

Krav analyse og objekt-orientert analyse

Bente Anda

22.09.2005

1

Overblikk 22/9 + 26/9

- 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

22.09.05

INF3120

2

Use case modellering

Use case modellen

- beskriver kravene til systemet,
- beskriver systemet sett fra kundens/brukerens perspektiv,
- beskriver 'hva' som skjer, ikke 'hvordan' det skjer og
- har blitt en "de facto" standard for håndtering av krav i systemutviklingsprosjekter.
- Kostnader ved feilhåndtering av krav er høye
→ Det er viktig med en god metode for å analysere krav.

22.09.05

INF3120

3

Stegene i use case modellering

1. Identifiser primære aktører, dvs. brukere, hardware komponenter eller andre (del)systemer som har mål de skal ha oppfylt ved bruk av dette systemet.
2. Identifiser use case fra aktørenes mål.
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.

22.09.05

INF3120

4

Use case modellen

- Hvert use case steg beskriver en enkelthandling mellom bruker og systemet
- En operasjon er et eller flere steg som må utføres samlet
- Et komplett use case består av flere ulike hendelsesforløp (flyt, scenarier)
- Use case modellen benyttes videre i prosjektplanlegging, objekt-orientert analyse og design og i testing.

22.09.05

INF3120

5

Restaurantsystem

The system shall support the processes of making reservations and allocating tables to customers, and it shall replace a manual system.

Customers contact the restaurant to make or cancel advance bookings, and a receptionist receives these calls. Bookings are entered for a particular table together with the number of covers. The restaurant runs three sittings in an evening: "Pre-theatre", "Dinner" and "Supper", but bookings can be made for time periods that span more than one sitting. A contact name and phone number is recorded for each booking.

When a party arrives, it is seated at its table by the head waiter. The corresponding booking is crossed out. If the party is seated at a table other than the one booked, an arrow is drawn from the original booking to the new table. Customers may cancel bookings in advance. The time by which a table must be vacated can also be recorded.

It is, of course, possible to eat without making an advance booking if a free table is available. This is recorded as a table occupancy, but no record of name or telephone number is made.

When new bookings are recorded, or changes are made to existing bookings, the display should be immediately updated, so that the staff is always working with the latest information available.

22.09.05

INF3120

6

[Oppgave 1]

Finn aktører for systemet

22.09.05

INF3120

7

[Oppgave 2]

■ 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.

22.09.05

INF3120

8

[Identifisere use case]

- Identifiser de use casene som oppfyller målene til de primære aktørene
- Et use case -
'One person - one place - one time'
- Et use case inneholder mange steg

22.09.05

INF3120

9

[Oppgave 3]

- Finn use casene til restaurantsystemet

22.09.05

INF3120

10

Et eksempel til:

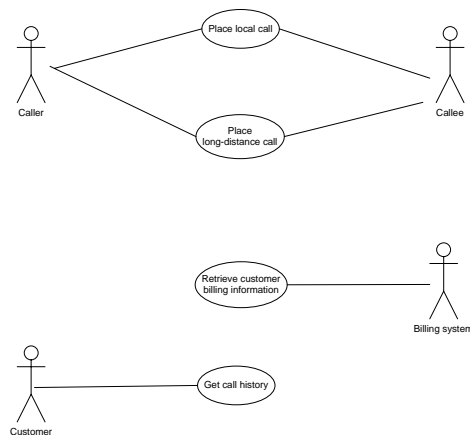
A telephone system shall allow callers to place local calls and long-distance calls. For long-distance calls the system shall always select the cheapest routing. The system shall provide continuously up-to-date call history for all accounts to its customers, as well as billing information to a separate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

22.09.05

INF3120

11

Telefonsystemet



22.09.05

INF3120

12

[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 alle er komplette (fully dressed).
 - Det er ofte ikke nødvendig å detaljere ut til alle use casene er komplette.

22.09.05

INF3120

13

[Hvordan skrive detaljerte 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'

22.09.05

INF3120

14

[Main success scenario for 'Record booking']

1. The receptionist enters a date
2. The system displays the bookings for that date
3. The receptionist enters the customer name and phone number, the time of the booking, the number of covers and the table reserved
4. The system records and displays the new booking

22.09.05

INF3120

15

[Use case på 3 nivåer]

1. Business use case
Eks. Et system for markedsførings-kampanjer kan ha 3 business use case: 'Registrer kunde', 'Gjennomfør kampanje for kunde' og 'Fakturer kunde'
2. User level use case
Use case som tilfredsstill mål for kunder
3. Subfunksjon use case
Forenkler use case ved å beskrive funksjoner som egne use case, for eksempel 'Logg inn', 'Registrer kundes adresse'

22.09.05

INF3120

16

[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

22.09.05

INF3120

17

[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 case
- Basis use case vet hvilke use case det inkluderer
- Include lar oss abstrahere ut felles oppførsel og forenkler den overordnede strukturen, men skaper avhengigheter mellom use casene.

22.09.05

INF3120

18

[Extends relasjonen]

- Alternativ oppførsel som utgår fra 'extension points' i use case 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 tilfredsstillende tilleggsmål.
- Basis use case er fullstendig definert uten extensions, disse utvider funksjonaliteten.
- Basis use case kjenner sine extended use case

22.09.05

INF3120

19

[Oppgave 4]

- Hvordan kan use casene til restaurant-systemet struktureres v.h.a. include og extends?

22.09.05

INF3120

20

[Oppgave 5]

- Lag komplette beskrivelse av "Record arrival" og "Table transfer"

22.09.05

INF3120

21

[Hvorfor use case modellering?]

- De funksjonelle kravene, dvs. hva systemet skal gjøre, må beskrives
- Metoden oppmuntrer til å stille riktige spørsmål til riktig tid
- Metoden er systematisk
 - Utviklere og kunder kan sjekke at use case modellen inneholder det som er nødvendig
- Det er enkelt å navigere i modellen
 - Får raskt overblikk over funksjonaliteten
 - Kan studere detaljer når det er nødvendig

22.09.05

INF3120

22

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.

22.09.05

INF3120

23

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

22.09.05

INF3120

24

[Use cases i prosjektplanlegging]

- Følgende brukes for å planlegge 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.

22.09.05

INF3120

25

[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.

22.09.05

INF3120

26

[Domenemodell forts.]

- 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.

22.09.05

INF3120

27

[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 substantivuttrykk i use case beskrivelsene
- Ta bort unødvendige klasser
 - Noen klasser kan vise seg å være attributter.

22.09.05

INF3120

28

Assosiasjoner

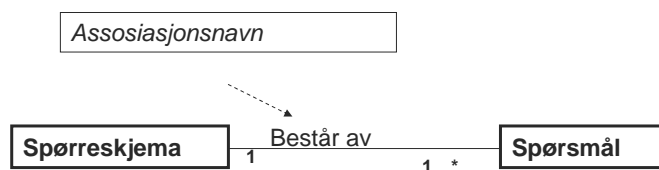
- En assosiasjon er en relasjon mellom to klasser som viser at det er en sammenheng mellom dem.
 - En assosiasjon kan ha et navn
- Brukes i domenemodellen hvis informasjon om relasjonen skal lagres.
- Hver ende av assosiasjonen kalles en rolle, en rolle kan ha:
 - Multiplisitet
 - Navn
 - Navigasjon

22.09.05

INF3120

29

Eksempel



Assosiasjonen går i begge retninger.

Liste over mulige assosiasjoner på s. 155

22.09.05

INF3120

30

[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

22.09.05

INF3120

31

[Oppgave 6]

- Lag en domenenmodell for restaurantsystemet

22.09.05

INF3120

32