

## მართვის მრავალდონიანი განაწილებული სისტემების დაპროექტება

თეიმურაზ სუხიაშვილი  
საქართველოს ტექნიკური უნივერსიტეტი

### რეზიუმე

განიხილება UML მეთოდოლოგიით განსაზღვრული პროგრამული სისტემის დაპროექტების გაფართოებული ვარიანტი, რომელშიც ხუთი თვალთახედვის საფუძველზე მოცემულია მართვის მრავალდონიანი განაწილებული სისტემების აგების ეტაპები. მოყვანილი მიდგომა ითვალისწინებს ინტეგრირებული კვლევის მექანიზმის შექმნას, რითაც სისტემის დამპროექტებელს საშუალება ეძლევა აგებული მოდელების საფუძველზე განახორციელოს პროცესების კომპიუტერული ანალიზი და მოახდინოს ვარიანტების რაოდენობრივი შეფასება მათგან ოპტიმალურის ამოსარჩევად.

**საკვანძო სიტყვები:** მრავალდონიანი განაწილებული სისტემა. ობიექტ ორიენტირებული ანალიზი. ობიექტ ორიენტირებული დაპროექტება. UML დიაგრამები.

### 1. შესავალი

ორგანიზაციულ-ადმინისტრაციული სისტემის ანალიზმა, სამდონიანი სასამართლო სისტემის მაგალითზე გვიჩვენა, რომ მისი განაწილებული მართვის ავტომატიზებული სისტემა მიეკუთვნება დიდი და რთული სისტემების კლასს, როგორც დაპროექტების, ასევე მისი პროგრამული რეალიზაციისა და ექსპლუატაციის თვალსაზრისით [1]. აქედან გამომდინარე, ასეთი მართვის სისტემების მოდელირების, დაპროექტებისა და აგების მიზნით აუცილებელია როგორც სისტემების ზოგადი თეორიის, სისტემური მიდგომის მეთოდების, ასევე სპეციალიზებული კიბერნეტიკული ხერხებისა და ინსტრუმენტების გამოყენება [2,3].

### 2. ძირითადი ნაწილი

ამჟამად სისტემების მოდელირების ობიექტ-ორიენტირებული მიდგომა და მის საფუძველზე დამუშავებული უნიფიცირებული მოდელირების ენის (UML) მეთოდოლოგია არის განაწილებული, რთული საგნობრივი სფეროებისა და მათი შესაბამისი დიდი პროგრამული პაკეტების (Large Application's Software Engineering with UML) დაპროექტებისა და პროგრამული რეალიზაციის ტექნოლოგია [3].

UML ტექნოლოგიით მართვის განაწილებული სისტემების არქიტექტურა A გამოისახება ხუთი წარმოდგენით (პენტაედრული მოდელით):

$A = Au \cup As \cup Ap \cup Ai \cup Ad$ , სადაც

Au - წარმოდგენაა პრეცედენტების თვალთახედვით (Use case view) – მოთხოვნების მოდელირებისათვის;  
As - წარმოდგენა პროექტირების თვალთახედვით (Design view) – სტრუქტურული მოდელირებისათვის;  
Ap - წარმოდგენა პროცესების თვალთახედვით (Process view) – პროცესების მოდელირებისათვის, რომლებიც ახდენს სისტემაში პარალელიზმისა და სინქრონიზაციის მექანიზმის ფორმირებას;

Ai - წარმოდგენა რეალიზაციის თვალთახედვით (Implementation view) – მზა ფიზიკური სისტემის აწყობისა და სტანდარტული პროდუქტის სახით გამოშვებისათვის;

Ad - წარმოდგენა განლაგების თვალთახედვით (Deployment view) – მასში შედის კვანძები, რომლებიც ახდენს აპარატულ საშუალებათა ტოპოლოგიის ფორმირებას და რომლებზეც სრულდება სისტემა.

თითოეული ამორჩეული სახეობის მიმართ საგნობრივი სფეროს სფეციფიკიდან გამომდინარე დგინდება, თუ რომელი მოდელირება საჭირო სისტემის ყველაზე არსებითი დეტალების გამოსახვისათვის რათა სრულად აისახოს მოთხოვნები ავტომატიზებული სისტემისადმი და მისი რეალიზების საშუალებები.

UML როგორც უნივერსალური საშუალება, კარგ მეთოდოლოგიურ ინსტრუმენტს წარმოადგენს, მაგრამ დაპროექტებისა და რეალიზაციის პროცესი თავიდან ბოლომდე დამოკიდებულია დამპროექტებელ-სპეციალისტზე. აქ ძირითადად ევრისტიკულ მოდელირებასთან გვაქვს საქმე, რამეთუ როლებისა და მათი ამოცანების დაგეგმარებიდან დაწყებული, კლასებისა და კლასთაშორის დიაგრამებისა და პროცესების აგებით დამთავრებული, შემდეგ პროგრამული კომპონენტების დახვეწითა და ტესტირებით, მთლიანად საბოლოო შედეგები ამ სპეციალისტზეა დამოკიდებული. გარდა ამისა, ამოცანათა გადაწყვეტის მრავალეტაპიანობა და კომპლექსურობა შესაძლებელს ხდის შედეგები მივიღოთ მრავალვარიანტული, გამომდინარე მათი კომბინატორული ხასიათისა. ევრისტიკული მიდგომით კი ვერ დავადგენთ, თუ რამდენად კარგია ჩვენს მიერ მიღებული შედეგები. ამ თვალსაზრისით საინტერესოა გადაწყვეტის სინტაქსური და სემანტიკური ეკვივალენტობის (მოდელირების) მეთოდებით დასმული ამოცანის გადაჭრა. განსაზღვროს მისი გადაწყვეტის ოპტიმალური მეთოდი, შეფასების კრიტერიუმები. მოცემულ ნაშრომში განვიხილავთ მრავალდონიანი, მართვის განაწილებული სისტემების დაპროექტება-რეალიზაციის ამოცანათა ერთობლიობას UML მოდელის ინტერპრეტაციის საფუძველზე. იგი ითვალისწინებს კომპლექსური კრიტერიუმების ერთობლიობის გამოვლენას, რომელიც UML მოდელის

ცალკეული ეტაპების ოპტიმიზაციას ემსახურება სპეციალური ლოგიკურ-ალგებრული, სიმრავლეთა თეორიისა და გრაფების, ავტომატების თეორიისა და პეტრის ქსელების, მონაცემთა ბაზებისა და დაპროგრამების ობიექტ-ორიენტირებული მეთოდებისა და ინსტრუმენტების გამოყენების ბაზაზე. ამგვარად, განიხილება პენტადრული მოდელის გაფართოებული ვარიანტი.

სასამართლო სისტემის შესწავლისა და ანალიზის მაგალითზე დადგენილი იქნა იმ მოდელების ერთობლიობა, რომელთა მეშვეობით შესაძლებლობა გვეძლევა სრულად გამოვსახოთ მართვის მრავალდონიანი განაწილებული სისტემების არქიტექტურა თითოეული თვალთახედვით. დაშუშავებული პენტადრული მოდელის 1-ლი (პრეცედენტების) თვალთახედვით საჭიროა დადგინდეს თუ რას უნდა აკეთებდეს სისტემა გარე მეთვალყურის თვალსაზრისით, იმისგან დამოუკიდებლად, თუ როგორ უნდა აკეთებდეს იგი ამას. შესაბამისად პრეცედენტების თვალთახედვით სისტემა სასურველია შეიცავდეს:

$$Au = \langle \{U_i\}, \{I_i\}, \{F_i\} \rangle, \text{ სადაც}$$

$\{U_i\}$  - სისტემის კონტექსტის, მოთხოვნების და მათი რეალიზების საშუალებების ამსახველი მოდელების სიმრავლეა;  $\{I_i\}$  - მართვის ნაკადების ამსახველი მოდელების სიმრავლეა;  $\{F_i\}$  - სამუშაო პროცესებისა და ოპერაციების ამსახველი მოდელების სიმრავლეა.

სისტემის კონტექსტის და მოთხოვნების წარმოდგენისათვის გამოიყენება პრეცედენტების მოდელი. მოცემული მიდგომის არსი მდგომარეობს პრეცედენტების ტრანსფორმაციაში, დაწყებული კომპლექსური პროცესებიდან (ზედა დონე) ფუნქციითა დეკომპოზიციამდე (ქვედა დონე), მათი შემდგომი რეალიზაციით.

კომპლექსური პროცესების ქვეშ ჩვენ ვგულისხმობთ მრავალდონიანი განაწილებული მართვის სისტემის როლების (ფუნქციური თანამშრომლების) მთავარ ფუნქციებს, რომლებიც ურთიერთკავშირშია ერთმანეთთან ზედა დონეზე და იმლება სტრუქტურულ ქვეფუნქციებად (დეკომპოზიციის საფუძველზე), რომელიც მოიცავს განსაზღვრულ ამოცანათა (ფუნქციითა) და მათი რეალიზებისათვის საჭირო ელემენტთა ერთობლიობას, მათ შორის მოქმედებათა აღწერით. ფუნქციურ ამოცანათა შესაბამის ელემენტთა მოქმედებების დასაპროექტებლად გამოვიყენებთ ურთიერთქმედებისა და მოღვაწეობის დიაგრამებს.

პენტადრული მოდელის მე-2 (დაპროექტების) თვალთახედვის სქემა პირველის საფუძველზე იქმნება და იგი სტატიკური სტრუქტურების დაპროექტებას ეხება. ეს არის კლასები და კლასთა დიაგრამები, რომელთა სქემების კონსტრუირება საკვლევი ობიექტის სინტაქსურ-სემანტიკური ანალიზისა და მისი ელემენტების სინთეზის მეთოდების საფუძველზე უნდა განვახორციელოთ. ამიტომ, დაპროექტების As თვალთახედვით – სისტემა უნდა შეიცავდეს სიტემის ლექსიკონის დამდგენ და გამოვლენილ არსებს შორის მიმართებების აღმწერ მოდელებს:

$$As = \langle \{K_i\}, \{R_i\}, \{S_i\} \rangle, \text{ სადაც}$$

$\{K_i\}$  - სისტემის აბსტრაქციების (კლასების) სიმრავლეა;  $\{R_i\}$  - კლასებს შორის მიმართებების სიმრავლე;  $\{S_i\}$  - ობიექტთა სასიცოცხლო ციკლის ამსახველი მოდელების სიმრავლეა.

სტატიკური სტრუქტურების, ანუ კლასთა დიაგრამების ოპტიმალური ვარიანტების დაპროექტება უმნიშვნელოვანესი ამოცანაა მუშა პროგრამული პაკეტების (კოდების) შესაქმნელად. ეს საკითხები, თავიანთი სირთულისა და ძნელადფორმალისებადი ამოცანების არსებობის გამო, აუცილებლად მოითხოვს ადამიან-მანქანის ერთობლივ (ავტომატიზებულ) რეჟიმში მუშაობის პროცედურებს. უნდა გადაწყდეს ლოკალური სტრუქტურების (ცალკეული კლასების შიგა სტრუქტურების) შედგენილობის განსაზღვრის ამოცანა, რომელიც საკვლევი ობიექტის სემანტიკურ ანალიზზე და სტრუქტურათა სინთეზზეა დაფუძნებული. კვლევის ინსტრუმენტებად კი ნაწილობრივ მოწესრიგებულ სიმრავლეთა და რელაციურ დამოკიდებულებათა თეორიები გამოგვადგება.

შემდეგ უნდა გადაწყდეს ლოკალურ სტრუქტურათა (კლასებს) შორის ბუნებრივად არსებული გენეტიკური და აგრეგატული კავშირების (გლობალური კავშირების) რეალიზაცია, ანუ კლასთა დიაგრამების ერთობლიობათა შექმნა. ბოლოს განისაზღვრება იმ მეთოდებისა და შეტყობინებების ერთობლიობათა სიმრავლეები, რომლებიც დაპროექტებულ კლასებზე და კლასთა დიაგრამებზე იქნება ორიენტირებული. ამ შემთხვევაში სტატიკური სტრუქტურები განსაზღვრული მეთოდების კვლევის ობიექტებს წარმოადგენს და მათი დამუშავების პროცესების აქტუალიზაცია სისტემის გარედან შემოსულ შეტყობინებებზეა დამოკიდებული.

თუ აღნიშნულ მსჯელობას შევაჯამებთ, განხილულ ეტაპზე – სტატიკურ სტრუქტურათა ოპტიმიზაციის ეტაპზე, დადგინდება კლასების, კლასთაშორის-კავშირების და მათი სასიცოცხლო ციკლების ამსახველი მოდელები. ესაა ის წერტილი, სადაც სტატიკური სტრუქტურები (მდგომარეობათა მოდელები) დინამიკური სტრუქტურებით (ყოფაქცევის მოდელები) უნდა გაფართოვდეს. ამის საფუძველი კი – მეთოდები ანუ ის ფუნქციებია, რომლებიც ინიციალიზირდება გარედან მოსული შეტყობინებების (მოთხოვნათა) საფუძველზე.

ავტომატიზებული სისტემის არქიტექტურა პროცესების თვალთახედვით Ap უნდა შეიცავდეს - პროცესების აღმწერ, პარალელიზმისა და სინქრონიზაციის მექანიზმების დამდგენ მოდელებს:

$$Ap = \langle \{P_i\}, \{W_i\} \rangle, \text{ სადაც}$$

$\{P_i\}$  - პროცესების სიმრავლეა;  $\{W_i\}$  - პროცესების შესრულების დასადგენი წესების სიმრავლე.

ამჯერად საქმე გვაქვს დინამიკურ პროცესებთან, რომლებიც, ობიექტ-ორიენტირებული მოდელირების ტერმინებით თუ დავახასიათებთ, სისტემის ყოფაქცევას აღწერს. პენტაედრული მოდელის განხილვა დინამიკური პროცესების თვალთახედვით მეტად მნიშვნელოვანი და შრომატევადი ეტაპია. იგი აერთიანებს როგორც სამუშაოთა გადართობას პარალელურ მართვის ნაკადებს შორის, ასევე კომუნიკაციისა და სინქრონიზაციის სწორი მექანიზმების დადგენას სისტემის აქტიურ და პასიურ ობიექტებს შორის. ამგვარად, ჩვენი ამოცანა ამ ეტაპზე მდგომარეობს განაწილებული მრავალდონიანი მართვის სისტემის დინამიკური პროცესების კვლევაში და მის საფუძველზე მეთოდებისა (ფუნქციების) და მანინციალიზებული შეტყობინებების ერთობლიობათა განსაზღვრასა და მათი ეფექტური დამუშავების სცენარებისა და ლოგიკური პროცედურების კონსტრუირებაში.

მართვის თითოეულ დონეზე შესაძლებელი და სასურველია საპროექტო შედეგების თვისობრივი და რაოდენობრივი ანალიზის ჩატარება და შეფასება. ამ მიზნით ვიყენებთ, ერთის მხრივ, მასობრივი მომსახურების მეთოდებს, კერძოდ მარკოვის პროცესებს, და მეორეს მხრივ, პეტრის სისტემური ქსელებით დინამიკური პროცესების მოდელირების ინსტრუმენტს. ასეთი ინტეგრირებული კვლევის მექანიზმის შექმნა ხელს უწყობს სისტემის დამპროექტებელს აგებული მოდელის საფუძველზე განახორციელოს პროცესების კომპიუტერული ანალიზი და მოახდინოს ვარიანტების რაოდენობრივი შეფასება მათგან ოპტიმალურის ამოსარჩევად.

მომდევნო ეტაპი ეხება პროგრამული პაკეტის კომპონენტების დამუშავებას. ესაა პროგრამული, რესურსული, სათავო ფაილებისა და სხვა კოდების შექმნა, რომლებიც თავსდება დასაპროექტებელ ფაილში. მრავალდონიანი, განაწილებული მართვის სისტემის არქიტექტურა რეალიზების თვალთახედვით  $A_i$  - უნდა შეიცავდეს პროგრამული კომპონენტების და მათ შორის კავშირების, შესანახი მონაცემების ლექსიკონის და მათ შორის სემანტიკური კავშირების აღმწერ მოდელებს:

$$A_i = \langle \{C_i\}, \{T_i\}, \{R_i\} \rangle, \text{ სადაც}$$

$\{C_i\}$  - პროგრამული კომპონენტების სიმრავლეა;  $\{T_i\}$  - შესანახი მონაცემების(ცხრილების) სიმრავლეა;  $\{R_i\}$  - მიმართებების სიმრავლე.

მართვის სისტემის პროგრამული უზრუნველყოფის დასამუშავებლად თავდაპირველად განისაზღვრება ოპერაციული სისტემის გარემო, მაგალითად, Windows ან Linux. შეირჩევა ძირითადი პლატფორმა, თუკი ამის საჭიროება არსებობს, მაგალითად, dot-NET შემთხვევაში, როცა გამოიყენება Visual Studio .NET პაკეტი.

ბოლოს, ჩვენს მიერ შერჩეული იქნება დაპროგრამების ენა, რომლის გარემოშიც მოხდება კოდების აგება. მაგალითად, ობიექტ-ორიენტირებული, ვიზუალურ კომპონენტებიანი ენები: C++, Visual Basic, C#, J++ .NET ან სხვა [5]. პროგრამული კოდების დამუშავება შესაძლებელია ავტომატიზებულ რეჟიმში და იგი რამდენიმე ეტაპად მიმდინარეობს. თავდაპირველად დამუშავდება პროგრამათა ტექსტური (CPP, CSH, ... ) და სათავო (H) ფაილები. ეს უკანასკნელი ავტომატურად იწერება თვით სისტემის მიერ, თუ ჩვენ ვიყენებთ, მაგალითად, UML-ის MsVisio ინსტრუმენტს, Visual Studio .NET პაკეტის სრულ ვერსიას [5].

მეორე ეტაპზე მუშავდება მონაცემთა ბაზის, მისი ცხრილებისა და ინდექსური ფაილების კომპონენტები, რომლებიც კონტექსტურ დიაგრამებში ერთიანდება და მზადდება სერვერ-მონაცემთა ბაზებში განსათავსებლად. ცენტრალურ სერვერ-ბაზასთან ერთად გათვალისწინებული უნდა იყოს სარეზერვო, არქივირებული ბაზის არსებობა. სერვერ-ბაზების ადმინისტრირების რეგლამენტი დამატებით იქნება შემუშავებული სისტემის დანერგვის შემდეგ, გამომდინარე ფუნქციური ავტომატიზებული სამუშაო ადგილების რაოდენობისა და მენეჯმენტის მიზნებიდან. მესამე დონე ეხება სისტემაში მუშა ფაილების (EXE) პაკეტების შექმნას. საყურადღებოა ასევე სისტემური დამხმარე და რესურსული (მაგ., DLL, RES და სხვ.) ფაილების შექმნა.

გარდა სპეციალური პროგრამული და საბაზო ფაილების დამუშავებისა, სისტემაში ფართოდ გამოიყენება სტანდარტული და სერვისული ბიბლიოთეკების პროგრამები, რაც ხორციელდება ობიექტ-ორიენტირებული და სტრუქტურული დაპროგრამების პრინციპებით. კომპონენტების თვალთახედვის დონე ითვალისწინებს აუცილებლად მათ ტესტირებას და საბოლოო მუშა სახემდე მიყვანას, აგრეთვე დოკუმენტირებისა და ინსტრუქციების მომზადებას მომხმარებლებისათვის, სისტემისა და მონაცემთა ბაზების ადმინისტრატორებისათვის.

პენტაედრული მოდელის ბოლო ეტაპი მდგომარეობს განაწილებული სისტემის გამოთვლითი რესურსების კონფიგურაციის დადგენისა და სისტემის პროგრამული კომპონენტების კონტექსტურად განთავსებაში კომპიუტერული ქსელის კვანძებში. შესაბამისად, ავტომატიზებული სისტემის არქიტექტურა განლაგების თვალთახედვით  $A_d$  უნდა შეიცავდეს კლიენტ-სერვერული და მთლიანად განაწილებული სისტემის აღმწერ მოდელებს:

$$A_d = \langle \{Z_i\}, \{R_i\}, \{C_i\} \rangle, \text{ სადაც}$$

$\{Z_i\}$  - კვანძების სიმრავლეა;  $\{R_i\}$  - კვანძებს შორის მიმართებებია (დამოკიდებულების, ასოციაციის);  $\{C_i\}$  - კვანძებზე განლაგებული პროგრამული კომპონენტების სიმრავლე.

პროგრამული კომპონენტების ქსელში განთავსების ამოცანა მნიშვნელოვანია, რამეთუ ქსელის ეფექტური მუშაობა სწორედ მასზეა დამოკიდებული. აქ საყურადღებოა როგორც პროგრამული, ასევე მონაცემთა ფაილების კლიენტ-სერვერ არქიტექტურის კვანძებში განლაგების საკითხი.

მრავალდონიანი, განაწილებული სისტემებისათვის მიზანშეწონილია კლიენტ-სერვერის ე.წ. „თხელი კლიენტის“ ვარიანტის განვიხილავა. მონაცემთა ბაზებისა და პროგრამული ფაილების ძირითადი ერთობლიობა ცენტრალურ სერვერზე იქნება განთავსებული, ხოლო კლიენტ კვანძებში სპეციალური საინტერფეისო პროგრამებისა და ლოკალური ფაილების პაკეტები განლაგდება. ცხადია უნდა შერჩეულ იქნას ყველაზე ეფექტური ვარიანტი პროგრამული კომპონენტების განთავსებისა ქსელში. ეფექტურობის ქვეშ უნდა ვიგულისხმოთ მთლიანი სისტემის მოქნილი და საიმედო მუშაობა, საერთო გამოყენების რესურსების (არხები, მონაცემთა ფაილები, პროგრამები და სხვ.) უკონფლიქტო მართვა, ტრანზაქციათა შესრულებისას ჩიხების გამორიცხვა, მომხმარებელთა მოთხოვნების დაკმაყოფილება დროის მისაღებ ინტერვალებში, მონაცემთა დუბლირების მინიმიზაცია და ა.შ. ამიტომ, არქიტექტურის ფორმირებისას ერთ-ერთ ამოცანას წარმოადგენს განაწილებული სისტემის ფიზიკური რესურსების, პროგრამულ და მონაცემთა კომპონენტების განთავსების მოდელის აგება სისტემური პეტრის ქსელების გამოყენებით, შემდეგ მათი ცალკეული ელემენტებისათვის რაოდენობრივი მახასიათებლების შერჩევა (დასაშვებ ცვლილებათა დიაპაზონებით) და ამ მოდელის ანალიზი, ქსელის შესასვლელზე განსხვავებული განაწილების კანონებით მიწოდებული მოთხოვნების შემთხვევაში, განსაკუთრებით პუასონის, ექსპონენციალური და ნორმალური განაწილებებისათვის.

გამოკვლევის შედეგები ანალიზის საფუძველზე მოგვცემს შესაძლებლობას განისაზღვროს მრავალდონიან განაწილებულ სისტემებში კლიენტ-სერვერ კონფიგურაციისათვის პროგრამული კომპონენტების ეფექტური განთავსების ალგორითმი, მოხდეს ვარიანტების რაოდენობრივი შეფასება და დადგინდეს მათ შორის ოპტიმალური. სისტემის დამუშავების პროცესი ითვალისწინებს არქიტექტურის მიმდევრობით დაზუსტებას პრეცედენტების ანალიზის, იტერაციული და ინკრემენტული კვლევის საფუძველზე. რაც გულისხმობს იმას, რომ მოდელის დამუშავება ხდება ინკრემენტულად (უმატებთ რა თითო ახალ ფრაგმენტს ყოველ ერთ ჯერზე) და იტერაციულად (ვიმეორებთ რა პროექტირების პროცესს ყოველი ახალი გაუმჯობესების შემდეგ).

### 3. დასკვნა

პროგრამული სისტემის დაპროექტების მოყვანილი გაფართოება, ინტეგრირებული კვლევის მექანიზმის საფუძველზე, საშუალებას აძლევს სისტემის დამპროექტებელს აგებული მოდელის საფუძველზე განახორციელოს პროცესების კომპიუტერული ანალიზი და მოახდინოს ვარიანტების რაოდენობრივი შეფასება მათგან ოპტიმალურის ამოსარჩევად.

### ლიტერატურა:

1. ახობაძე მ., ბოსიკაშვილი ზ., გოგიჩაიშვილი გ., სურგულაძე გ., სუხიაშვილი თ., ღვინფაძე გ. სასამართლო საქმეთა წარმოების ქსელური მართვის ავტომატიზებული სისტემა. სტუ, თბ., 2006
2. სუხიაშვილი თ. ავტომატიზებული მართვის თეორიული საფუძვლები. სტუ, თბ., 2005
3. . . . . UML-2 . . . . . 2008
4. სურგულაძე გ.. დაპროგრამების ვიზუალური მეთოდები და ინსტრუმენტები (UML, MsVisio, C++Builder) სტუ, თბ., 2005.

## DESIGNING OF THE MULTILEVEL DISTRIBUTED CONTROL SYSTEMS

Temur Sukhiashvili

Georgian Technical University

### Summary

In the represented article there is considered the expanded version of designing the software systems, defined by UML technology. This method allows a construction of the multilevel distributed control systems on the basis of five representations are considered stages of. The resulted approach provides creation of the integrated mechanism of research which allows a system's designer to make the computer analysis of processes on the basis of the constructed models and to make a quantitative estimation of variants for revealing optimum.

UML