

პროგრამული ინჟინერიის განვითარების თანამედროვე ტექნოლოგიები

ეკატერინე თურქია

საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

განიხილება პროგრამული ინჟინერიის ტექნოლოგიის განვითარების თანამედროვე კონცეფციები, UML-ის სრულყოფის ძირითადი მიმართულებები მეტა-მოდელირების პრინციპებისა და თანამედროვე სტანდარტების მიხედვით. აღწერილია პროგრამული ინჟინერიის უახლესი, განვითარებადი პარადიგმა - მოდელური ინჟინერიის ტექნოლოგია (Model-Driven Development-MDD), გაანალიზებულია ამ ტექნოლოგიის გამოყენების საზღვრები და ასახვის ფორმები თანამედროვე ინსტრუმენტულ საშუალებებში.

საკვანძო სიტყვები: მეტა-მოდელირება. UML პროფილი. Clabject. MOF. მოდელური ინჟინერიის არქიტექტურა. MDA.

1. შესავალი

საინფორმაციო სისტემების განვითარების თვალსაზრისით, სამეცნიერო კვლევები მიმართულია სხვადასხვა მოდელირების სისტემების, პროგრამული კოდის, მონაცემებისა და ინფორმაციის ინტეგრაციის, ტრანსფორმაციის, მიგრაციისა და ურთიერთთავსებადობის ტექნოლოგიების სრულყოფისკენ. გამომდინარე აქედან თანამედროვე მოდელირების ენების მიმართ ჩნდება მაქსიმალურად ადაპტირებისა და აბსტრაქციის მაღალი დონით წარმოდგენის მოთხოვნები.

ამ მიმართულებით, მნიშვნელოვანი განვითარება ჰპოვა UML-ენამ, რომელიც პროგრამული ინჟინერიის მოქნილ და ფართოდ გავრცელებულ ტექნოლოგიად ითვლება. UML-ის სრულყოფის მიზნით OMG-ჯგუფის მიერ შემუშავდა პროგრამული უზრუნველყოფის ტექნოლოგიური დამუშავების განვითარებადი პარადიგმა - მოდელური ინჟინერია ანუ მოდელებით მართვადი არქიტექტურა (Model-Driven Development-MDD), რომელიც დღესდღეობით, მასშტაბური სისტემების კომპოზიციისა და ინტეგრაციის პრობლემების გადაჭრის მოქნილ ტექნოლოგიად ითვლება. MDD საშუალებას იძლევა გარდაქმნას მოდელი პლატფორმადამოუკლებელ არტეფაქტებად, მაღალი დონის პროგრამულ ტექნოლოგიებში გამოსაყენებლად (მაგალითად, EJB, .NET) [1].

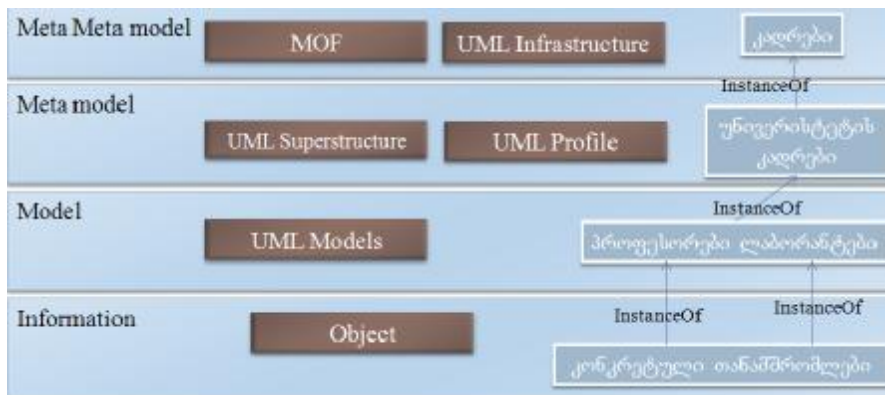
როგორც წესი, პროგრამული უზრუნველყოფის სისტემების რეალიზაცია მოითხოვს სისტემის მრავალვარიანტულ კვლევას (აღწერა, დაგეგმვა, დამუშავება, რეალიზაცია, ტესტირება, ვალიდაცია, დანერგვა) მოდელირების სხვადასხვა ტიპისა და შესაბამისად, სხვადასხვა ინსტრუმენტული საშუალებების გამოყენებით. მოდელის სხვადასხვა ტიპების გამოყენების ფარგლებში ძირითადად ერთიანდება პროცესების კოორდინაციის, დინამიკური სიმულაციის, სერვის-ორიენტირებული მოდელირების, ობიექტ-ორიენტირებული დაპროექტებისა და ა.შ. სტანდარტები (მაგალითად, CASE, RAD, BPMN, WFMS, PetriNet, DSM, SOA). მიუხედავად

იმისა, რომ აღნიშნული მოდელები სრულყოფილად ახდენენ სისტემის აღწერას, ჩნდება რიგი პრობლემური საკითხები, რაც ძირითადად უკავშირდება მოდელირებული ელემენტებისა და მონაცემების ინტეგრაციას, არაფორმალური და ფორმალური მოდელების ურთიერთაქმადობასა და კომპლექსურ განხილვას.

ამ თვალსაზრისით, მოდელური ინჟინერიის ტექნოლოგიის კონცეფცია სხვადასხვა ტიპის მოდელების ურთიერთტრანსფორმაცია, რაც ტექნიკურად XML ენაზე ბაზირებული სინტაქსური ანალიზატორისა (XMI) და მეტა-მოდელირების პრინციპების (MOF- Meta Object Facility) მიხედვით ხორციელდება.

2. ძირითადი ნაწილი

მეტა-მოდელირების კონცეფცია იხილავს მოდელების ფორმალური აღწერისა და აბსტრაქციის ოთხ იერარქიულ დონეს: რეალური ობიექტი აღიწერება მოდელით (სტანდარტიზებული გრაფიკული ნოტაციები, ენის სინტაქსი და სემანტიკა), მოდელის აღწერა ხდება მეტა-მოდელით ანუ სუპერ სტრუქტურით (შეთანხმებული გრაფიკული ნოტაციები, ენის სინტაქსი და სემანტიკა), მეტა-მოდელი აღიწერება მეტა-მეტა მოდელით (გაფართოებადი გრაფიკული ნოტაციები, ენის სინტაქსი და სემანტიკა), ხოლო მეტა-მეტა მოდელი (შაბლონები) თვითაღწერადია (ნახ.1).

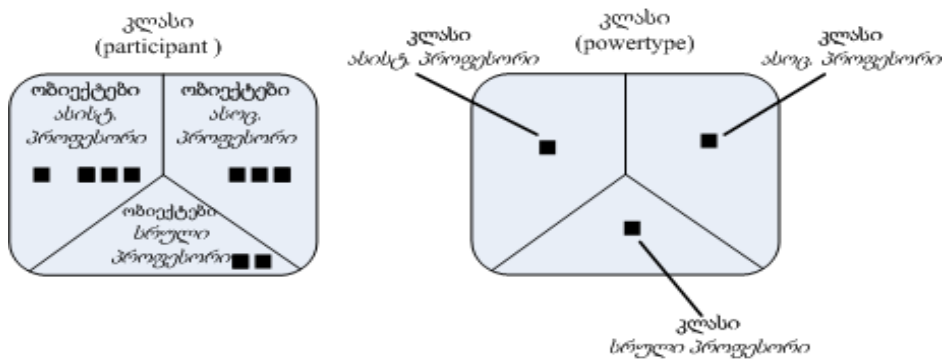


ნახ.1. მეტა-მოდელირების კონცეფცია

ამ კონცეფციის მიხედვით, ობიექტი, მოდელი და მეტა მოდელი შესაბამისად, არის მოდელის, მეტა-მოდელისა და მეტა-მეტა მოდელის ეგზემპლარი ანუ კონკრეტიზაცია. მაგალითად, არსთა დამოკიდებულების მოდელის გრაფიკული ნოტაციები - არსი, რელაცია და ატრიბუტი აღიწერება მეტა-მოდელის „სუპერ“ ელემენტებით: მართკუთხედი, ისარი და ოვალი. ამ თვალსაზრისით, მოდელური ინჟინერიის სისტემები, სხვადასხვა ტიპის მოდელების ურთიერთსახვას პრაქტიკულად ახორციელებს მოდელების ტრანსფორმაციის საშუალებებით და სპეციალიური „plugin“-ების შემუშავებით (მნიშვნელოვნად აქტუალურია, მოდელირების სხვადასხვა ინსტრუმენტული საშუალებების გამოყენების ფარგლებში), რასაც ფაქტობრივად, საფუძვლად უდევს მეტა-მოდელირების კონცეფციის გამოყენება – მოდელების ე.წ. ზოგადი სტრუქტურული აღწერა [2].

მოდელური ინჟინერიის ბაზაზე, UML ენამ განავითარა სხვადასხვა ძირეული მიმართულებები: დაიხვეწა და გაფართოვდა სტატიკური და დინამიკური დიაგრამების ნაირსახეობა; შეიქმნა თავისუფალი სტერეოტიპების გამოყენებისა და შექმნის შესაძლებლობები (UML პროფილი); ჩამოყალიბდა და განვითარდა მეტა-მოდელირების პრინციპები – ინფრასტრუქტურა (UML მოდელის შიგა სტრუქტურის რეორგანიზაცია, ბაზური კლასების გაფართოება), სუპერსტრუქტურა (კომპონენტური დამუშავება, მეტა-მოდელის გაფართოება, სხვადასხვა სტანდარტებთან შესაბამისობის შესაძლებლობა), ობიექტების შეზღუდვების ენა (OCL, Object Constraint Language), დიაგრამების ურთიერთგაცვლის ფორმატი (XMI).

ამავდროულად, მეტა-მოდელირების კონცეფციის ბაზაზე განისაზღვრა მეტა-იერარქიის ე.წ. ღრმა ეგზემპლარისა და მატერიალიზაციის ცნებები, რის საფუძველზეც შემოდის ახალი სტანდარტი Clabject (ობიექტ-ორიენტირებული პარადიგმა), რაც კლასის ორმაგი თვისების აღწერის მექანიზმია. მაგალითად, ობიექტის „კონკრეტული თანამშრომელი“ ატრიბუტია – გვარი, ღრმა ეგზემპლარისა და მატერიალიზაციის ცნებების მიხედვით, მეტა-კლასს ”უნივერსიტეტის კადრები”, ასევე გააჩნია ატრიბუტი – გვარი (ნახ.1). აზრობრივად, კლასი, შესაძლებელია იყოს გამოყენებული როგორც კლასი, ისე ობიექტი. მე-2 და მე-3 ნახაზებზე ასახულია, Clabject-ის ფუნქციონირების პრინციპი.

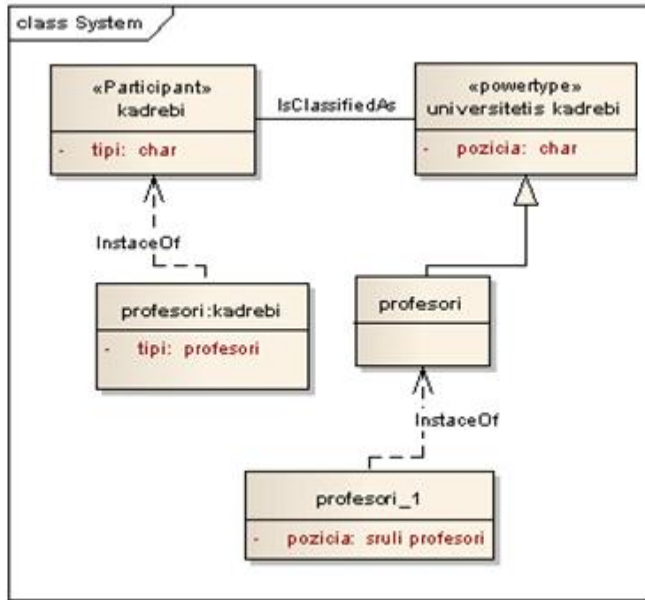


ნახ.2. Clabject-ის ფუნქციონირების პრინციპი

Clabject – ის გამოყენება, ჯერჯერობით მხოლოდ მოდელირების სისტემების დონეზე რეალიზებული (კლასის სტერეოტიპის სახეობები: power type - ძლიერი ტიპი, participant – მონაწილე ტიპი), მიმდინარეობს კვლევები ობიექტ-ორიენტირებული პროგრამირების სისტემებში დასანერგად [2,3].

UML მეტამოდელის შიგა სტრუქტურის რეორგანიზაციის, მოდულური სახით გაფართოების ამაღლებისა და სხვადასხვა სტანდარტებთან შესაბამისობის სარეალიზაციოდ, OMG ჯგუფის მიერ ჩამოყალიბდა, UML პროფილის შემდეგი მიმართულებები [4]:

- ▶ Platform Independent Model (PIM) & Platform Specific Model (PSM) for Software Radio Components (also referred to as UML Profile for Software Radio);
- ▶ UML Profile for CORBA® and CORBA® Component Model (CCM) UML Profile for Enterprise Application Integration (EAI);
- ▶ UML Profile for Enterprise Distributed Object Computing (EDOC);
- ▶ UML Profile for Modeling QoS and Fault Tolerance Characteristics and Mechanisms;
- ▶ UML Profile for Schedulability, Performance and Time;
- ▶ UML Profile for System on a Chip (SoC);
- ▶ UML Profile for Systems Engineering (SysML);
- ▶ UML Testing Profile.



ნახ.3. Clabject-ის ფუნქციონირების მაგალითი

მნიშვნელოვანია, რომ UML2 მოდელირების ინსტრუმენტულ საშუალებებში არ მოხდა ამ ცვლილებების ერთობლივი ასახვა. ძირითადად, გამოიყენება ისეთი მოქნილი სისტემები, როგორცაა – Sparx System EA, Visual Paradigm for UML, IBM Rational Software Architect, PowerDesigner, MagicDraw UML, EclipseUML Studio, WebSphere Development Studio.

რიც სისტემებში ყურადღება ექცევა UML პროფილის სრულყოფას სხვადასხვა კუთხით (პროგრამული სკრიპტის გამოყენებით, გრაფიკული ობიექტის ჩასმით,

სტერეოტიპების ვიზუალური ცვლილებებით და სხვ.), რიგი სისტემები ანვითარებენ პროგრამული კოდის გენერაციის მიმართულებას პლატფორმადამოკიდებული მიმართულებით. ასეთი ცალსახა მხარდაჭერის ძირითადი პრობლემა შესაძლოა იყოს ამ ტექნოლოგიის სიახლე. ფაქტობრივად, ლიტერატურულ წყაროებში ხაზგასმულია UML2 ტექნოლოგიის რთული, სწრაფი და მრავალმხრივი განვითარება, რის გამოც ჯერ კიდევ სრულად არ არის ჩამოყალიბებული და კვლევის ობიექტად რჩება UML2-ის სტანდარტის წესდება და მეტა-მეტა მოდელირების გამოყენების საზღვრები.

3. დასკვნა

როგორც წესი, ყოველ მოდელს გააჩნია აბსტრაქციის შემოსაზღვრული დონე, სხვადასხვა სემანტიკა და იყენებს განსხვავებულ სინტაქსს. გამომდინარე აქედან, მოდელის ტრანსფორმაცია გარდაქმნის მხოლოდ კონკრეტულ მოდელს სხვა კონკრეტულ მოდელად. ხშირ შემთხვევაში იკარგება სტრუქტურული ინფორმაცია, რაც იწვევს მონაცემთა ინტეგრაციისა და სინქრონიზაციის პრობლემებს. ეს ნიშნავს, რომ ტრანსფორმირებული მოდელის ელემენტებს შორის არ არსებობს სემანტიკური ურთიერთკავშირი, შეუძლებელია ერთი ტიპის მოდელის არსის ან ობიექტის გამოყენება მეორე მოდელში, ვინაიდან მეტა-მოდელი აღწერს მოდელის აბსტრაქტულ ნაწილს და არა შინაარსს. ფაქტობრივად, მოდელის ტრანსფორმაციის გამოყენება ართულებს სისტემის მთლიანობის დაცვას და შესაბამისად სისტემის სრულ ანალიზის. ამდენად, პრობლემატურ საკითხად ჯერ კიდევ რჩება სხვადასხვა ტიპის მოდელის ერთიანი ინტეგრირებული გარემოს შექმნა, სადაც მოდელის ელემენტებსა და არსებს შორის სრული სემანტიკური ურთიერთკავშირი განხორციელდება. მიუხედავად ამისა, მოდელური ინჟინერიის ტექნოლოგიის გამოყენება ჭრის

მრავალ პრობლემატურ საკითხს რთული პროგრამული სისტემების მოდელირების სფეროში პლატფორმადამოუკიდებელი და დომენზე ორიენტირებული მოდელირების მიმართულებით.

ლიტერატურა:

1. Jablonski St., Volz B., Turkia E. Integrating (Meta-) Models in a Unified Context. The Intern. Scient. Conf. Devoted to the 80-th anniversary acad. I.V.Prangishvili, 2010
2. Volz B., Jablonski St. Towards an Open Meta Modeling Environment, 10th Workshop on Domain-Specific Modeling (DSM 2010)
3. Henderson-Sellers B., Gonzalez-Perez C. Connecting Powertypes and Stereotypes”, Journal of Object Technology, Vol.4, no.7, pp. 83-96, 2005
4. http://www.omg.org/technology/documents/profile_catalog.htm

MODERN TECHNOLOGIES OF SOFTWARE ENGINEERING DEVELOPMENT

Turkia Ekaterine

Georgian Technical University

Summary

In the Article the modern concepts of development of the software -engineering technology as well as the main direction of perfection of UML under modern standards and principles of meta-modeling are considered. The newest developing paradigm of software engineering MDD (Model-Driven Development) is described and the application thresholds of this technology along with the reflection forms in modern modeling tools are analysed.

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАЗВИТИЯ ПРОГРАММНОЙ ИНЖИНИРИИ

Туркия Е. Г.

Грузинский Технический Университет

Резюме

В статье рассмотрены современные концепции развития технологии Программной Инженерии, основные направления совершенствования UML по современным стандартам и принципам мета-моделирования. Описана новейшая развивающаяся парадигма Программной Инженерии - Технология Модельной Инженерии (MDD - Model-Driven Development), проанализированы границы использования этой технологии и формы отражения в современных инструментальных средствах моделирования.