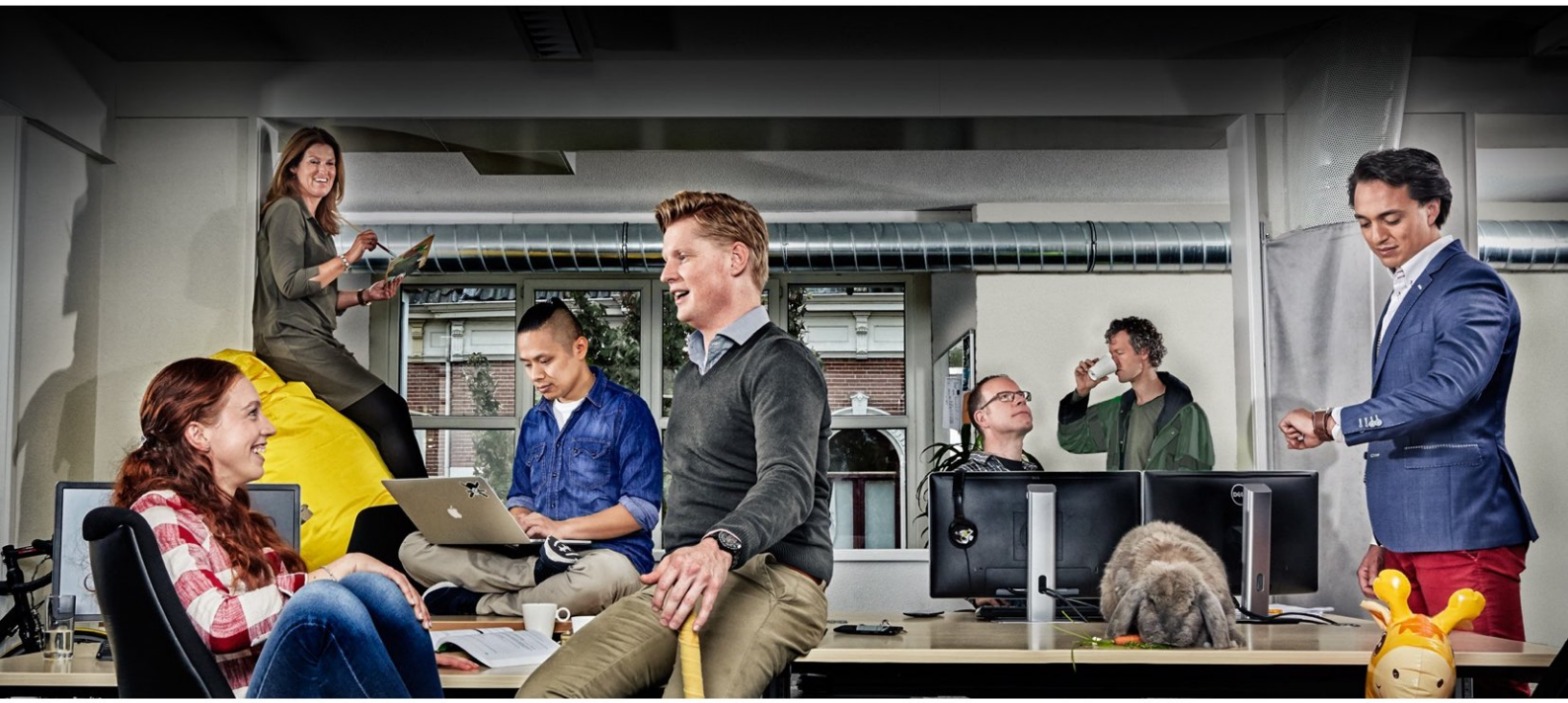
topicus
ACADEMY



TA-301

Object Oriented Analysis & Design

Studiegids

TA301 Object Oriented Analysis and Design

Topicus Academy

Inhoudsopgave

Inleiding

Object Oriented Analysis and Design levert de mogelijkheid om je probleem te analyseren en een oplossing te formuleren op een gestructureerde manier zodanig dat er een onderhoudbaar programma ontstaat.

Deze cursus werkt aan de hand van het boek [LAR] *Applying UML and Design Patterns* van *Craig Larman*. De 2e editie is al wat oud, maar kan gratis als PDF gedownload worden.

We gebruiken in deze cursus niet alles uit het boek en zullen er relatief vluchtig door de stof gaan.

Doelen

Na deze cursus kan je van een probleembeschrijving een model bouwen en op basis van dat model software gaan schrijven. Je leert de verschillende diagram technieken zoals Domain Model, System Sequence Diagram toe te passen bij het uitwerken van je probleem.

Inhoud

De opbouw van deze cursus is als volgt:

- leren te achterhalen wat de vereisten zijn voor de oplossing
- deze om te zetten in een vaste structuur
- hiervan een domeinmodel te maken
- de juiste plek voor de onderdelen in je domeinmodel te vinden
- je domeinmodel om te zetten naar een werkend programma

Als afsluiting zal je een opdracht uitwerken om van een probleemomschrijving naar een werkende applicatie te komen.

Bronnen

- [] Craig Larman, Applying UML and Design Patterns, 2nd Edition, ISBN 13: 9780130925695, https://www.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf
- [] Scott McEwen, Requirements: An introduction, IBM DeveloperWorks, https://www.ibm.com/developerworks/rational/library/4166.html
- [] Tim Ottinger en Jeff Langer, FURPS+, http://agileinaflash.blogspot.nl/2009/04/furps.html

[LAR], [McEwen] en [FURPS] zijn ook via de Topicus Academy website te downloaden als PDF onder het vak TA301 → downloads.

Chapter 1. Requirements analyse

Requirements zijn de vereisten waaruit je probleem bestaat waaraan invulling gegeven moet worden door jouw te ontwerpen systeem. Deze vereisten zijn onder te verdelen in een aantal categorieën, waarvan de meest bekende het onderscheid is tussen:

- functionele eisen (*functional requirements*), en
- niet-functionele eisen (*non-functional requirements*)

Daarnaast is in de OOAD wereld de lijst van vereisten onderverdeeld in het acroniem *FURPS*, en die is weer uitgebreid met nog meer elementen onder het acroniem *FURPS+*.

In de onderstaande teksten leer je meer over de verschillende soorten vereisten die van een systeem verlangd kunnen worden.

1.1. Lezen

Lees de volgende delen uit [LAR]:

- Hoofdstuk 1: *Object-Oriented Analysis and Design*
- Hoofdstuk 3: *Case Study: The NextGen POS System.*

  Het POS systeem staat voor "Point of Sale", ook bekend als kassasysteem. Aangezien het boek dit voorbeeld gebruikt is deze introductie handig om door te nemen.

- Hoofdstuk 5: *Understanding Requirements*
- Hoofdstuk 7: *Identifying other Requirements*

  Lees alleen 7.1 tot en met *7.5 Commentary: Vision*

Lees daarna het artikel [McEwen] *Requirements: An introduction.*

In de tekst van [LAR] wordt uitgegaan van een iteratief proces voor het ontwerpen en ontwikkelen van het systeem. In het boek wordt hiervoor het *Unified Process* gebruikt, wat uitgelegd staat in hoofdstuk 2 *Iterative Development and the Unified Process*. Aangezien we binnen Topicus en de meeste bedrijven geen gebruik meer maken van het *UP*, slaan we dit hoofdstuk over. Je kan het wel lezen voor achtergrond informatie, maar het is geen essentiële kennis voor OOAD.

1.2. Opdrachten

1. *Lees [FURPS+]. Wat zou jouw uitleg van FURPS+ zijn? Geef een vertaling van één zin van elk van de letters in FURPS+ in je eigen woorden.*

   F:

   ```
   ----------------------------------------------------------------
   ```

U:

```
---------------------------------------------------------------
```

R:

```
---------------------------------------------------------------
```

P:

```
---------------------------------------------------------------
```

S:

```
---------------------------------------------------------------
```

2. *Welke van de begrippen needs, features en requirements horen bij het solution domain?*
    1. needs, features en requirements
    2. needs en requirements
    3. features en requirements
    4. alleen requirements
3. *Wat wordt bedoeld met 'elicit needs from stakeholders'?*
    1. Het boven tafel krijgen van de needs van belanghebbenden.
    2. Het documenteren van de needs van belanghebbenden.
    3. Het verifiëren van de needs van belanghebbenden.
    4. Het vertalen van de needs van belanghebbenden naar software features.
4. *Wat leg je vast m.b.v. use cases?*
    1. needs
    2. features
    3. functional requirements
    4. non-functional requirements
5. *Welke term wordt gebruikt voor 'the ability to describe and follow the life of a requirement, in both forwards and backwards direction'?*

    ---
6. *In een project wordt gewerkt volgens een iteratieve en incrementele aanpak. Aan het begin van het project heb je de requirements opgesteld. Op welke momenten in het project kunnen de requirements gewijzigd worden?*
    1. Op elk gewenst moment.

2. Na elke iteratie.
3. Niet meer, de requirements liggen vast.

7. *Geef voor elk van de volgende niet-functionele eisen aan in welke categorie (volgens FURPS) het valt?*

   NFR 1. Het systeem moet 75% van de webpagina's binnen 2 seconden tonen. Voor de overige pagina's mag het nooit langer dan 5 seconden duren.

   NFR 2. Een ervaren internetgebruiker moet binnen 2 minuten op basis van zijn selectiecriteria een hotelkamer kunnen boeken, ook als het de eerste keer is dat hij de hotelreserveringssite bezoekt.

   NFR 3. Het systeem moet in hooguit 2 dagen met een extra Europese taal uitgebreid kunnen worden.

   NFR 4. Het systeem moet 1.200 gebruikers tegelijkertijd aan kunnen met een piek van 2.500 gebruikers op de eerste werkdag van iedere maand. Tijdens de piek mag de snelheid met maximaal 20% dalen.

   NFR 5. Het systeem moet laten zien dat het bezig is als het meer dan 2 seconden nodig heeft om zichtbaar op een actie van de gebruiker te reageren.

Chapter 2. Van requirements naar usecases

Wanneer je weet aan wat voor eisen je systeem moet voldoen kan je deze uitwerken in procesbeschrijvingen. Een semi formele notatie hiervoor zijn use cases, zoals beschreven in de UML (*Unified Modelling Language*).

In de praktijk zal je waarschijnlijk weinig *fully dressed* use cases meer tegenkomen, behalve als je je carriere voortzet bij een meer formelere organisatie. Use cases zijn nog wel relevant om te leren om zo je analyse van het systeem te verduidelijken.

Modernere, meer lichtgewicht varianten van use cases zijn *User Stories*. Deze worden o.a. in Scrum gebruikt en beschrijven functionaliteit volgens het formaat:

> *Als **rol** wil ik **eis** zodat **gevolg**.*

Bijvoorbeeld: *Als pizzakoerier wil ik het afleveradres duidelijk zien zodat ik de pizza makkelijk kan bezorgen op het juiste adres.*

In de leesstof vind je meer informatie over Use Cases en User Stories.

2.1. Lezen

Lees uit [LAR] de volgende delen:

- Hoofdstuk 6: Use-case Model: Writing Requirements in Context

Lees de volgende introductie artikelen over User Stories:

- Mountain Goat Software, User Stories
- Scott Ambler, Agile Modelling, User Stories

Er zijn hele boeken geschreven over hoe je use cases en user stories het beste kan schrijven. Het is geen exacte wetenschap en het enige dat we kunnen doen is deze vaker te schrijven om er zo beter in te worden.

2.2. Opdrachten

Hieronder vind je een aantal casussen die je moet uitwerken naar use cases en user stories.

2.2.1. Casus 1: Cursusomgevingen beheren

Wanneer er een cursus voor ParnasSys of SOMtoday gegeven wordt door een trainer heeft deze voor haar studenten een omgeving nodig waarin de studenten kunnen werken. Voor de cursus wil de trainer de omgeving kunnen inrichten zodat de studenten goed kunnen starten met de training. De cursusomgeving wordt aangemaakt wanneer de trainer vraagt om de omgeving. Hiervoor wordt door Robin een database server en een daaraan gekoppelde applicatie server ingericht. Wanneer Robin klaar is met de inrichting stuurt hij een email naar de trainer met daarin de gegevens van de omgeving (URL naar de applicatie). Als de training gegeven is moet de omgeving ook weer

opgeruimd worden om zo ruimte te maken op de server voor volgende trainingen. Robin stuurt een mailtje naar de trainer een dag voordat de omgeving opgeruimd wordt ter bevestiging dat ze gaan verdwijnen. Het kan zo zijn dat de trainer de cursusomgeving nog wat langer nodig heeft, om een vervolg training te geven of omdat de cursus uitgesteld is.

1. *Identificeer de actor(en) en de use-case(s). Geef duidelijke namen en omschrijvingen.*
   Actor(en):

   + Alle use-case(s):

2. *Teken een use-case diagram*
   (bijv. met https://draw.io, of https://plantuml.com)
3. *Beschrijf alle gevonden use-case(s) in brief format*
   (gebruik bijv. *Google Docs*)
4. *Beschrijf (één van) de gevonden use-case(s) in full format*
   Gebruik het *single column* formaat
5. *Beschrijf de gevonden use-cases ook in User Stories formaat*
   Gebruik dus *Als **rol** wil ik **eis** zodat **gevolg**.* als formaat.

2.2.2. Casus 2: Kaaswinkel

De kaaswinkel "Op de Brink" uit Deventer wil graag het assortiment ook online kunnen verkopen. Een bezoeker moet dus kunnen zien welke kazen er allemaal in de winkel aanwezig zijn. Wanneer de bezoeker een kaas gevonden heeft die haar lekker lijkt, voegt ze deze kaas toe aan de winkelmand. De kaas kan in eenheden van 100gr toegevoegd worden. Pas wanneer er meer dan 300gr besteld is kan er afgerekend worden om zo niet te weinig te hoeven versturen. Per honderd gram komt er €0,50 aan verzendkosten bij met een minimum van €3,-. Wanneer de bezoeker klaar is met het toevoegen van kazen aan de winkelwagen en overgaat tot bestellen moet er natuurlijk een verzendadres ingevoerd worden en moet er betaald worden.

Betaling kan via iDEAL, Bitcoin of onder rembours bij aflevering. In het geval van iDEAL wordt er een iDEAL transactie gestart bij *Currence*. In deze transactie wordt het ID van de winkel, de transactie ID van de order en het bedrag verzonden. Hierna krijgt de winkel verder geen inzage in de transactie, anders dan dat aan het eind een bericht gestuurd wordt naar de winkel dat er betaald is. Bij Bitcoin betaling wordt er een QR-code gegenereerd met daarin versleuteld het bedrag dat afgerekend moet worden en het Bitcoin id van de winkel. De klant scant de QR-code met haar *Bitcoin Wallet* en bevestigt de betaling. Het Bitcoin systeem stuurt dan een bericht naar de winkel dat de betaling gelukt is. Betaling onder rembours kost €3,- extra.

1. *Identificeer de actor(en) en de use-case(s). Geef duidelijke namen en omschrijvingen.*
   Actor(en):

   + Alle use-case(s):

2. *Teken een use-case diagram*
   (bijv. met https://draw.io, of https://plantuml.com)
3. *Beschrijf alle gevonden use-case(s) in brief format*

(gebruik bijv. *Google Docs*)

4. *Beschrijf (één van) de gevonden use-case(s) in full format*
   Gebruik het *single column* formaat
5. *Beschrijf de gevonden use-cases ook in User Stories formaat*
   Gebruik dus *Als **rol** wil ik **eis** zodat **gevolg**.* als formaat.

2.2.3. Casus 3: Pakketbedrijf

Het pakketbedrijf "Van Hot naar Her" bouwt een systeem waarmee pakketten on-line getraceerd kunnen worden. Een klant registreert het ophaaladres, ophaaldatum en afleveradres bij het systeem. Als het pakket zwaarder is dan 100kg moet er €100 extra betaald worden en kan het alleen met een vrachtwagen vervoerd worden. De klant ontvangt een track'n'trace nummer waarmee in het systeem de status van het pakket kan worden uitgelezen. Een pakketbezorger krijgt dan een melding dat het pakket opgehaald moet worden en als deze bij het ophaaladres gekomen is wordt geregistreerd dat het pakket opgehaald is. De bezorger gaat dan zijn ronde afmaken en rijdt naar het distributiecentrum. Daar wordt de wagen uitgeladen en krijgen de pakketjes de status "In distributiecentrum".

De volgende dag wordt door de bezorger een vrachtwagen volgeladen met pakketjes die bezorgd moeten worden. Wanneer de bezorger een pakketje laadt wordt de status weer aangepast naar "Onderweg". De bezorger gaat dan z'n ronde doen en wanneer een pakketje bij een afleverpunt bezorgd is verandert de status naar "Bezorgd". Als het pakketje niet afgeleverd kon worden gaat deze terug naar het distributiecentrum en wordt de volgende dag geprobeerd het pakketje te bezorgen.

1. *Identificeer de actor(en) en de use-case(s). Geef duidelijke namen en omschrijvingen.*
   Actor(en):

   + Alle use-case(s):

2. *Teken een use-case diagram*
   (bijv. met https://draw.io, of https://plantuml.com)
3. *Beschrijf alle gevonden use-case(s) in brief format*
   (gebruik bijv. *Google Docs*)
4. *Beschrijf (één van) de gevonden use-case(s) in full format*
   Gebruik het *single column* formaat
5. *Beschrijf de gevonden use-cases ook in User Stories formaat*
   Gebruik dus *Als **rol** wil ik **eis** zodat **gevolg**.* als formaat.

Chapter 3. Van usecases naar een domeinmodel

Nu je weet wat je applicatie allemaal ongeveer moet doen door het bepalen van de use cases of user stories (de functionele vereisten), en hoe je dat moet doen (de non-functionele vereisten), ga je een model maken van hoe je applicatie zich moet structureren.

Hiertoe maak je een domeinmodel waarin de essentiële domeinbegrippen en -concepten uitgewerkt worden en tot elkaar in relatie gezet worden.

3.1. Lezen

Lees de onderstaande hoofdstukken uit [LAR] over *domain models*:

- Hoofdstuk 10

Na dit hoofdstuk kan je al beginnen met de onderstaande opdrachten over het vinden van de domeinbegrippen.

Deze hoofdstukken horen bij elkaar voor het tekenen van de domeinmodellen:

- Hoofdstuk 11
- Hoofdstuk 12

Je kan dan de opdrachten afmaken.

3.2. Opdrachten

Hieronder vind je een aantal casussen die je moet uitwerken in een domeinmodel.

3.2.1. Casus 1: Cursusomgevingen beheren

Deze casus bouwt voort op casus 1 "Cursusomgevingen beheren" uit de vorige module. Lees die casus tekst ook weer aandachtig.

Een cursus wordt gegeven voor een product van Topicus (Onderwijs), zoals *ParnasSys* of *SOMtoday*. De cursus heeft een code en een naam. Wanneer door de cursusleider de omgeving wordt aangevraagd wordt er vastgelegd voor welke cursus en product de omgeving wordt gebruikt.

Bij de aanvraag worden de startdatum en de eigenaar ingevuld. Wanneer de omgeving klaar is wordt de einddatum gezet en de URL waarop de omgeving beschikbaar is wordt vastgelegd en verzonden in de email naar de aanvrager. De einddatum wordt op 2 weken na de startdatum gezet. Voor de omgeving wordt een afspraak in de agenda van de eigenaar verzonden met daarin een unieke code (UUID) om de afspraak te kunnen identificeren.

Om een omgeving aan te maken moet er een URL aangeroepen worden op een server bij Justin. Deze URL is verschillend per product, bijv. https://foreman.edu.codes/create/parnassys.

Van de eigenaar wordt de naam en het emailadres vastgelegd zodat we de eigenaar via email kunnen bereiken voor notificaties en de afspraken.

Een omgeving kan als status "aangevraagd", "in gebruik", "wordt verwijderd" of "verwijderd" hebben. Wanneer de servers van de omgeving verwijderd zijn krijgt de omgeving in de applicatie de status "verwijderd". De omgevingsaanvragen blijven bewaard om zo later het gebruik in te kunnen zien.

1. *Identificeer de mogelijke domein concepten en domein attributen door middel van Noun Phrase Identification. Doorloop alle stappen en probeer zo tot de uiteindelijke lijst van domein concepten en attributen te komen. Laat de tussenliggende stappen zien.*

   Nouns

2. *Teken een domein class model. Ken de attributen aan de juiste concepten toe, en teken de juiste associaties. Let erop dat je de juiste multipliciteiten aangeeft tussen de verschillende concepten.*

   Domeinmodel diagram:

3.2.2. Casus 2: Kaaswinkel

Deze casus bouwt voort op casus 2 "Kaaswinkel" uit de vorige module. Lees die casus tekst ook weer aandachtig.

De kaaswinkel slaat het assortiment aan kazen op met de naam, een omschrijving en de prijs per 100gr. Wanneer een klant de bestelling afrondt worden de winkelwagen regels opgeslagen als orderregels. Deze bestaan uit de kaas die aangeschaft is, de hoeveelheid en het totaalbedrag van de regel. Een order bestaat uit minimaal één orderregel. Van de order wordt het totaalbedrag, de verzendkosten, eventuele betalingskosten, en wie het besteld heeft. De betaalwijze wordt ook geregistreerd zodat de winkel kan besluiten te stoppen met de betaalmethode als er te weinig gebruik van gemaakt wordt. Bij de koper wordt het verzendadres opgeslagen zodat de winkel de bestelling kan verzenden.

1. *Identificeer de mogelijke domein concepten en domein attributen door middel van Noun Phrase Identification. Doorloop alle stappen en probeer zo tot de uiteindelijke lijst van domein concepten en attributen te komen. Laat de tussenliggende stappen zien.*

   Nouns

2. *Teken een domein class model. Ken de attributen aan de juiste concepten toe, en teken de juiste associaties. Let erop dat je de juiste multipliciteiten aangeeft tussen de verschillende concepten.*

   Domeinmodel diagram:

3.2.3. Casus 3: Pakketbedrijf

Deze casus bouwt voort op casus 3 "Pakketbedrijf" uit de vorige module. Lees die casus tekst ook weer aandachtig.

Een pakje wordt bij "Van Hot naar Her" geregistreerd door het gewicht vast te leggen, het ophaaladres en het afleveradres. Een adres is een straatnaam, huisnummer, postcode en plaats. Een pakketje krijgt ook een track and trace code (UUID).

"Van Hot naar Her" heeft verschillende distributiecentra. Een distributiecentrum heeft naam en een adres. Als een pakketje bij een distributiecentrum wordt afgeleverd wordt dit geregistreerd met het aflevermoment. Wanneer het pakketje weer onderweg is wordt dit ook met een tijdstip geregistreerd. Je kan dus altijd vragen waar een pakketje zich bevindt op een willekeurig tijdstip.

Een pakket begint met "klaar voor ophalen". Als de bezorger het pakketje opgehaald heeft, verandert de status naar "Onderweg", en wordt het pakketje toegevoegd aan de lading van de bestelbus of -vrachtwagen. Wanneer het pakketje afgeleverd is in het distributiecentrum dan is de status "In distributiecentrum", en is het pakketje verwijderd uit de lading van de vrachtwagen. Wanneer het pakketje onderweg is naar het afleveradres verandert de status weer naar "Onderweg" en wordt het weer aan de lading van de vrachtwagen toegevoegd. Als het pakketje bezorgd is dan is de status "Bezorgd", en is het pakketje weer verwijderd uit de lading.

Een bezorger identificeert zich met een code en een naam. De bestelbus of vrachtwagen van de bezorger heeft een maximale belasting qua gewicht, en een maximale snelheid. Een vrachtwagen heeft een maximale snelheid van 90km/u, en een maximaal gewicht van 10.000kg; een bestelbus heeft een maximale snelheid van 130km/u en een maximaal gewicht van 2.000kg, daarnaast mag een pakketje voor een bestelbus maximaal 100kg wegen. Dit geldt niet voor een vrachtwagen.

Omdat "Van Hot naar Her" altijd de dienstverlening wil kunnen verbeteren is het noodzakelijk om alle stappen te kunnen zien dat een pakketje doorlopen heeft: van aanmelding tot aflevering. Hierdoor kunnen ze zien hoe lang het duurt voordat een pakketje afgeleverd wordt en hoe vaak bijvoorbeeld iemand niet thuis is.

1. *Identificeer de mogelijke domein concepten en domein attributen door middel van Noun Phrase Identification. Doorloop alle stappen en probeer zo tot de uiteindelijke lijst van domein concepten en attributen te komen. Laat de tussenliggende stappen zien.*

   Nouns

2. *Teken een domein class model. Ken de attributen aan de juiste concepten toe, en teken de juiste associaties. Let erop dat je de juiste multipliciteiten aangeeft tussen de verschillende concepten.*

   Domeinmodel diagram:

   == Domeinmodel patterns

Chapter 4. Van domeinmodel naar software