

კლასების დაპროექტება ობიექტთა მდგომარეობის გათვალისწინებით

თეიმურაზ სუხიაშვილი, ირაკლი შურღაია
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

განიხილება ობიექტ-ორიენტირებული ანალიზისა და დაპროექტების ეტაპების ბიზნეს-პროცესების მოდელირების საკითხები, კლასთა ერთობლიობის საფუძველზე აგებული სისტემური პროექტის ადაპტაცია რეალიზაციის სფეროსთან ყველა არაფუნქციონალური მოთხოვნის გათვალისწინებით. ობიექტ-ორიენტირებული დაპროექტების ეტაპის მნიშვნელოვანი ნაწილია კლასების კონსტრუირება, რომელიც გულისხმობს ატრიბუტებისა და ოპერაციების თვისებათა დადგენას, აგრეთვე იმ შეტყობინებათა გამოვლენას, რომლებზეც უნდა მოხდეს მათი რეაგირება. ამ თვალსაზრისით მნიშვნელოვანია კლასთა მდგომარეობების და იმ მოვლენების მოდელირება, რომლებიც განაპირობებს მდგომარეობათა შეცვლას. განსაკუთრებით ეს ეხება სისტემებს, რომლებშიც გვაქვს მდგომარეობისაგან დამოკიდებული ობიექტები ქცევის რთული დინამიკით.

საკვანძო სიტყვები: ობიექტ-ორიენტირებული დიაგრამა. კლასი. ობიექტი. მდგომარეობა. მოვლენა. კომპოზიცია. ქცევის მოდელი. მოქმედების არე.

1. შესავალი

ობიექტ – ორიენტირებული დაპროექტება ძირითადად მოიცავს სისტემის არქიტექტურის და ელემენტების დაპროექტებასთან დაკავშირებულ მოღვაწეობას.

სისტემის არქიტექტურის დაპროექტებისას ხდება “ანალიზის” კლასებს შორის ურთიერთქმედების ანალიზი, ქვესისტემებისა და ინტერფეისების გამოვლენა, არქიტექტურული დონეების ფორმირება, მართვის ნაკადების სტრუქტურის დაპროექტება და სისტემის განაწილებული კონფიგურაციის დაპროექტება.

სისტემის ელემენტების დაპროექტებისას კი უნდა მოვახდინოთ გამოყენებითი შემთხვევების აღწერის დაზუსტება (ურთიერთქმედებისა და კლასების დიაგრამების მოდიფიცირება კლასებისა და ქვესისტემების დაპროექტებისას ახლად გამოვლენილი ფაქტორების გათვალისწინებით), კლასების დაპროექტება და მონაცემთა ბაზის დაპროექტება.

საფუძველს ყველა ჩამოთვლილ მოღვაწეობებისათვის წარმოადგენს კლასების დაპროექტება, რომლის დროსაც

უნდა მოვახდინოთ „ანალიზის“ კლასების შემადგენელი ელემენტების (ოპერაციებისა და ატრიბუტების თვისებების(ხედავ, მოქმედების არე, ჯერადობა) დაზუსტება და იმ მოვლენების (სიგნალების) დადგენა, რომლებზეც უნდა რეაგირებდეს იგი თავისი მოვალეობების შესასრულებლად.

2. ძირითადი ნაწილი

პრობლემის გადასაწყვეტად განსაკუთრებით, თუ სისტემაში გვაქვს მდგომარეობისაგან დამოკიდებული ობიექტები ქცევის რთული დინამიკით სასურველია ავავოთ მოდელი, რომელიც აღწერს ობიექტის მდგომარეობებს და გადასვლებს მათ შორის. ეს მოდელი წარმოდგება მდგომარეობის დიაგრამის სახით. საულისტრაციო მაგალითისათვის განვიხილოთ კლასის CourseOffering(კურსებზე დარეგისტრირება) ქცევა[2]. მდგომარეობის დიაგრამა იგება სამ ეტაპად:

1 ეტაპი. მდგომარეობების იდენტიფიცირება. მდგომარეობათა გამოვლენის ნიშნებია ობიექტის ატრიბუტების მნიშვნელობების შეცვლა და კავშირების გაწყვეტა სხვა ობიექტებთან. ასე მაგალითად, ობიექტი CourseOffering შეიძლება იმყოფებოდეს მდგომარეობაში Open (კურსზე მიღება გახსნილია) მანამდე, სანამ მასზე დარეგისტრირებულ სტუდენტთა რაოდენობა არ გადააჭარბებს 10-ს, ხოლო როგორც კი გადააჭარბებს 10-ს, ობიექტი გადადის მდგომარეობაში Closed (კურსზე მიღება დახურულია). ამას გარდა, ობიექტი CourseOffering შესაძლებელია იმყოფებოდეს მდგომარეობაში Unassigned (იგი არავის არ მიყავს ანუ არ არსებობს კავშირი Professor – ის რომელიმე ობიექტთან) ან Assigned (ასეთი კავშირი არსებობს).

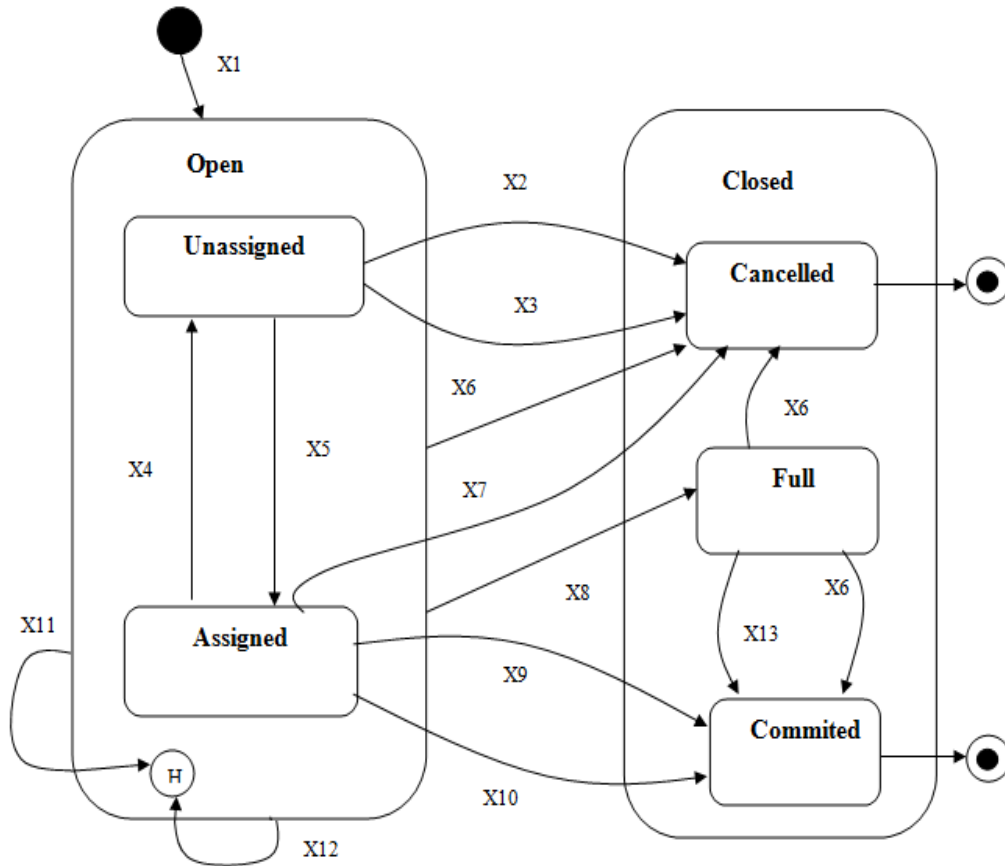
ეტაპი 2. მოვლენათა იდენტიფიცირება. მოვლენები, როგორც წესი დაკავშირებული არიან გარკვეული ოპერაციების შესრულებასთან. კლასი CourseOffering მოვლენების განაწილების შედაგად გამოყენებითი შემთხვევა „ავირჩიოთ კურსები სწავლებისათვის“ ანალიზისას განისაზღვრა ორი ოპერაცია – addProfessor და removeProfessor, დაკავშირებულნი გარკვეული პროფესორის მიერ კურსების არჩევასთან (ახალი კავშირის შექმნა) და არჩეული კურსზე უარის თქმა (კავშირის გაწყვეტა). ამ ოპერაციებს შეუსაბამებენ ორ მოვლენას – addProfessor და removeprofessor.

ეტაპი 3. მდგომარეობებს შორის გადასვლების იდენტიფიცირება. გადასვლები გამოიძახებიან მოვლენებით. მაშასადამე, Unassigned და Assigned ერთდებიან ორი გადასვლით (ნახ.1).

CourseOffering ობიექტის ქცევის შემდგომი დეტალიზებას მიყვავართ მდგომარეობის დიაგრამის აგებამდე (ნახ.1). მოცემულ დიაგრამაზე გამოყენებულია მოდელირების ისეთი შესაძლებლობები, როგორიც არის კომპოზიციური (composite state) და ისტორიული მდგომარეობები (history state). მოცემულ შემთხვევაში კომპოზიციური მდგომარეობებია Open და Closed, ხოლო ჩართული მდგომარეობებია – Unassigned, Assigned, Cancelled (კურსი გაუქმდა), Full (კურსი შევსებულია) და Committed (კურსი ჩართულია განრიგში). კომპოზიციური მდგომარეობები საშუალებას იძლევიან გავამარტივოთ დიაგრამა, შევამციროთ რა გადასვლების რაოდენობა, რამდენადაც ჩართული მდგომარეობები მემკვიდრეობით იძენენ კომპოზიციური მდგომარეობების ყველა თვისებებსა და გადასვლებს.

ისტორიული მდგომარეობა (დიაგრამაზე აღნიშნულია წრეში ჩაწერილი H ასოთი) – ეს პსევდომდგომარეობაა, რომელიც აღადგენს წინამდებარე აქტიურ მდგომარეობას კომპოზიტურ მდგომარეობაში. იგი საშუალებას აძლევს კომპოზიტურ მდგომარეობას Open დაიმასხვროს

რომელი ჩართული მდგომარეობა (Unassigned an Assigned) იყო მიმდინარე Open – დან გამოსვლის მომენტში, იმისათვის რომ ნებისმიერი გადასვლა Open – ში (add student ან remove student) ბრუნდებოდეს სწორედ ამ ჩართულ მდგომარეობაში, და არა საწყის მდგომარეობაში.



ნახ.1. მდგომარეობის დიაგრამა კომპოზიციური მდგომარეობებით

დიაგრამის გადატვირთვის თავიდან ასაცილებლად, ქვემოთ მოყვანილია იმ მოვლენების აღწერა, რომლებიც განაპირობებენ ერთი მდგომარეობიდან გადასვლას მეორეში:

- X1 – სტუდენტების რაოდენობა=0;
- X2 - რეგისტრაციის დახურვა;
- X3 – დახურვა;
- X4 - პროფესორის დამატება;
- X5 - პროფესორის ამოგდება;
- X6 – დამთავრება;
- X7 – დახურვა[სტუდენტების რაოდენობა<3];
- X8 – სტუდენტების რაოდენობა =10;
- X9 – რეგისტრაციის დახურვა[სტუდენტების რაოდენობა>=3];
- X10 – დახურვა {სტუდენტების რაოდენობა>=3};
- X11 – სტუდენტის დამატება /სტუდენტის–რაოდენობა=სტუდენტის–რაოდენობა+1;
- X12 – სტუდენტის დამატება /სტუდენტის–რაოდენობა=სტუდენტის–რაოდენობა+1;
- X13 - რეგისტრაციის დახურვა[პროფესორები დაინიშნა].

3. დასკვნა

თუ სისტემაში გვაქვს მდგომარეობისაგან დამოკიდებული ობიექტები ქცევის რთული დინამიკით. ასეთი კლასებისათვის, შემაღენელი ელემენტების სრულყოფილი აღწერისათვის, სასურველია ავაგოთ მოდელი, რომელიც აღწერს ობიექტის მდგომარეობებს და გადასვლებს მათ შორის. ეს მოდელი წარმოსდგება მდგომარეობის დიაგრამის სახით.

ლიტერატურა:

1. სუხიაშვილი თ. ავტომატიზებული მართვის თეორიული საფუძვლები. სტუ. თბ., 2005
2. სუხიაშვილი თ. პროგრამული სისტემის დამუშავების CASE საშუალებები. ელ-სახელმძღ., სტუ. თბ., 2013. <http://gtu.ge/books/CASE.pdf>
3. სუხიაშვილი თ. სასამართლო სამოქალაქო სამართლის საქმეთა მდგომარეობათა ანალიზი სამართალწარმოების ავტომატიზებულ სისტემაში. პერ. სამეცნ. ჟურნ. „ინტელექტი“, №1 (18), 2004, გვ.43-47.

DESIGN OF CLASSES TAKING INTO ACCOUNT CONDITIONS OF OBJECTS

Sukhiashvili Temur, Shutgaia Irakli
Georgian Technical University

Summary

The purpose of object-oriented design is adaptation of the preliminary system project (a set of classes of "analysis"), making a stable basis of architecture of system, to the realization environment taking into account all nonfunctional requirements. The considerable part in activity of object-oriented design is occupied by design of classes which means establishment of properties (visibility, a scope, frequency rate) attributes and operations, and identification of those messages on which they have to show reaction. Modeling of conditions of classes and those phenomena which cause change of conditions is in this regard considerable, especially it concerns to systems in which there are objects dependent on conditions with difficult dynamics of behavior.

ПРОЕКТИРОВАНИЕ КЛАССОВ С УЧЕТОМ СОСТОЯНИЙ ОБЪЕКТОВ

Сухиаშვილი Т.
Грузинский Технический Университет

Резюме

Рассматриваются вопросы моделирования бизнес-процессов этапов объектно-ориентированного анализа и проектирования, адаптации системного проекта, построенного на основе совокупности классов, к среде реализации с учетом всех нефункциональных требований. Значительной частью объектно-ориентированного проектирования является конструирование классов, что подразумевает установление свойств атрибутов и операций, а также выявление тех сообщений, на которые они должны реагировать. В этом отношении представляется значительным моделирование состояний классов и тех событий, которые обуславливают изменение состояний. Особенно это касается систем, в которых есть зависимые от состояний объекты со сложной динамикой поведения.