

Use case modellering

Metode for å identifisere og beskrive de funksjonelle kravene til et system.

Bente Anda

21.09.2004

1

Modellering i INF3120

- Fordypning i objekt-orientert analyse og design
- Bygger på UML- kunnskapen fra INF102
- Fokus på prinsipper for analyse og design (ikke UML syntaks)

INF3120 vs. INF3220 (IN265)

Fokus INF3220:

- Samspill mellom systemet og organisasjonen der det skal brukes.
- Kravspesifikasjon i form av teknikker for å klargjøre krav og beskrive systemer.
- UML brukes for å beskrive analyse og design, men uten en spesiell prosess.

■ Fokus INF3120:

- Planlegging og gjennomføring av systemutviklings- prosjekter.
- Analyse og design når en (overordnet) kravspesifikasjon foreligger.
- "God" design ved hjelp av UML

Use case dreven utvikling

■ Analyse

1. Identifiser aktører og deres mål
2. Lag et use case diagram
3. Lag høynivå (brief) use case beskrivelser
3. Utdyp beskrivelsen av hvert use case med beskrivelser av normal hendelsesflyt og variasjoner.

■ Design

- (Identifiser objekter og fordel ansvar mellom dem)
4. Lag sekvensdiagram for viktige use case
 5. Lag klassediagram på use case nivå
 6. Lag klassediagram på systemnivå

Use case modellering

Use case modellen

- beskriver kravene til systemet,
 - beskriver systemet sett fra kundens 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.

Use case modellering forts.

- Use case modellen består av use case diagram og tekstlige beskrivelser av de enkelte use casene.
- Use case diagrammet gir et overblikk over aktører og use cases og dermed over funksjonaliteten til systemet.
- Den tekstlige beskrivelsen av et use case viser hvordan en aktør når et mål ved bruk av systemet.
- Hvert use case steg beskriver en enkelthandling (transaksjon) mellom bruker og systemet
- Et komplett use case består av flere ulike hendelsesforløp (flyt, scenarier)
- Use case modellen benyttes videre i prosjektplanlegging og objekt-orientert analyse og design.

Detaljering i iterasjoner

- En høy nivå use case modell består av diagram og en kort beskrivelse av alle use case
- En uformell modell har "main success" scenarier på de viktigste use case
- Variasjoner og feilsituasjoner finnes ved hjelp av "brainstorming"
- Use casene detaljeres ut til alle er komplette (fully dressed)

Eksempel: Spørreskjemagenerator

Kravspesifikasjon:

Et meningsmålingsinstitutt ønsker å få laget et system der spørreskjema er på Internett/Web. Systemet skal gjøre det enkelt å legge et spørreskjema ut på Web, enkelt for andre å fylle ut skjemaene på Web. Svarene skal lagres på et format som kan eksporteres til andre verktøy (f eks "strukturert tekst" som kan importeres til et regneark). Deltakerne skal kunne lagre svarene underveis og fortsette utfyllingen av skjemaet senere. Til noen av spørsmålene er det nødvendig å lese en del tekst. Meningsmålingsinstituttet ønsker å ha en enkel oversikt over svarene som er kommet inn, f eks hvor mange som har svart på de ulike spørsmålene. Det som skal lages er altså ikke et enkelt spørreskjema på Web, men en "spørreskjema-generator" for Web.

Identifiser aktører og use cases

- Identifiser aktører ved å se på brukere av systemet
 - Personer i roller
 - Andre systemer
 - Hardware (I/O, CPU..)
- Identifiser aktørens mål (behov) ved bruk av systemet
- Aktører som har et mål ved bruk av systemet er **primære aktører**. Aktører som bare gir input til systemet er **sekundære aktører (supporting actor)**.
- Identifiser de use casene som oppfyller målene
 - Et use case -
'One person - one place - one time'

Hvordan skrive use case

- Beskriv hva som gjøres, ikke hvordan det gjøres
 - Skriv hendelsesflyten som en nummerert liste på formen
 1. <Ansatt> <ber om> at <spørreskjema>< blir lagret>
 - 2.<Systemet><lagrer><skjemaet>
 - 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'

Aktører og målbeskrivelse

Aktør	Mål 1	Mål 2	Mål 3
Ansatt	Generer spørreskjema	Publiser spørreskjema	Vis statistikk
Deltager	Velg spørreskjema	Svar på spørreskjema	
Web-browser	Lagre spørreskjema		

21.09.04

INF3120

11

Aktører i et embedded, real-time system

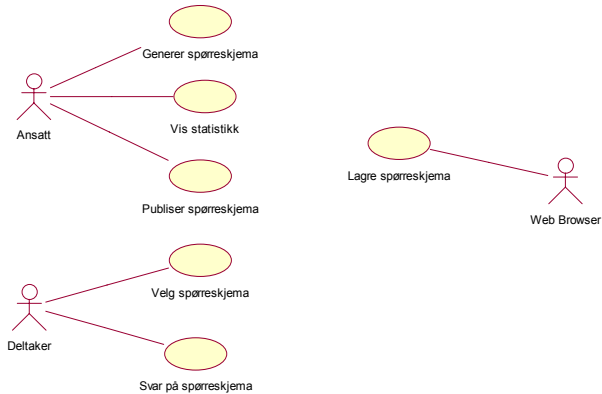
- **Compact Flash Card**
Compact Flash card used to load new firmware to the controller.
- **Power Switch**
Switches power on after power failure.
- **Intended Stall**
Stall timer which triggers takeover in a redundant system.
- **Bootmain**
Models the product that is responsible for doing the basic hardware initialization at start-up.
- **Safety Module Update Code in the Processor Module**
Code in the processor module responsible for firmware upgrade of the safety module.
- **User**
Presses the Init button to restart the system.

21.09.04

INF3120

12

Overordnet use case modell - spørreskjemagenerator



21.09.04

INF3120

13

Main success scenario for 'Generer Spørreskjema'

1. Systemet ber om overskrift, innledning og antall spørsmål
2. Ansatt skriver inn nødvendig informasjon
3. Systemet sjekker at alle felt er utfylt
4. Systemet viser et spørreskjema der tekst til spørsmål skal fylles inn
5. Ansatt skriver inn tekst og svaralternativ
6. Systemet sjekker at riktig antall spørsmål har fått tekst
7. Ansatt ber om at spørreskjemaet blir lagret
8. Systemet lagrer spørreskjemaet

21.09.04

INF3120

14

Use case relasjoner

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

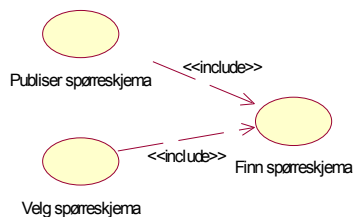
- **Include-relasjonen:**
Et use case kan være en del av ett flere andre use case.
- **Extend-relasjonen:**
Et use case som beskriver tilleggsoppførsel som utføres under gitte omstendigheter

21.09.04

INF3120

15

Include relasjonen



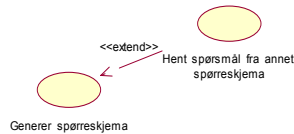
- To eller flere use cases kan ha en felles del. Denne delen kan da legges ut i et eget use case som disse use casene kan inkludere.
- Basis use caset vet hvilke use case det inkluderer
- Include lar oss abstrahere ut felles oppførsel og forenkler den overordnede strukturen, men skaper avhengigheter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

21.09.04

INF3120

16

Extend 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

21.09.04

INF3120

17

Use case 'Svar på spørreskjema'

Aktør: Deltaker

Trigger: Deltaker velger å svare på undersøkelsen

1. Systemet viser spørreskjemaet
2. Deltaker svarer på spørsmålene
3. Systemet sjekker om alle spørsmålene er besvart
4. Deltaker bekrefter svarene
5. Systemet sender skjemaet og genererer en kvittering

Variasjoner:

3a. Alle spørsmål er ikke besvart

1. Systemet ber deltaker fylle ut resten av spørsmålene

21.09.04

INF3120

18

Komplett use case beskrivelse av 'Generer spørreskjema'

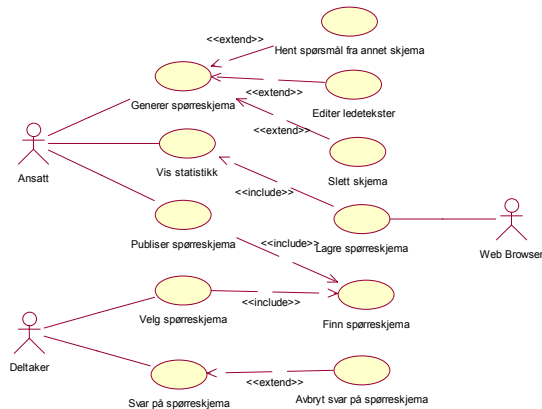
Use Case	Generer spørreskjema
Aktor	Ansatt
Trigger	Ansatt ønsker å opprette et nytt spørreskjema
Pre-betingelser	Ansatt har valgt å sette opp et nytt spørreskjema
Post-betingelser	1. Nytt spørreskjema opprettet eller 2. Ansatt har fått feilmelding
Normal hendelsesflyt	1. Systemet ber om overskrift, innledning og antall spørsmål som spørreskjemaet skal bestå av. 2. Ansatt skriver inn nødvendig informasjon 3. Systemet sjekker at alle felt er utfylt Extension point: UC "Hent spørsmål fra annet spørreskjema" 4. Systemet viser et spørreskjema der tekst til spørsmål skal fylles inn. 5. Ansatt skriver inn tekst og evt. svaralternativ til hvert av spørsmålene 6. Systemet sjekker at riktig antall spørsmål har fått tekst 7. Ansatt ber om at spørreskjema blir lagret 8. Systemet lagrer spørreskjemaet
Variasjoner	3a. Alle felt er ikke tilfredsstillende utfylt. 3a1. Systemet informerer ansatt om hvilke felt som ikke er utfylt, og går ikke videre før dette har blitt ordnet. 6a. Alle de angitte spørsmålene har ikke fått en tekst. 6a1. Systemet informerer sekretæren om hvilke spørsmål som ikke har fått tekst, og går ikke videre før dette har blitt ordnet.
Relatert informasjon	Svar på spørsmål kan være fritekst eller avkrysningsbokser med alternativer.

21.09.04

INF3120

19

Include og extend use cases



21.09.04

INF3120

20

”Concrete” use case style

- Beskriver den konkrete prosessen med brukergrensesnitt og objekter.
- Denne formen brukes vanligvis ikke, bare når det foreligger et design.
- Eksempel for use case ”Svar på spørreskjema”
 1. Systemet viser spørreskjemaet
Gjenta 2 og 3 til alle spørsmål er besvart
 2. Systemet markerer neste spørsmål som skal besvares og plasserer markøren ved svaret.
 3. Deltaker svarer på spørsmålet
 4. Systemet sjekker om alle spørsmålene er besvart
 5. Deltaker trykker på knappen ”Bekreft svarene”
 6. Systemet sender skjemaet og viser en kvittering

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

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.

Ikke-funksjonelle krav

- Eksempler på ikke-funksjonelle krav er krav til sikkerhet, ytelse, responstid, kostnader eller detaljer rundt hvilke hardware eller software komponenter som skal brukes.
- Ikke-funksjonelle krav som gjelder et spesielt use case kan skrives i det.
- Ikke-funksjonelle krav som gjelder flere use case eller hele systemet beskrives i et eget dokument.

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

21.09.04

INF3120

25

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.

(Kap. 8 i Larman)

21.09.04

INF3120

26

Estimering basert på use case

- **Use case poeng metoden:**
 - Hver aktør og hvert use case kategoriseres i henhold til kompleksitet og tilordnes en vekt.
 - Kompleksiteten til et use case måles i antall steg (forutsetter jevn detaljeringsgrad)
 - Ujusterte use case poeng beregnes ved å legge sammen vektene for hver aktør og hvert use case.
 - De ujusterte use case poengene justeres basert på verdiene til 13 tekniske faktorer og 8 omgivelsesfaktorer.
 - Tilslutt multipliseres de justerte use case poengene med en produktivitetsfaktor.

21.09.04

INF3120

27

Tekniske faktorer

Factor number	Factor description	Weight
T1	Distributed system	2
T2	Response or throughput performance objective	1
T3	End-user efficiency	1
T4	Complex internal processing	1
T5	Code must be reusable	1
T6	Easy to install	0.5
T7	Easy to use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Includes special security features	1
T12	Provides direct access for third parties	1
T13	Special user training facilities are required	1

21.09.04

INF3120

28

Omgivelsesfaktorer

Factor number	Factor description	Weight
F1	Familiar with RUP	1.5
F2	Application experience	0.5
F3	Object-oriented experience	1
F4	Lead analyst capability	0.5
F5	Motivation	1
F6	Stable requirements	2
F7	Part-time workers	-1
F8	Difficult programming language	-1

Produsere et estimat

- Ujustert aktør vekt, UAW, beregnes ved å legge sammen vektene for hver aktør
- Ujustert use case vekt, UUCW, beregnes tilsvarende
- Ujusterte use case poeng, UUCP, = UAW + UUCW
- Teknisk faktor, TCF, = $.6 + (.01 * \sum_{1..13} T_n * Vekt_n)$
- Omgivelsesfaktor, EF, = $1.4 + (-.03 * \sum_{1..8} F_n * Vekt_n)$
- UCP = UUCP * TCF * EF
- Estimat = UCP * Produktivitets faktor

Eksempel

- Use case modellen på foil 20:
 - 3 aktører; Ansatt og deltaker er mennesker som kommuniserer via et grafisk brukergrensesnitt, Web browser er et annet system med et enkelt grensesnitt mot vårt system.
 - 11 use case; "Svar på spørreskjema" har 6 steg, "Generer spørreskjema" har 10 steg. Tilordner kompleksitet til de 9 andre use casene.
 - Tilordner verdier til tekniske og omgivelsesfaktorer.
 - Produktivitetsfaktoren settes bl.a. basert på omgivelsesfaktorene.

21.09.04

INF3120

31

Actors Weight Factors	Description	Weight	Enter Count	Weighted Value	Comment	
Simple_Actor	Simple program interface	1	1	1		
Average_Actor	Complex program interface	2	0	0		
Complex_Actor	Graphical interface	3	2	6		
Total Actor Weight			1	7		
Use Cases Weight Factors (Bases on the number of transactions in a use case)						
Simple_Use_Case	Description	Weight	Enter Count	Weighted Value	Comment	
Simple_Use_Case	3 or fewer transactions	5	3	15		
Average_Use_Case	4 to 7 transactions	10	5	50		
Complex_Use_Case	more than 7 transactions	15	3	45		
Transaction Based Factors				110		
Unadjusted Use Case Points				117		
Technical Weight Factors						
Rating Scale is 0 to 5						
	Description	Weight	Enter Rating	Weighted Value	Reason	
T1	Distributed System	2	0	0		
T2	Response or throughput performance objectives	1	3	3		
T3	End-user efficiency (online)	1	3	3		
T4	Complex internal processing	1	1	1		
T5	Code must be reusable	1	4	4		
T6	Easy to install	0.5	2	1		
T7	Easy to use	0.5	3	1.5		
T8	Portable	2	0	0		
T9	Easy to change	1	3	3		
T10	Concurrent	1	0	0		
T11	Includes special security features	1	1	1		
T12	Provides direct access for third parties	1	0	0		
T13	Special user training facilities are required	1	1	1		
Technical Factors				18.5		
Technical Complexity Factor (TCF)	.6 + (.01*Technical Factor)			0.785		
Environmental Factors for Team and Weights						
Rating Scale is 0 to 5						
	Description	Weight	Enter Rating	Weighted Value	Reason	Experience Stability
F1	Familiar with the Rational Unified Process	1.5	4	6		0
F2	Application experience	0.5	4	2		1
F3	Object-Oriented Experience	1	2	2		1
F4	Lead analyst capability	0.5	2	1		1
F5	Motivation	1	2	2		1
F6	Stable requirements	2	3	6		0
F7	Part-time workers	-1	0	0		0
F8	Difficult programming language	-1	2	-2		0
Environmental Factor				17		4
				0.89		
Use Case Points				81.74255		
Man hours per use case point				28		
Estimated Manhours in Project				2288.7774		