AGL: Incorporating Behavioral Aspects into Domain-Driven Design

Duc-Hanh Dang[a,b,*], Duc Minh Le[c], Van-Vinh Le[a]

[a] *Department of Software Engineering, VNU University of Engineering and Technology, Vietnam*
[b] *Vietnam National University, Hanoi*
[c] *Department of Information Technology, Swinburne Vietnam, FPT Univeristy*

Abstract

Context: Domain-driven design (DDD) aims to iteratively develop software around a realistic domain model. Recent research in DDD has been focusing on using annotation-based domain-specific languages (aDSLs) to build the domain model. However, within current approaches behavioral aspects, that are often represented using UML Activity and State machine diagrams, are not explicitly captured in the domain model.
Objective: The focus of this paper is to introduce a new approach for incorporating behavioral aspects into domain models within the Domain-Driven Design (DDD) approach. The proposed approach involves using a new activity graph language (AGL) as an aDSL for representing behavioral aspects within a unified domain model. This integration of AGL and the previously developed aDSL (DCSL to represent domain models) aims to achieve three important features of DDD: feasibility, productivity, and understandability.
Method: Our approach involves building a unified class model in DCSL within a domain-driven architecture, which uses the annotation attachment feature of the host programming language (such as Java) to attach AGL activity graphs directly to the activity class of the unified class model, resulting in a unified domain model. In this work, we define the abstract and concrete syntax of AGL. To demonstrate our method, we use a Java framework called jDomainApp and evaluate AGL through a case study to show that it is expressive and practical for real-world software.
Results: This paper presents two contributions. Firstly, it proposes a mechanism to include behavioral aspects in a unified domain model by introducing a new aDSL called AGL to represent domain behaviors. Secondly, it presents a unified modeling method for domain-driven software development.
Conclusion: Our method significantly extends the state-of-the-art in DDD in two important fronts: constructing a unified domain model for both structural and behavioral aspects of domain models and bridging the gaps between model and code.

Keywords: Domain-driven design (DDD); Module-based Architecture; Domain-specific language (DSL); UML/OCL-based domain modelling; Attribute-oriented Programming (AtOP)

*Corresponding author
Email addresses: hanhdd@vnu.edu.vn (Duc-Hanh Dang), duclm20@fe.edu.vn (Duc Minh Le), vinhskv@gmail.com (Van-Vinh Le)