

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

LÊ VĂN VINH

NGHIÊN CỨU CÁC KỸ THUẬT
BIỂU DIỄN VÀ CHUYỂN ĐỔI MÔ HÌNH
CHO THIẾT KẾ HƯỚNG MIỀN

Chuyên ngành: Kỹ thuật phần mềm

Mã số: 9480103

TÓM TẮT LUẬN ÁN TIẾN SĨ KỸ THUẬT PHẦN MỀM

Hà Nội - 2026

Công trình được hoàn thành tại:

Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

Người hướng dẫn khoa học: PGS.TS. Đặng Đức Hạnh

Phản biện 1: PGS.TS. Nguyễn Mạnh Hùng

Nơi công tác: Học viện Công nghệ Bưu chính Viễn Thông

Phản biện 2: TS. Nguyễn Thanh Hùng

Nơi công tác: Trường CNTT và Truyền thông, ĐHBKHN

Phản biện 3: PGS.TS. Nguyễn Việt Hùng

Nơi công tác: Học viện Kỹ thuật Quân sự

Luận án được bảo vệ tại Hội đồng chấm luận án cấp Đại học Quốc

Gia họp tại: vào hồi ... giờ ngày ... tháng ... năm 2026

Có thể tìm hiểu luận án tại:

- Thư viện Quốc gia Việt Nam,
- Trung tâm thông tin - Thư viện, Đại học Quốc gia Hà Nội.

Hà Nội - 2026

Tóm tắt

Sự gia tăng về độ phức tạp, quy mô và tính không đồng nhất của các hệ thống phần mềm hiện đại đặt ra nhu cầu cấp thiết về các phương pháp phát triển phần mềm có khả năng gắn kết chặt chẽ tri thức miền với quá trình triển khai. Khoảng cách ngữ nghĩa giữa mô hình miền và hệ thống thực thi trở thành một thách thức cốt lõi. Trong bối cảnh đó, thiết kế hướng miền (DDD) được xem là một phương pháp tiếp cận hiệu quả, nhấn mạnh phát triển phần mềm lập dựa trên mô hình miền giàu ngữ nghĩa, phản ánh logic cốt lõi và các quy tắc của miền. DDD đề cao việc sử dụng nhất quán ngôn ngữ phổ quát (UL), qua đó tăng cường sự hiểu biết chung và giao tiếp giữa chuyên gia miền và nhóm phát triển trong toàn bộ vòng đời phần mềm. Cùng với DDD, các ngôn ngữ chuyên biệt miền (DSL) được sử dụng rộng rãi nhằm mã hóa trực tiếp các khái niệm miền trong đặc tả và mã nguồn, góp phần nâng cao tính chính xác, khả năng bảo trì và mở rộng của phần mềm. Gần đây, nhiều nghiên cứu tập trung vào DSL nội sinh dựa trên chú thích (aDSL) nhằm xây dựng mô hình miền có khả năng thực thi. Tuy nhiên, các tiếp cận hiện có vẫn bộc lộ nhiều hạn chế: khía cạnh hành vi thường được mô tả tách rời bằng biểu đồ hoạt động hoặc máy trạng thái UML; các ràng buộc OCL phức tạp chưa được tích hợp đầy đủ; và việc sử dụng các DSL theo các mối quan tâm không đồng nhất (như hành vi, bảo mật) làm phân mảnh mô hình miền, suy giảm tính nhất quán ngữ nghĩa và khả năng tự động hóa.

Xuất phát từ các thách thức đó, luận án đề xuất các kỹ thuật biểu diễn và chuyển đổi mô hình cho thiết kế hướng miền nhằm xây dựng mô hình miền hợp nhất (UDML) giàu thông tin và có khả năng thực thi. Cụ thể, luận án đề xuất các kỹ thuật biểu diễn tích hợp cấu trúc, hành vi, bảo mật và ràng buộc OCL vào mô hình miền, kèm theo đặc tả và kiểm chứng ngữ nghĩa hình thức. Trên cơ sở đó, một phương pháp tích hợp các DSL theo các mối quan tâm không đồng nhất vào mô hình miền hợp nhất được xây dựng, cùng với chuỗi chuyển đổi mô hình nhằm sinh tự động bản mẫu phần mềm, đảm bảo bảo toàn ngữ nghĩa và khả năng truy vết.

Để đánh giá hiệu quả của phương pháp đề xuất, luận án xây dựng công cụ hỗ trợ và tiến hành thực nghiệm trên các ví dụ điển hình, cho thấy tính khả thi và hiệu quả của phương pháp trong thực tiễn.

Từ khóa: Thiết kế hướng miền, Kỹ thuật biểu diễn, chuyển đổi mô hình, UDML, DSL, aDSL.

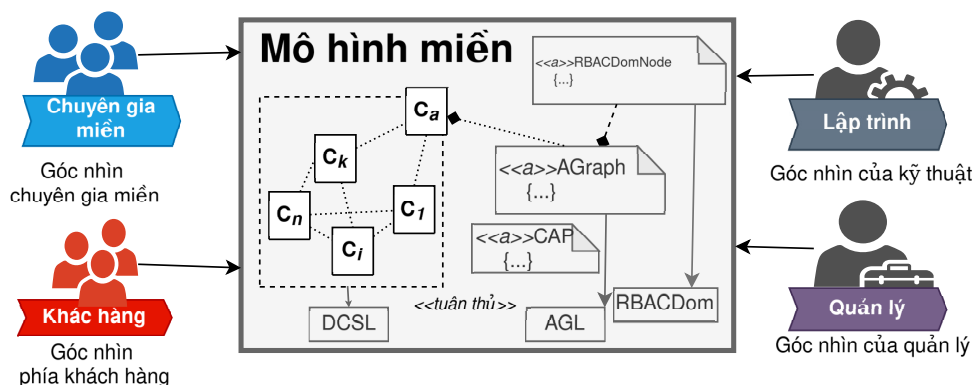
Chương 1

GIỚI THIỆU

1.1 Đặt vấn đề

1.1.1 Bối cảnh nghiên cứu về các kỹ thuật biểu diễn và chuyển đổi mô hình cho thiết kế hướng miền

Trong bối cảnh các hệ thống phần mềm ngày càng phức tạp và đa dạng miền ứng dụng, các phương pháp phát triển dựa chủ yếu vào cách tiếp cận hướng mã nguồn, thiếu tự động hóa bộc lộ nhiều hạn chế về chi phí, thời gian và khả năng bảo trì. Thiết kế hướng miền (DDD) nhấn mạnh vai trò trung tâm của mô hình miền (DM) như một lõi ngữ nghĩa và ngôn ngữ chung giữa các bên liên quan. Theo đó, các góc nhìn khác nhau trên cùng một hệ thống cần được quy chiếu về một mô hình miền thống nhất, như được minh họa trong Hình 1.1. Hình này cho thấy mô hình miền đóng vai trò lõi ngữ nghĩa, liên kết và điều phối các góc nhìn khác nhau của hệ thống (như chuyên gia miền, lập trình, khách hàng và quản lý), qua đó đảm bảo tính nhất quán và làm cơ sở cho việc chuyển đổi sang hệ thống thực thi.



Hình 1.1: Các góc nhìn khác nhau trên cùng một mô hình miền.

Tuy nhiên, trong thực tiễn, mô hình miền thường được đặc tả bằng UML/OCL và hiện thực hóa thủ công, dẫn đến khoảng cách giữa đặc tả miền và hệ thống thực thi. Bên cạnh đó, các cách tiếp cận hiện có vẫn chưa hỗ trợ đầy đủ việc biểu diễn, tích hợp và chuyển đổi một cách có hệ thống các khía cạnh cấu trúc, hành vi và ràng buộc nghiệp vụ trong một mô hình miền hợp nhất có khả năng thực thi.

1.1.2 Các thách thức trong nghiên cứu các kỹ thuật biểu diễn và chuyển đổi mô hình cho thiết kế hướng miền

Luận án xác định ba thách thức chính. Thứ nhất, thiếu cơ chế biểu diễn thống nhất các khía cạnh cấu trúc, hành vi và ràng buộc nghiệp vụ trong một mô hình miền hợp nhất có khả năng thực thi. Thứ hai, việc tích hợp các mối quan tâm thông qua các DSL còn phân mảnh, chưa có cơ chế hợp nhất đảm bảo tính nhất quán ngữ nghĩa. Thứ ba, các chuỗi chuyển đổi mô hình từ đặc tả đến hiện thực chưa có tính hệ thống và chưa bảo đảm bảo toàn ngữ nghĩa. Từ đó, khoảng trống nghiên cứu là thiếu một phương pháp tổng thể cho biểu diễn, tích hợp và chuyển đổi mô hình miền một cách nhất quán và có khả năng thực thi.

1.2 Mục tiêu, đối tượng, phương pháp, phạm vi nghiên cứu

1.2.1 Mục tiêu nghiên cứu

Luận án đề xuất các kỹ thuật biểu diễn và chuyển đổi mô hình trong DDD nhằm thu hẹp khoảng cách giữa mô hình miền và hệ thống thực thi. Các kỹ thuật này cho phép biểu diễn đầy đủ các khía cạnh cấu trúc, hành vi, bảo mật và các ràng buộc OCL, đồng thời hỗ trợ sinh tự động phần mềm, đồng thời bảo đảm chất lượng và tuân thủ nguyên lý DDD. Các mục tiêu nghiên cứu cụ thể 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 DSL theo mối 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.

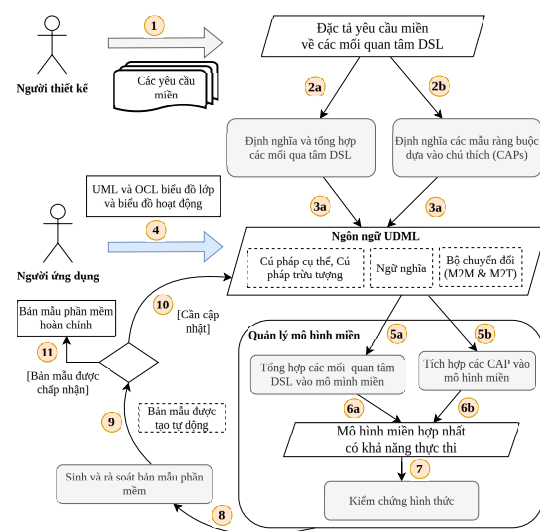
1.2.2 Đối tượng, phương pháp và phạm vi nghiên cứu

Đối tượng nghiên cứu: Các kỹ thuật biểu diễn mô hình miền và chuyển đổi mô hình trong thiết kế hướng miền. Phương pháp nghiên cứu: Phân tích tài liệu, mô hình hóa DSL, thiết kế, kiểm chứng hình thức–xây dựng ngôn ngữ và chuyển đổi mô hình, kết hợp thực nghiệm và đánh giá trên các ca nghiên cứu điển hình. Phạm vi nghiên cứu: Tập trung vào DSL cho DDD, áp dụng cho DSL nội sinh và ngoại sinh, mở rộng miền cấu trúc bằng tích hợp hành vi, ràng buộc, hợp nhất nhiều mối quan tâm miền, đánh giá qua CourseMan, OrderMan, ProcessMan và OJS.

1.3 Nội dung nghiên cứu

Luận án tập trung nghiên cứu các kỹ thuật biểu diễn, hợp nhất và chuyển đổi mô hình miền (DM) trong bối cảnh thiết kế hướng miền. Các mối quan tâm miền như cấu trúc, hành vi, bảo mật và các ràng buộc OCL được đặc tả thông qua các DSL và được tích hợp vào một mô hình miền hợp nhất (UDML) với đặc tả cú pháp và ngữ nghĩa đầy đủ. Trên cơ sở đó, luận án đề xuất các cơ chế chuyển đổi mô hình và nền tảng ngữ nghĩa phục vụ kiểm chứng hình thức, hướng tới xây dựng các mô hình miền có khả năng thực thi và hỗ trợ sinh tự động bản mẫu phần mềm.

Hình 1.2 minh họa tổng quan phương pháp đề xuất. Trong đó, các mối quan tâm miền được đặc tả bằng DSL và các mẫu ràng buộc được xây dựng, tích hợp vào mô hình miền ban đầu để suy dẫn mô hình miền hợp nhất. Mô hình này sau đó được đặc tả ngữ nghĩa hình thức và kiểm chứng trước khi được sử dụng làm đầu vào cho chuỗi chuyển đổi mô hình nhằm sinh tự động bản mẫu phần mềm. Quy trình được thực hiện theo hướng lặp, cho phép hiệu chỉnh mô hình và các đặc tả liên quan dựa trên phản hồi.



Hình 1.2: Tổng quan phương pháp đề xuất cho kỹ thuật biểu diễn và chuyển đổi mô hình.

1.4 Các đóng góp chính của luận án

Luận án có các đóng góp chính như sau: (i) Đề xuất phương pháp biểu diễn mô hình miền có khả năng thực thi, cho phép tích hợp các khía cạnh cấu trúc, hành vi, bảo mật và ràng buộc nghiệp vụ, đồng thời xây dựng nền tảng ngữ nghĩa và cơ chế kiểm chứng hình thức cho mô hình. (ii) Đề xuất Ngôn ngữ mô hình miền hợp nhất (UDML), được xây dựng dựa trên tiếp cận siêu mô hình và cây cú pháp trừu tượng, và được kiểm chứng hình thức bằng Event-B. (iii) Đề xuất các kỹ thuật chuyển đổi mô hình cho phép sinh tự động các bản mẫu phần mềm từ mô hình miền, đồng thời đảm bảo bảo toàn ngữ nghĩa trong quá trình chuyển đổi.

Chương 2

CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU

Chương này trình bày cơ sở lý thuyết và tổng quan các nghiên cứu liên quan đến biểu diễn mô hình miền, tích hợp các mối quan tâm và chuyển đổi mô hình trong thiết kế hướng miền. Mục tiêu là làm rõ các hạn chế của các tiếp cận hiện có trong việc biểu diễn đầy đủ cấu trúc, hành vi, ràng buộc và bảo mật, cũng như trong việc hợp nhất các mối quan tâm và bảo toàn ngữ nghĩa khi sinh phần mềm.

2.1 Tổng quan về thiết kế hướng miền

Thiết kế hướng miền (DDD) là phương pháp phát triển phần mềm dựa trên mô hình miền phản ánh tri thức nghiệp vụ. Mô hình miền (DM) đóng vai trò trung tâm, vừa là lõi ngữ nghĩa vừa định hướng hiện thực phần mềm. Theo đó, DM cần đảm bảo khả năng biểu đạt tri thức miền và khả năng chuyển hóa thành mã nguồn. DDD nhấn mạnh việc sử dụng ngôn ngữ phổ quát (UL) nhằm duy trì sự nhất quán giữa mô hình và hệ thống. Tuy nhiên, trong thực tiễn, DM thường được đặc tả bằng UML/OCL và tách rời khỏi mã nguồn, dẫn đến khoảng cách ngữ nghĩa giữa mô hình và hệ thống thực thi.

2.1.1 Mô hình miền hướng thực thi

Trong nhiều hệ thống hiện nay, DM chủ yếu mang tính mô tả và chưa có khả năng thực thi trực tiếp. Hành vi miền thường bị tách khỏi mô hình và được hiện thực trong mã hoặc UML độc lập. Các tiếp cận dựa trên ngôn ngữ chuyên biệt miền (DSL), đặc biệt là DSL nội sinh dựa trên chú thích (aDSL), cho phép gắn mô hình vào mã nguồn và thu hẹp khoảng cách mô hình–triển khai. Tuy nhiên, các tiếp cận này vẫn hạn chế trong việc tích hợp hành vi phức tạp và nhiều mối quan tâm trong một DM thống nhất.

2.1.2 Ngôn ngữ chuyên biệt miền

Ngôn ngữ chuyên biệt miền (DSL) là công cụ quan trọng để biểu diễn DM với mức trừu tượng cao và sát với UL. DSL có thể là ngoại sinh hoặc nội

sinh. Trong DDD, DSL giúp liên kết mô hình với mã nguồn và hỗ trợ tự động hóa phát triển phần mềm.

Các DSL nội sinh, đặc biệt aDSL, cho phép nhúng trực tiếp ngữ nghĩa miền vào lớp miền và hỗ trợ sinh tự động phần mềm. Tuy nhiên, việc tích hợp đồng thời nhiều khía cạnh như cấu trúc, hành vi và ràng buộc trong một mô hình thống nhất vẫn còn hạn chế.

2.2 Các hướng tiếp cận biểu diễn mô hình miền

UML/OCL là các công cụ phổ biến để biểu diễn DM, trong đó UML mô tả cấu trúc và OCL đặc tả ràng buộc. Tuy nhiên, cách tiếp cận này phân tách các khía cạnh của mô hình và thiếu khả năng hỗ trợ thực thi trực tiếp.

Hành vi miền thường được mô tả bằng UML hoặc DSL riêng biệt, dẫn đến khó khăn trong việc hợp nhất với cấu trúc miền. Các tiếp cận aDSL cho phép gắn hành vi vào mã nguồn nhưng chưa cung cấp cơ chế biểu diễn hành vi nghiệp vụ một cách hình thức và nhất quán.

Các chính sách bảo mật như RBAC thường được tích hợp thủ công và thiếu cơ chế kiểm chứng hình thức, gây khó khăn trong việc đảm bảo tính nhất quán và an toàn.

2.3 Các hướng tiếp cận tích hợp các mối quan tâm trong mô hình miền

Một mối quan tâm là một khía cạnh tương đối độc lập của miền như cấu trúc, hành vi hoặc bảo mật. Trong thực tế, nhiều mối quan tâm cần được tích hợp trong cùng một mô hình miền. Khi mỗi mối quan tâm được đặc tả bằng một DSL riêng, bài toán đặt ra là hợp nhất các đặc tả này thành một mô hình thống nhất, có khả năng thực thi và bảo toàn ngữ nghĩa.

Các hướng tiếp cận hiện có bao gồm hợp nhất dựa trên siêu mô hình và dựa trên cây cú pháp trừu tượng (AST). Đối với bảo mật, các chính sách như RBAC thường được xử lý tách rời hoặc tích hợp thủ công, dẫn đến thiếu nhất quán với hành vi miền; các tiếp cận hình thức hiện có cũng chủ yếu xem bảo mật như một mối quan tâm độc lập, chưa được tích hợp chặt chẽ vào mô hình miền. Nhìn chung, các tiếp cận hiện tại chưa cung cấp một cơ chế hợp nhất cho phép tích hợp đồng thời các mối quan tâm về cấu trúc, hành vi, ràng buộc và bảo mật trong một DM thống nhất, có khả năng thực

thi và được kiểm chứng hình thức. Đây chính là khoảng trống mà luận án hướng tới giải quyết thông qua việc xây dựng mô hình miền hợp nhất.

2.4 Các thao tác chuyển đổi mô hình miền

Mục này trình bày các thao tác chuyển đổi mô hình miền trong khuôn khổ kỹ nghệ phần mềm hướng mô hình, nhấn mạnh vai trò của chuyển đổi mô hình trong việc liên kết các mức trừu tượng khác nhau của mô hình miền.

Nhiều tiếp cận hiện nay mới tập trung vào sinh giao diện hoặc một phần chức năng, trong khi chưa thiết lập được một chuỗi chuyển đổi đầy đủ từ mô hình yêu cầu đến mô hình miền thực thi và bản mẫu phần mềm hoàn chỉnh. Vì vậy, nhu cầu nghiên cứu các bộ chuyển đổi mô hình miền có khả năng tích hợp đồng thời cấu trúc và hành vi, làm nền tảng cho sinh tự động bản mẫu phần mềm trong các chương tiếp theo.

2.5 Hướng tiếp cận sử dụng AI/LLM

Các mô hình ngôn ngữ lớn (LLM) có thể hỗ trợ sinh mô hình từ đặc tả ngôn ngữ tự nhiên. Tuy nhiên, các tiếp cận này thiếu nền tảng ngữ nghĩa hình thức, không đảm bảo tính nhất quán và khả năng truy vết, do đó chưa phù hợp để xây dựng DM làm trung tâm trong DDD.

2.6 Tổng kết chương

Các nghiên cứu hiện có cho thấy nhiều tiếp cận nhằm hỗ trợ mô hình hóa, thực thi và kiểm chứng mô hình miền (DM). Tuy nhiên, các giải pháp này vẫn còn phân tán, thiếu một khuôn khổ thống nhất đặt DM làm lõi ngữ nghĩa trung tâm, cho phép tích hợp đồng thời các mối quan tâm như cấu trúc, hành vi và bảo mật, đồng thời hỗ trợ khả năng thực thi và chuyển đổi mô hình theo tinh thần DDD. Điều này đặt ra nhu cầu phát triển các kỹ thuật biểu diễn và hợp nhất DM giàu ngữ nghĩa, cùng với các cơ chế kiểm chứng và chuyển đổi mô hình bảo toàn ngữ nghĩa, nhằm thu hẹp khoảng cách giữa mô hình miền và hệ thống thực thi. Đây chính là động lực và nền tảng cho các kỹ thuật và phương pháp được đề xuất trong các chương tiếp theo của luận án.

Chương 3

KỸ THUẬT BIỂU DIỄN MÔ HÌNH MIỀN

3.1 Kỹ thuật tích hợp ràng buộc OCL vào mô hình miền

Trong DDD, ràng buộc nghiệp vụ là yếu tố then chốt để bảo toàn ngữ nghĩa miền. Tuy nhiên, UML/OCL thường tách rời khỏi mô hình thực thi; DSL ngoại sinh khó tích hợp và bảo trì; còn aDSL hiện có chỉ hỗ trợ các ràng buộc đơn giản, khó bao phủ OCL phức tạp. DCSL khắc phục một phần nhưng chưa hỗ trợ tái sử dụng và tích hợp các mẫu ràng buộc nâng cao.

Luận án đề xuất CAP (Constraint Annotation Pattern) – kỹ thuật biểu diễn ràng buộc OCL phức tạp dưới dạng các mẫu chú thích tham số hóa. Mỗi CAP đại diện cho một nhóm ràng buộc có cấu trúc tương tự, bảo đảm ánh xạ một–một với OCL để giữ nguyên ngữ nghĩa, đồng thời cho phép tích hợp trực tiếp vào mô hình miền hợp nhất và hỗ trợ sinh bản mẫu phần mềm. Phương pháp CAP gồm hai giai đoạn: (i) xây dựng danh mục CAP bằng cách khái quát hóa các ràng buộc OCL phổ biến; và (ii) áp dụng CAP trong DCSL mở rộng, hợp nhất với các mối quan tâm khác (như AGL) để tạo mô hình miền hợp nhất có thể thực thi và sinh nguyên mẫu tự động.

3.1.1 Tích hợp mẫu CAP vào mô hình miền

Khía cạnh cú pháp. CAP được xem như một *mẫu tham số hóa* để biểu diễn các ràng buộc OCL trong ngữ cảnh mô hình miền. Mỗi lần áp dụng CAP tương ứng với việc gán giá trị cụ thể cho các tham số của mẫu để tạo ra một ràng buộc OCL cụ thể. Các tham số của mẫu được khai báo và cung cấp thông qua cơ chế chú thích; vì vậy, DCSL được mở rộng để có thể đặc tả và quản lý CAP một cách chính quy trong cùng một khung mô hình hóa.

Về cú pháp, một mẫu CAP bao gồm các thành phần chính sau:

Tên mẫu: Định danh duy nhất dùng để quản lý và truy xuất CAP trong danh mục.

Mô tả: Giải thích ngắn gọn mục đích và ngữ nghĩa của nhóm ràng buộc OCL mà CAP biểu diễn.

Mẫu tham số hóa: Biểu diễn một dạng OCL tham số, tuân thủ cú pháp OCL và đại diện cho một họ ràng buộc có cấu trúc tương tự. Mẫu bao gồm:

- *Cấu trúc biểu đồ lớp:* phần cấu trúc liên quan để xác lập ngữ cảnh OCL, với các tham số cho lớp, thuộc tính và quan hệ.
- *Mẫu OCL:* biểu thức OCL ở dạng tham số hóa.
- *Đặc tả chú thích:* tập chú thích có cấu trúc dùng để gán tham số và sinh ra biểu thức OCL cụ thể.

Ví dụ: Minh họa cách áp dụng CAP trong một bài toán miền cụ thể.

Khía cạnh ngữ nghĩa. Ngữ nghĩa của CAP được xác định tương ứng trực tiếp với ngữ nghĩa của OCL. Mỗi thể hiện CAP ánh xạ một-một tới một bất biến OCL trong mô hình miền, do đó việc đánh giá một ràng buộc dựa trên CAP cho kết quả lô-gic tương đương với việc đánh giá bất biến OCL tương ứng.

Definition 1 (Ứng dụng CAP). *Cho CD là biểu đồ lớp biểu diễn mô hình miền, và OCL_{CD} là tập các bất biến OCL hợp lệ có thể xác định trên CD . Cho CAP_i là một mẫu CAP được đặc tả bởi mẫu OCL tham số hóa T_i với không gian tham số \mathcal{D}_i . Một ứng dụng CAP trên CD là quá trình gán một bộ tham số $d_i \in \mathcal{D}_i$ cho T_i để thu được một bất biến OCL hợp lệ trên CD . Quá trình này được mô hình hóa bởi ánh xạ sinh: $\mathcal{G} : (CAP_i, d_i) \mapsto oclInv \in OCL_{CD}$, trong đó $\mathcal{G}(CAP_i, d_i)$ là bất biến OCL cụ thể được sinh ra từ việc thể hiện mẫu CAP T_i với bộ tham số d_i . \square*

3.1.2 Áp dụng mẫu CAP và sinh bản mẫu phần mềm

Luận án đề xuất quy trình áp dụng các mẫu chú thích ràng buộc (CAP) nhằm chuyển đổi các ràng buộc miền thành các mô hình miền hợp nhất có khả năng thực thi. Các ràng buộc được đặc tả ở mức trừu tượng thông qua chú thích CAP, sau đó được tái tạo tự động thành các biểu thức OCL và tích hợp vào mô hình miền. Mô hình hợp nhất thu được được kiểm chứng và sử dụng làm đầu vào cho chuỗi chuyển đổi mô hình nhằm sinh tự động bản mẫu phần mềm. Quy trình được thực hiện theo hướng lặp, cho phép đánh giá và hiệu chỉnh mô hình dựa trên phản hồi của chuyên gia miền.

3.2 Kỹ thuật tích hợp hành vi vào mô hình miền

Phần này trình bày kỹ thuật tích hợp khía cạnh hành vi vào mô hình miền nhằm hỗ trợ sinh tự động phần mềm theo tiếp cận thiết kế hướng miền. Đề xuất tập trung vào việc gắn kết trực tiếp hành vi miền với DM thông qua một DSL nội sinh dựa trên chú thích (aDSL), từ đó nâng cao khả năng biểu đạt và khả năng thực thi của mô hình miền.

3.2.1 Tổng quan về phương pháp đề xuất

Phương pháp đề xuất là một quy trình lặp gồm ba bước. Trước hết, các yêu cầu miền được đặc tả thông qua mô hình miền và biểu đồ hoạt động UML, tương ứng với các góc nhìn cấu trúc và hành vi, sau đó được hợp nhất thành một mô hình miền thống nhất gồm DCSL và AGL. Tiếp theo, DM hợp nhất được sử dụng để sinh tự động các bản mẫu phần mềm và được đánh giá bởi chuyên gia miền. Dựa trên phản hồi, mô hình được hiệu chỉnh và quy trình được lặp lại cho đến khi đạt yêu cầu.

Kỹ thuật tích hợp hành vi miền dựa trên hai nguyên tắc: (i) xác định các hành động nguyên tử cho mỗi lớp miền; và (ii) biểu diễn hành vi như sự phối hợp giữa các mô-đun thông qua biểu đồ hoạt động. Để đảm bảo tính nhất quán ngữ nghĩa, các hành vi được giới hạn trong một tập các mẫu cơ bản như tuần tự, rẽ nhánh, song song và hợp nhất.

Mô hình lớp hợp nhất mở rộng DM bằng cách gắn hành vi vào cấu trúc thông qua các lớp hoạt động, liên kết với các lớp dữ liệu và điều khiển. Cách biểu diễn này cho phép đồng bộ hóa trạng thái hành vi với trạng thái miền, từ đó hình thành một mô hình miền hợp nhất có khả năng thực thi.

3.2.2 Ngữ nghĩa hành động mô-đun

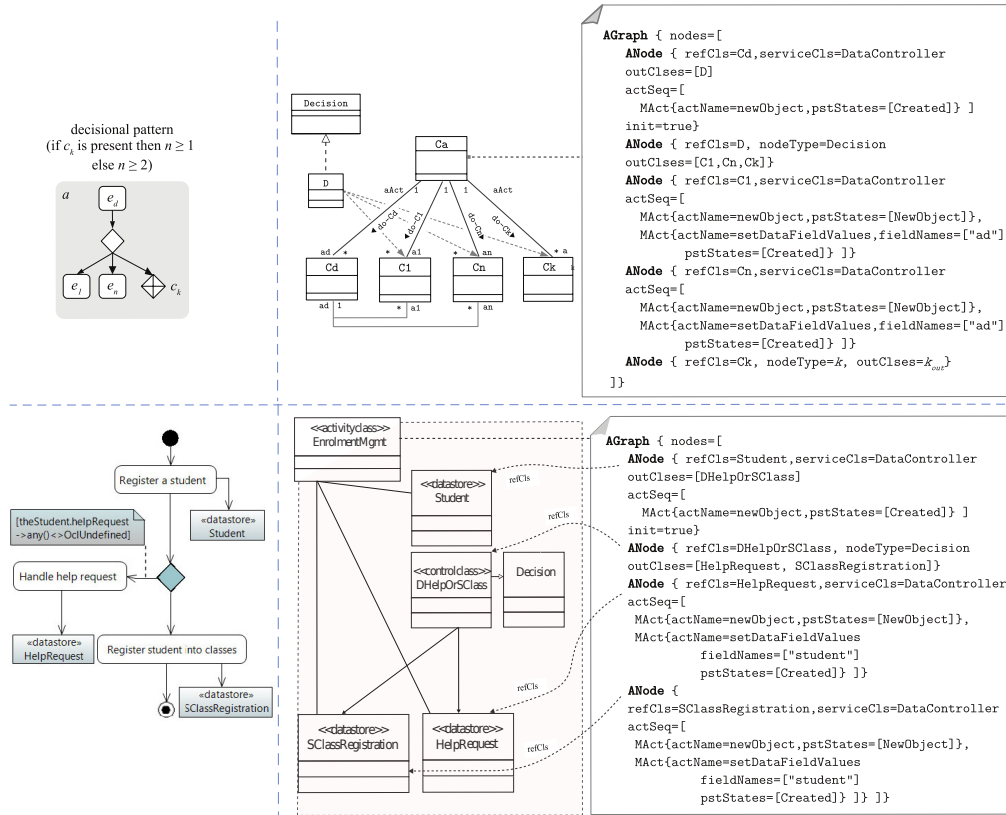
Hành vi miền được đặc tả thông qua các hành động mô-đun, được xây dựng từ các hành động nguyên tử. Mỗi hành động nguyên tử biểu diễn một thao tác cơ bản trên đối tượng miền, với các tiền điều kiện và hậu điều kiện tương ứng. Các hành động nguyên tử có thể được kết hợp thành các chuỗi hành động nhằm mô tả các kịch bản nghiệp vụ phức tạp hơn. Cách tiếp cận này cho phép mô hình hóa hành vi miền một cách có cấu trúc và hỗ trợ suy luận về trạng thái hệ thống trong quá trình thực thi.

3.2.3 Các mẫu hành vi miền

Luận án đề xuất cách tiếp cận dựa trên mẫu nhằm tích hợp hành vi miền vào mô hình miền một cách có hệ thống. Mỗi hành vi miền, được mô tả bằng biểu đồ hoạt động UML, được chuyển đổi thành một mô hình miền hợp nhất thông qua hai thành phần: (i) mở rộng mô hình lớp với các lớp hoạt động và các thành phần liên quan; và (ii) đặc tả lô-gic hành vi tương ứng dưới dạng đồ thị hoạt động được gắn trực tiếp vào mô hình. Quá trình này được thực hiện thông qua các mẫu hành vi miền, đóng vai trò như các khuôn mẫu chuẩn hóa việc ánh xạ giữa hành vi và cấu trúc. Các mẫu hành vi được xây dựng dựa trên năm mẫu cơ bản của UML, bao gồm: tuần tự, rẽ nhánh, phân nhánh song song, hợp nhất và đồng bộ. Mỗi mẫu xác định một cơ chế tích hợp hành vi vào mô hình lớp hợp nhất thông qua các lớp hoạt động, các lớp điều khiển và các liên kết tương ứng, đồng thời đảm bảo sự đồng bộ giữa trạng thái thực thi và trạng thái miền.

Hình 3.1 minh họa mẫu hành vi miền dạng quyết định, trong đó cấu trúc rẽ nhánh của biểu đồ hoạt động UML được ánh xạ trực tiếp vào mô hình lớp hợp nhất. Lớp hoạt động đóng vai trò điều phối, trong khi lớp quyết định hiện thực hóa lô-gic rẽ nhánh và các lớp dữ liệu tương ứng biểu diễn các nhánh kết quả. Ở mức thực thi, hành vi được đặc tả bằng AGL thông qua các nút hành động (ANode), mỗi nút tương ứng với một bước xử lý trong kịch bản nghiệp vụ. Các nhánh quyết định được ánh xạ thành các kịch bản hành vi khác nhau, cho phép mô hình miền hợp nhất phản ánh đầy đủ lô-gic điều khiển và có khả năng thực thi.

Việc áp dụng các mẫu hành vi miền được thực hiện thông qua ba bước chính: (i) biểu diễn biểu đồ hoạt động và mô hình lớp mẫu dưới dạng đồ thị; (ii) xác định ánh xạ giữa các nút hành động và các lớp miền; và (iii) đặc tả hành vi mô-đun tương ứng dưới dạng các chuỗi hành động. Cơ chế này đảm bảo sự liên kết chặt chẽ giữa hành vi và cấu trúc, đồng thời duy trì tính nhất quán ngữ nghĩa trong mô hình miền hợp nhất. Kết quả là một mô hình miền hợp nhất trong đó hành vi được tích hợp trực tiếp vào các lớp miền thông qua các đặc tả AGL, tạo nền tảng cho việc sinh tự động phần mềm theo tiếp cận DDD.



Hình 3.1: Đặc tả mẫu hành vi miền cho mẫu quyết định.

3.2.4 Ngôn ngữ hành vi miền dựa trên mô-đun

Ngôn ngữ AGL được đề xuất như một aDSL nhằm đặc tả hành vi miền trực tiếp trên mô hình miền. AGL cho phép biểu diễn các đồ thị hoạt động thông qua các chú thích gắn với các lớp miền, từ đó tích hợp hành vi vào mô hình một cách liền mạch. Ngôn ngữ này được xây dựng dựa trên một siêu mô hình cú pháp trừu tượng và một cú pháp cụ thể dạng chú thích, cho phép nhúng trực tiếp vào các ngôn ngữ lập trình hướng đối tượng. Cách tiếp cận này giúp đơn giản hóa việc đặc tả hành vi và hỗ trợ sinh tự động phần mềm từ mô hình miền hợp nhất.

3.3 Tổng kết chương

Chương 3 đã đề xuất hai kỹ thuật biểu diễn nhằm xây dựng mô hình miền đầy đủ thông tin cho DDD: (i) AGL để tích hợp hành vi miền vào mô hình miền hợp nhất thông qua lớp hoạt động, hành động mô-đun và các mẫu hành vi miền dựa trên UML Activity; và (ii) CAP để tích hợp các ràng buộc OCL phức tạp vào mô hình miền bằng các mẫu chú thích tham số hóa, bảo toàn ngữ nghĩa và hỗ trợ tái sử dụng.

Chương 4

PHƯƠNG PHÁP TÍCH HỢP CÁC MỐI QUAN TÂM VÀO MÔ HÌNH MIỀN

4.1 Phương pháp biểu diễn mô hình miền hợp nhất dựa vào siêu mô hình

Mục này trình bày một khung ngữ nghĩa hợp nhất cho mô hình miền, cho phép tích hợp có hệ thống các mối quan tâm về cấu trúc, hành vi và bảo mật. Cách tiếp cận dựa trên siêu mô hình nhằm đảm bảo tính mô-đun, khả năng mở rộng và hỗ trợ diễn giải ngữ nghĩa hình thức, từ đó làm nền tảng cho kiểm chứng và sinh tự động phần mềm.

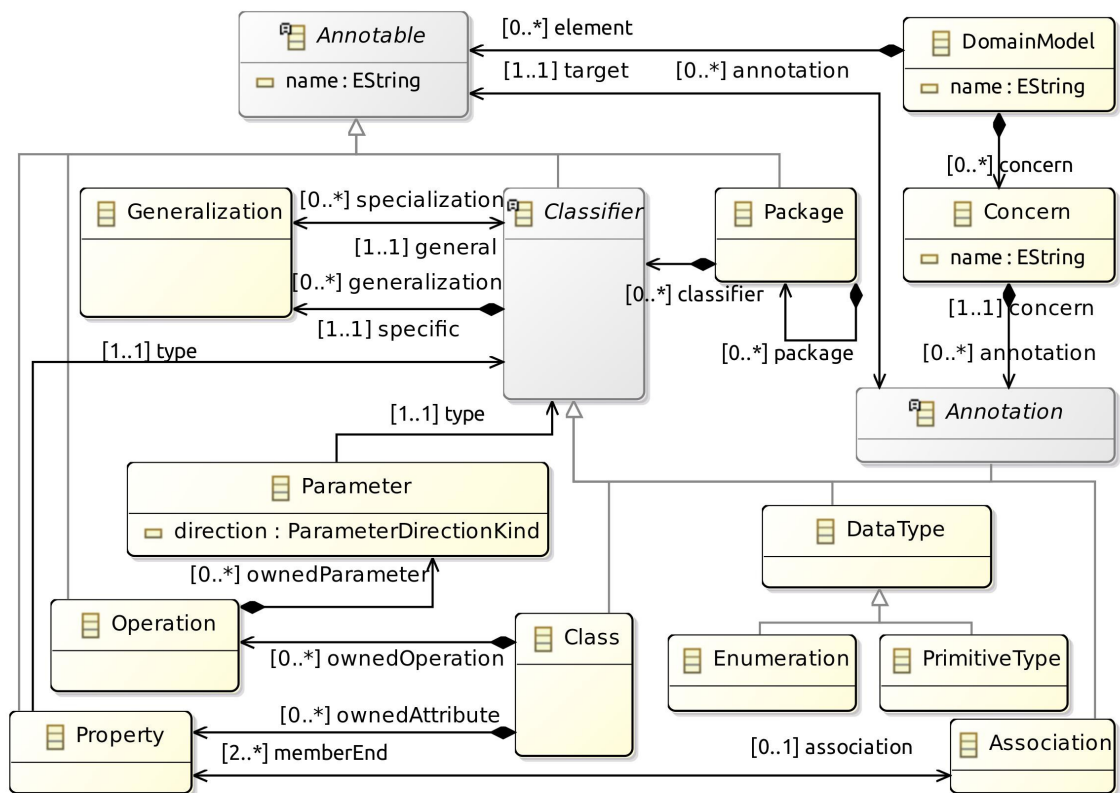
4.1.1 Tổng quan về phương pháp đề xuất

Phương pháp đề xuất mang tên mô hình hóa miền hợp nhất (UDMM) và được tổ chức thành ba bước chính: (i) siêu mô hình hóa và hợp thành các mối quan tâm thông qua các DSL chuyên biệt và cơ chế chú thích trong UDML; (ii) định nghĩa ngữ nghĩa hình thức bằng phép chuyển đổi từ UDML sang Event-B; và (iii) kiểm chứng hình thức bằng công cụ Rodin.

Cách tiếp cận này cho phép tích hợp các DSL theo mối quan tâm (DCSL, AGL, RBACDom) vào một mô hình miền hợp nhất có khả năng thực thi, đồng thời đảm bảo tính đúng đắn thông qua các nghĩa vụ chứng minh. Luận án đã định nghĩa siêu mô hình của UDML lõi dựa trên các siêu khái niệm của UML, như được minh họa trong Hình 4.1. Bốn siêu khái niệm mới—DomainModel, Concern, Annotation và Annotable—được bổ sung nhằm hỗ trợ việc hợp thành các mối quan tâm với DM và hợp nhất DSL theo mối quan tâm chuyên biệt trong UDML.

4.1.2 Biểu diễn mô hình miền hợp nhất

UDML cung cấp một siêu mô hình rút gọn làm lớp hợp thành trung gian, cho phép liên kết các mô hình miền chuyên biệt theo mối quan tâm vào một ngữ cảnh thống nhất. Mỗi DSL giữ tính độc lập về cú pháp và ngữ nghĩa, trong khi việc tích hợp được thực hiện thông qua các ánh xạ tường minh dựa trên chú thích.



Hình 4.1: Siêu mô hình UDML lõi cho mô hình miền hợp nhất.

Cách tiếp cận này làm rõ tương tác giữa các mối quan tâm và tạo điều kiện cho việc phân tích ngữ nghĩa và kiểm chứng hình thức trên mô hình hợp nhất.

Tích hợp các ràng buộc cấu trúc

DCSL được sử dụng để đặc tả các ràng buộc cấu trúc thông qua cơ chế chú thích gắn với các phần tử UDML. Các khái niệm như `DCClass`, `DAttr`, `DAssoc` và `DOpt` được ánh xạ trực tiếp tới các phần tử UML tương ứng.

Ngữ nghĩa của DCSL được định nghĩa thông qua: (i) ánh xạ hình thức từ các chú thích đến phần tử UDML; và (ii) tập các ràng buộc tính đúng đắn cấu trúc phải được bảo toàn trong mọi trạng thái thực thi. Cách tiếp cận này đảm bảo rằng các ràng buộc cấu trúc được tích hợp chặt chẽ vào mô hình miền mà không phá vỡ cấu trúc lõi.

Tích hợp hành vi miền

AGL được sử dụng để mô hình hóa hành vi miền dưới dạng đồ thị hoạt động có khả năng thực thi. Mỗi hành vi được biểu diễn bằng một `ActivityGraph`, trong đó các nút tương ứng với các hành động nguyên tử và các cạnh xác định luồng điều khiển.

Ngữ nghĩa hình thức của AGL được xây dựng dựa trên hệ chuyển trạng thái, trong đó một phép thực thi là chuỗi các nút hợp lệ thỏa mãn các điều kiện tiên/hậu trạng thái. Việc gắn các nút với lớp miền cho phép liên kết chặt chẽ giữa hành vi và cấu trúc. Đặc biệt, các nút có thể gắn chú thích, tạo điểm tích hợp trực tiếp với các mối quan tâm khác như bảo mật.

Tích hợp mối quan tâm bảo mật

`RBACDom` được giới thiệu như một DSL chuyên biệt để đặc tả kiểm soát truy cập theo RBAC. Các chính sách bảo mật được gắn vào các nút hành vi thông qua các chú thích `RbacDomNode`, cho phép ràng buộc việc thực thi hành vi theo ngữ cảnh ủy quyền. `RBACDom` hỗ trợ: đặc tả vai trò, quyền và phiên; các chiến lược kết hợp luật (`denyOverrides`, `permitOverrides`, `firstApplicable`); các ràng buộc phân tách nhiệm vụ (`SoD`).

Ngữ nghĩa của `RBACDom` được định nghĩa như một điều kiện kích hoạt bổ sung trên các nút hành vi, đảm bảo rằng chỉ các chuyển trạng thái hợp lệ về bảo mật mới được thực thi.

Ánh xạ giữa AGL và RBACDom

AGL và `RBACDom` được tích hợp thông qua ánh xạ ở mức nút, trong đó `RBACDom` không thay đổi cấu trúc hành vi mà chỉ giới hạn khả năng kích hoạt các nút.

Definition 2 (Tương ứng nút `RBACDom`). *Giả sử $G = (N, E)$ là một đồ thị hoạt động AGL với $N \subseteq ND$. Việc tích hợp `RBACDom` ở mức nút vào G được định nghĩa thông qua ánh xạ toàn phần $targetNode : RbacDomNode \rightarrow ND$ ánh xạ mỗi $rna \in RbacDomNode$ tới một nút đích duy nhất. Một nút $n \in N$ được kích hoạt trong một ngữ cảnh thực thi khi và chỉ khi tất cả các ràng buộc RBAC và ràng buộc phân tách nhiệm vụ được đặc tả bởi mọi rna sao cho $targetNode(rna) = n$ đều được thỏa mãn. \square*

Ngữ nghĩa hợp nhất được xác định bằng cách kết hợp: luồng điều khiển từ AGL; các điều kiện bảo mật từ RBACDom. Điều này đảm bảo tính mô-đun đồng thời duy trì ngữ nghĩa thực thi nhất quán.

4.2 Phương pháp biểu diễn mô hình miền hợp nhất dựa vào cây cú pháp

Phần này đề xuất một cách tiếp cận thay thế dựa trên cây cú pháp trừu tượng (AST) để tích hợp các DSL theo mỗi quan tâm. Khác với cách tiếp cận siêu mô hình, phương pháp này tích hợp trực tiếp ở mức cú pháp trừu tượng, hướng tới việc xây dựng mô hình miền có khả năng thực thi.

4.2.1 Tổng quan về phương pháp đề xuất

Phương pháp được tổ chức thành quy trình lặp bốn bước, bao gồm: (i) thiết kế DSL cho từng mỗi quan tâm; (ii) tích hợp DSL vào UDML; (iii) sinh phần mềm; (iv) đánh giá và cải tiến. Quy trình này tách biệt rõ giai đoạn thiết kế ngôn ngữ và giai đoạn ứng dụng, hỗ trợ phát triển lặp và thích ứng với thay đổi yêu cầu.

4.2.2 Biểu diễn và tích hợp các mối quan tâm

Phần này định nghĩa cú pháp và ngữ nghĩa của UDML dưới dạng một ngôn ngữ hợp nhất được xây dựng từ lõi UDML và tập các DSL theo mỗi quan tâm.

Cây cú pháp trừu tượng của UDML

UDML được hình thức hóa dưới dạng một cây cú pháp trừu tượng (AST), trong đó: - các nút biểu diễn các khái niệm miền; - các cạnh biểu diễn quan hệ cấu trúc và tham chiếu; - các DSL mở rộng AST một cách tăng dần.

Cú pháp cụ thể được định nghĩa như một phép chiếu từ AST sang các ký hiệu trực quan hoặc văn bản, trong khi ngữ nghĩa được xác định bằng cách ánh xạ các nút và cạnh sang miền ngữ nghĩa hợp nhất.

Cơ chế tích hợp các DSL theo mỗi quan tâm vào UDML

Thuật toán hợp nhất DSL thực hiện tích hợp tăng dần các DSL vào UDML thông qua: mở rộng cú pháp trừu tượng; hợp nhất cú pháp cụ thể; tích hợp ngữ nghĩa theo mô-đun; kiểm tra nhất quán xuyên mỗi quan tâm.

Kết quả là một mô hình miền hợp nhất bảo toàn tính mô-đun, hỗ trợ truy vết và có khả năng thực thi.

4.3 Ngữ nghĩa của mô hình hợp nhất

4.3.1 Định nghĩa hình thức các mô hình UDML.

Định nghĩa hình thức các mô hình UDML như các mô hình miền có khả năng thực thi hợp nhất, thu được bằng cách kết hợp nhiều mối quan tâm trực giao nhưng có liên hệ ngữ nghĩa với nhau.

Definition 3 (Mô hình UDML). *Một mô hình UDML được định nghĩa là một bộ: $\mathcal{M} = \langle \mathcal{U}, \mathcal{S}, \mathcal{B}, \mathcal{R}, \mathcal{C} \rangle$ trong đó: \mathcal{U} là siêu mô hình kết hợp lõi của UDML, cung cấp các trừu tượng cho mô-đun hóa mỗi quan tâm và liên kết chú thích, đồng thời độc lập với ngữ nghĩa thao tác; \mathcal{S} là mô hình mối quan tâm cấu trúc DCSL, bao gồm các chú thích và các ràng buộc của DCSL được gắn với các phần tử cấu trúc của mô hình miền; \mathcal{B} là mô hình mối quan tâm hành vi AGL, bao gồm các đồ thị hoạt động mà các nút và cạnh của chúng đặc trưng cho sự tiến hóa hành vi cho phép và các ranh giới thực thi cho các hành động miền; \mathcal{R} là mô hình mối quan tâm bảo mật RBACDom, đặc tả các ràng buộc ủy quyền và SoD nhằm hạn chế khả năng thực thi của các phần tử hành vi (đặc biệt là việc thực thi nút); \mathcal{C} là tập các ràng buộc toàn cục, bao gồm các điều kiện hợp lệ và các ràng buộc nhất quán xuyên mối quan tâm phải được bảo toàn bởi mọi phép thực thi mô hình. \square*

Ngữ nghĩa của mô hình UDML được định nghĩa dựa trên khái niệm hình chụp trạng thái, trong đó mỗi trạng thái là sự kết hợp giữa trạng thái lõi và trạng thái của các DSL theo mối quan tâm. Ngữ nghĩa thực thi được hình thức hóa như một hệ chuyển trạng thái, trong đó: AGL xác định không gian chuyển trạng thái; RBACDom ràng buộc khả năng thực thi; DCSL và các ràng buộc toàn cục đảm bảo tính hợp lệ của trạng thái.

Một kết quả quan trọng là tính tương thích ngữ nghĩa giữa UDML và AGL, đảm bảo rằng các phép thực thi được bảo toàn khi tích hợp các mối quan tâm.

Definition 4 (Ánh xạ các hình chụp). *Cho \mathcal{M} là một mô hình UDML và Σ_{AGL} là tập các hình chụp do đặc tả AGL của nó sinh ra. Một hình chụp của AGL $q \in \Sigma_{AGL}$ tương ứng với một hình chụp của UDML $\sigma \in \Sigma$, ký hiệu $q \sim \sigma$, khi và chỉ khi: $q = \pi_{AGL}(\sigma)$, trong đó $\pi_{AGL} : \Sigma \rightarrow \Sigma_{AGL}$ là phép*

chiếu trích xuất thành phần hành vi AGL từ $S_{DSL_c}(\sigma)$. Ánh xạ này cho phép liên kết ngữ nghĩa giữa trạng thái hành vi của AGL và trạng thái hợp nhất của UDML. \square

4.3.2 Ánh xạ định nghĩa ngữ nghĩa sang Event-B

Không gian trạng thái của UDML được đặc trưng bởi tập các hình chụp Σ , trong đó mỗi trạng thái của máy Event-B tương ứng với một hình chụp hợp lệ của mô hình UDML. Thông qua phép chiếu π_{AGL} , mỗi hình chụp $\sigma \in \Sigma$ xác định một hình chụp hành vi $q \in \Sigma_{AGL}$, qua đó bảo toàn ngữ nghĩa thực thi của các phép chuyển trạng thái được đặc tả trong AGL. Ngược lại, mỗi sự kiện Event-B biểu diễn một chuyển trạng thái khả dĩ trong quan hệ \rightarrow , đồng thời phản ánh sự tiến hóa của thành phần hành vi được ánh xạ từ AGL. Mô hình UDML được chuyển đổi sang Event-B nhằm hỗ trợ mô phỏng và kiểm chứng hình thức. Phép ánh xạ này bảo toàn cả cấu trúc và hành vi của mô hình.

Thuật toán ánh xạ từ UDML sang Event-B (UDML2Event-B) thực hiện chuyển đổi theo ba pha: cấu trúc, hành vi và bảo mật, sau đó tích hợp các mối quan tâm thông qua việc tăng cường điều kiện bảo vệ của các sự kiện. Các nghĩa vụ chứng minh được sinh ra và giải bằng Rodin nhằm xác lập tính đúng đắn của mô hình.

4.4 Tổng kết chương

Chương đã đề xuất hai cách tiếp cận tích hợp mối quan tâm vào mô hình miền: dựa trên siêu mô hình và dựa trên AST. Cả hai đều cho phép hợp nhất các DSL cấu trúc, hành vi và bảo mật vào một mô hình miền thống nhất có khả năng thực thi và kiểm chứng hình thức.

Đặc biệt, việc tích hợp RBACDom và ánh xạ sang Event-B cung cấp một cơ chế kiểm chứng mạnh mẽ, đảm bảo tính nhất quán và an toàn của hệ thống ngay từ giai đoạn thiết kế.

Chương 5

PHƯƠNG PHÁP SINH TỰ ĐỘNG BẢN MẪU PHẦN MỀM

5.1 Tổng quan phương pháp

Phần này trình bày phương pháp sinh tự động bản mẫu phần mềm từ mô hình yêu cầu thông qua mô hình miền hợp nhất (RM2UDM) và các kỹ thuật chuyển đổi mô hình có khả năng thực thi (AD2AGL/M2T). Phương pháp gồm ba giai đoạn chính: (i) xây dựng mô hình miền hợp nhất từ UML/OCL, (ii) chuyển đổi sang mô hình miền thực thi và sinh mã nguồn, và (iii) sinh bản mẫu phần mềm kết hợp quy trình lập dựa trên phản hồi của chuyên gia miền.

5.2 Phương pháp sinh tự động bản mẫu phần mềm từ mô hình yêu cầu

Phần này mô tả kỹ thuật xây dựng mô hình hợp nhất UDM từ mô hình yêu cầu RM và bộ chuyển đổi RM2UDM, bao gồm (i) nguyên tắc xây dựng UDM và (ii) các luật chuyển đổi cùng cơ chế cài đặt.

5.2.1 Xây dựng mô hình UDM từ mô hình yêu cầu

UDM gồm hai thành phần: (1) khía cạnh cấu trúc tương ứng với DCSL; và (2) khía cạnh hành vi tương ứng với AGL.

5.2.2 Xây dựng bộ chuyển đổi mô hình RM2UDM

Bộ chuyển đổi RM2UDM được cài đặt theo hướng chuyển đổi mô hình sang mô hình (M2M) bằng ATL nhằm chuyển mô hình yêu cầu (RM) sang mô hình miền hợp nhất (UDM). Mô hình kết quả tuân thủ siêu mô hình UDML và có thể tiếp tục được ánh xạ tự động sang mã nguồn (ví dụ Java) để cài đặt bản mẫu phần mềm. Phương pháp thực hiện ánh xạ hoạt động sang ActivityClass/ActivityGraph; ánh xạ các nút và cạnh của biểu đồ hoạt động sang Node/Edge kèm tham chiếu lớp miền; ánh xạ các nút điều khiển sang các DomainPattern; liên kết hành vi với lớp miền thông qua

actSeq để đồng bộ trạng thái thực thi; ánh xạ lớp, thuộc tính và ràng buộc của biểu đồ lớp sang DCSL; và ánh xạ lớp miền sang cấu hình mô-đun.

5.3 Chuyển đổi từ đặc tả yêu cầu sang mô hình miền hợp nhất

5.3.1 Bộ chuyển đổi AD2AGL

Luận án đề xuất thuật toán *AD2AGL* để sinh đặc tả *AGL⁺* từ *AD* và *CD*. Thuật toán duyệt theo chiều sâu trên đồ thị hoạt động, nhận diện loại nút kế tiếp (Decision/Sequential/Fork/Merge/Join) và gọi các hàm sinh tương ứng để bổ sung đặc tả *AGL*; sau đó duyệt *CD* để sinh phần *DCSL*.

5.3.2 Sinh đặc tả mô hình miền thực thi (*AGL⁺*)

Luận án sử dụng kỹ thuật chuyển đổi mô hình–sang–văn bản (M2T) để sinh đặc tả *AGL⁺* từ các mô hình mức cao. Công cụ Acceleo được lựa chọn để hiện thực bộ sinh mã thông qua các mẫu (template) dựa trên siêu mô hình, cho phép ánh xạ trực tiếp các phần tử mô hình sang mã nguồn.

Các luật chuyển đổi được đặc tả dưới dạng template trong Acceleo, hỗ trợ chuyển đổi biểu đồ hoạt động (*AD*) và biểu đồ lớp (*CD*) sang đặc tả *AGL⁺*, qua đó tạo mô hình miền có khả năng thực thi. Cụ thể, thuật toán duyệt các nút trong *AD*, sử dụng `oclIsKindOf()` để xác định loại nút và gọi các hàm sinh mã tương ứng, đồng thời cập nhật mô hình miền *DM*.

5.4 Tổng kết chương

Chương này đề xuất bộ chuyển đổi từ mô hình yêu cầu UML sang mô hình miền hợp nhất có khả năng thực thi trong thiết kế hướng miền. Thông qua *AGL⁺* và cơ chế *AD2AGL*, hành vi được tích hợp vào mô hình miền, trong khi phép chuyển đổi *RM*→*UDM* hỗ trợ sinh tự động bản mẫu phần mềm. Cách tiếp cận này thiết lập liên kết chặt chẽ giữa yêu cầu và hiện thực, hình thành một quy trình chuyển đổi hoàn chỉnh, nhất quán và có khả năng áp dụng cho các miền ứng dụng chuyên biệt.

Chương 6

THỰC NGHIỆM VÀ ĐÁNH GIÁ

6.1 Giới thiệu

Chương này sử dụng CourseMan làm ví dụ thúc đẩy để minh họa cách tích hợp hành vi (AGL) và ràng buộc (CAP) vào mô hình miền UML/OCL nhằm tạo mô hình miền hợp nhất có thể thực thi theo DDD. Nội dung tập trung giới thiệu công cụ hỗ trợ và đánh giá các đề xuất theo bốn câu hỏi nghiên cứu RQ1–RQ4 về: tính biểu đạt, nỗ lực mô hình hóa, khả năng biểu đạt ràng buộc và mức độ áp dụng thực tiễn.

6.2 Công cụ hỗ trợ

Công cụ hỗ trợ kỹ thuật tích hợp khía cạnh hành vi vào mô hình miền:

Công cụ hiện thực các mẫu hành vi lõi của AGL (tuần tự, quyết định, phân nhánh, hợp nhất, gộp) trên CourseMan và OrderMan. Mỗi mẫu được ánh xạ thành mô hình hợp nhất và sinh GUI tương ứng, minh họa cơ chế điều khiển luồng, truyền đối tượng giữa các hành động và tổ chức giao diện phản ánh cấu trúc mô-đun. Việc hiện thực dựa trên JDA với đặc tả AGL nhúng bằng chú thích Java và được xử lý tại thời gian chạy.

Công cụ hỗ trợ kỹ thuật tích hợp ràng buộc vào mô hình miền:

Công cụ CAP/UDML được xây dựng trên JetBrains MPS, hỗ trợ định nghĩa và hợp nhất các mẫu CAP vào mô hình miền thông qua thao tác trực quan, đồng thời sinh mã Java cho khung JDA. Công cụ cho phép kiểm tra ràng buộc nghiệp vụ tại thời gian chạy (ví dụ giới hạn tín chỉ), minh họa khả năng tái sử dụng và thực thi trực tiếp các ràng buộc từ mô hình.

Công cụ hỗ trợ phương pháp tổng hợp các DSL theo mỗi quan tâm vào mô hình miền hợp nhất có thể thực thi: Công cụ UDMM hỗ trợ hai hướng tích hợp DSL theo mỗi quan tâm: (i) dựa trên siêu mô hình (EMF, Sirius, ATL/Accelio) và (ii) dựa trên AST (JetBrains MPS). Cả hai đều cho phép hợp nhất các DSL như DCSL, AGL, RBAC vào một mô hình

thống nhất và sinh mã thực thi trên JDA, được minh họa qua nghiên cứu tình huống **CourseMan**.

6.3 Thực nghiệm và đánh giá

Kỹ thuật biểu diễn mô hình miền tích ràng buộc: CAP được đánh giá về mức độ biểu đạt và khả năng áp dụng thực tế so với DCSL và các khung DDD phổ biến. Kết quả cho thấy CAP hỗ trợ hiệu quả phần lớn các ràng buộc nghiệp vụ dạng bất biến và tham số hóa, giảm mã kiểm tra thủ công. Các ràng buộc mang tính hành vi động được xác định nằm ngoài phạm vi hiện tại.

Kỹ thuật biểu diễn mô hình miền tích hợp hành vi: AGL tích hợp DCSL (AGL+) được đánh giá trên CourseMan, ProcessMan và OrderMan, và so sánh với các framework DDD dựa trên chú thích và biểu đồ hoạt động UML. Kết quả cho thấy AGL+ biểu đạt tốt cả cấu trúc và hành vi, duy trì gắn kết giữa luồng điều khiển và thao tác miền, đồng thời giảm đáng kể công sức hiện thực thủ công.

Kỹ thuật tích hợp các DSL theo mỗi quan tâm vào mô hình miền Phương pháp tích hợp dựa trên UDML được đánh giá qua các ca nghiên cứu, cho thấy khả năng hợp nhất các DSL theo mỗi quan tâm vào một mô hình miền thống nhất mà không làm biến dạng cấu trúc lõi. Cơ chế kết hợp DSL ngoại sinh và aDSL đảm bảo nhất quán giữa mô hình và hiện thực, trong khi hợp nhất dựa trên AST hỗ trợ tích hợp linh hoạt và đồng bộ đa góc nhìn.

Kỹ thuật sinh tự động bản mẫu phần mềm: Chuỗi chuyển đổi RM2UDM và sinh mã Java được đánh giá trên CourseMan và OrderMan. Kết quả cho thấy mô hình hợp nhất có thể sinh phần lớn mã nguồn tự động; các chỉnh sửa thủ công còn lại chủ yếu do giới hạn của UML/OCL đầu vào và phạm vi luật chuyển đổi.

6.4 Tổng kết chương

Chương này chứng minh tính khả thi và giá trị thực tiễn của các kỹ thuật AGL, CAP và UDML thông qua công cụ hỗ trợ, nghiên cứu tình huống và đánh giá thực nghiệm, làm rõ khả năng xây dựng ngôn ngữ mô hình miền hợp nhất có thể thực thi theo DDD.

Chương 7

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kỹ thuật biểu diễn và chuyển đổi mô hình cho thiết kế hướng miền. Mặc dù trong thực tế đã có một số phương pháp đề xuất tập trung vào biểu diễn mô hình miền có thể thực thi. Kỹ thuật biểu diễn mô hình miền được tích hợp cả khía cạnh hành vi và các ràng buộc OCL phức tạp vào mô hình miền hợp nhất. Trong UDML xem xét các khía cạnh đó như là một DSL theo mỗi quan tâm và luận án đã đề xuất kỹ thuật tích hợp các DSL theo mỗi quan tâm thành một mô hình miền hợp nhất có thể thực thi, ngữ nghĩa được kiểm chứng hình thức. Để thu được mô hình miền có thể thực thi, luận án đã trình bày kỹ thuật chuyển đổi mô hình từ các đặc tả bậc cao đến mô hình hóa các khía cạnh là DSL theo mỗi quan tâm sang mã nguồn (nhúng trực tiếp vào ngôn ngữ lập trình máy chủ).

7.1 Các đóng góp của luận án

Sau một thời gian nghiên cứu và giải quyết bài toán này, đã nâng cao đáng kể trạng thái hiện tại của DDD bằng cách thu hẹp khoảng cách giữa mô hình và mã nguồn, đồng thời xây dựng một mô hình miền thống nhất, luận án đã có một số những đóng góp nhất định như sau.

1. Đóng góp thứ nhất là đề xuất kỹ thuật tích hợp khía cạnh hành vi, ràng buộc phức tạp vào mô hình miền hướng đến sinh tự động phần mềm theo tiếp cận thiết kế hướng miền.
2. Đóng góp thứ hai là đề xuất phương pháp tích hợp các mối quan tâm vào mô hình miền có thể thực thi theo thiết kế hướng miền bằng cách xây dựng một ngôn ngữ hợp nhất mô hình miền UDML đầy đủ cú pháp, ngữ nghĩa bao gồm theo tiếp cận siêu mô hình và phương pháp cây cú pháp trừu tượng. Ngữ nghĩa thực thi của UDML được kiểm chứng hình thức bằng Event-B.
3. Đóng góp thứ ba là đề xuất một chuỗi chuyển đổi mô hình để sinh tự động bản mẫu phần mềm.

Các kết quả của luận án đã được công bố trong các công trình khoa học đăng tải tại các kỷ yếu hội thảo quốc tế.

7.2 Hướng phát triển

Trong mục này, luận án thảo luận về một số hạn chế và những đề xuất cải tiến nghiên cứu được xem xét trong tương lai.

- Hoàn thiện công cụ UDML dưới dạng plugin, đồng thời xây dựng các cú pháp biểu diễn đồ họa cho đặc tả hành vi, bảo mật và các mẫu chú thích ràng buộc (CAP), qua đó nâng cao khả năng sử dụng và hỗ trợ người dùng.
- Tiếp tục hoàn thiện kỹ thuật biểu diễn mô hình miền bằng cách mở rộng CAP cho các miền ứng dụng thực tế, thông qua việc xây dựng thư viện các mẫu ràng buộc hỗ trợ biểu thức OCL phức tạp, góp phần nâng cao khả năng áp dụng, tích tích hợp, hiệu năng thực thi và khả năng bảo trì của mô hình miền.
- Mở rộng khung phương pháp cho các DSL chuyên biệt theo từng mối quan tâm, đồng thời hỗ trợ các cơ chế phân quyền động và phụ thuộc ngữ cảnh đa dạng hơn.
- Tích hợp các kỹ thuật AI/LLM vào quy trình mô hình hóa miền nhằm hỗ trợ giai đoạn khởi tạo mô hình, đặc biệt trong việc sinh nháp đặc tả và gợi ý các thành phần từ mô tả nghiệp vụ; các đặc tả này cần được chuẩn hóa trong các DSL và kiểm chứng bằng các cơ chế hình thức để đảm bảo tính đúng đắn và nhất quán.

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN TỚI LUẬN ÁN

1. [Vinh1] 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. [Vinh2] Van-Vinh Le, Nghia-Trong Be, Duc-Hanh Dang. On Automatic Generation of Executable Domain Models for Domain-Driven Design. *Proceedings of the 15th International Conference on Knowledge and Systems Engineering (KSE)*, IEEE, 2023. DOI: 10.1109/KSE59128.2023.10299453. (WoS/Scopus).
3. [Vinh3] Van-Vinh Le, Duc-Hanh Dang. An Approach to Composing Concerns for an Executable Unified Domain Model. *Proceedings of the 18th International Conference on Research, Innovation and Vision for the Future (RIVF)*, IEEE, 2024, pp. 415–419. DOI: 10.1109/RIVF64335.2024.11009109. (WoS/Scopus).
4. [Vinh4] Le Van Vinh, Dang Duc Hanh. RM2UDM: A method for automatically generating functional prototypes from requirement models. *Proceedings of the 17th National Conference on Fundamental and Applied Information Technology Research (FAIR)*, pp. 734–741, 2024. ISBN: 978-604-357-304-6, DOI: 10.15625/vap.2024.0271.
5. [Vinh5] 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. *Proceedings of the 14th International Symposium on Information and Communication Technology (SOICT)*, Springer, 2025. (WoS/Scopus, accepted).
6. [Vinh6] Van-Vinh Le, Duc-Hanh Dang. Constraint Annotation Patterns for Prototype Generation within Domain-Driven Design. *e-Infomatica Software Engineering Journal*, 2025. ISSN: 2084-4840. (WoS/Scopus, Q3, submitted).
7. [Vinh7] 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.*

Danh mục này gồm 07 công trình.