

არაფუნქციონალური მოთხოვნების მოდელირება ბიზნეს-პროცესების მართვისას

თეიმურაზი სუხიაშვილი, ირაკლი შურაია
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

მართვის თანამედროვე პროგრამულ სისტემებს უნდა გააჩნდეთ ფართო ფუნქციონალური შესაძლებლობები, კერძოდ მრავალ მომხმარებელთან ერთდროულად მუშაობის უნარი, სისტემის გამოყენების მოხერხებულობა, უზრუნველყოფილი იყოს ჩართული ონლაინ დახმარებით, საჭირო ინსტრუქციების მისაღებად. ამიტომ მართვის სისტემის დამუშავებისას სასურველია საპრობლემო სფეროს როგორც ფუნქციონალური, ისე არაფუნქციონალური (საიმედოება, წარმადობა, უშიშროება) მოთხოვნების მოდელირება. ყოველ საპრობლემო სფეროს აქვს მხოლოდ მისთვის დამახასიათებელი თავისებურებანი, რაც გამოხატულებას პოულობს ფუნქციონალურ მოთხოვნებში (Business UseCase). რაც შეეხება არაფუნქციონალურ მოთხოვნებს. მათი მისადაგება სხვა სისტემებთან შესაძლებელია საკმაოდ დიდი ალბათობით. ამიტომ, მათი რეალიზებისათვის დამუშავების პროცესში შეგვიძლია ვისარგებლოთ დამუშავებული მსგავსი სისტემის კომპონენტებით, რითაც დავზოგავთ დროს და ხარჯებს სისტემის დასამუშავებლად.

საკვანძო სიტყვები: მართვის სისტემა. ბიზნეს-პროცესი. ფუნქციონალური მოთხოვნა. არაფუნქციონალური მოთხოვნა. საიმედოება. წარმადობა. უსაფრთხოება.

1. შესავალი

თანამედროვე პირობებში რთული ორგანიზაციული მართვის ავტომატიზებული სისტემა აღჭურვილი უნდა იქნას მთელი რიგი ფუნქციონალური შესაძლებლობებით. კერძოდ, სისტემა უნდა უზრუნველყოფდეს მუშაობის მრავალმომხმარებლის რეჟიმს, რამდენადაც მსგავსი სისტემები მუშავდება განაწილებული სამუშაო ადგილებით, კლიენტ-სერვერ ორგანიზაციით. გამოყენების მოხერხებულობა, სისტემის მომხმარებლის ინტერფეისი უნდა იყოს მარტივი და არ მოითხოვდეს მომხმარებლისაგან, რომელსაც აქვს კომპიუტერული განათლება, დამატებით შესწავლას. სისტემის ყოველი ფუნქცია უნდა უზრუნველყოფილი იყოს ჩართული ონლაინ დახმარებით, რომელიც უნდა შეიცავდეს ინსტრუქციას სისტემასთან მუშაობისათვის. ამიტომ, მართვის ავტომატიზებული სისტემის დამუშავება მოითხოვს როგორც ფუნქციონალური (Business Use Case), ისე არაფუნქციონალური მოთხოვნების მოდელირებას. მაგალითად, უმაღლეს სასწავლებელში სტუდენტთა დასწრებისა და მოსწრების აღრიცხვის კომპიუტერული სისტემისათვის, რომელშიც თითოეული მომხმარებლისათვის (პროფესორი, სტუდენტი, დეკანატის მუშაკი) შექმნილია კომპიუტერული სამუშაო ადგილი, უნდა გათვალისწინებული იქნას **საიმედოება** (სისტემა უნდა იყოს სამუშაო მდგომარეობაში 24 სთ კვირაში 7 დღე, უქმად დგომის დრო არ უნდა აღემატებოდეს 10%. საიმედო მუშაობის საშუალო დრო არ უნდა აღემატებოდეს 300 სთ-ს), **წარმადობა** (სისტემა უნდა აკავებდეს დაახლოებით 2000 მომხმარებელს, რომლებიც ერთდროულად იმუშავებენ მონაცემთა ცენტრალურ ბაზასთან, და დაახლოებით 500 მომხმარებელს, რომლებიც ერთდროულად იმუშავებენ ლოკალურ სერვერებთან, სისტემა უნდა უზრუნველყოფდეს მონაცემთა ბაზასთან მიმართვას დაყოვნების დროით არა უმეტეს 10 წმ.), **უშიშროება** (სისტემა არ უნდა აძლევდეს საშუალებას სტუდენტებს შეცვალონ ნებისმიერი მონაცემი, ასევე პროფესორებმა მოახდინონ კონკრეტული ჯგუფის სტუდენტთა მონაცემების მოდიფიცირება, რომლებიც სხვა პროფესორების მიერ არის არჩეული. მხოლოდ საგნის წამყვან პროფესორს აქვთ უფლება დაუსვან შეფასებები

სტუდენტებს. მხოლოდ რეგისტრატორს შეუძლია შეცვალოს ნებისმიერი ინფორმაცია სტუდენტების შესახებ).

ყოველივე ეს მიგვანიშნებს იმ ფაქტზე, რომ თავიდან მოვახდინოთ ბიზნეს-პროცესების ფუნქციების მოდელირება, რათა მომავალი სისტემის ფუნქციონალური შესაძლებლობებიდან დადგინდეს არაფუნქციონალური მოთხოვნები და მოვახდინოთ მათი მოდელირება. ასეთ მიდგომას აქვს გარკვეული უპირატესობები. კერძოდ, სისტემის დამუშავების პროცესში შეგვიძლია ვისარგებლოთ დამუშავებული მსგავსი სისტემის კომპონენტებით მისი არაფუნქციონალური მოთხოვნების რეალიზებისათვის, რითაც დავზოგავთ დროს და ხარჯებს სისტემის დამუშავებისათვის. ხოლო თუ დამუშავების პროცესში მოხდება ცვლილება ვიპოვოთ მისთვის მისადაგებული კომპონენტები და ჩავანაცვლოთ არსებული.

ამის განხორციელებისათვის საჭიროა გარკვეული მექანიზმი ფუნქციონალური და არაფუნქციონალური მოთხოვნების ურთიერთშეთავსებისათვის საწყისი ვერსიის მისაღებათ.

2. ძირითადი ნაწილი

სისტემისადმი ფუნქციონალური მოთხოვნები მოდელირდება და დოკუმენტირდება პრეცედენტების (Use Case) მეშვეობით, რომლებიც ითვალისწინებენ შემდეგს:

- პრეცედენტი აფიქსირებს შეთანხმებას პროექტის მანაწილეებისა სისტემის ქცევასთან;
- პრეცედენტი აღწერს სისტემის ქცევას სხვადასხვა პირობებისას, როდესაც სისტემა პასუხობს ერთერთი მონაწილის დაკვეთას, რომელიც იწოდება ძირითად მოქმედ პირად.
- ძირითადი მოქმედი პირი ახდენს სისტემასთან ურთიერთქმედების ინიცირებას, იმისათვის რომ მიღწეულ იქნას გარკვეული მიზანი. სისტემა პასუხობს, იცავს რა ყველა მონაწილის ინტერესებს.

პრეცედენტების აღწერისას არსებობს სიზუსტის ოთხი დონე:

- მოქმედი პირები და მიზნები (დგინდება ყველა მოქმედი პირები და მათი მიზნები, რომელსაც უზრუნველყოფს სისტემა);
- პრეცედენტის მოკლე დახასიათება (ერთ სტრიქონში) ან მოვლენათა ძირითადი ნაკადი (მოსალოდნელი შეცდომების ანალიზის გარეშე);
- უარყოფის (შეფერხების) პირობები (მოვლენათა ძირითად ნაკადში მოსალოდნელი შეცდომების აღძვრის ადგილების ანალიზი);
- შეფერხებების დამუშავება (მოვლენათა ალტერნატიული ნაკადების აღწერა).

მოყვანილი პოსტულატები აღიწერება მოღვაწეობის დიაგრამებით. კლასები, რომლებიც იქმნება ბიზნეს-პროცესების ანალიზის ეტაპზე, გამოხატავენ სისტემისადმი ფუნქციონალურ მოთხოვნებს და ახდენენ საგნობრივი სფეროს მოდელირებას. ანალიზის კლასების ერთობლიობა წარმოადგენს სისტემის საწყის კონცეპტუალურ მოდელს.

ბიზნეს – მოდელიდან პრეცედენტების მოდელის საწყის ვერსიაზე გადასვლისას სრულდება შემდეგი წესები:

- ყოველი შემსრულებლისათვის ბიზნეს – ანალიზის მოდელში, რომელიც პერსპექტივაში განდება ახალი სისტემის მომხმარებელი, პრეცედენტების მოდელში იქმნება მოქმედი პირი ასეთივე დასახელებით. მოქმედი პირების შემადგენლობაში ჩაირთვება ასევე გარე სისტემები, რომლებიც ბიზნეს – პროცესებში ინფორმაციის წყაროს პასიურ როლს თამაშობენ;
- პრეცედენტი მოცემული მოქმედი პირისათვის იქმნება შესაბამისი შემსრულებლის მოთხოვნათა ანალიზის საფუძველზე (უმარტივეს შემთხვევაში შემსრულებლის ყოველი

ოპერაციისათვის იქმნება პრეცედენტი, რომელიც მოახდენს მოცემული ოპერაციის რეალიზებას სისტემაში).

მოდელის ასეთი საწყისი ვერსია აღწერს სისტემის მინიმალურ ვარიანტს, რომლის მომხმარებლები არიან მხოლოდ ბიზნეს – პროცესების შემსრულებლები. თუ მომავალში სისტემის განვითარებისას მისი უშუალო მომხმარებლები იქნებიან ბიზნეს – პროცესების მოქმედი პირები, პრეცედენტთა მოდელი დაიწყებს მოდიფიცირებას.

მოყვანილი წესების გამოყენებას მივყავართ ახალი მოქმედი პირების გამოჩენასთან სისტემის საწყისი ვერსიისათვის, შესაბამისად ახალი მმართველი (Control) კლასების ფორმირებასთან, რომლებიც უზრუნველყოფენ ობიექტების ქცევის კოორდინაციას სისტემაში. როგორც წესი, ყოველი პრეცედენტისათვის განისაზღვრება ერთი მმართველი კლასი. ყოველი წყვილისათვის “მოქმედი პირი – პრეცედენტი” განისაზღვრება ერთი მოსაზღვრე კლასი. მოსაზღვრე კლასები (Boundary) ასრულებენ შუამავლების როლს გარე ობიექტებსა და სისტემას შორის ურთიერთქმედებისას. მოსაზღვრე კლასების ტიპებია: მომხმარებლის ინტერფეისი(ინფორმაციის გაცვლა მომხმარებელთან ინტერფეისის დეტალების გარეშე – ლილაკები, ცხრილები, ფანჯრები), სისტემური ინტერფეისი და აპარატული ინტერფეისი(გამოყენებული პროტოკოლები მათი რეალიზაციის გარეშე). იმისათვის, რომ მარეალიზებული კომპონენტები უფრო ზუსტად შევარჩიოთ დამუშავებულ მსგავს სისტემებში, შეგვიძლია ვისარგებლოთ ურთიერთქმედების დიაგრამებით. პირველ რიგში იგება დიაგრამა, რომელიც აღწერს მონაცემთა ძირითად ნაკადს (ერთი ან რამოდენიმე) და მის დაქვემდებარებულ ნაკადებს. ყოველი ალტერნატიული ნაკადისათვის იგება ცალკე დიაგრამა. მაგალითად, შეცდომების დამუშავება, შესრულების დროის კონტროლი, არასწორად შეტანილი მონაცემების დამუშავება.

იმისათვის, რომ უზრუნველყოთ მრავალი მომხმარებლის ერთდროული მუშაობა, რომელთაგან თვითუფილი ბადებს სისტემაში დამოუკიდებელ პროცესს. უნდა გამოვიყენოთ აქტიური კლასები – კლასი სტერეოტიპით „process“ და „thread“. აქტიური კლასი ფლობს საკუთარ პროცესს ან ნაკადს და შეუძლია მმართველი ზემოქმედების ინიცირება. კავშირები პროცესებს შორის მოდელირდება როგორც დამოკიდებულება. ნაკადები შესაძლებელია არსებობდეს მხოლოდ პროცესების შიგნით, ამიტომ კავშირები პროცესებსა და ნაკადებს შორის მოდელირდება როგორც კომპოზიცია. პროცესებისა და ნაკადების რეალიზაცია უზრუნველყოფილია ოპერაციული სისტემის საშუალებით.

3. დასკვნა

როგორც ორგანიზაციული სისტემების მართვის ავტომატიზებული სისტემის დამუშავებისას თავიდან სასურველია დამუშავდეს მათში მიმდინარე ბიზნეს-პროცესების მოდელი და მოხდეს მისი შეთავსება არაფუნქციონალურ მოთხოვნებთან საწყისი ვერსიის მისაღებათ.

ლიტერატურა:

1. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. Серия “Объектно- ориентированные технологии в программировании”. Москва, 2004
2. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем. Москва. Финансы и статистика. 2006
3. სუხიაშვილი თ., ბიზნეს - პროცესების მართვის განაწილებული, მრავალდონიანი სისტემების კონფიგურაციის მოდელირება. საქართველოს ტექნიკური უნივერსიტეტის 90 წლისთავისადმი მიძღვნილი საერთაშორისო სამეცნიერო კონფერენციის შრომები. თბილისი, 19-21 სექტემბერი, 2012. გვ.423-425.

MODELING OF NEFUNKTSIONALNY REQUIREMENTS AT MANAGEMENT OF BUSINESS PROCESSES

Sukhiashvili Teimuraz, Shurgaia Irakli
Georgian Technical University

Summary

Modern program control systems have to possess wide functionality, in particular ability of simultaneous work with many users, convenience of use, security built in online support, for receiving necessary to the instruction. Therefore at development of the system of management it is desirable to model both functional(Busines UseCase), and nonfunctional (reliability, productivity, safety) requirements. Each problem environment possesses features peculiar only to them that is shown in functional requirements that касается nonfunctional requirements their prepodoby to other system is possible very high probability. Therefore, for their realization in process development can use components of the developed similar system, thereby with we save time and costs of system development.

МОДЕЛИРОВАНИЕ НЕФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ ПРИ УПРАВЛЕНИИ БИЗНЕС-ПРОЦЕССАМИ

Сухиашвили Т., Шургая И.
Грузинский Технический Университет

Резюме

Современные программные системы управления должны обладать широкими функциональными возможностями, в частности, способностью одновременной работы с многими пользователями, удобством использования, обеспеченностью встроенными онлайн поддержками для получения нужных инструкции. Поэтому при разработке системы управления желательно моделировать как функциональные (Business Use Case), так и нефункциональные (надежность, производительность, безопасность) требования. Каждая проблемная сфера обладает особенностями, свойственные только ей, что проявляется в функциональных требованиях (Business Use Case). Что касается нефункциональных требований их приложении к другим системам возможно с весьма большой вероятностью. Поэтому, для их реализации в процессе разработке можно воспользоваться компонентами разработанной подобной системы, тем самым будут сэкономлены время и затраты на разработку системы.