

მონაცემთა ბაზის ლოგიკური სქემის დამუშავება ბიზნესპროცესების კომპიუტერული მართვისათვის

თეიმურაზ სუხიაშვილი. ირაკლი შურღაია
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

ობიექტ-ორიენტირებული სისტემების ძირითად სამშენებლო ბლოკს წარმოადგენს ობიექტი - კლასი. ამიტომ ბიზნეს-პროცესების მართვის კომპიუტერული სისტემების დამუშავებისას, იმისათვის, რომ უზრუნველყოფილი იყოს ფუნქციონალური მოთხოვნების ეფექტური შესრულება და ამავე დროს, არაფუნქციონალური მოთხოვნების (გამოყენების მოხერხებულება, საიმედოობა, წარმადობა, უსაფრთხოება) სრულად დაკმაყოფილება, დიდი მნიშვნელობა აქვს კლასების გამოვლენას, მოვალეობების რაციონალურ განაწილებას და მათ შორის რეალური მიმართებების დადგენას. სტატიამ განიხილა უნივერსიტეტის ბიზნესპროცესებისთვის მონაცემთა ბაზის ლოგიკური სქემის დადგენის საშუალება, რომელიც ეყრდნობა რაციონალური უნიფიცირებული პროცესის (RUP) ფარგლებში მოდელირების უნიფიცირებული ენის (UML) გამოყენებას.

საკვანძო სიტყვები: ბიზნესპროცესი. UML. პრეცედენტების დიაგრამა. სცენარი. კომუნიკაცია. სინქრონიზაცია. კლასი. ობიექტი.

1. შესავალი

მონაცემთა ბაზის ლოგიკური სქემის დადგენა იწყება ბიზნესპროცესების ანალიზის ეტაპზე. საფუძველს ამისათვის წარმოადგენს პრეცედენტების დიაგრამა (Use case), რომელშიც ასახულია როგორც ფუნქციონალური, ისე არაფუნქციონალური (სისტემის საიმედოობა, წარმადობა, უსაფრთხოება) მოთხოვნები სისტემისადმი. ცალკეული პრეცედენტების მოქმედების სცენარების (ძირითადი, ალტერნატიული) შესწავლისა და ანალიზის საფუძველზე უნდა მოვახდინოთ ძირითადი აბსტრაქციების იდენტიფიკაცია, რომლებიც მონაწილეობს აღნიშნულ სცენარებში.

ჩვენი მაგალითი ეხება უმაღლეს სასწავლებელში სტუდენტთა კურსებზე რეგისტრაციისა და მოსწრების აღრიცხვის ამოცანას. „კურსებზე დარეგისტრირების“ გამოყენებითი შემთხვევის სპეციფიკაცია ასე გამოიყურება:

დასახელება: კურსებზე დარეგისტრირება.

მოკლე აღწერა: მოცემული გამოყენებითი შემთხვევა საშუალებას აძლევს სტუდენტს დარეგისტრირდეს შემოთავაზებულ კურსებზე მიმდინარე სემესტრში.

ძირითადი სცენარი:

1. სტუდენტი მიდის დეკანატის მოსამსახურესთან და გადასცემს მას შევსებულ ფორმას კურსებზე რეგისტრაციისათვის.

2. დეკანატის მოსამსახურე ადასტურებს ფორმის შევსების სისწორეს.

3. დეკანატის მოსამსახურე ადასტურებს, რომ სტუდენტმა შეასრულა წინასწარი მოთხოვნები ყოველი არჩეული კურსისათვის (გარკვეული კურსების გავლა), ასევე თავისუფალი ადგილების არსებობას.

4. დეკანატის მოსამსახურე შეაქვს სტუდენტის გვარი არჩეულ კურსების სიაში.

5. დეკანატის მოსამსახურე ავსებს სტუდენტის გრაფიკს კურსებზე მიმდინარე სემესტრში და გადასცემს მას სტუდენტს.

ალტერნატიული სცენარი:

2ა. არასწორად არის შევსებული რეგისტრაციის ფორმა.

დეკანატის მოსამსახურე უბრუნებს სტუდენტს ფორმას შეცდომების გასწორებისათვის.

3ა. არ არის შესრულებული წინასწარი მოთხოვნები ან კურსი შევსებულია.

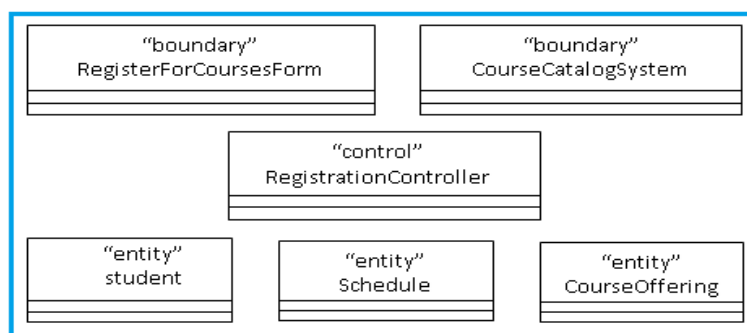
თუ დეკანატის მოსამსახურე აღმოაჩენს, რომ სტუდენტს არ შეუძლებია აუცილებელი წინასწარი მოთხოვნები ან მის მიერ არჩეული კურსი შევსებულია (უკვე ჩაწერილია 10 სტუდენტი), მაშინ იგი სთავაზობს სტუდენტს შეიცვალოს თავისი არჩევანი ან წაილოს ფორმა და დაუბრუნდეს მას მოვლიანებით.

ძირითადი (არაფორმალური) საშუალება არსების იდენტიფიკაციისა – ეს აბსტრაქციების პონაა, რომლებიც აღწერენ ფიზიკურ ან მატერიალურ ობიექტებს, პროცესებს და მოვლენებს, ადამიანთა როლებს, ორგანიზაციებს და სხვა მცნებებს. ერთადერთ ფორმალურ საშუალებათ არსთა იდენტიფიკაციისა არის საგნობრივი სფეროს ტექსტური აღწერილობების ანალიზი, აღწერიდან არსებითი სახელების გამოყოფა და მათი არჩევა როგორც “კანდიდატებისა” აბსტრაქციების როლზე. ყოველ არსს უნდა გააჩნდეს დასახელება, გამოსახული არსებითი სახელით მხოლოდით რიცხვში. თუ მივეყვებით ამ რეკომენდაციებს, განვსაზღვრავთ რეგისტრაციის სისტემისათვის ანალიზის ხუთ კლასს სტუდენტი(Student), პროფესორი (Professor), სასწავლო გრაფიკი (Schedule), კურსი (Course), შეთავაზებული კურსი (CourseOffering).

მოვალეობების განსაზღვრისათვის მოცემულ ეტაპზე კლასებს ვყოფთ სამი ტიპის კლასებად:

- **მოსაზღვრე კლასები (Boundary)** - შუამავლები გარე ობიექტებსა და სისტემას შორის ურთიერთქმედებისას. როგორც წესი, ყოველი წყვილისათვის „მოქმედი პირი – პრეცედენტი” განისაზღვრება ერთი მოსაზღვრე კლასი.
- **კლასები – არსი (Entity)** – დასამუშავებელი სისტემის ძირითადი აბსტრაქციები (ცნებები). კლასები – არსის გამოვლენის წყაროებია აბსტრაქციები, რომლებიც პასუხს აგებს მონაცემთა შენახვასა და მანიპულირებაზე;
- **მმართველი კლასები (Control)** – უზრუნველყოფს ობიექტების ქცევის კოორდინაციას სისტემაში. შეიძლება არ იყოს ზოგიერთ პრეცედენტში და შეზღუდულია მარტივი მანიპულაციებით შენახულ მონაცემებთან. როგორც წესი, ყოველი პრეცედენტისათვის განისაზღვრება ერთი მმართველი კლასი.

ანალიზის კლასები გამოხატავს სისტემისადმი ფუნქციურ მოთხოვნებს და ახდენს საგნობრივი სფეროს მოდელირებას. ანალიზის კლასების ერთობლიობა სისტემის საწყისი კონცეპტუალური მოდელია. თითოეული პრეცედენტისათვის უნდა მოხდეს მასში მონაწილე კლასების იდენტიფიკაცია და გადავანაწილოთ მოვალეობები კლასებს შორის. კლასების ერთობლიობა შემთხვევისთვის „კურსებზე დარეგისტრირება” რეალიზებაში მოყვანილია 1-ელ ნახაზზე.



ნახ.1

გამოყოფილი სამი ტიპის კლასების დანიშნულებიდან გამომდინარე, შესაძლებელია დავახსნათოთ მოვალეობები მათ შორის:

- მოსაზღვრე კლასები პასუხს აგებენ ურთიერთქმედებაზე სისტემის გარე სამყაროსთან (მოქმედი პირები – სტუდენტი - RegisterForCoursesForm, კურსების კატალოგი - CoursesCatalogSystem);

- კლასი არსები პასუხს აგებენ მონაცემთა შენახვასა და მანიპულირებაზე(Student, Schedule, CourseOffering);
- მმართველი კლასები კოორდინაციას უწევენ პრეცედენტების მოვლენათა ნაკადებს(RegistrationController).

"entity" Student	"entity" CourseOffering	"entity" Schedule
Name Address Student()	Number: String = "100" startTime: Time endTime: Time days: Enum numStudents: Int	semester
// add schedule() // get schedule() // delete schedule() // has pre.requisites()	// add student() // remove student() // still open?()	// delete() // submit() // save() // any conflicts?() // create with offerings() // update with new selections()

ნახ.2

მოვალეობების უფრო დეტალური განაწილება (კლასების ოპერაციების სახით) სრულდება ურთიერთქმედების დიაგრამებით. პირველ რიგში იგება დიაგრამა, რომელიც აღწერს მონაცემთა ძირითად ნაკადს (ერთს ან რამდენიმეს) და მის დაქვემდებარებულ ნაკადებს. ყოველი ალტერნატიული ნაკადისათვის იგება ცალკე დიაგრამა. მაგალითად: შეცდომების დამუშავება, შესრულების დროის კონტროლი, არასწორად შეტანილი მონაცემების დამუშავება.

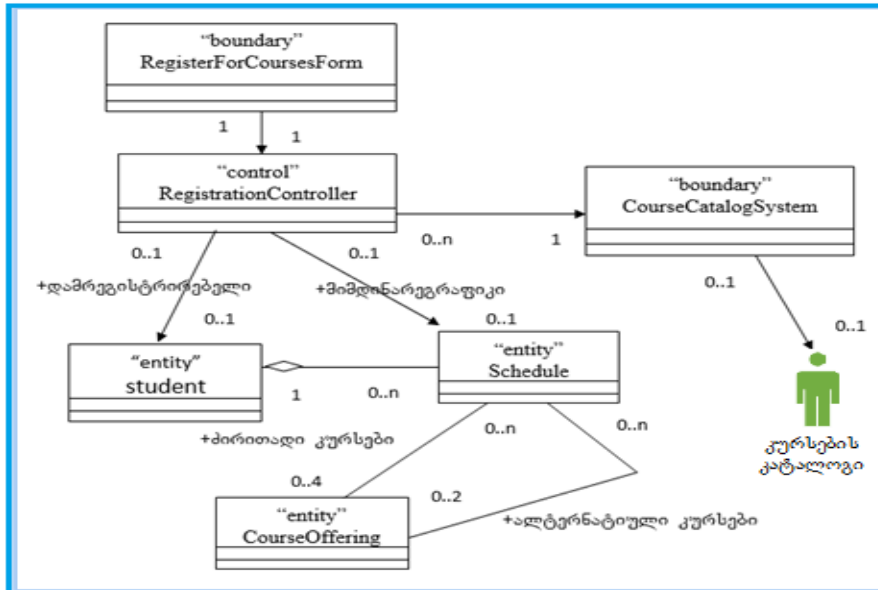
ურთიერთქმედების დიაგრამის აგების პროცესში ვახდენთ შეტყობინებების შესაბამისობას კლასების ოპერაციებთან, რის შედეგადაც ჩნდება „ანალიზის“ ოპერაციები. ანალიზის კლასების ატრიბუტები განისაზღვრება საგნობრივი სფეროს შესახებ ცოდნისა და სისტემისადმი მოთხოვნების საფუძველზე. კლასებს შორის კავშირები (ასოციაცია) განისაზღვრება კოოპერაციის დიაგრამების ანალიზის საფუძველზე. თუ ორი ობიექტი ურთიერთქმედებს (ცვლიან შეტყობინებებს), მათ შორის კოოპერაციის დიაგრამაზე უნდა არსებობდეს კავშირი (ურთიერთქმედების გზა), რომელიც გარდაიქმნება ორმიმართულებიან ასოციაციაში შესაბამის კლასებს შორის. თუ შეტყობინებები რომელიმე ობიექტებს შორის გადაიცემა მხოლოდ ერთი მიმართულებით, მაშინ შესაბამისი ასოციაციისათვის ინიშნება მიმართულების ნავიგაცია.

გარდა ამისა, ანალიზდება და ზუსტდება ასოციაციები კლას-არსებს შორის. დგინდება ასოციაციის სიმძლავრე, შეიძლება გამოყენებულ იქნას მრავლობითი ასოციაციები, აგრეგაციები, განზოგადება და ასოციაცია კლასები.

მე-3 ნახაზზე ნაჩვენებია კლასების გაფართოებული დიაგრამა პრეცედენტისათვის „კურსებზე დარეგისტრირება“ (ატრიბუტებისა და ოპერაციების გარეშე). ასოციაციები მოსაზღვრე და მმართველ კლასებს შორის, ასევე მმართველ კლასებსა და კლას-არსებს შორის შემოტანილია კოოპერაციის დიაგრამების ანალიზის საფუძველზე და მდგრადი სტრუქტურული (სემანტიკური) კავშირები არსებს შორის გამოხატავს კავშირებს, რომლებიც დინამიურად აღიძვრება შესაბამის ობიექტებს შორის მართვის ნაკადში (დანართის მუშაობის პროცესში). რამდენადაც ასოციაციისათვის ეს არ არის დამახასიათებელი, მომავალში(პროექტირების პროცესში) ისინი შესაძლებელია გარდაიქმნენ დამოკიდებულებაში.

2. ძირითადი ნაწილი

მონაცემთა ბაზის ლოგიკური სქემის დამუშავების პროცესი გრძელდება დაპროექტების ეტაპზე. იგი გულისხმობს: საპროექტო კლასების დეტალიზირებას, ოპერაციებისა და ატრიბუტების დაზუსტებას, კლასების მდგომარეობების მოდელირებას, კლასებს შორის კავშირების დაზუსტებას.



ნახ.3

2.1. საპროექტო კლასების დეტალიზება

ყოველი მოსაზღვრე კლასი გარდაიქმნება კლასების გარკვეულ ნაკრებში თავისი დანიშნულებიდან გამომდინარე. ეს შესაძლებელია იყოს მომხმარებლის ინტერფეისის ელემენტების ნაკრები, დამოკიდებული დამუშავების გარემოს შესაძლებლობებზე, ან კლასების ნაკრები, რომელიც ახდენს სისტემურ ან აპარატურ ინტერფეისს.

კლასი – არსი, წარმადობისა და მონაცემთა დაცვის მოსაზრებებიდან გამომდინარე შესაძლებელია დაიყოს რიგ კლასებად. საფუძველს დაყოფისათვის წარმოადგენს კლასებში ატრიბუტების სხვადასხვა სიხშირით ან ხედვით გამოყენება. ასეთი ატრიბუტები, როგორც წესი, გამოიყოფიან ცალკე კლასებათ.

მმართველი კლასები, რომლებიც ახდენს ინფორმაციის უბრალო გადაცემას მოსაზღვრე კლასებიდან არსებზე, შეიძლება გაძევებულ იქნას. შენარჩუნებული იქნება კლასები, რომლებიც ასრულებენ არსებით სამუშაოს მონაცემთა ნაკადების სამართავად (ტრანზაქციების მართვა, განაწილებული დამუშავება და ა.შ.).

2.2. ატრიბუტებისა და ოპერაციების დაზუსტება

კლასების მოვალეობები, რომლებიც განისაზღვრა ანალიზის პროცესში, „ანალიზის“ ოპერაციების სახით, გარდაიქმნება ოპერაციებში, რომლებიც რეალიზებული უნდა იყოს კოდში.

ამისათვის, ყოველ ოპერაციას ენიჭება მოკლე სახელი, რომელიც ახასიათებს მის შედეგს. განისაზღვრება ოპერაციების მთელი სიგნატურა (UML ენაში მიღებული ნოტაციის შესაბამისად). იქმნება ოპერაციის მოკლე დასახელება, მისი ყველა პარამეტრის შინაარსის გათვალისწინებით. იქმნება ოპერაციათა ხილვადობა: public, private ან protected. განისაზღვრება ოპერაციათა მოქმედების არე: ვგზემპლარი (ობიექტის ოპერაცია) ან კლასიფიკატორი (კლასების ოპერაცია).

შესაძლებელია აღიწეროს ოპერაციის შესრულების ალგორითმი (აქტიურობის ან ურთიერთქმედების დიაგრამებით).

თუ სისტემაში გვაქვს მდგომარეობისაგან დამოკიდებული ობიექტები ქცევის რთული დინამიკით, მაშინ კლასების ატრიბუტების დასაზუსტებლად (ნახ.4), ატრიბუტის დასახელების გარდა მიეთითება მისი ტიპი და მნიშვნელობა დუმილით (არააუცილებელი).

“entity” Student
-name : string -address : string <<class>> - nextAailID : Int -studentId : int -dateOfBirth : Date
+ getTuition() : double + addSchedule(theSchedule : Schedule) + getSchedule(forSemester : Semester) : Schedule + deleteSchedule(forSemester : Semester) + hasPrerequisites(forCourseOffering : CourseOffering) : boolean // passed(the CourseOffering : CourseOffering) : Boolean <<class>> + getNextAailID() : Int + getStudentID() : Int + getName() : string + getAddress() : string

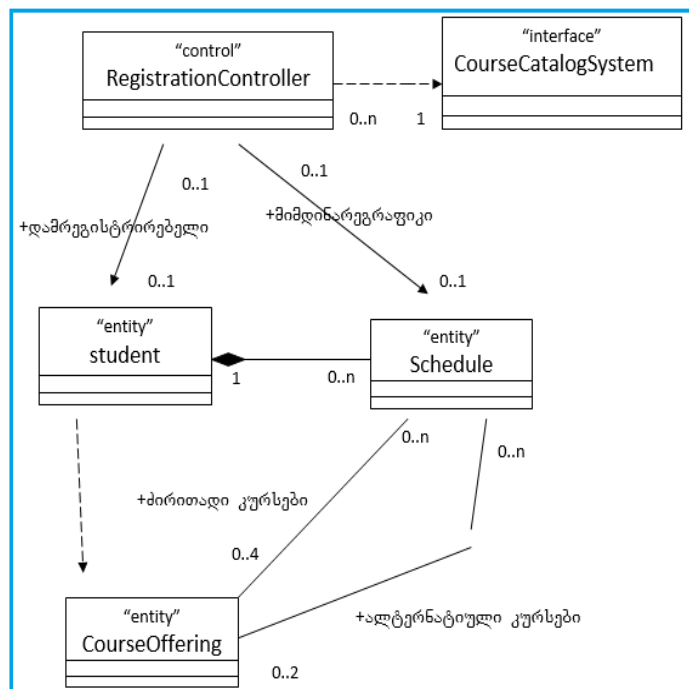
ნახ.4

გათვალისწინებულია შეთანხმება ატრიბუტთა დასახელების შესახებ, რომელიც მიღებული იყო პროექტში და რეალიზაციის ენაში. მიეთითება ატრიბუტების ხილვადობა: public, private ან protected. აუცილებლობის შემთხვევაში განისაზღვრება წარმოებული (გამოთვლითი) ატრიბუტები. მათთვის შესაძლებელია ავავოთ მოდელი, რომელიც აღწერს ობიექტის მდგომარეობებს და გადასვლებს მათ შორის. ეს მოდელი აისახება მდგომარეობის დიაგრამის სახით.

2.3. კლასებს შორის კავშირების დაზუსტება

პროექტირების პროცესში კავშირები კლასებს შორის (ასოციაცია, აგრეგაცია და განზოგადება) მოითხოვს დაზუსტებას. ასოციაცია მოსაზღვრე და მმართველ კლასებს შორის გამოხატავენ კავშირებს, რომლებიც დინამიურად არიძვრებიან შესაბამის ობიექტებს შორის მართვის ნაკადში. ასეთი კავშირებისათვის საკმარისია უზრუნველყოთ კლასების ხედვა, ამიტომ ისინი გარდაიქმნიან დამოკიდებულებაში. თუ ზოგიერთი ასოციაციისათვის არ არის აუცილებელი ორმომართულებიანი კავშირისა, მაშინ შემოიღანება ნავიგაციის მიმართულება. აგრეგაციები, რომლებსაც აქვთ კომპოზიციის თვისებები, გარდაიქმნიან კომპოზიციურ კავშირებში.

კავშირების გარდაქმნის მაგალითი მოყვანილი რეკომენდაციების შესაბამისად კლასებისათვის პრეცედენტიდან „კურსებზე დარეგისტრირება“ მოყვანილია მე-5 ნახაზზე. ასოციაცია მმართველ და მოსაზღვრე კლასებს შორის გარდაიქმნა დამოკიდებულებათ. აგრეგაცია კლასებს შორის Student და Schedule ფლობს კომპოზიციის თვისებას. ნავიგაციის მიმართულება ასოციაციებზე კლასებს შორის Schedule და CourseOffering შემოტანილია შემდეგი მოსაზრებებით: არ არის აუცილებელი გრაფიკების ცხრილის მიღება, რომლებშიც არის რომელიმე კურსი; გრაფიკების რაოდენობა შედარებით ცოტაა კონკრეტული კურსების რაოდენობასთან შედარებით.



ნახ.5

3. დასკვნა

მონაცემთა ბაზის დამუშავების მოყვანილი საშუალება უზრუნველყოფს ბიზნეს-პროცესების მართვის ისეთი სისტემის შექმნას, რომლითაც სრულად იქნება რეალიზებული ფუნქციონალური და არაფუნქციონალური მოთხოვნები.

ლიტერატურა - References - Литература:

1. Арлоу Дж., Нейштадт А. (2008). UML/2 и Унифицированный процесс. 2-е изд., Практический объектно-ориентированный анализ и проектирование. Санкт-Петербург - Москва, 2008.

2. სუხიაშვილი თ. პროგრამული სისტემის დამუშავების CASE საშუალებები. ლაბ.პრაქტ., [gvhttp://gtu.ge/Learning/EIBooks/ims_books.php](http://gtu.ge/Learning/EIBooks/ims_books.php) 20013, 130 გვ.

DEVELOPMENT OF THE LOGICAL SCHEME OF THE DATABASE FOR COMPUTER MANAGEMENT OF BUSINESS PROCESSES

Sukhiashvili Teimuraz, Shurgaia Irakli

Georgian Technical University

Summary

The main building construction unit of object-oriented systems represents object - a class. Therefore developing computer systems of management of business processes in order that provide effective implementation of all functional requirements and at the same time full satisfaction nonfunctional (usability, reliability, a productivity, safety) the requirement, definition of classes, rational distribution of duty and establishment of the real relations between them is of great importance. In article the way of establishment of the logical scheme of the database for business process (registration of students in higher educational institutions on courses and the accounting of progress) within the rational unified process (RUP) with use of the unified language of a modeling (UML) is offered.

РАЗРАБОТКА ЛОГИЧЕСКОЙ СХЕМЫ БАЗЫ ДАННЫХ ДЛЯ КОМПЬЮТЕРНОГО УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССОВ

Сухиаშვილი Т., Шурგაია И.

Грузинский Технический Университет

Резюме

Основным строительным блоком объектно-ориентированных систем является объект - класс. Поэтому разрабатывая компьютерные системы управления бизнес-процессов, для того, чтобы обеспечить эффективное выполнение всех функциональных требований и при этом полностью удовлетворить нефункциональные (удобство использования, надежность, производительность, безопасность) требования, имеет большое значение определение классов, рациональное распределение обязанностей и установление реальных отношений между ними. В статье предлагается способ установления логической схемы базы данных для бизнес-процесса(регистрация студентов в высших учебных заведениях на курсы и учет успеваемости) в рамках рационального унифицированного процесса(RUP) с использованием унифицированного языка моделирования (UML).