

TRÍCH YẾU LUẬN ÁN

1. Thông tin chung về luận án

- Tên tác giả: Lê Văn Vinh
- Tên luận án: Nghiên cứu các kỹ thuật biểu diễn và chuyển đổi mô hình cho thiết kế hướng miền
- Ngành khoa học của luận án: Kỹ thuật phần mềm Mã số: 9480103
- Tên đơn vị đào tạo: Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

2. Nội dung bản trích yếu

2.1. Mục đích nghiên cứu

Luận án nghiên cứu đề xuất các kỹ thuật biểu diễn và chuyển đổi mô hình cho thiết kế hướng miền, nhằm thu hẹp khoảng cách giữa mô hình miền ở mức khái niệm và mô hình miền có khả năng thực thi. Luận án hướng đến việc biểu diễn đầy đủ các khía cạnh của miền, bao gồm các khía cạnh như cấu trúc, hành vi và chính sách bảo mật; Đồng thời nghiên cứu các kỹ thuật sinh tự động phần mềm đúng yêu cầu, đảm bảo chất lượng và tuân thủ nguyên lý thiết kế hướng miền (DDD). Các nội dung chính của luận án bao gồm:

- Mở rộng mô hình miền để tích hợp hành vi, bảo mật miền và các ràng buộc OCL, tạo ra mô hình miền giàu thông tin và có khả năng thực thi.
- Xây dựng ngôn ngữ mô hình miền hợp nhất (UDML) nhằm tích hợp có hệ thống các ngôn ngữ chuyên biệt miền (DSL) theo mỗi quan tâm vào một mô hình miền hợp nhất.
- Nghiên cứu bộ chuyển đổi mô hình nhằm gia tăng tự động hóa phát triển phần mềm từ mô hình miền trong bối cảnh DDD.

2.2. Đối tượng nghiên cứu

Đối tượng nghiên cứu của luận án là mô hình miền trong DDD, cùng với các kỹ thuật dựa trên DSL để biểu diễn, tích hợp và chuyển đổi mô hình miền, nhằm hỗ trợ khả năng thực thi, kiểm chứng hình thức và sinh phần mềm tự động.

2.3. Phương pháp nghiên cứu

Luận án đề xuất áp dụng phương pháp mô hình hóa ngôn ngữ chuyên biệt miền, thiết kế, kiểm chứng hình thức, phát triển lặp lại dựa trên một mô hình miền để xây dựng ngôn ngữ và chuyển đổi mô hình, kết hợp thực nghiệm và đánh giá trên các ca nghiên cứu điển hình. Tiếp đó, Luận án tập trung giải quyết bài toán nghiên cứu các kỹ thuật biểu diễn và chuyển đổi mô hình cho thiết kế hướng miền.

2.4. Các kết quả chính và kết luận

Luận án đạt được các kết quả nghiên cứu sau:

- Đề xuất phương pháp và công cụ hỗ trợ biểu diễn mô hình miền trong thiết kế hướng miền, cho phép đặc tả một cách tường minh và nhất quán các khía cạnh cấu trúc, hành vi và ràng buộc nghiệp vụ. Phương pháp này cho phép tăng khả năng diễn đạt của mô hình miền, đảm bảo sự thống nhất của các bên liên quan về yêu cầu phần mềm.
- Đề xuất kỹ thuật hợp nhất các ngôn ngữ chuyên biệt miền theo các mối quan tâm chuyên biệt vào một mô hình miền hợp nhất có khả năng thực thi. Trên cơ sở đó, luận án xây dựng nền tảng ngữ nghĩa phục vụ kiểm chứng hình thức, nhằm đảm bảo tính nhất quán ngữ nghĩa của mô hình miền hợp nhất.
- Đề xuất kỹ thuật và công cụ hỗ trợ chuyển đổi mô hình để thao tác trên mô hình miền hợp nhất, cho phép sinh tự động các bản mẫu phần mềm có khả năng thực thi, đồng thời bảo toàn ngữ nghĩa miền và chất lượng chuyển đổi, phù hợp với các kịch bản ca sử dụng cụ thể trong các miền ứng dụng chuyên biệt.

Các kết quả này có ý nghĩa hỗ trợ đặc tả chính xác các mô hình miền, đồng thời hiện thực hóa các thao tác tự động trên mô hình miền, bao gồm, sinh tự động mã nguồn, kiểm tra tính tuân thủ với thiết kế và cài đặt. Kết quả này có ý nghĩa gia tăng tự động hóa trong phát triển phần mềm và đã được công bố trong kỷ yếu hội nghị và các bài báo tạp chí được lập chỉ mục ISI/Scopus:

1. Duc-Hanh Dang, Duc Minh Le, Van-Vinh Le. AGL: Incorporating behavioral aspects into domain-driven design. *Information and Software Technology*, 163, 107284, 2023. DOI: 10.1016/j.infsof.2023.107284. (WoS/Scopus, Q1)
2. Van-Vinh Le, Nghia-Trong Be, Duc-Hanh Dang. On Automatic Generation of Executable Domain Models for Domain-Driven Design. *Proc. 15th. Int. Conf. Knowledge and Systems Engineering (KSE)*, IEEE, 2023. DOI: 10.1109/kse59128.2023.10299453. (WoS/Scopus)
3. Van-Vinh Le, Duc-Hanh Dang. An Approach to Composing Concerns for an Executable Unified Domain Model. *Proc. 18th. Int. Conf. Computing and Communication Technologies (RIVF)*, pages 415–419, IEEE, 2024. DOI: 10.1109/RIVF64335.2024.11009109. (WoS/Scopus)
4. Le Van Vinh, Dang Duc Hanh. RM2UDM: A method for automatically generating functional prototypes from requirement models. *Proc 17th. Nat. Conf Fundamental and Applied Information Technology Research (FAIR)*, pages 734–741, 2024. ISBN: 978-604-357-304-6, DOI: 10.15625/vap.2024.0271.
5. Van-Vinh Le, Nhat-Hoang Nguyen, Duc-Quyen Nguyen, Duc-Hanh Dang. A Method for Composing Concerns into a Unified Domain Model in Domain-Driven Design. *Proc.*

14th Int. Symp. on Information and Communication Technology (SOICT), Springer, 2025. (WoS/Scopus, accepted)

6. Van-Vinh Le, Duc-Hanh Dang. Constraint Annotation Patterns for Prototype Generation within Domain-Driven Design. *e-Informatica Software Engineering Journal*, 2025. ISSN: 2084-4840. (WoS/Scopus, Q3, submitted)
7. Van-Vinh Le, Nhat-Hoang Nguyen, Duc-Quyen Nguyen, Duc-Hanh Dang. A Semantic Framework and Tool Support for Unified Executable Domain Models in UDML: A Case Study on the RBAC Concern. *VNU Journal of Science: Computer Science and Communication Engineering*, Vol 42 No 1, pages 79-114, 2026. ISSN: 2615-9260, DOI: 10.25073/2588-1086/vnucsce.6743.

Hà Nội, ngày tháng năm 20...

NGHIÊN CỨU SINH
(Ký và ghi rõ họ, tên)

CÁN BỘ HƯỚNG DẪN
(Ký và ghi rõ họ, tên)

XÁC NHẬN CỦA CƠ SỞ ĐÀO TẠO

THE ABSTRACT OF DOCTORAL THESIS

1. General information about the thesis

Author: Le Van Vinh

Title of the doctoral thesis: An Investigation of Model Representation and Transformation Techniques for Domain-Driven Design

Field of study: Software Engineering Code: 9480103.01

Training institution: University of Engineering and Technology, Vietnam National University

2. Abstract content

2.1. *Research Purpose*

Proposing techniques for representing domain models in the context of Domain-Driven Design (DDD), enabling the explicit and consistent specification of structural, behavioral, and business-constraint aspects of the business domain. These techniques extend the expressive power of domain models and provide a foundation for subsequent model integration, formal verification, and model transformation.

Proposing a technique for unifying domain-specific languages (DSLs) associated with heterogeneous concerns into a unified, executable domain model. On this basis, the dissertation establishes a semantic foundation for formal verification, aiming to ensure the semantic consistency of the unified domain model at the design stage.

Proposing model transformation techniques based on the unified domain model, enabling the automatic generation of executable software prototypes while preserving domain semantics and transformation quality, and supporting domain-specific usage scenarios in specialized application domains.

2.2. *Research Object*

The research object of this dissertation is the domain model in DDD, together with DSL-based techniques for representing, integrating, and transforming domain models, with the aim of supporting executability, formal verification, and automatic software generation.

2.3. *Research methods*

For achieving the research purpose, methods for domain modeling, DSL design, semantic integration of heterogeneous concerns, model transformation, and formal verification are employed. In addition, a design-and-implementation-based research methodology is

adopted, including the development of supporting tools and empirical evaluation through representative case studies.

2.4. Major Results and Conclusions

This thesis presents a model-driven approach to narrowing the gap between domain modeling and executable software development in the context of DDD. The research focuses on three core problems: (i) expressive and executable representation of domain models, (ii) semantic integration of heterogeneous domain concerns, and (iii) semantics-preserving model transformation for automated software generation. The major results and conclusions of the dissertation are summarized as follows.

First, the dissertation proposes a set of techniques for rich and executable domain model representation in DDD. We identified that existing DDD-based modeling approaches mainly emphasize structural aspects and provide limited support for explicitly capturing domain behavior, business constraints, and security policies in a unified manner. To address this limitation, we proposed: (1) *Activity Graph Language (AGL)* for representing domain behavior, (2) *Constraint Annotation Patterns (CAP)* for integrating complex business constraints derived from OCL, and (3) *RBACDom* for modeling role-based access control policies at the domain level. These techniques extend annotation-based domain-specific languages (aDSLs) to allow domain models to capture structural, behavioral, and policy-related semantics consistently. As a result, domain models become richer in semantics while remaining closely aligned with the ubiquitous language and feasible for execution-oriented development.

Second, the dissertation proposes UDML (Unified Domain Model Language) as a unified domain modeling language that integrates multiple domain concerns into a single executable domain model. We formulated the problem of concern integration in DDD as a semantic unification problem, where heterogeneous DSLs—each addressing a specific concern such as structure, behavior, or security—must be composed without semantic conflicts. To solve this problem, we introduced an annotation-based composition mechanism that merges concern-specific DSLs at the abstract syntax and semantic levels. On this basis, we defined a unified semantic foundation for UDML, enabling formal reasoning and consistency checking over the integrated domain model. This result establishes domain models as the semantic core of the development process, rather than as disconnected design artifacts.

Finally, the dissertation proposes a semantics-preserving model transformation framework that enables automated generation of executable software prototypes from unified domain models. We designed a systematic transformation chain from requirement models to unified domain models and further to executable software artifacts. The proposed transformations preserve domain semantics, including behavioral logic, constraints, and security rules, across different abstraction levels.

The framework supports both model-to-model (M2M) and model-to-text (M2T) transformations and is implemented on top of a modular software architecture. Experimental results from multiple case studies—including CourseMan, OrderMan, ProcessMan, and OJS—demonstrate that the proposed transformations can automatically generate executable software prototypes that conform to domain requirements and DDD principles.

The dissertation implements a prototype toolchain to validate the feasibility and applicability of the proposed approaches. Through empirical evaluation on representative domain scenarios, the results show that the proposed techniques effectively support executable domain modeling, semantic integration, and automated software generation. The findings confirm that the approach significantly reduces manual effort, improves semantic consistency between models and implementations, and enhances the maintainability and evolvability of DDD-based software systems.

In conclusion, this dissertation contributes a coherent and systematic framework for domain modeling, semantic integration, and model transformation in DDD. By placing the domain model at the center of both semantics and execution, the proposed approach bridges the long-standing gap between domain specification and software realization, and provides a solid foundation for future research on executable domain models and model-driven DDD.

These results have been published in conference proceedings and ISI/Scopus-indexed journal papers.

1. Duc-Hanh Dang, Duc Minh Le, Van-Vinh Le. AGL: Incorporating behavioral aspects into domain-driven design. *Information and Software Technology*, 163, 107284, 2023. DOI: 10.1016/j.infsof.2023.107284. (WoS/Scopus, Q1)
2. Van-Vinh Le, Nghia-Trong Be, Duc-Hanh Dang. On Automatic Generation of Executable Domain Models for Domain-Driven Design. *Proc. 15th. Int. Conf. Knowledge and Systems Engineering (KSE)*, IEEE, 2023. DOI: 10.1109/kse59128.2023.10299453. (WoS/Scopus)
3. Van-Vinh Le, Duc-Hanh Dang. An Approach to Composing Concerns for an Executable Unified Domain Model. *Proc. 18th. Int. Conf. Computing and Communication Technologies (RIVF)*, pages 415–419, IEEE, 2024. DOI: 10.1109/RIVF64335.2024.11009109. (WoS/Scopus)
4. Le Van Vinh, Dang Duc Hanh. RM2UDM: A method for automatically generating functional prototypes from requirement models. *Proc 17th. Nat. Conf Fundamental and Applied Information Technology Research (FAIR)*, pages 734–741, 2024. ISBN: 978-604-357-304-6, DOI: 10.15625/vap.2024.0271.
5. Van-Vinh Le, Nhat-Hoang Nguyen, Duc-Quyen Nguyen, Duc-Hanh Dang. A Method for Composing Concerns into a Unified Domain Model in Domain-Driven Design. *Proc.*

14th Int. Symp. on Information and Communication Technology (SOICT), Springer, 2025. (WoS/Scopus, accepted)

6. Van-Vinh Le, Duc-Hanh Dang. Constraint Annotation Patterns for Prototype Generation within Domain-Driven Design. *e-Informatica Software Engineering Journal*, 2025. ISSN: 2084-4840. (WoS/Scopus, Q3, submitted)
7. Van-Vinh Le, Nhat-Hoang Nguyen, Duc-Quyen Nguyen, Duc-Hanh Dang. A Semantic Framework and Tool Support for Unified Executable Domain Models in UDML: A Case Study on the RBAC Concern. *VNU Journal of Science: Computer Science and Communication Engineering*, Vol 42 No 1, pages 79-114, 2026. ISSN: 2615-9260, DOI: 10.25073/2588-1086/vnucsce.6743.

HaNoi, day month year

DOCTORAL CANDIDATE
(Signature and full name)

SUPERVISOR
(Signature and full name)

CONFIRMATION FROM THE TRAINING UNIVERSITY