

Inleiding

Het doel van analyse en ontwerp van software is om een helder beeld te krijgen van deze software, en om hierover met elkaar te kunnen communiceren.

Een middel om dit doel te bereiken is een verzameling afbeeldingen (**diagrammen**), die kort en bondig het volgende beschrijven:

- **use cases** (gebruiksgevallen) die gebruikers kunnen uitvoeren
- **gegevens** die met de software worden bekeken en/of bewerkt
- **(bedrijfs)processen** waar deze gegevens doorheen lopen
- **rollen en autorisaties** die gebruikers van de software hebben
- **gebruikersschermen**
- **navigatie** tussen deze gebruikersschermen
- **koppelingen** tussen systemen en componenten
- **volgorde** waarin gebeurtenissen plaatsvinden

Dit boek toont hoe deze diagrammen gemaakt kunnen worden.

Deze diagrammen vormen samen het **model** van een applicatie.

De meeste beschreven modelleertechnieken zijn onderdeel van een internationaal gestandaardiseerde modelleertaal, de **Unified Modeling Language (UML)**. UML wordt beheerd en onderhouden door een internationale organisatie, de Object Management Group (OMG).

Om deze modelleertechnieken in een context te plaatsen, worden in dit boek ook enkele niet-UML-gerelateerde aspecten van het analyse- en ontwerpproces beschreven.

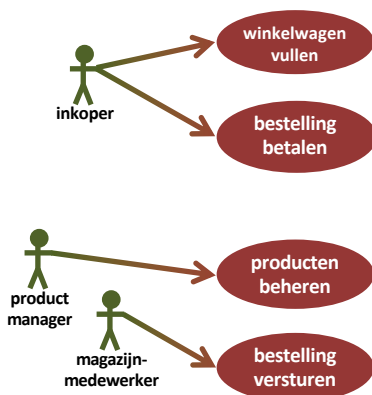
1 Overzicht creëren met een use case diagram

Use cases geven een overzicht van de informatie die nodig is om van start te kunnen gaan met informatie-analyse en ontwerp.

Use cases beschrijven **wat** de applicatie moet doen, zonder in te gaan op **hoe** de applicatie dit doet. Een use case diagram bevat:


- Menselijke **actoren**, in de vorm van draadpoppetjes. Dit zijn de **rollen** die menselijke gebruikers van het systeem kunnen hebben.
- **Use cases**, in de vorm van ellipsen. Dit zijn de handelingen die een gebruiker kan uitvoeren. De naam van een use case bevat altijd een werkwoord (wat doet de gebruiker), en meestal een zelfstandig naamwoord (waarmee doet de gebruiker dit). De naam van een use case wordt vaak de naam van een menu-item, knop en/of scherm in de applicatie.
- Pijlen van een actor naar de use cases die hij/zij kan uitvoeren.

Voorbeeld



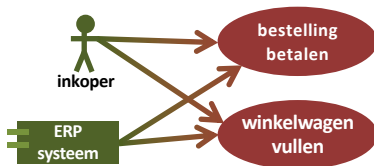
Dit use case diagram beschrijft een webshop, waarin een inkoper een winkelwagen kan vullen, een bestelling kan betalen, etc.

1.1 Systeem actor

Indien een use case kan worden uitgevoerd door een extern geautomatiseerd systeem, dan wordt dit systeem gerepresenteerd door een **systeem actor** (*Engels: system actor*): 

Voorbeeld

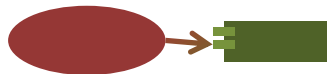
In een webshop kan een geautomatiseerd ERP (Enterprise Resource Planning) systeem een winkelwagen vullen en een bestelling betalen. Dit ERP-systeem is een systeem actor:



1.2 Primary en secondary actor

Een actor die een use case uitvoert is een **primary actor**, en heeft een **pijl naar de use case**.

Indien er voor het uitvoeren van een use case ook een andere actor nodig is, is dat een **secondary actor**, en heeft de use case een **pijl naar de secondary actor**:



Voorbeeld

Om de use case 'bestelling betalen' uit te voeren, heeft deze use case het iDeal systeem van een bank nodig. Dit iDeal systeem is dan een secondary actor, en de use case heeft een pijl naar het iDeal systeem.

