

სისტემაში მიმდინარე დინამიკური პროცესების მოდელირება ობიექტ-ორიენტირებული დაპროექტებისას

თეიმურაზ სუხიაშვილი, ირაკლი შურაია
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

ყველა რეალური სისტემა ამა თუ იმ ზომით დინამიურია, და მათი დინამიურობა განპირობებულია მოვლენებით, რომლებიც ხდება სისტემის შიგნით ან მის გარეთ. ნებისმიერი წარმოსახვა, რომელიც შესაძლებელია მოხდეს ობიექტზე, მოდელირდება როგორც მოვლენა(Event). UML გვაძლევს საშუალებას მოვახდინოთ მოვლენათა მოდელირება და შესაბამისად აღვწეროთ სისტემაში მიმდინარე დინამიური პროცესები.

საკვანძო სიტყვები: მოვლენა. მოდელი. სიგნალი. ასინქრონული. სინქრონული. დიაგრამა. ოპერაცია. კლასი. მდგომარეობა. გამორიცხვის მეთოდი.

1. შესავალი

რეალური სამყარო ყოველდღიურად ქმნის ახალ წარმოსახვებს, რომლებიც გავლენას ახდენენ ჩვენს საპრობლემო სფეროზე. აბსოლუტურად სტატიური სისტემა არ არსებობს და არც არის რაიმე ინტერესის მატარებელი, რამდენადაც მათში არაფერი არ ხდება. სხვადასხვა წარმოსახვებს, რომლებიც ხდება გარე სამყაროში, უწოდებენ მოვლენებს. ყოველი მოვლენა აღწერს არსებით ფაქტს, რომელიც ლოკალიზებულია დროში და სივრცეში. ავტომატების კონტექსტში მოვლენა ეს სტიმულია, რომელსაც შეუძლია გამოიწვიოს გადასვლა ერთი მდგომარეობიდან მეორეში. მოვლენები შესაძლებელია იყვნენ შიდა და გარე. გარე მოვლენები გადაიცემა სისტემასა და მის მომხმარებელს შორის(მაგალითად, ღილაკის დაჭერა). შიდა სიგნალები გადაიცემა სისტემაში არსებულ ობიექტებს შორის. მოვლენათა რიცხვს მიეკუთვნება სიგნალები, გამოძახება, დროის შუალედის ამოწურვა და მდგომარეობის შეცვლა. მოვლენები შესაძლებელია იყვნენ სინქრონული და ასინქრონული, რომელთა მოდელირებაც სისტემაში მიმდინარე პროცესების მოდელირების შემადგენელი ნაწილია.

2. ძირითადი ნაწილი

მოვლენა - **სიგნალი** ეს დასახელებული ობიექტია, რომელიც ასინქრონულად აღიძვრება ერთი ობიექტით და მიიღება მეორეთი. გამორიცხვა, რომელიც ფართოდ გამოიყენება თანამედროვე დაპროგრამების ენების უმეტესობაში, ყველაზე გავრცელებული სახეა შიდა სიგნალებს შორის. ობიექტების შექმნა და მოსპობაც სიგნალის ნაირსახეობას მიეკუთვნება.

სიგნალები შესაძლებელია წარმოვადგინოთ კლასების სახით და დავაკავშიროთ განზოგადების მიმართებით. რაც საშუალებას იძლევა მოვახდინოთ მოვლენათა იერარქიის მოდელირება, რომელშიც ერთნი არიან ზოგადი, ხოლო სხვები კერძო.

სიგნალი შეიძლება გაიგზავნოს როგორც მდგომარეობის გადასვლის მოქმედება ავტომატში ან როგორც შეტყობინების გაგზავნა ურთიერთქმედებისას. ოპერაციების შესრულებისას ასევე შეიძლება გაიგზავნოს სიგნალი. პრაქტიკაში, როდესაც მოდელირდება კლასი ან ინტერფეისი, ქცევის სპეციფიკაციების მნიშვნელოვანი ნაწილია, მითითება იმისა, თუ რა სიგნალების გაგზავნა შეუძლია მათ. იმისათვის, რომ მიუთითონ ოპერაციის მიერ სიგნალის გაგზავნა, შეიძლება ვისარგებლოთ მიმართებით **დამოკიდებულება** სტერეოტიპით **send**.

გამოძახების მოვლენები. თუ სიგნალის მოვლენა წარმოადგენს სიგნალის აღძვრას, გამოძახების მოვლენა განკუთვნილია ოპერაციის შესრულების აღწერისათვის. ორივე შემთხვევაში მოვლენას შეუძლია გამოიწვიოს მდგომარეობის შეცვლა ავტომატში. იმ დროს როდესაც სიგნალი წარმოადგენს ასინქრონულ მოვლენას, გამოძახების მოვლენა ჩვეულებრივ სინქრონულია. ეს ნიშნავს იმას, რომ როდესაც ერთი ობიექტი ინიცირებას ახდენს ოპერაციის შესრულებაზე მეორე ობიექტზე, რომელსაც გააჩნია თავისი ავტომატი, მართვა გადაიცემა გამომგზავნიდან მიმღებზე, აქუშავდება შესაბამისი გადასვლა, შემდეგ ოპერაცია სრულდება, მიმღები გადადის ახალ მდგომარეობაში და უბრუნებს მართვას გამომგზავნს.

დროითი და ცვლილების მოვლენები. დროითი მოვლენა წარმოადგენს დროითი შუალედის ამოწურვას. მოდელირებისას იგი შეიძლება წარმოვადგინოთ გასაღებური სიტყვით after(შემდეგ), მას მოსდევს გამოსახულება, რომელიც ითვლის დროის გარკვეულ შუალედს. გამოსახულება შეიძლება იყოს მარტივი ან რთული. თუ არ არის მითითებული, დროის ათვლა იწყება მიმდინარე მდგომარეობაში შესვლის მომენტიდან.

ცვლილების მოვლენით აღიწერება მდგომარეობის შეცვლა ან გარკვეული პირობის შესრულება. იგი წარმოიდგინება გასაღებური სიტყვით when, რომელსაც მოსდევს ბულის გამოსახულება. ასეთი გამოსახულება შესაძლებელია აბსოლუტური დროის მომენტის ან პირობის შესამოწმებლად (მაგალითად, when $t < 5$).

სიგნალისა და გამოძახების მოვლენებში მონაწილეობენ სულ მცირე ორი ბიექტი : ობიექტი, რომელიც აგზავნის სიგნალს ან ახდენს ოპერაციის ინიცირებას, და ობიექტი, რომელზეც არის ეს მოვლენა დამისამართებული. რამდენადაც სიგნალები თავისი ბუნებით ასინქრონულნი არიან, კლასის ყოველ ეგზემპლარს შეუძლია გაუგზავნოს სიგნალი მიმღებ ობიექტს ან მოახდინოს მასში ოპერაციის ინიცირება. გაუგზავნის რა სიგნალს მიმღებს, იგი აგრძელებს თავის მართვის ნაკადს, არ ელოდება მისგან პასუხს. გამოძახების მოვლენის დროს კი მისი სინქრონული ხასიათის გამო, თუ ობიექტი ახდენს ოპერაციის ინიცირებას, იგი უნდა დაელოდოს პასუხს მიმღებისაგან.

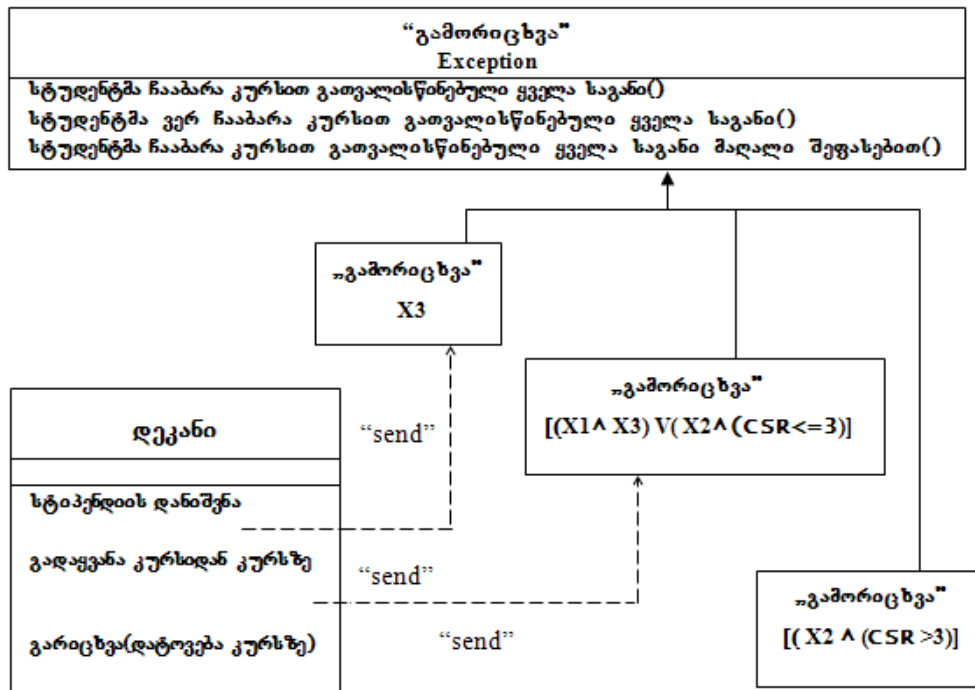
ნებისმიერი კლასის ნებისმიერი ეგზემპლარი შესაძლებელია იყოს გამოძახების მოვლენის ან სიგნალის მიმღები. თუ ეს სინქრონული მოვლენაა, მაშინ გამგზავნი და მიმღები იმყოფებიან მდგომარეობაში **რანდეუ** ოპერაციის შესრულების მთელი პერიოდის განმავლობაში. ეს ნიშნავს, რომ გამომგზავნის მართვის ნაკადი ბლოკირებულია მიმღების მართვის ნაკადით, სანამ ოპერაცია არ დამთავრდება. თუ ეს სიგნალია, მაშინ გამგზავნი და მიმღები არ შედიან **რანდეუ** მდგომარეობაში: გამგზავნი აგზავნის სიგნალს, მაგრამ არ ელოდება პასუხს მიმღებისაგან.

გამორიცხვები (Exception). კლასების ქცევის ვიზუალიზების მნიშვნელოვან ნაწილს წარმოადგენს გამორიცხვების სპეციფიცირება, რომლებსაც შეუძლიათ ალაგზნონ მისი ოპერაციები. თუ გაქვთ კლასი ან ინტერფეისი, მაშინ ოპერაციები, რომლებიც შესაძლებელია მათზე გამოვიძახოთ ნათლად ჩანს აღწერიდან, მაგრამ იმის გაცემა თუ რომელ გამორიცხვებს ალაგზნებენ ისინი არ არის ადვილი, თუ ეს ნათლად არ არის მითითებული მოდელში.

გამორიცხვები წარმოადგენენ სიგნალების კერძო შემთხვევებს და მოდელირებიან სტერეოტიპული კლასებით (ნახ.1). გამორიცხვები შესაძლებელია მიუერთოდ ოპერაციების სპეციფიკაციებს. გამორიცხვების მოდელირება წარმოადგენს ოპერაციას, გარკვეული აზრით საწინააღმდეგოს სიგნალთა სიმრავლის მოდელირებისა. სიგნალთა ოჯახის მოდელირების ძირითადი მიზანია – მოვახდინოთ იმის სპეციფიცირება, თუ რომელი სიგნალები შეუძლია მიიღოს აქტიურმა ობიექტმა; გამორიცხვების მოდელირების მიზანია უჩვენოთ, რომელი გამორიცხვები შეიძლება აღიძვრეს ობიექტით თავისი ოპერაციებიდან.

მაგალითად, 1-ელ ნახაზზე წარმოდგენილია გამორიცხვების იერარქიული მოდელი, რომლებიც შესაძლებელია აღიძვრას კლასით **dekani**. ამ იერარქიის თავში იმყოფება აბსტრაქტული კლასი **Exception**, ხოლო ქვევით სპეციალიზებული გამორიცხვები. ნახაზზე მიღებულია შემდეგი აღნიშვნები:

- X1 – სტუდენტმა ჩააბარა კურსით გათვალისწინებული ყველა საგანი;
 - X2 – სტუდენტმა ვერ ჩააბარა კურსით გათვალისწინებული ყველა საგანი;
 - X3 – სტუდენტმა ჩააბარა კურსით გათვალისწინებული ყველა საგანი მაღალი შეფასებით.
- CSR = ჩაუბარებელი საგნების რაოდენობა.
- $X1 = (S1 \geq 51) \wedge (S2 \geq 51) \wedge \dots \wedge (Sn \geq 51)$;
- $X2 = (S1 \leq 51) \vee (S2 \leq 51) \vee \dots \vee (Sn \leq 51)$;
- $X3 = (S1 \geq 90) \wedge (S2 \geq 90) \wedge \dots \wedge (Sn \geq 90)$;
- სადაც,
- S1, S2, . . . Sn – კურსზე ჩასაბარებელ საგნებში მიღებული შეფასებები.



ნახ.1

3. დასკვნა

მოდელირების განხილული საშუალებები უზრუნველყოფს სრულყოფილად აღიწეროს საპრობლემო სფეროში მიმდინარე დინამიური პროცესები, რაც შემდგომ უზრუნველყოფს შემუშავებული მოდელების ავტომატიზებულ რეალიზებას.

ლიტერატურა:

1. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. Серия “Объектно- ориентированные технологии в программировании”. Москва, 2004.
2. სუხიაშვილი თ., განაწილებული სისტემების მოდელირება პროცესების თვალთახედვით. სტუ-ს შრომები, 2006, № 4(437), გვ. 206-207.

МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИХ ПРОЦЕССОВ СИСТЕМЫ ПРИ ОБЪЕКТНО-ОРИЕНТИРОВАННОМ ПРОЕКТИРОВАНИИ

Сукиашвили Т., Шургая И.
Грузинский Технический Университет

Резюме

Все реальные системы в той или иной мере динамичны, и их динамичность обусловлена событиями, происходящими внутри или вне системы. Любое явление, которое может произойти на объекте, моделируется как событие. UML предоставляет средства моделирования событий и, соответственно, дает возможность описать динамические процессы, протекающие в системе.

MODELING OF DYNAMIC PROCESSES IN SYSTEM AT OBJECT-ORIENTED DESIGNING

Sukhiashvili Teimuraz, Shurgaia Irakli
Georgian Technical University

Summary

All real systems are to some extent dynamical, and their dynamism is caused by the events occurring inside or out of system. Any phenomenon which can occur on object, is simulated as event. UML gives tools for simulation of events and accordingly possibility to describe dynamic processes proceeding in system.