[simula . research laboratory]

Kravanalyse og objekt-orientert analyse

Bente Anda
Forsker, Simula Research Laboratory
1. Amanuensis II, IFI

11.09.2006

[simula . research laboratory]

Overblikk 11/9 + 14/9

- Kravanalyse med use case og system sekvens diagrammer
- Objekt-orientert analyse med domene modell
- Objekt-orientert design med sekvens- og klassediagrammer
- Fokus på prinsipper, ikke syntaks

[simula . research laboratory]

Use case modellen

- beskriver kravene til systemet (hva skal det gjøre og hvem skal bruke det),
- beskriver systemet sett fra kundens/brukerens perspektiv,
- beskriver interaksjonen mellom brukeren og systemet (på en måte som brukeren kan forstå),
- beskriver 'hva' som skjer, ikke 'hvordan' det skjer,
- dokumenterer omfanget til systemet,
- har blitt en "de facto" standard for håndtering av krav i systemutviklingsprosjekter og
- er starten på analysefasen. Use case modellen benyttes videre i prosjektplanlegging, objekt-orientert analyse og design og i testing.

[simula . research laboratory]

Use case modellen forts.

- Use case model = Use case diagram + Use case beskrivelser
- Use case diagram = gir en enkel oversikt, men er ikke komplett.
- Use case beskrivelser = strukturerte tekstlige beskrivelser, ingen standard.

[simula . research laboratory]

Finn systemets omgivelser og omfang

- Hvilken funksjonalitet må systemet inkludere og hvilken kan eksluderes?
- Hvordan skal dette systemet forholde seg til andre systemer i vår arkitektur?
- Hvem skal bruke dette systemet?
- Hvem eller hva er systemet avhengig?
- Hvilke produkter og/eller resultater skal systemet tilby?
- Hvorfor trenger brukerne/andre systemer den funksjonaliteten som dette systemet tilbyr?

[simula . research laboratory]

Aktører

- tegnes som fyrstikkmennesker med et navn,
- er rollene som mennesker, andre systemer eller hardware komponenter har når de kommuniserer med ett eller flere use case,
- den eller det (person, maskin etc. som interagerer med et use case,
- ikke det samme som titler eller personer,
  - mennesker med en jobb tittel kan spille rollen til mange aktører
  - en aktør kan representere flere jobb titler
- kan delta i mer enn et use case,
- kan, men må ikke være representert med en klasse I klassediagrammet.

Oppgave 1

•Finn aktører for dette systemet:

A fire detection system monitors a series of fire detection sensors for signs of fire, and when one is detected it rings an alarm, sets off a set of sprinklers, and notifies the local fire department.

Use Case

- tegnes som ellipser med et navn i eller under ellipsen,
- beskriver en sekvens av handlinger som systemet utfører for å oppnå et observerbart resultat av verdi for en aktør,
  - hvert **use case steg** beskriver en enkelthandling mellom aktør og system
  - en **operasjon** er ett eller flere steg som må utføres samlet
  - et komplett use case består av flere ulike hendelsesforløp (flyt, scenarier)
- beskriver eksternt synlig og testbar systemoppførsel,
- navnet er vanligvis et aktivt verb og en substantivsetning,
- må ikke interagerer direkte med aktører.

[simula . research laboratory]

Assosiasjoner

- Linje mellom aktør og use case
- Kan ha piler for å vise hvor kommunikasjonen er initiert (pilen peker fra initiator).
- Representer kommunikasjonslinkene mellom en instans av et use case og en instans av en aktør.
- Identifiserer interaksjoner mellom aktører og use case.

[simula . research laboratory]

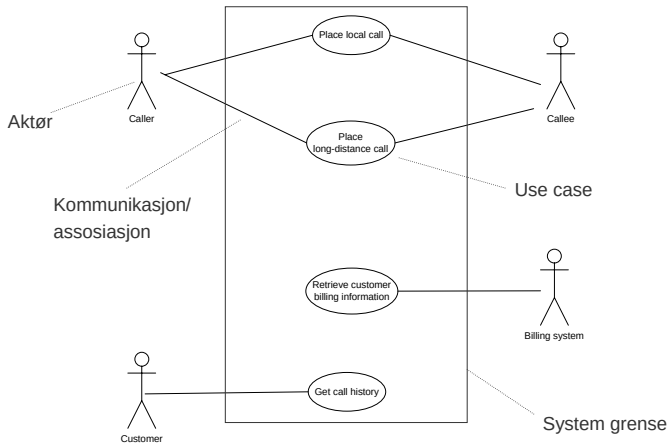
Stegene i use case modellering

1. Identifiser primære aktører
2. Identifiser use case fra aktørenes mål.
   - Et use case -
     'One person - one place - one time'
3. Identifiser sekundære aktører, dvs. aktører som ikke har egne mål, men som er nødvendige for å gjennomføre use casene.
4. Tegn use case diagram. Dette gir et overblikk over aktører og use cases og dermed over funksjonaliteten til systemet.
5. Lag tekstlige beskrivelser av use casene. Disse viser hvordan aktører når mål ved bruk av systemet.

Et eksempel til:

A telephone system shall allow callers to place local calls and long-distance calls. For long-distance calls the system shall always select the cheapest routing. The system shall provide continuously up-to-date call history for all accounts to its customers, as well as billing information to a separate billing system.

Telefonsystemet



[simula . research laboratory]

Detaljering i iterasjoner

1. En høy nivå use case modell består av diagram og en kort beskrivelse av alle use case
2. En uformell modell har “main success” scenarier på de viktigste use case
3. Variasjoner og feilsituasjoner finnes ved hjelp av “brainstorming”
4. Use casene detaljeres ut til de er komplette (fully dressed).
   - Det er ofte ikke nødvendig å detaljere ut til alle use casene er komplette.
   - Use casene kan detaljeres ut v.h.a. tekst eller et annet diagram som collaboration, sequence eller state chart. Tekst er mest egnet hvis use case skal brukes i kommunikasjon med bruker.

[simula . research laboratory]

Hvordan skrive detaljerte tekstlige use case

- Beskriv hva som gjøres, ikke hvordan det gjøres
- Skriv hendelsesflyten som en nummerert liste på formen
  1. <Head waiter> <enters> <the current date>
  2. <System><displays><bookings for that date>
- Finn riktig detaljeringsnivå
  - Beskriv kun 1 hendelse per steg
  - Vanligvis beskrives ikke detaljer om brukergrensesnitt.
    - Vanligvis benyttes essensielle use case ref. Larman

  Eksempel: Ikke ’Aktør trykker på ’Send’-knappen’

[simula . research laboratory]

Main success scenario for ’Place local call ’

Trigger: The Caller lifts the receiver

1. The Caller enters the number to be called
2. The system connects the Caller’s phone to the requested device

   1. The system analyzes the digits of the entered number and determines the network address of Callee
   2. The system determines whether a connection can be established between Caller and Callee
   3. The system establishes the connection
   4. The system rings Callee’s phone

   ← subflow
3. The call is made
4. The connection is terminated
5. The details of the call are recorded

[simula . research laboratory]

Pre- og postbetingelser

Er avgjørende for å kunne gjøre funksjonell, black-box testing av systemet

Use case ”Place Local Call”:

Precondition:

The Caller’s device has a connection to the system, signal is present

Post condition:

The connection between the Caller and the Callee has been terminated and all call details have been recorded.

Variasjoner

No answer

If the Callee does not answer, the Caller replaces the handset and the use case ends.

Line Busy

If the Callee's line is in use, the system rings the busy tone. The Caller then replaces the handset and the use case ends.

Feilsituasjoner

Number Dialled Not Known

The system cannot identify a receiving device from the number dialed. An error message is transmitted to Caller, and the use case ends.

The Signal is Lost

The carrier/signal is lost during the call. The use case ends.

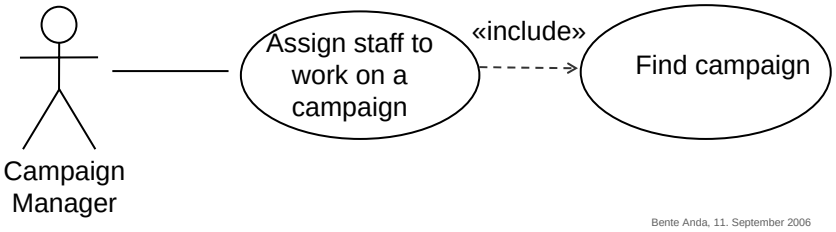
Use case relasjoner

Use case modellen utvides gjennom flere iterasjoner med mer funksjonalitet, variasjoner og feilsituasjoner.

- **Include-relasjonen:**
  Et use case kan være en del av ett flere andre use case.
- **Extend-relasjonen:**
  Et use case som beskriver tilleggsoppførsel som utføres under gitte omstendigheter

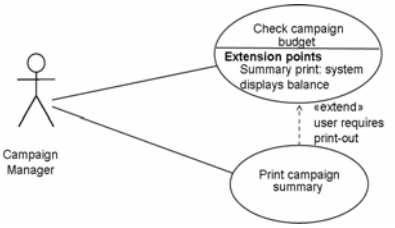
Include relasjonen

- To eller flere use cases kan ha en felles del. Denne delen kan da legges ut i et eget use case som disse use casene kan inkludere.
  - Include kan også brukes for å forenkle store use case med mange steg
  - Include kan videre brukes for å håndtere hendelser som kan forekomme når som helst i utførelsen av use caset
- Basis use caset vet hvilke use case det inkluderer
- Include lar oss abstrahere ut felles oppførsel og forenkler den overordnede strukturen, men skaper avhengigheter mellom use casene.



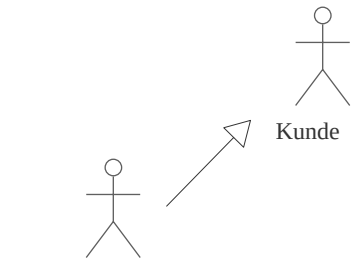
Extends relasjonen

- Alternativ oppførsel som utgår fra 'extension points' i use caset kan enten skrives som eget use case, eller som en variasjon. Hvert use case steg er et potensielt extension point.
- Variasjoner beskriver hva som skjer ved avvik i normal flyt.
- Extends use case beskriver hvordan tilfredsstille tilleggsmål.
- Basis use caset er fullstendig definert uten extensions, disse utvider funksjonaliteten.
- Basis use caset kjenner sine extended use case



Relasjoner mellom aktører

- Kan definere generelle aktører (som f.eks. `Kunde`) og spesialisere disse (som f.eks. `Ekstern kunde`) v.h.a. en generaliseringsrelasjon.



(Kan gjøre alt det som `Kunde` *kan + mer)*

Utfordringer

- Kravene blir til mens de beskrives
- Kravene til systemet endres underveis

  → Use case modellen må oppdateres
- Kravene kan bli beskrevet med ujevn detaljeringsgrad. Dette motvirkes ved å detaljere ut i iterasjoner.

[simula . research laboratory]

Posisjonering

- Teknikken forutsetter en visjon av systemet som skal lages.
- Størrelsen og kompleksiteten til systemet som skal lages avgjør hva som må foreligge før use case modellen kan utformes.
  - Beskrivelse av forretningsprosesser som skal støttes
  - Beskrivelse av systemets kontekst
- Use case modellen brukes videre i utviklingsprosessen.
  - Planlegging og estimering
  - Design
  - Testing

[simula . research laboratory]

Use cases i prosjektplanlegging

- Planlegg hvilke use case som skal realiseres i hvilke iterasjoner:
  - Realiser use casene i henhold til hvor viktige de er og/eller hvor vanskelige de antas å være å implementere.
  - Normal hendelsesflyt realiseres først, deretter variasjonene.
  - Estimerer hvor mange use case (eller hendelsesflyt og variasjoner) som kan realiseres i en iterasjon.

[simula . research laboratory]

Domenemodell

- Domenemodellen brukes i objekt-orientert analyse
- Domenemodellen viser konsepter i applikasjonsdomenet, konseptuelle klasser, og forholdet mellom dem: ideer, ting, objekter.
- Hensikten med domenemodellen er å forstå konseptene og få en oversikt over terminologi.
- Domenemodellen beskrives med UML klassediagrammer uten metoder, og utarbeides gjennom flere iterasjoner.
- Use case modell og domenemodell utformes parallelt.
- Domenemodellen fanger opp informasjonen om entiteter som er beskrevet i use casene.
- Use casene presiseres ved utforming av domenemodellen.
- Domenemodellen er et viktig verktøy for å sjekke at use casene er beskrevet med riktig detaljeringsnivå.
- Det er bedre å spesifisere for mange konseptuelle klasser enn for få.

[simula . research laboratory]

Hvordan finne domeneklasser?

- Lag en liste over kandidater til klasser:

2 forskjellige tilnærminger:

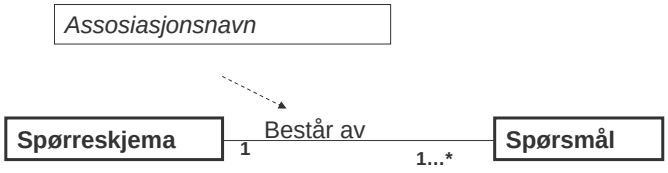
  - Lag en liste over kandidater til domeneklasser (for eksempel basert på liste s.140-141)
  - Finn substantiver og substantivutrykk i use case beskrivelsene

- Ta bort unødvendige klasser
  - Noen klasser kan vise seg å være attributter.

Assosiasjoner

- En assosiasjon er en relasjon mellom to klasser som viser at det er en sammenheng mellom dem.
  - En assosiasjon kan ha et navn
- Brukes i domenemodellen hvis informasjon om relasjonen skal lagres.
- Hver ende av assosiasjonen kalles en rolle, en rolle kan ha:
  - Multiplisitet
  - Navn
  - Navigasjon

Eksempel



Assosiasjonen går i begge retninger.

Liste over mulige assosiasjoner på s. 155

Attributter

- Inkluder de attributter som det må lagres informasjon om i følge use casene
- Attributter er vanligvis datatyper, f.eks

  int, Boolean, string, dato
- Modeller konsepter som klasser, ikke attributter. Hvis det er tvil, lag en klasse