეობაში იმყოფება ან უწყვეტი ქმედება სრულდება. მდგომარეობა განისაზღვრება, როგორც ატრიბუტებისა და დამოკიდებულებების მნიშვნელობების ნაკრები. თითოეულ მდგომარეობასთან ერთი ან მეტი ქმედებაა დაკავშირებული, რომლებსაც მათი შეცვლა შეუძლიათ.

მდგომარეობის შეცვლა ობიექტის მდგომარეობაში ცვლილებებს განსაზღვრავს იმ ქმედების შედეგად, რომელიც მოცემულ მდგომარეობაში მყოფ ობიექტში წარმოიშობა. ყოველ მდგომარეობის შეცვლას, უნიკალური სახელი უნდა ჰქონდეს.

მდგომარეობის შეცვლის აღწერა შემდეგი ატრიბუტებით ხდება:

დანიშნულება – ობიექტის მდგომარეობაა, რომელშიც ობიექტი მდგომარეობის შეცვლის შემდეგ გადავა.

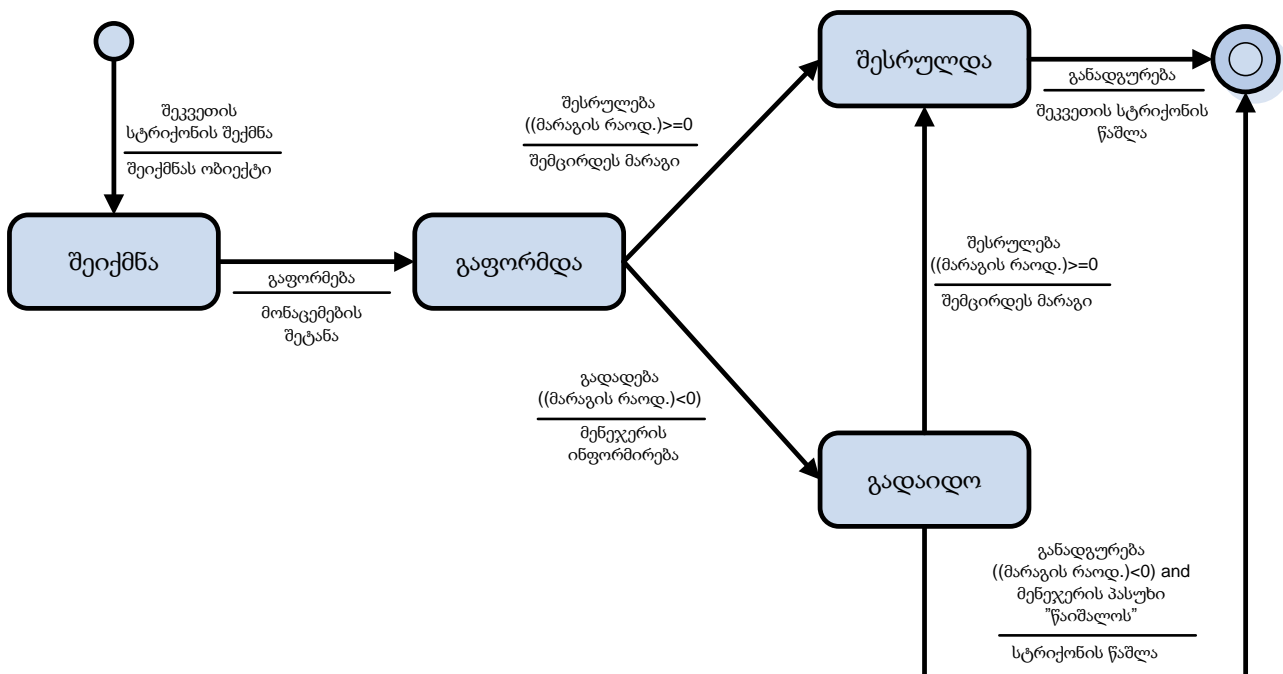
გადასვლის გამომწვევი მიზეზი – ქმედების სახელი, რომელიც მდგომარეობის შეცვლას იწვევს. გადასვლის გამომწვევი ქმედება შეიძლება იყოს გარე, რომელიც აქტიორის მიერ ხორციელდება, ან შიგა, რომელიც სხვა ობიექტების ქცევასთანაა დაკავშირებული, ან დროებითი, რომელიც დროის მოცემული ინტერვალის გასვლასთან არის დაკავშირებული.

გადასვლის პირობა – ლოგიკური გამოსახულება, რომელიც მდგომარეობის შეცვლის ასარჩევად უნდა შემოწმდეს. გადასვლის პირობა ერთი საწყისი მდგომარეობისათვის ურთიერთგამომრიცხავი უნდა იყოს.

ქმედება – ინფორმაციულად აღწერილი ქმედების არსი, რომელიც მდგომარეობის შეცვლის დროს უნდა შესრულდეს.

მდგომარეობის შეცვლა გრაფიკულად გამოსახება წრფით, რომელზეც ერთ-ერთი შემდეგი ატრიბუტი მაინც მიეთითება: გადასვლის გამომწვევი მიზეზი, გადასვლის პირობა და ქმედება.

მაგალითი. ობიექტისათვის "შეკვეთის სტრიქონის" მდგომარეობის დიაგრამა.



შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის TQM Diagram შაბლონის საშუალებით არის შესრულებული.

ობიექტთა ურთიერთქმედების დიაგრამა

ობიექტთა ურთიერთქმედების დიაგრამა (**Interaction diagram**) გამოყენების თითოეული პრეცედენტისათვის შემდეგი ორი ფორმით შეიძლება იქნეს აგებული:

- მიმდევრობითი დიაგრამის (**sequence diagram**) სახით;
- კოოპერატიული დიაგრამის (**collaboration diagram**) სახით.

მიმდევრობით დიაგრამაში ობიექტთა ურთიერთქმედება იმ ობიექტებს შორის ისრების საშუალებით გამოისახება, რომელიც ერთი ობიექტიდან მეორემდე მოქმედებას ან შეტყობინებას შეესაბამება. ისრის ნომერი მიმდევრობაში მოქმედების ნომერს შეესაბამება.

კოოპერატიული მოქმედების დიაგრამა ცხრილის სახით, შემდეგი წესების გათვალისწინებით იქმნება:

1. ცხრილის სვეტებში პრეცედენტის გამოყენების რეალიზაციაში მონაწილე ყველა ტიპის ობიექტი მიეთითება;

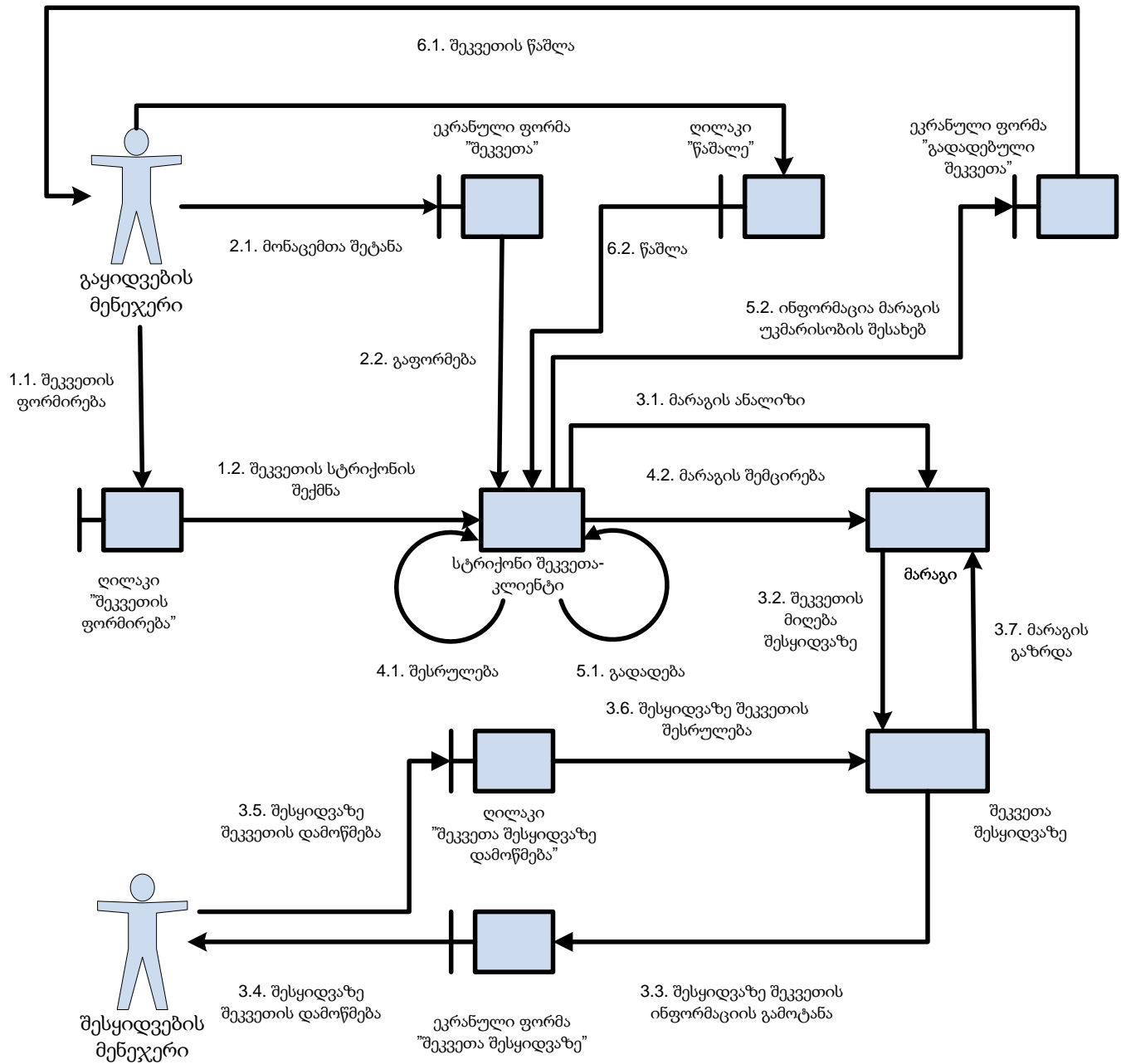
2. ისრები წარწერით ჰორიზონტალურად ისმება, რომლებიც ერთი ოპერაციის ფარგლებში ობიექტების ურთიერთქმედებას გამოხატავს. ეს ისარი უჩვენებს, რომ პირველი ობიექტი ოპერაციის შესრულების ფარგლებში, მეორე ობიექტს მოქმედების შესრულების აუცილებლობის შესახებ შეტყობინებას უგზავნის. შეტყობინების მიღებისთანავე მეორე ობიექტი მოქმედებას ასრულებს;

3. სტრიქონებისა და სვეტის გადაკვეთაზე ვერტიკალურად პირობითად აისახება დროის მონაკვეთი, რომლის განმავლობაში ობიექტზე სრულდება მოქმედება.

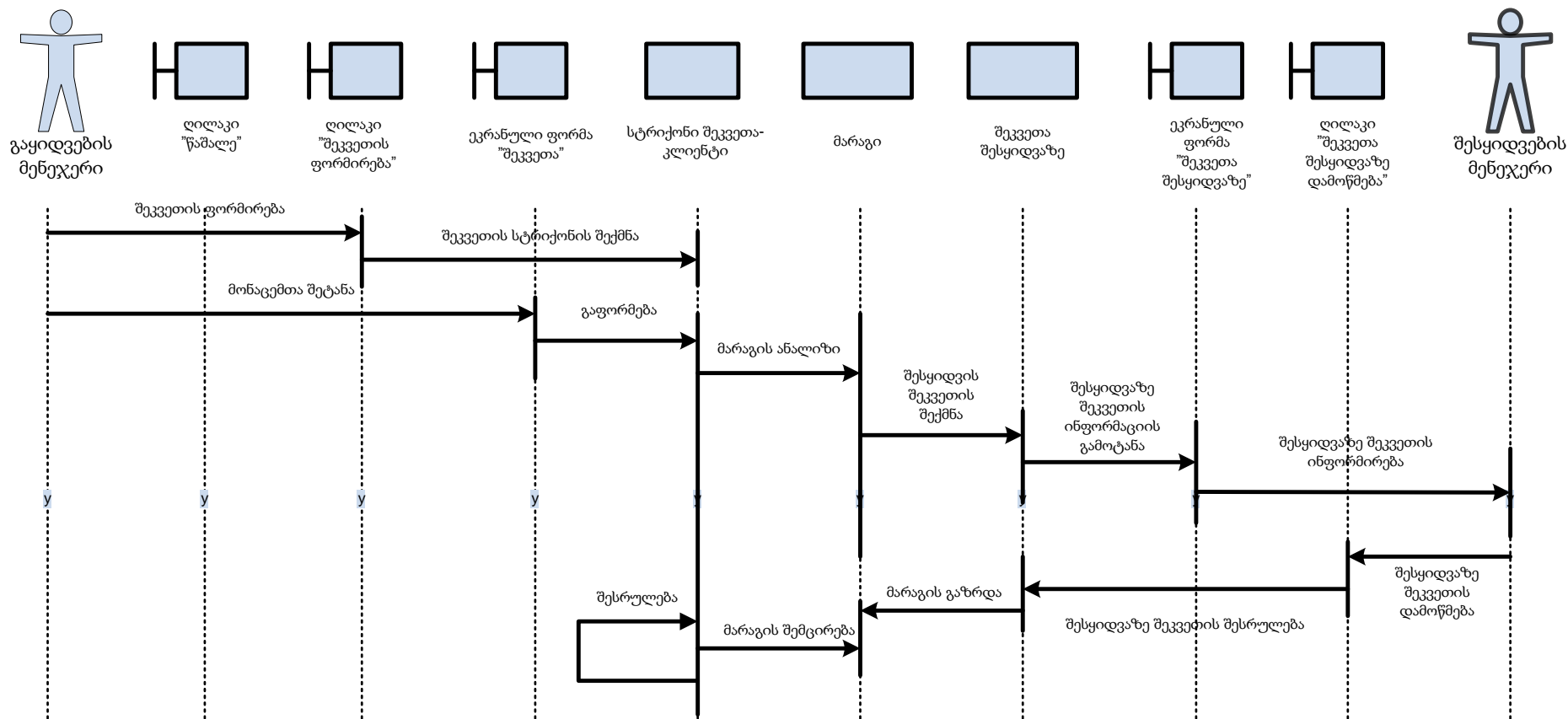
ობიექტთა ურთიერთქმედების დიაგრამა ოპერაციათა შესრულების მიმდევრობას, განშტოების, ციკლური გამეორების, მოქმედებათა პარალელობასა და სხვა ნაწილში, დეტალურად არ ასახავს.

მაგალითი. "შესრულდეს კლიენტის შეკვეთა" პრეცედენტისათვის
 ობიექტთა ურთიერთქმედების დიაგრამა მიმდევრობითი ფორმის სახით:

5.3. ინფორმირებულობა გადაღებული შეკვეთის შესახებ



მაგალითი. "შესრულდეს კლიენტის შეკვეთა" პრეცედენტისათვის ობიექტთა ურთიერთქმედების დიაგრამა კოოპერატიული მოქმედების ფორმის სახით:



179

შენიშვნა. ზემოთ მოყვანილ მაგალითებში დიაგრამები Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის Data Flow Diagram და Work Flow Diagram შაბლონების გამოყენებითაა შესრულებული.

მოღვაწეობის დიაგრამა

მოღვაწეობის დიაგრამა (Activity diagram) საშუალებას იძლევა ობიექტთა ურთიერთქმედების დიაგრამის ნაკლოვანებანი გამოვასწოროთ. მოღვაწეობაში იგულისხმება რაიმე სამუშაო, რომლის დეკომპოზიციაც მოქმედებათა ერთობლიობის სახით არის შესაძლებელი. მოღვაწეობის დიაგრამას შეუძლია ობიექტების ურთიერთქმედება გამოყენების რამდენიმე პრეცედენტის საშუალებით, კერძოდ, ობიექტთა დამუშავების რეალიზაციის სტანდარტული და ალტერნატიული გზები ცალ-ცალკე გამოსახოს.

ერთი მოღვაწეობის შესაბამის ბლოკს შეუძლია რამდენიმე ქმედება გამოსახოს და შეიძლება მისი დეკომპოზიცია მოხდეს. ქვემოთ, მოღვაწეობის დიაგრამაში გამოყენებული ცნებები და მათი გრაფიკული აღნიშვნებია მოყვანილი:



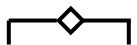
მოღვაწეობა (activity)



ნაკადი მოღვაწეობიდან მოღვაწეობამდე



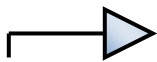
ნაკადის დაყოფა პარალელურ და თავისუფალ შესრულებად მოღვაწეობებად



ამოხსნა (გადაწყვეტა)



სინქრონიზაცია

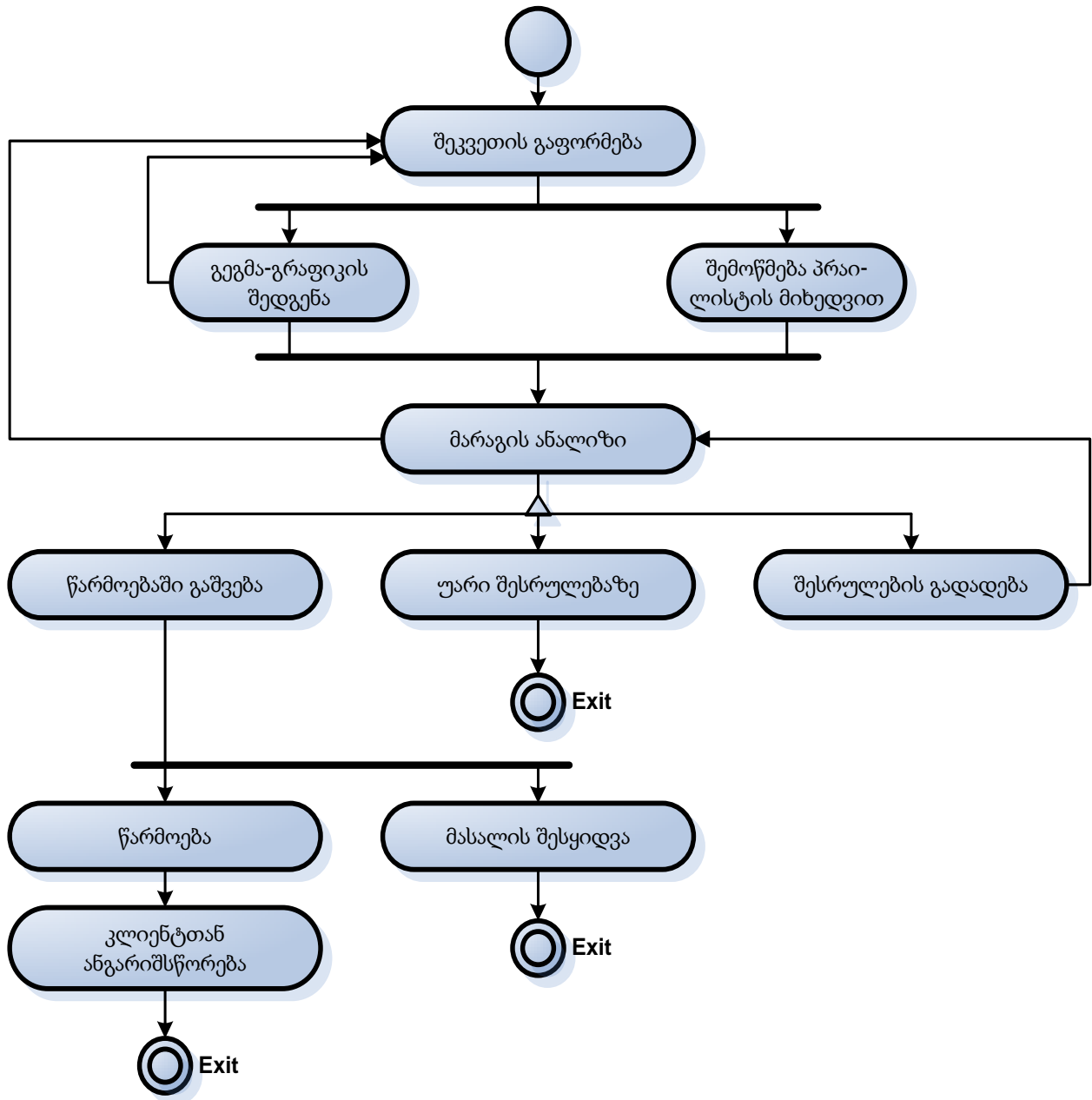


იტერაცია



Exit დასასრული

მაგალითი. კლიენტის შეკვეთის შესრულების პროცესის ამოცანის, მოღვაწეობის დიაგრამა.



შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის Basic Flowchart შაბლონის საშუალებითაა შესრულებული.

პაკეტების დიაგრამა

ობიექტზე ორიენტირებულ მიდგომაში პაკეტი ურთიერთდაკავშირებულ ობიექტთა კლასების სიმრავლეს მოიცავს. გამოყენების ერთმა პრეცედენტმა შეიძლება ობიექტთა კლასი სხვადასხვა პაკეტიდან მოითხოვოს. ჩვეულებრივ, ობიექტთა კლასი ერთ პაკეტს ენიშნება, მაგრამ სხვადასხვა მიზნების მიღწევის პოზიციიდან, შეიძლება სხვადასხვა პაკეტის შემადგენლობაში შედიოდეს.

ობიექტთა კლასის დაჯგუფების პაკეტური ტექნოლოგია საშუალებას იძლევა გავამარტივოთ:

- საინფორმაციო სისტემის დამუშავება და ექსპლუატაცია;
- ტიპური კომპონენტების, მათი განმეორებითი გამოყენების პოზიციიდან, მოქნილი ადაპტაცია;
- საინფორმაციო სისტემის მომხმარებელ-სერვერის არქიტექტურის ოპტიმიზაცია.

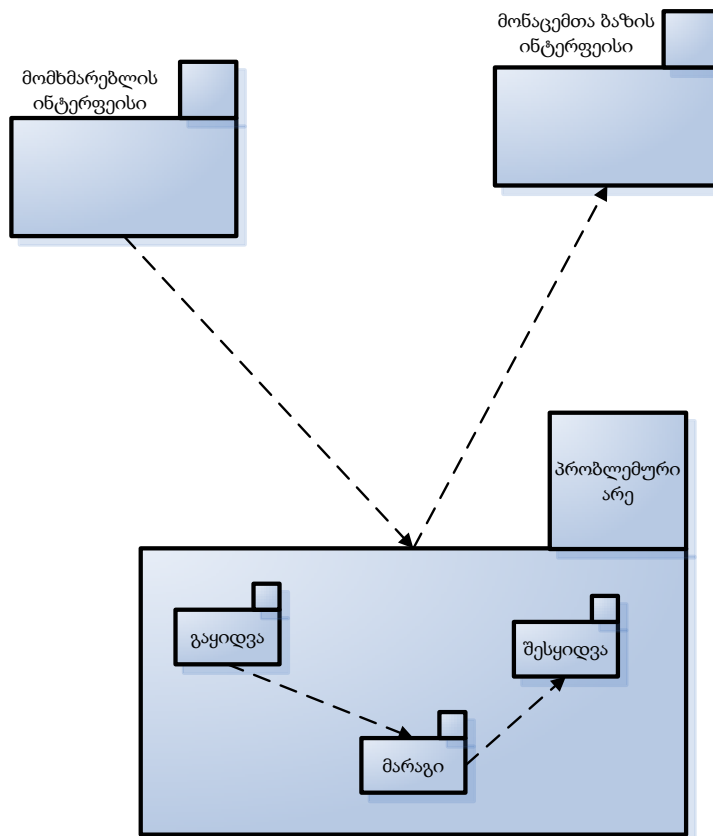
საინფორმაციო სისტემებს ჩვეულებრივ, ფუნქციურ და უზრუნველყოფის პაკეტებად ყოფენ. ფუნქციური პაკეტები, რომლებიც გადასაწყვეტ პრობლემებს (ამოცანებს) შეესაბამება, ერთ საერთო პაკეტში - "პრობლემური არე", ერთიანდება. თითოეული პაკეტი, თავის მხრივ, შეიძლება ქვეპაკეტებად დაიყოს. ობიექტთა კლასი, რომელთაც რამდენიმე ქვესისტემა ესაჭიროება, დამოუკიდებელი პაკეტის სახით გამოიყოფა. ერთ პაკეტში, როგორც წესი, არა უმეტეს 20-სა, ჩვეულებრივ 5-15 კომპონენტი ნაწილდება.

უზრუნველყოფის თვალსაზრისით, საინფორმაციო სისტემებს ხუთ ძირითად პაკეტად ყოფენ:

- "ინტერფეისი", რომლის ობიექტები ინფორმაციის შეტანა-გამოტანისა და ქვესისტემებს შორის შეტყობინებების მიმოცვლის საკითხებში მომხმარებლის საინფორმაციო სისტემასთან ურთიერთქმედების ფუნქციის რეალიზებას ახდენს;
- "მონაცემთა ბაზა", რომლის ობიექტები გარე მეხსიერებაში არსებულ მონაცემებზე მიმართვას ასრულებს;
- "ამოცანების მართვა", რომლის ობიექტები ობიექტთა დამუშავების დისპეტჩერიზაციისა და მარშრუტიზაციის ფუნქციას ასრულებს;
- "უტილიტები", რომელთა ობიექტები დამხმარე ფუნქციებს ასრულებს;

- უზრუნველყოფის პაკეტი, ანუ ისინი, რომლებიც "მომხმარებელ-სერვერის" არქიტექტურის პრინციპით მუშაობენ, სერვერის ფუნქციას ასრულებენ ფუნქციური ობიექტ-მომხმარებლისათვის. ამგვარად, უზრუნველყოფის პაკეტი მომხმარებელს საინფორმაციო სისტემის პროგრამულ-ტექნიკური რეალიზაციის დეტალების ცოდნისაგან ათავისუფლებს.

მაგალითი. პაკეტების დიაგრამა.



შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Business Process კატეგორიის Basic Flowchart შაბლონის საშუალებითაა შესრულებული.

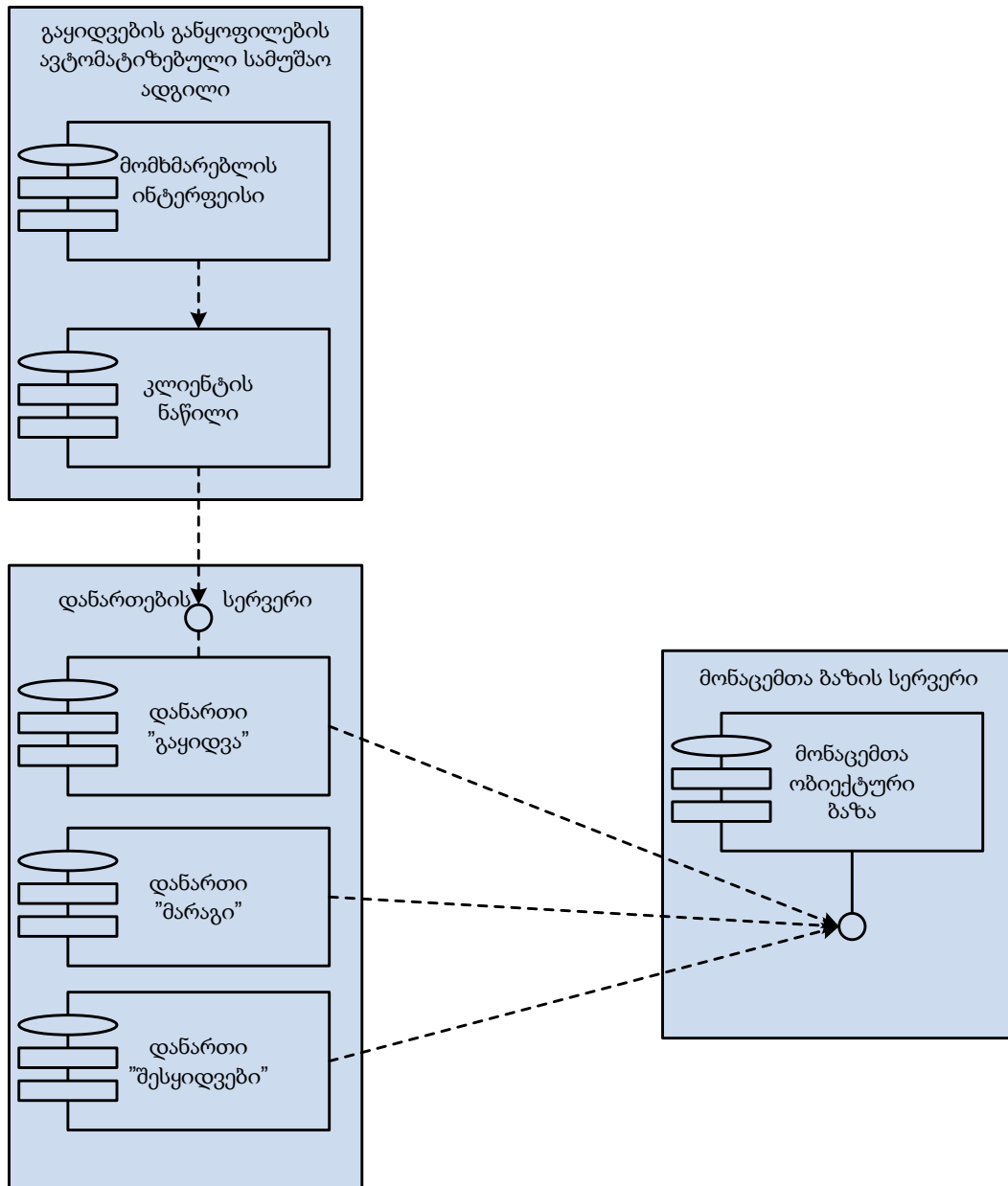
კომპონენტთა და განთავსების დიაგრამები

კომპონენტთა დიაგრამა (**Component diagram**) ასახავს პროგრამული კომპონენტების დამოკიდებულებას, რომელიც საწყისი, კომპილირებულ და შესრულებად ობიექტთა პროგრამული კოდების სახით გამოიხატება. როგორც წესი, ერთი კომპონენტი ობიექტთა კლასის ერთი პაკეტის პროგრამულ კოდს შეესაბამება.

კომპონენტს თავის შემადგენლობაში აქვს ობიექტთა ინტერფეისული კლასი, რომლის საშუალებითაც კომპონენტის დანარჩენი ობიექტების კლასზე მიმართვა ხორციელდება. ინტერფეისის საშუალებით სხვა კომპონენტების ობიექტები მიმართავს არა განხილული კომპონენტების კონკრეტულ ობიექტს, არამედ მის ინტერფეისულ ობიექტს. ამგვარად, კომპონენტებს შორის ურთიერთდამოკიდებულება მარტივდება, რადგან კომპონენტზე სხვა კომპონენტიდან მიმართვისას ამ კომპონენტის შიგა სტრუქტურის ცოდნა არ არის საჭირო. კომპონენტს შეიძლება რამდენიმე ინტერფეისი ჰქონდეს.

განთავსების დიაგრამა (**Deployment diagram**) გამოთვლითი ქსელის კვანძების მიხედვით კომპონენტების განლაგების ტოპოლოგიას გამოსახავს. ცალკეული კომპონენტი ყოველთვის ერთ კომპიუტერ-სერვერზე არის განთავსებული. ერთ კომპიუტერ-სერვერზე შეიძლება რამდენიმე კომპონენტი განთავსდეს.

მაგალითი. კომპონენტთა და განთავსების დიაგრამა.



შენიშვნა. ზემოთ მოყვანილ მაგალითში დიაგრამა Windows-ის დიაგრამებისა და ბლოკ-სქემების რედაქტორის Microsoft Visio-ის Block Diagram კატეგორიის Basic Diagram შაბლონის საშუალებით არის შესრულებული.

CASE-საშუალებების ძირითადი შესაძლებლობა

თანამედროვე CASE-საშუალებები, საინფორმაციო სისტემების დაპროექტების მხარდამჭერი ტექნოლოგიების ფართო სპექტრს მოიცავს: ანალიზისა და დოკუმენტირების მარტივი ინსტრუმენტებიდან დაწყებული და მთელი სასიცოცხლო ციკლის პროგრამული უზრუნველყოფის მომცავი სრული ავტომატიზაციით დამთავრებული.

საინფორმაციო სისტემების დამუშავების ყველაზე შრომატევადია ანალიზისა და დაპროექტების ეტაპები, როდესაც CASE-საშუალებები ტექნიკური გადაწყვეტილებებისა და საპროექტო დოკუმენტაციის მომზადების ხარისხს უზრუნველყოფს. ამასთან, მნიშვნელოვან როლს ასრულებს ინფორმაციის ვიზუალური წარმოდგენის მეთოდები. ეს გულისხმობს სტრუქტურული ან სხვა დიაგრამების რეალურ დროში აგებას, მრავალფეროვანი პალიტრის გამოყენებას, სინტაქსის წესების მრავალმხრივ შემოწმებას. საგნობრივი არის გრაფიკული მოდელირება დეველოპერს საშუალებას აძლევს არსებული საინფორმაციო სისტემა თვალნათლივ შეისწავლოს, შესაბამისი მიზნისა და არსებული შეზღუდვების მიხედვით გადააწყოს იგი.

CASE-საშუალებებს ყველაზე სრულყოფილი სახით შემდეგი მახასიათებლები აქვს:

- ერთიანი გრაფიკული ენა. CASE-ტექნოლოგია ყველა პროექტის მონაწილეებს, მათ შორის მომხმარებელს, უზრუნველყოფს ერთი ძლიერი, მკაფიო და ინტუიციური გრაფიკული ენით, რომელიც საშუალებას იძლევა მარტივი და მკაფიო სტრუქტურის ხელმისაწვდომი კომპონენტები მივიღოთ. ამასთან, აღნიშნული პროგრამები წარმოდგენილია ორგანოზომილებიანი სქემით (უფრო მარტივია გამოსაყენებლად, ვიდრე მრავალგვერდიანი აღწერები), რომელიც დამკვეთს საშუალებას აძლევს მონაწილეობა მიიღოს დამუშავების პროცესში, ხოლო დეველოპერს - ითანამშრომლოს საგნობრივი არის ექსპერტებთან, სისტემური ანალიტიკოსების, დიზაინერებისა და პროგრამისტების აზრი გაიზიაროს, უმარტივეს მათ ხელმძღვანელობის წინაშე პროექტის დაცვას და ამავე დროს, სისტემის თანმდევი

დოკუმენტაციისა და მასში ცვლილებების შეტანის გამარტივებას უზრუნველყოფს;

- პროექტის ერთიანი მონაცემთა ბაზა. CASE-ტექნოლოგიის საფუძველია - პროექტის მონაცემთა ბაზის (რეპოზიტორის) პროექტის შესახებ ყველა იმ არსებული ინფორმაციის შენახვა, რომლებიც დეველოპერებთან ერთად შეიძლება მათი წვდომის უფლებების შესაბამისად იქნეს გამოყენებული. რეპოზიტორის შინაარსი არა მარტო სხვადასხვა სახის ინფორმაციულ ობიექტებს მოიცავს, არამედ მათ კომპონენტებს შორის დამოკიდებულებას, ასევე ამ კომპონენტების გამოყენების ან დამუშავების წესებს. რეპოზიტორი შეიძლება სხვადასხვა ტიპის ობიექტებს ინახავდეს: სტრუქტურულ დიაგრამებს, ეკრანებისა და მენიუების განსაზღვრებებს, ანგარიშების პროექტებს, მონაცემთა აღწერას, დამუშავების ლოგიკას, მონაცემების, მათი ორგანიზაციისა და გადამუშავების მოდელებს, საწყის კოდებს, მონაცემთა ელემენტებს და ა. შ.;
- საშუალებების ინტეგრაცია. რეპოზიტორის საფუძველზე, CASE-საშუალებების ინტეგრაცია და დეველოპერებს შორის სისტემური ინფორმაციის განაწილება ხდება. ამავე დროს, რეპოზიტორის შესაძლებლობები რამდენიმე დონის ინტეგრაციას უზრუნველყოფს: მთელი საშუალებებზე საერთო სამომხმარებლო ინტერფეისი, საშუალებებს შორის მონაცემთა გადაცემა, სასიცოცხლო ციკლის ფაზების წარმოდგენის ერთიანი სისტემის საშუალებით დამუშავების ეტაპების, მონაცემთა გადაცემისა და რესურსების სხვადასხვა პლატფორმებს შორის ინტეგრაციას;
- პროექტის კოლექტიური დამუშავებისა და მართვის მხარდაჭერა. CASE-ტექნოლოგიის პროექტზე უჭერს მხარს გუნდურ მუშაობას, რომელიც ქსელში მუშაობის, პროექტის ნებისმიერი ფრაგმენტის განვითარების ან/და მოდიფიკაციის მიზნით ექსპორტისა და იმპორტის, ასევე დაგეგმვის, მონიტორინგის, მართვისა და ურთიერთქმედების საშუალებას უზრუნველყოფს, ანუ იმ ფუნქციებს, რომლებიც პროექტის განვითარებისა და შენარჩუნებისათვისაა აუცილებელი. ამ ფუნქციების რეალიზება ასევე რეპოზიტორის საფუძველზე ხორციელდება; კერძოდ,

რეპოზიტორის საშუალებით შეიძლება განხორციელდეს უსაფრთხოების კონტროლი (შეზღუდვების და წვდომის შეღავათები), ვერსიისა და ცვლილებების კონტროლი და ა.შ.;

- მოდელირება. CASE-ტექნოლოგიის საშუალებით შეიძლება სწრაფად შეიქმნას მომავალი სისტემის მოდელები (პროტოტიპები), რაც მომხმარებელს საშუალებას აძლევს დამუშავების ადრეულ ეტაპზე შეაფასოს, თუ რამდენად აწყობს და მისაღებია იგი პოტენციური მომხმარებლისათვის;
- დოკუმენტაციის გენერაცია. პროექტის ყველა დოკუმენტაცია რეპოზიტორის ბაზაზე ავტომატურად გენერირდება (ჩვეულებრივ, არსებული სტანდარტების მოთხოვნების შესაბამისად). CASE-ტექნოლოგიის უცილობელი უპირატესობაა, რომ დოკუმენტაცია არსებულ ვითარებას ყოველთვის შეესაბამება, რადგან პროექტში განხორციელებული ნებისმიერი ცვლილება რეპოზიტორში ავტომატურად აისახება;
- პროექტის შემოწმება (ვერიფიკაცია). CASE-ტექნოლოგია პროექტის დამუშავების საწყის ეტაპზე სისრულისა და თანამიმდევრულობის ავტომატურ შემოწმებას და კონტროლს უზრუნველყოფს, რაც მთლიანობაში, წარმატებაზე ახდენს გავლენას;
- პროგრამის კოდის ავტომატური გენერაცია. პროგრამის კოდის გენერაცია რეპოზიტორის საფუძველზე ხდება და მაღალი დონის ენებზე ტექსტების 85-90% ავტომატურად შექმნის საშუალებას გვაძლევს;
- სისტემის თანმდევი დოკუმენტაცია და რეინჟინერინგი. CASE-ტექნოლოგიის შემთხვევაში, სისტემის თანმდევი დოკუმენტაცია პროექტის და არა პროგრამის კოდის თანმდევი დოკუმენტაციით ხასიათდება.

რეინჟინერინგისა და ინჟინერინგის საპირისპირო საშუალებებით შესაძლებელია სისტემის კოდებისაგან შეიქმნას მისი მოდელი და მოხდეს მიღებული მოდელის ინტეგრირება პროექტში, კოდების შეცვლის დროს დოკუმენტაციის ავტომატური განახლება, კოდების რედაქტირების შემთხვევაში, სპეციფიკაციის ავტომატური შეცვლა განხორციელდეს და ა.შ.

ყველა CASE-საშუალება ზემოთ აღნიშნულ შესაძლებლობას მხარს არ უჭერს. ამიტომ, როგორც წესი, CASE-საშუალებებს ყველა იმ პროგრამულ პროდუქტს მიაკუთვნებენ, რომლებიც პროცესის სასიცოცხლო ციკლის ამა თუ იმ ერთობლიობის პროგრამული უზრუნველყოფის ავტომატიზაციას ახდენს და შემდეგი ძირითადი მახასიათებლები აქვს:

- საინფორმაციო სისტემის აღწერისა და დოკუმენტირებისათვის ხელმისაწვდომია მძლავრი გრაფიკული ინსტრუმენტები, რომლებიც მოსახერხებელი ინტერფეისით უზრუნველყოფს დეველოპერს და მისი შემოქმედებითი პოტენციალის განვითარებას ხელს შეუწყობს;
- CASE-საშუალებათა ცალკეული კომპონენტების ინტეგრაცია, რომელიც საინფორმაციო სისტემების დამუშავების პროცესის მართვას უზრუნველყოფს;
- სპეციალურად ორგანიზებული საპროექტო მეტა-მონაცემების საცავის (რეპოზიტორი) გამოყენება.

საინფორმაციო სისტემების პროტოტიპული დაპროექტება (RAD-ტექნოლოგია)

იმ კორპორაციულ ეკონომიკური საინფორმაციო სისტემის გამოჩენასთან დაკავშირებით, რომელიც "მომხმარებელი-სერვერი"-ს არქიტექტურაზეა ბაზირებული, ჩვეულებრივ, მომხმარებლისა და სერვერული ნაწილის პარალელური დამუშავების ხარჯზე დანართების დაჩქარებული დამუშავების შესაძლებლობა წარმოიშობა. მაგრამ არაერთგვაროვან გარემოში ასეთი რთული დანართების დამუშავება საკმაოდ დიდ სიძნელებთანაა დაკავშირებული, რის გამოც ამ უპირატესობის რეალური გამოყენება არც ისე იოლია. ამ პირობებში, ტრადიციული კასკადური მეთოდით საინფორმაციო სისტემის დამუშავება შეიძლება საკმაოდ გაიწელოს დროში, ხოლო შედეგი გარანტირებული არ არის.

საინფორმაციო სისტემის დამკვეთის ძირითადი სურვილია მინიმალური დანახარჯებით მაღალი ხარისხის მზა დანართები მიიღოს. ამასთან, დამკვეთი, რომელმაც სისტემის დამუშავებისათვის საკმაოდ სახსრები ჩადო, დაინტერესებულია დამუშავების პროცესი აკონტროლოს. სისტემის ექსპლუატაციაში გაშვების მომენტში მისი ხარისხის ძირითად კრიტერიუმს ის წარმოადგენს, თუ რამდენად სრულად აკმაყოფილებს მოცემული სისტემა დამკვეთის მოთხოვნებს.

დასამუშავებელი საინფორმაციო სისტემის მაღალი ხარისხის ერთ-ერთ პირობას წარმოადგენს დამუშავების პროცესში საბოლოო მომხმარებლის აქტიური ჩართვა, რამაც პროტოტიპული დაპროექტების მეთოდოლოგიაში პოვა ასახვა. ასეთი მეთოდოლოგიის ბირთვს წარმოადგენს RAD (Rapid Application Development) დანართების სწრაფი დამუშავება.

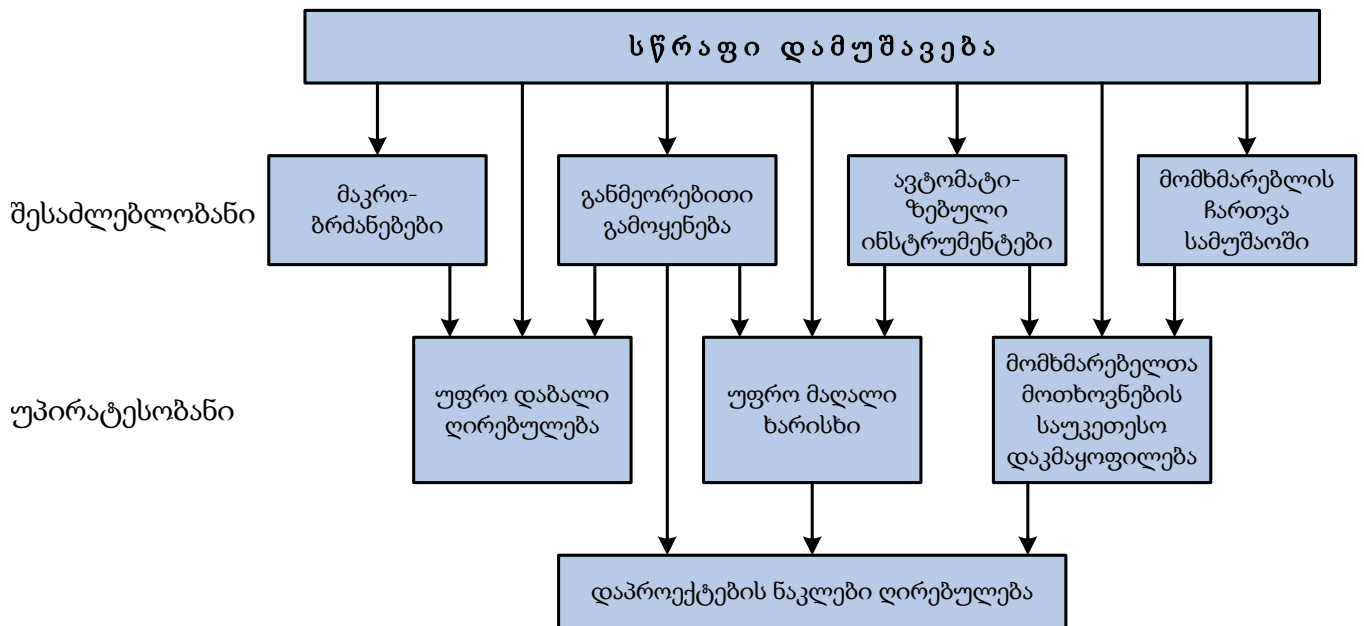
რთული კორპორაციული საინფორმაციო სისტემების დამუშავებისას სასურველია დამუშავების მთელი პერიოდის განმავლობაში მომხმარებლებმა დამპროექტებლებთან ერთად იმუშაონ. სისტემების დამუშავების ხარისხისა და ეფექტურობის ამაღლების ერთ-ერთი გზაა პროტოტიპული დაპროექტების ტექნოლოგიის გამოყენება.

მოცემული ტექნოლოგია სისტემის რეალიზაციის ადრეულ სტადიაში უზრუნველყოფს მოქმედი ინტერაქტიური მოდელის, ე.წ. სისტემა-

პროტოტიპის შექმნას, რომელიც საშუალებას მოგვცემს თვალნათლივ ვაჩვენოთ მომხმარებელს მომავალი სისტემა, დავაზუსტოთ მისი მოთხოვნები, ოპერატიულად მოვახდინოთ ინტერფეისული ელემენტების მოდიფიკაცია: შეტყობინების შეტანის ფორმები, მენიუ, გამომავალი დოკუმენტები, დიალოგის სტრუქტურა, რეალიზებული ფუნქციების შემადგენლობა.

სისტემა-პროტოტიპებთან მუშაობის დროს მომხმარებელი მომავალი სისტემის შესაძლებლობებს რეალურად აღიქვამს და მონაცემების დამუშავების უფრო ხელსაყრელ რეჟიმს განსაზღვრავს, რაც დასამუშავებელი სისტემის ხარისხს უფრო ამაღლებს.

განვიხილოთ საინფორმაციო სისტემა-პროტოტიპის სწრაფი დამუშავების ძირითადი შესაძლებლობა და უპირატესობა:



RAD დანართების სწრაფი დამუშავებისათვის გამოყენებული ყველა მეთოდი იმავდროულად, პროდუქციის მაღალ ხარისხსა და დამუშავების დაბალ ღირებულებას ემსახურება. ამ მეთოდების რიცხვს მიეკუთვნება:

1. დანართების იტერაციებით დამუშავება;
2. სასიცოცხლო ციკლის ყოველ ეტაპზე, შემდეგ ეტაპზე გადასასვლელად, სამუშაოს სრულად დამთავრება არ არის აუცილებელი;
3. სისტემის დაპროექტებასა და დამუშავებაში მომხმარებლის აუცილებელი ჩართვა;

4. სამუშაოთა შესრულების მაღალი პარალელობა;
5. პროექტის ნაწილების განმეორებითი გამოყენება;
6. CASE-საშუალებების აუცილებელი გამოყენება, რომელიც ანალიზისა და დაპროექტების ეტაპზე ტექნიკურ ერთიანობას უზრუნველყოფს;
7. კონფიგურაციის მართვის საშუალებათა გამოყენება, რომელიც პროექტში ცვლილებების შეტანასა და მზა სისტემის გამოყენებას ამარტივებს;
8. ავტომატური გენერატორების (ოსტატების) გამოყენება;
9. იმ პროტოტიპების გამოყენება, რომელიც საშუალებას იძლევა მომხმარებლის მოთხოვნათა გამოკვლევა და დაკმაყოფილება სრულად მოხდეს;
10. პროექტის ტესტირება და შემდგომი განვითარება, იმავდროულად პროტოტიპის რამდენიმე ვერსიის დამუშავება.

თითოეული ზემოთ ჩამოთვლილი მეთოდი, ცალ-ცალკე სიჩქარის გაზრდის, ხარისხის გაუმჯობესების საშუალებას იძლევა, მაგრამ მხოლოდ მათი ერთობლივი გამოყენება იძლევა დამუშავების პროცესში ხარისხობრივ ცვლილებებს.

თითოეულ ეტაპზე სამუშაოს არასრული დამთავრება შემდგომ ეტაპზე გადასვლის საშუალებას გვაძლევს. საინფორმაციო სისტემის დამუშავების იტერაციული საშუალებების გამოყენების შემთხვევაში, დარჩენილი სამუშაო შეიძლება იტერაციის შემდეგ საფეხურზე დამთავრდეს. მთავარი ამოცანაა – მომხმარებელს შრომისუნარიანი სისტემა, რაც შეიძლება სწრაფად ვუჩვენოთ და ამით დაზუსტებისა და დამატებითი მოთხოვნების პროცესის გააქტიურება მოხდეს.

საინფორმაციო სისტემის RAD-ტექნოლოგიების საშუალებით დამუშავების პროცესის ძირითად პრობლემას შემდგომ ეტაპზე გადასვლის მომენტის განსაზღვრა წარმოადგენს. ამის გადასაწყვეტად საციცოცხლო ციკლის ყოველ ეტაპზე დროის შეზღუდვა უნდა იქნეს შემოტანილი. გადასვლა გეგმის მიხედვით უნდა განხორციელდეს, იმ შემთხვევაშიც კი, როდესაც მთელი დაგეგმილი სამუშაო არ შესრულებულა. გეგმა სტატისტიკური ინფორმაციისა და დამპროექტებლების გამოცდილების მიხედვით დგება.

პროტოტიპული დაპროექტების ტექნოლოგიის რეალიზაციისათვის იმ მაღალდონიანი ინსტრუმენტული საშუალებების გამოყენებაა აუცილებელი,

რომლებიც სისტემის პროტოტიპის ფუნქციურ ვერსიად სწრაფად გადაქცევისა და შემდგომ მათში აუცილებელი შესწორებების საშუალებას იძლევა.

ასეთი ინსტრუმენტული საშუალებები შეიძლება პირობითად ორ კლასად დავყოთ, როგორცაა: განვითარებულ მონაცემთა ბაზების მართვის სისტემებში დანართების სწრაფი დამუშავების ინსტრუმენტები – კლასი DEVELOPER და დანართების სწრაფი დამუშავების ინტეგრირებული ინსტრუმენტები – კლასი BUILDER.

ამ კლასის ინსტრუმენტებს შეიძლება მივაკუთვნოთ 4GL (დანართის კომპონენტების გენერატორი) საშუალებანი:

- მონაცემთა ბაზის ცხრილების გენერატორი;
- შეტანა-გამოტანის ფორმათა გენერატორი;
- შეკითხვათა გენერატორი;
- ანგარიშების გენერატორი;
- მენიუს გენერატორი.

ასეთი გენერატორები თითქმის ყველა მონაცემთა ბაზების მართვის სისტემებში არსებობს, როგორც პერსონალურ Access, FoxPro, Paradox, ისე მონაცემთა ბაზების საწარმოო სერვერების გარემოში (Oracle, Informix, Adabas D და სხვ.).

BUILDER კლასის განმასხვავებელი ნიშანი ისაა, რომ მოცემული კლასის ინსტრუმენტების CASE-საშუალებებთან ინტეგრაცია ძალზე მარტივად ხდება და დანართთა სწრაფი დამუშავების ერთიან გარემოს წარმოადგენს. BUILDER კლასის ინტეგრირებულ ინსტრუმენტებს შეიძლება მივაკუთვნოთ Power Builder Enterprise (Power Soft), Delphi (Borland), Builder Ci ++ და სხვ.

განვიხილოთ დანართების სწრაფი დამუშავების მონაცემთა ბაზის მართვის სისტემის Access-ის ინსტრუმენტების გარემო, რომელიც რიგ ოსტატებს (კონსტრუქტორებს) მოიცავს.

- ცხრილების ოსტატის (კონსტრუქტორი) დანიშნულებაა მონაცემთა ბაზის ცხრილების სტრუქტურისა და მათი ურთიერთკავშირის სწრაფი შექმნა;
- შეტანა-გამოტანის ფორმების ოსტატი (კონსტრუქტორი) შესაძლებლობას იძლევა სწრაფად შექმნათ მონაცემთა ბაზაში ინფორმაციის შესატანი სხვადასხვა ტიპის ეკრანული ფორმები;

- შეკითხვების ოსტატი (კონსტრუქტორი) შესაძლებლობას იძლევა შევქმნათ სხვადასხვა სირთულის კითხვები;

- ანგარიშების ოსტატი (კონსტრუქტორი) შესაძლებლობას იძლევა ანგარიშები შევქმნათ რამდენიმე ცხრილის ან შეკითხვის ბაზაზე;

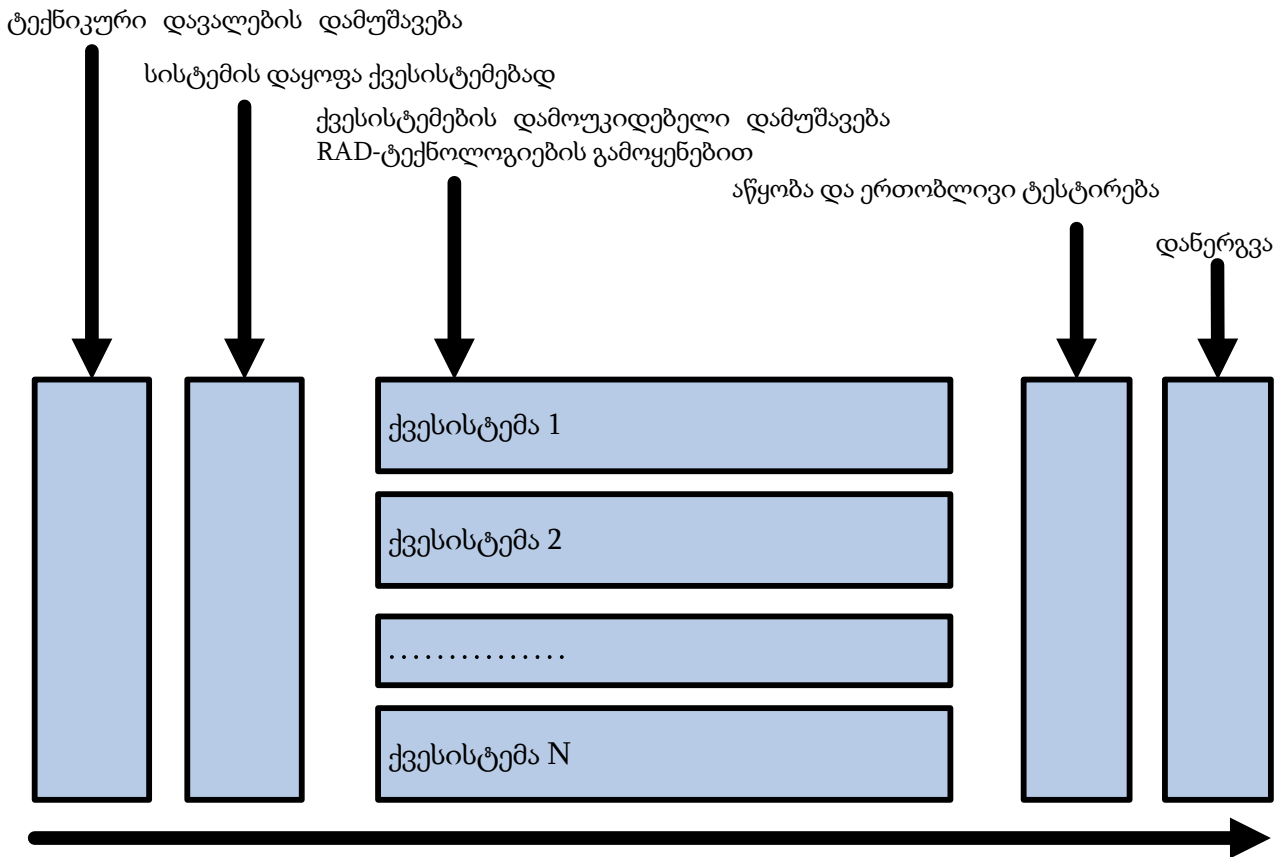
- ღილაკებიანი ფორმების ოსტატი (კონსტრუქტორი) შესაძლებლობას იძლევა დავამუშაოთ მმართველი ინტერფეისული ობიექტები, რომლებიც დანართებთან მუშაობის სამართავად გამოიყენება.

ოსტატთან მუშაობა მის ნაბიჯ-ნაბიჯ მომართვას, ხოლო კონსტრუქტორთან მუშაობა საინფორმაციო სისტემის შემადგენელი დასაპროექტებელი კომპონენტების კონსტრუქციული ელემენტების დეტალურ განსაზღვრას ნიშნავს.

RAD-ტექნოლოგიების საშუალებით საინფორმაციო სისტემის დამუშავების სასიცოცხლო ციკლი, ტექნიკური დავალებისა და სისტემის დეკომპოზიციის ფორმირების შემდეგ გვთავაზობს ქვესისტემების დამოუკიდებელ დამუშავებას კომპლექსური საინფორმაციო სისტემის შემდგომი აწყობით, ტესტირებითა და დანერგვით (ნახ. 27).

RAD-ტექნოლოგიების გამოყენების გამოცდილება გვიჩვენებს, რომ სისტემა-პროტოტიპის გამოყენებით ტექნოლოგიური პროცესების დაპროექტების ორგანიზაციის ორი საბაზო ვარიანტი არსებობს.

პირველ ვარიანტში სისტემა-პროტოტიპის დამუშავება საინფორმაციო სისტემის დამუშავებისადმი წაყენებულ იმ მოთხოვნათა საუკეთესო სპეციფიკაციებისათვის გამოიყენება, რომელთა დამუშავების შემდეგ თვით პროტოტიპი უკვე საჭირო აღარ არის. ამ შემთხვევაში, ტრადიციულად დამუშავდება "ამოცანის დასმა", რომლის დოკუმენტაცია არის სისტემა-პროტოტიპის სპეციფიკაცია. მომხმარებლისათვის პროტოტიპის დემონსტრირებისა და გადამუშავების შემდეგ მუშავდება ახალი "ამოცანის დასმა", რომელიც უკვე მოქმედი საინფორმაციო სისტემის დამუშავების საფუძველი გახდება.



ნახ. 27. საინფორმაციო სისტემის RAD-ტექნოლოგიების საშუალებით დამუშავების სასიცოცხლო ციკლი

პირველი ვარიანტის ძირითად ნაკლოვანებას წარმოადგენს სისტემა-პროტოტიპის არაეფექტური გამოყენება, რაც იმაში გამოიხატება, რომ პროტოტიპი საინფორმაციო სისტემის შემდგომი დამუშავებისას აღარ გამოიყენება, მას შემდეგ, რაც მან თავისი პირველი ამოცანა შეასრულა – პროექტში გაურკვევლობები აღმოფხვრა.

მეორე ვარიანტი გვთავაზობს პროგრამული პროდუქციის ექსპლუატაციისათვის სისტემა-პროტოტიპის მზა სახით იტერაციულ განვითარებას. სისტემა-პროტოტიპის იტერაციული დამუშავება სისტემა-პროტოტიპის შექმნა/მოდულიზაციას, მომხმარებლისათვის მის დემონსტრირებასა და შეთანხმებას, სისტემის მიმართ ახალი სპეციფიკაცია-მოთხოვნების, ახალი მოდიფიკაციის დამუშავებას და სხვა მოიცავს, ვიდრე არ იქნება შექმნილი მზა დანართი. სისტემა-პროტოტიპის კომპონენტების დოკუმენტაციას უშუალოდ თვით სპეციფიკაციები წარმოადგენს.

საინფორმაციო სისტემის დამუშავებისადმი პროტოტიპული მიდგომის იტერაციული გამოყენება უზრუნველყოფს დაპროექტებისათვის საჭირო რესურსების ეკონომიას, და რაც მთავარია, მკვეთრად ამცირებს დამუშავებისა და ექსპლუატაციისათვის მზა სისტემის დანერგვის დროს. ამასთან, პროტოტიპული ტექნოლოგიის ძირითად ღირსებას წარმოადგენს საინფორმაციო სისტემის დანერგვის დროს გადამუშავების მოცულობის მკვეთრი შემცირება, რომელიც დაპროექტების ტრადიციული მეთოდების შემთხვევაში თითქმის რეალიზაციის პირველადი ხარჯების ტოლი იყო.

შინაარსი

შესავალი.....	3
საინფორმაციო სისტემები	6
მონაცემთა ბაზები	9
CASE-საშუალებები	10
დამუშავების საშუალებები	12
საინფორმაციო სისტემის ცნება	13
საინფორმაციო სისტემების კლასიფიკაცია.....	18
კლასიფიკაცია მასშტაბის მიხედვით.....	18
ერთეული საინფორმაციო სისტემები	18
ჯგუფური საინფორმაციო სისტემები	19
კორპორატიული საინფორმაციო სისტემები	19
კლასიფიკაცია გამოყენების სფეროს მიხედვით.....	19
კლასიფიკაცია ორგანიზაციის წესის მიხედვით	21
არქიტექტურა ფაილ-სერვერი	22
არქიტექტურა კლიენტ-სერვერი	24
მრავალდონიანი არქიტექტურა.....	27
Internet/Intranet-ტექნოლოგიები.....	29
საინფორმაციო სისტემის უზრუნველყოფის ქვესისტემები.....	29
ორგანიზაციული უზრუნველყოფის ქვესისტემა	30
სამართლებრივი უზრუნველყოფის ქვესისტემა	30
ტექნიკური უზრუნველყოფის ქვესისტემა.....	31
მათემატიკური უზრუნველყოფის ქვესისტემა.....	31
პროგრამული უზრუნველყოფის ქვესისტემა	32
ინფორმაციული უზრუნველყოფის ქვესისტემა.....	32
ლინგვისტიკური უზრუნველყოფის ქვესისტემა	33
ტექნოლოგიური უზრუნველყოფის ქვესისტემა.....	33

საინფორმაციო სისტემების გამოყენების სფერო და მისი რეალიზაციის მაგალითები.....	35
საბუღალტრო აღრიცხვა	35
ფინანსური ნაკადების მართვა.....	36
საწყობის, ასორტიმენტის, შესყიდვების მართვა	36
საწარმოო პროცესების მართვა	36
მარკეტინგის მართვა.....	37
დოკუმენტთა ბრუნვა	37
წარმოების ოპერატიული მართვა	37
ფირმის შესახებ ინფორმაციის მიწოდება.....	38
საინფორმაციო სისტემის სასიცოცხლო ციკლი	39
პროექტების კლასიფიკაცია.....	40
საინფორმაციო სისტემის დაპროექტების ძირითადი ფაზები	42
საინფორმაციო სისტემის სასიცოცხლო ციკლის მოდელები	45
კასკადური მოდელი	45
იტერაციული მოდელი.....	47
სპირალური მოდელი	47
I ეტაპი – სისტემური ანალიზი.....	51
II ეტაპი – სისტემური სინთეზი	56
საჭირო რესურსების განსაზღვრა	57
ინფორმაციის კლასიფიკაცია	58
საინფორმაციო სისტემის პოტენციურ მომხმარებელთა კლასიფიკაცია.....	61
არსებული ინფორმაციის ანალიზი.....	62
განსაკუთრებით აუცილებელი ინფორმაციის ამორჩევა	63
საინფორმაციო სისტემის დაპროექტება	64
პროგრამული უზრუნველყოფის დამუშავება	70
III სტადია – დანერგვა	72
IV სტადია – ექსპლუატაცია და პროექტის მომსახურება.....	75
კორპორაციული საინფორმაციო სისტემები.....	77

საინფორმაციო სისტემების სპეციფიკა	77
საინფორმაციო სისტემის ამოცანები.....	78
საინფორმაციო სისტემის აგების პრობლემები	85
მოთხოვნები ტექნიკური საშუალებების მიმართ.....	91
ინფორმაციული დანართების არქიტექტურის ზოგადი კლასიფიკაცია.....	92
ფაილ-სერვერული დანართები.....	92
კლიენტ-სერვერული დანართები	93
Intranet – დანართები	94
მონაცემთა საცავები (Data Warehousing), მონაცემების დამუშავების ოპერატიული და ანალიტიკური სისტემები	98
ინტეგრირებული განაწილებული დანართები	102
ფაილ-სერვერული დანართების დაპროექტების, დამუშავების, თანმხლები დოკუმენტების საშუალებანი და მეთოდოლოგია	105
ფაილ-სერვერული დანართების დამუშავების ტრადიციული საშუალებანი და მეთოდოლოგიები.....	105
დაპროგრამების სისტემები და ბიბლიოთეკები	105
პერსონალურ კომპიუტერებზე მბმს საფუძველზე დანართის დამუშავების საშუალებები და მეთოდები.....	107
ფაილ-სერვერული დანართების დამუშავების ახალი საშუალება.....	107
თანამედროვე საშუალებების საერთო დახასიათება.....	108
ახალი მიდგომის მაგალითები	110
MS Access პაკეტი.....	110
სისტემა Visual FoxPro.....	112
დაპროგრამების გარემო CA-VISUAL OBJECTS	112
ფაილ-სერვერული დანართების კლიენტ-სერვერულ გარემოში გადატანა	113
მონაცემთა ბაზებთან წვდომის ბიბლიოთეკები	114
პროტოკოლი ODBC და მისი რეალიზაცია.....	115
დანართის გამსხვილება (Upsizing).....	117

კლიენტ-სერვერული დანართების დაპროექტების საშუალებები და მეთოდები. დამუშავება და თანმხლები დოკუმენტები.....	118
კლიენტ-სერვერული დანართების შექმნის ბაზური საშუალებები არქიტექტურაში “კლიენტ-სერვერი”.....	118
წაშლის პროცედურების გამოძახება	118
RPC პროტოკოლი და მისი რეალიზაცია	119
პროტოკოლი XDR.....	120
TCP/IP პროტოკოლების სტეკი, როგორც RPC-ის საფუძველი	120
მონაცემთა ბაზების სერვერები	122
მონაცემთა ბაზის სერვერის ცნება	122
მონაცემთა ბაზის სერვერის ბაზური არქიტექტურა	123
მონაცემთა ბაზების ენები	123
SQL ენა - SQL-სერვერის საბაზო ინტერფეისი	124
მოდულთა ენა, თუ ჩამენებული SQL?.....	125
რელაციურ მონაცემთა ბაზების დაპროექტების კლასიკური მიდგომა	126
ფუნქციური და სხვა დამოკიდებულებები.....	127
მონაცემთა ბაზების კონცეპტუალური მოდელები და სქემები	128
საინფორმაციო სისტემების დაპროექტების CASE-სისტემები	130
CASE-სისტემების დანიშნულება და სახესხვაობა	130
Silverrun	132
JAM	136
ობიექტზე ორიენტირებული CASE-საშუალებები (Rational Rose).....	141
მონაცემთა ბაზების ფიზიკური დაპროექტება	143
Intranet-დანართის დაპროექტების, დამუშავების, თანმხლები დოკუმენტაციის საშუალებები და მეთოდოლოგია	144
CASE-ტექნოლოგიების ძირითადი ცნებები და მათი კლასიფიკაცია	145
საინფორმაციო სისტემების სტრუქტურაზე ორიენტირებული დაპროექტება	150
ფუნქციური სპეციფიკაციების დიაგრამა.....	152
მონაცემთა ნაკადის დიაგრამა.....	153
მდგომარეობათა გადასვლის დიაგრამა	157
ინფორმაციულ-ლოგიკური მოდელი ”არსი-კავშირი” (ER-დიაგრამა).....	161

პროგრამული დანართის სტრუქტურის დიაგრამა	166
საინფორმაციო სისტემების ობიექტზე ორიენტირებული დაპროექტება	169
გამოყენების პრეცედენტის დიაგრამა	170
ობიექტთა კლასის დიაგრამა.....	173
მდგომარეობის დიაგრამა	175
ობიექტთა ურთიერთქმედების დიაგრამა.....	177
მოღვაწეობის დიაგრამა.....	180
პაკეტების დიაგრამა	182
კომპონენტთა და განთავსების დიაგრამები	184
CASE-საშუალებების ძირითადი შესაძლებლობა.....	186
საინფორმაციო სისტემების პროტოტიპული დაპროექტება (RAD-ტექნოლოგია)	
.....	190

რედაქტორი ი. სემიკინა

გადაეცა წარმოებას 26.06.2012. ხელმოწერილია დასაბეჭდად
10.09.2012. ქაღალდის ზომა 60X84 1/16. პირობითი ნაბეჭდი თაბახი 12,5.
ტირაჟი 100 ეგზ.

საგამომცემლო სახლი ”ტექნიკური უნივერსიტეტი”,
თბილისი, კოსტავას 77



Verba volant,
scripta manent