

[Home](#) [Article](#) [Analisis Pembentukan Class Diagram ...](#)

Analisis Pembentukan Class Diagram dengan menggunakan metode Domain Modelling

Analisis Pembentukan Class Diagram dengan menggunakan metode Domain Modelling

Introduction

Class diagram merupakan salah satu diagram utama dari Unified Modelling Language (UML) untuk menggambarkan class atau blueprint object pada sebuah sistem. Pada class diagram juga digambarkan bagaimana interaksi hubungan antar class dalam sebuah konstruksi piranti lunak seperti hubungan asosiasi, agregasi, komposisi, dan inheritance. Standarisasi pemakaian class diagram yang ter up to date pada diagram UML 2.0.

Dalam penggambaran class diagram, biasanya kita sebagai developer terkadang tidak tahu harus mulai menggambarkan class tersebut dari mana. Analisis pembentukan class diagram merupakan aktivitas inti yang sangat mempengaruhi arsitektur piranti lunak yang dirancang hingga ke tahap pengkodean. Bila kita salah dalam menganalisa class diagram dan tidak sesuai dengan problem-domain atau area permasalahan yang ingin kita buat solusinya, maka akan berakibat saat pemeliharaan atau maintenance kode sumber menjadi lebih sulit dan bisa juga berdampak pada performa piranti lunak yang dibuat. Desain class diagram yang tidak baik juga mengakibatkan susahnya pengembangan piranti lunak dikarenakan arsitektur kode yang kurang bagus dan copy-paste kode sumber yang sama dalam satu arsitektur

Domain Modelling

Domain Modelling merupakan teknik pengidentifikasian object-object pada kata benda yang terdapat pada daftar requirement yang diklasifikasikan pada area (domain) permasalahan yang sama untuk dijadikan candidate class pada class diagram. Analisis yang harus pertama kali dilakukan adalah analisis domain model daripada analisis use case diagram bersifat abstract dan ambigu untuk dianalisa dan pada akhirnya use case diagram harus dibuat secara konkrit pada konteks model object. Bentuk domain model merupakan fondasi dari bagian statis dari sebuah model sistem, sedangkan use case merupakan bagian dinamis dari sebuah model sistem. Domain model mendeskripsikan struktur arsitektur dari sebuah sistem yang statis, sedangkan use case diagram mendeskripsikan fungsi atau tingkah laku dari sebuah sistem.

Titik awal untuk memulai domain modeling adalah dari requirement atau kebutuhan sistem dari user/client. Pengumpulan daftar kebutuhan requirement yang baik akan menghasilkan domain model yang baik pula. Berikut adalah panduan bagaimana mendapatkan requirement yang baik dari user :

- Gunakan diagram model yang dapat menggambarkan requirement dan korelasi atau hubungan dengan requirement lainnya, contohnya gunakan diagram Mind Map untuk menggambarkan requirement. Penggunaan mind map dapat meningkatkan penelusuran requirement yang lebih mendetil
- Hubungkan requirement kepada use-case diagram untuk mengetahui sebuah requirement akan diimplementasikan pada fitur use-case yang mana
- Hindari penggunaan statement requirement yang disfungsional yang tidak menggambarkan fungsi dari sebuah sistem
- Hindari penggunaan bahasa teknis seperti void,int agar statement requirement dapat dengan mudah dibaca
- Tulis rencana skenario test case pada setiap requirement agar requirement yang dihasilkan dapat dilakukan pengujian
- Klasifikasikan requirement kedalam tipe yang berbeda-beda untuk membentuk domain permasalahan
- Perlakukan setiap requirement yang didapatkan sebagai hal yang sangat penting

Sedangkan untuk domain modeling terdapat panduan yang dapat diikuti:

- Analisa domain model harus berfokus kepada object dunia nyata yang terdapat pada sebuah domain atau area permasalahan.
- Pada saat penggambaran hubungan antar model, gunakan hubungan class diagram generalisasi sebagai abstraksi dari sebuah object dan agregasi untuk mengidentifikasikan suatu class merupakan komponen atau attribute dari class lain

SCHOOL OF COMPUTER SCIENCE

software dan aktivitas terakayasa software tidak hanya terpaku ke dalam pendefinisian domain model

- Selalu gunakan abstraksi seperti generalisasi pada class diagram pada pendefinisian domain model agar domain model lebih dinamis dalam menghadapi perubahan requirement
- Analisis domain model jangan dimulai dari analisis data model. Ini dikarenakan domain model merupakan candidate class diagram yang memiliki attribute (data) dan behavior (fungsi), sedangkan data model merupakan candidate dari sebuah tabel pada database
- Selalu mulai dari analisis domain model lalu diikuti dengan analisis use case
- Domain model nanti nya akan semakin berkembang dan detil pada class diagram, jadi bisa dipastikan bahwa domain model bukan merupakan hasil akhir dari class diagram.
- Untuk domain model awal, pastikan analisa yang dilakukan berfokus kepada domain (area) permasalahan. Jangan dicampur dengan class-class yang berhubungan dengan user interface

Case Study

Berikut case study mengenai pembuatan domain model dalam kasus internet bookstore yang diambil dari buku Use Case Driven Object Modelling with UML karya Rosenberg dan Stephens (2007). Pada buku ini dijelaskan bahwa pengidentifikasian domain model dari high-level requirement. Berikut high-level requirement yang digunakan untuk ekstraksi domain model atau domain class. Object-object yang diidentifikasi pada statement requirement terdapat pada kata benda dan frase kata benda yang dibold. Requirement ditulis dalam bahasa inggris seperti yang ada pada contoh buku:

1. The bookstore will be web based initially, but it must have a sufficiently flexible architecture that alternative front-ends may be developed (Swing/applets, web-service, etc)
2. The bookstore must be able to sell books, with orders accepted over the Internet
3. The user must be able to add books into an online shopping chart, prior to checkout.
 - a. Similarly, the user must be able to remove items from the shopping chart
4. The user must be able to maintain wish lists of books that he or she wants to purchase later
5. The user must be able to cancel orders before they have shipped
6. The user must be able to pay by credit card or purchase order
7. It must be possible for the user to return books
8. The bookstore must be embeddable into associate partners' website using mini catalog, which are derived from overall master catalog stored in a central database
 - a. The mini-catalog must be defined in XML, as they will be transferred between this and (later to be defined) systems

SCHOOL OF COMPUTER SCIENCE

-
- remembers the user's detail (name, address, credit card details) at login
- a. The system shall maintain a list of accounts in its central database
 - b. When a user logs in, his / her password must always be matched against the password in the master account list
10. The user must be able to search for books by various search method-title, author, keyword, or category-then view the books details
 11. It must be possible for the user to post review comments should appear on the book details screen. The review should include a customer rating, which usually along with the book title in book lists
 - a. Book reviews must be moderated-that is, checked and "OK'd" by a member of staff before they have published on the website
 - b. Longer review should be truncated on the book details screen; the customer may click to view the full review on a separate page
 12. It must be possible for staff to post editorial reviews of books. These should appear on the book detail screen.
 13. The bookstore shall allow third-party sellers (e.g., second hand bookstore) to add their own individual book catalogs, These are added into overall master book catalog so that seller's book are included in search result
 14. The bookstore must be scalable, with the following specific requirements:
 - a. The bookstore must be capable of maintaining user account for up to 100,000 customer in its first six months, then a further 1,000,000 after that
 - b. The bookstore must be capable of serving up to 1,000 simultaneous users (10,000 after six months)
 - c. The bookstore must be able to accommodate up to 100 search request per minute (1,000/minutes after six months)
 - d. The bookstore must be able to accommodate up to 100 purchases per hours (1,000/hours after six months)

Dari pengidentifikasian domain class di atas, berikut list candidate class diagram yang masih banyak kata benda yang berulang dan ambigu:

Associate Partner	Customer Account	Order
Author	Customer Rating	Password
Book	Database	Purchase Order
Book Catalog	Editorial Review	Review Comment
Book Details	Internet	Search Method
Book List	Item	Search Result
Book Review	Keyword	Seller
Bookstore	List of Accounts	Shipping Fulfillment System

SCHOOL OF COMPUTER SCIENCE

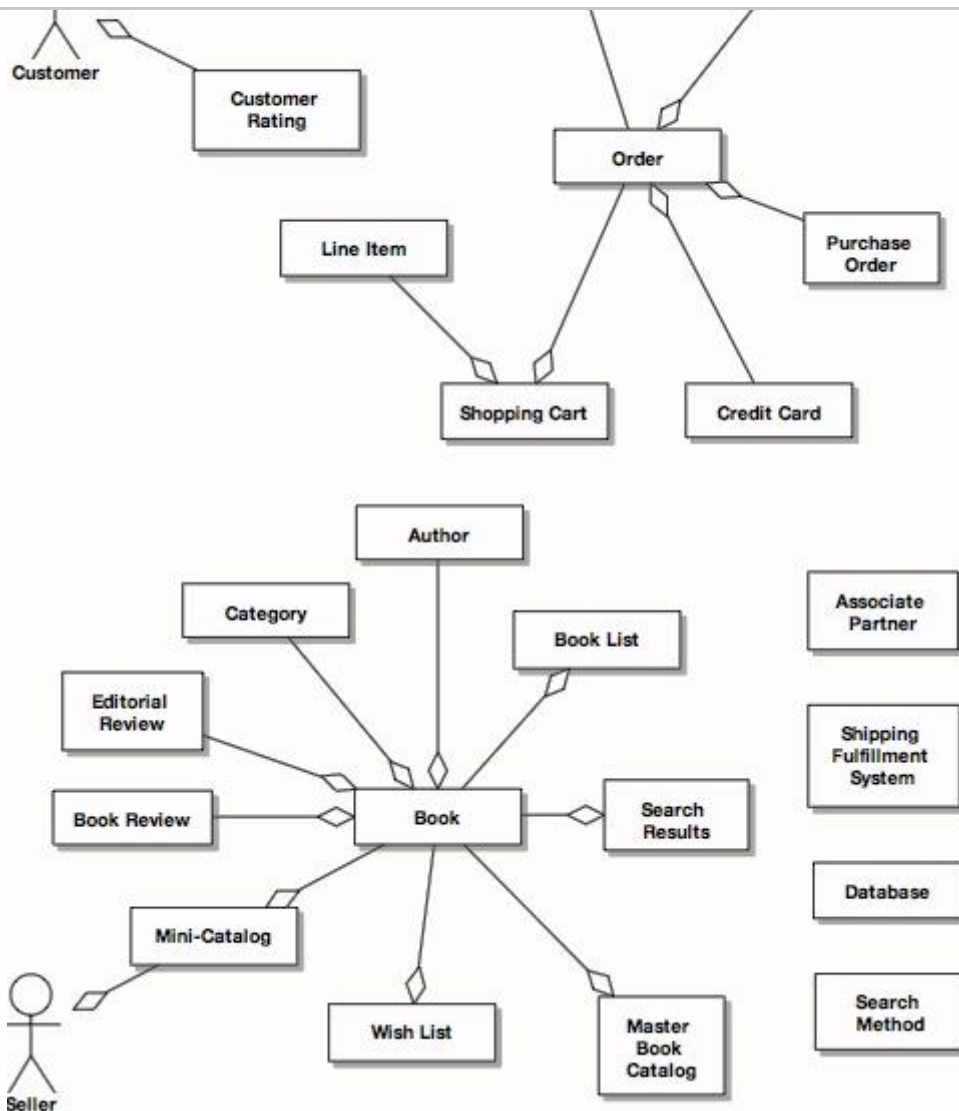
Credit Card	Master Catalog	User Account
Customer	Mini-Catalog	Wish List

Daftar domain class di atas masih ada kata-kata benda yang mirip dengan satu sama lain, dan ada juga kata benda yang dapat dijadikan atribut pada kata benda (object) yang lain. Berikut hasil penyaringan dari kata benda yang unik yang dapat dijadikan candidate class:

Associate Partner	Customer Account	Order
Author	Customer Rating	Purchase Order
Book	Database	Search Method
Book List	Editorial Review	Search Result
Book Review	Line Item	Shipping Fulfillment System
Category	Master Account List	Shopping Cart
Checkout	Master Book Catalog	Wish List
Credit Card	Mini Catalog	

Langkah berikutnya adalah penentuan hubungan agregasi antar tiap class untuk mengetahui apakah sebuah class merupakan komponen dari class lain. Hasil analisis untuk hubungan agregasi dalam domain class kasus ini sebagai berikut:

SCHOOL OF COMPUTER SCIENCE



First Domain Model for Internet Bookstore (Stephens & Rosenberg, 2007)

Akan ada domain class yang berdiri sendiri yang tidak berhubungan dengan class lainnya seperti Associate Partner, Shipping Fulfillment System, Database, dan Search Method. Artikel ini hanya membahas sampai langkah penentuan relasi domain class tipe agregasi. Langkah selanjutnya penentuan hubungan relasi asosiasi (bila ada) dan hubungan generalisasi untuk abstraksi class.

Kesimpulan

Domain Modelling merupakan salah satu metodologi dari software engineering yang digunakan untuk mengidentifikasi candidate class pada class diagram dan hubungan antar class. Metodologi ini tidak cukup untuk menggambarkan class diagram secara utuh, misalkan pengidentifikasian attribute object dan pengidentifikasian behavior object dengan menggunakan sequence diagram pada setiap bulatan item use case diagram.

References

RELATED CONTENT

PERANGKAT AJAR RAMBU-RAMBU LALU LINTAS

3D PRINTER

SHARE THIS

School of Computer Science
BINUS UNIVERSITY

Copyright © BINUS UNIVERSITY. All rights reserved.