Modellbasert utvikling

Christian Herzog
Knut Sagli

Hvem er vi?

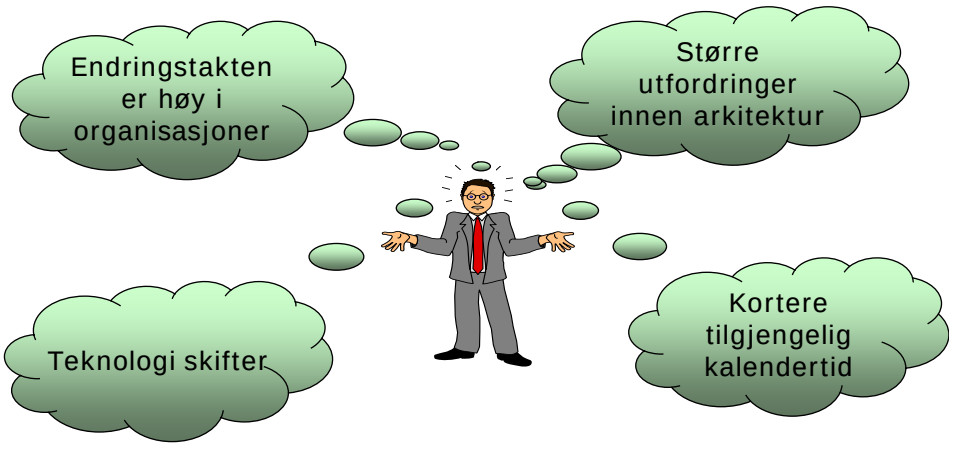
- Christian Herzog, seniorkonsulent i Esito
- Knut Sagli, sjefskonsulent i Esito
- Esito utvikler verktøy og leverer spisskompetanse i prosjekter

Agenda

- Generelt om metoder og modeller
  - Modelldrevet utvikling
  - Boehms spiral
  - Smidige prosesser og modellering
  - Arkitektur
- Gjennomgang av en reell modelldrevet utviklingsmetode
  - Faser og aktiviteter
  - Fokus på modeller
- Praktisk eksempel med Genova

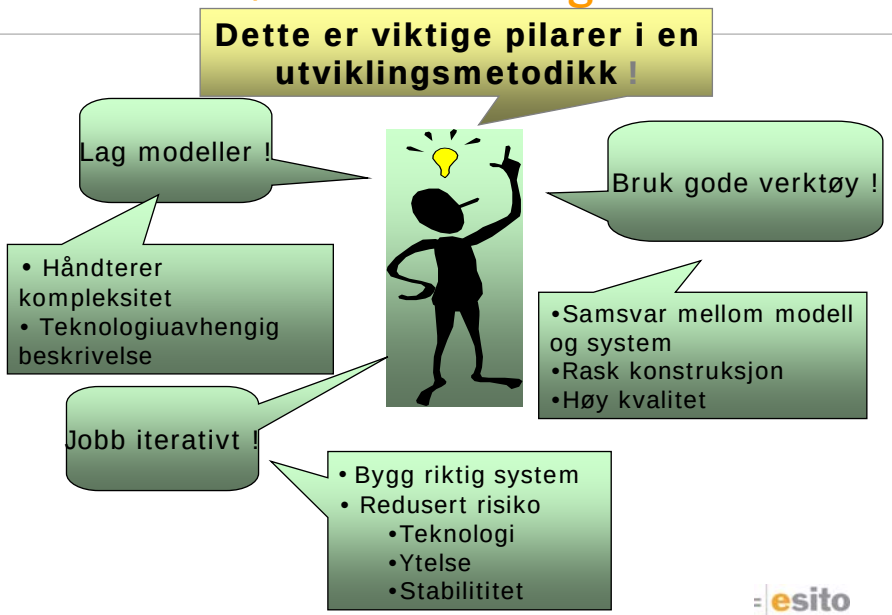
esito

Utfordringer i systemutvikling



esito

Hvordan møter vi utfordringene?



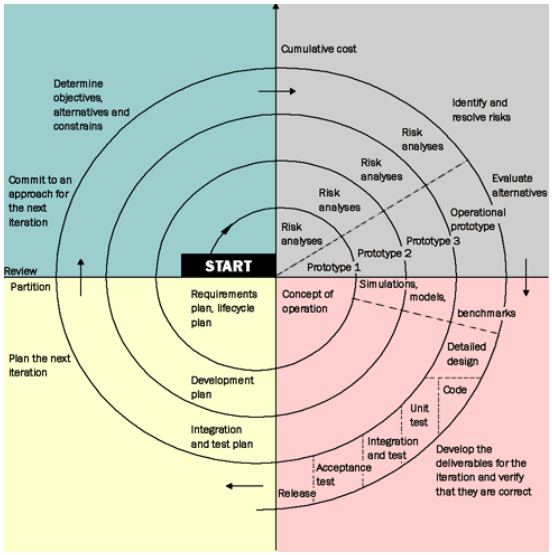
esito

Modelldrevet utvikling

- **Problemorientert**
  - fokus på domene, ikke teknologi/implementasjon, forståelse
- **Produktivitet**
  - lag løsningen fortere, kode mindre
  - hjelp til programmering, generere det trivielle
- **Kvalitet**
  - riktig med en gang
  - mest mulig automatisk programmering
- **Uavhengighet**
  - ikke bundet til målmiljø
  - samme modell brukes til mange ting
- **Vedlikeholdbarhet**
  - 70% av kostnadene i et systems livsløp er vedlikeholdskostnader

esito

Boehms spiral



esito

Spiralmodell

- En runde i modellen representerer en fase (etter hvert iterasjon)
- De fire kvadranter
  - Definere mål (for denne fasen) og identifisere risiko
  - Vurdere og redusere risiko
  - Utvikling og vurdering
  - Planlegging
- Risiko synliggjort og eksplisitt håndtert

esito

Inkrement og iterasjon

- Inkrement: En arbeidspakke, gjerne et funksjonsområde eller samling Use case
- Iterasjon: Gjennomløp av sett av aktiviteter (prosesskomponenter) som i seg selv er et miniprosjekt, innenfor et inkrement

esito

Smidige metoder (Agile methods)

Et forsøk på definisjon av smidig programvareutvikling:

An iterative and incremental (evolutionary) approach
performed in a highly collaborative manner
with "just enough" ceremony
that produces high quality software
which meets the changing needs of its stakeholders

- Scott Ambler

esito

Smidige metoder og spiralmodellen

- Smidige metoder og spiralmodellen
  - Samme fokus på evolusjonær utvikling
  - Prosjektplanen er i stadig endring
- Spiralmodellen har fokus på å håndtere de største risikoer først
- Smidige prosesser har fokus på å produsere den biten kunden ønsker seg mest (som gir størst verdi) først.
- Smidige prosesser bruker risikoreduserende teknikker som en naturlig del av prosessen. De har ikke samme fokus på risiko*analyse* som spiralmodellen.

esito

Smidige prosesser og modellering

- Du finner ofte en antagelse om at modellering ikke hører hjemme i en smidig prosess.
- Det finnes ingen god grunn til at det skal være slik. (Agile model driven design med Scott Ambler, Kent Beck i eXtreme Programming eXplained)
- Modellering og kodegenerering passer godt inn i tankegangen
- Prosessen som vil bli presentert bruker vi selv i en smidig kontekst.

esito

Smidige prosesser og modellering 2

- Modeller bare det nødvendige, verken mer eller mindre
- Fokuser på modeller som gir direkte verdi i forhold til sluttresultatet.
  - Modeller som brukes til kodegenerering kan (bør) betraktes som kode
    - I vår prosess vil klassemodeller, objektseleksjoner og dialogmodeller betraktes som kode.
  - Modeller som brukes som nødvendig dokumentasjon
    - Alle systemer må ha en form for dokumentasjon, også de som er laget smidig
    - Det gjelder bare å finne et passende nivå og en passende form
    - Levende modeller kan være enkel og god dokumentasjon
- Modellene som skal brukes ut over analyse/forståelsesfasen må vedlikeholdes.

esito

Vedlikehold av modeller

- Modeller som ikke vedlikeholdes dør fort
  - Bruk minst mulig tid på dem
  - De gjør skade om noen tror at de er levende
- Modeller vedlikeholdes dersom noen ser en klar personlig gevinst i at det gjøres, ikke ellers
  - Modellene brukes av de som vedlikeholder dem
  - Modellene har en rolle ut over å være analyseverktøy
    - Kodegenerering er et typisk eksempel
- Om en modell ikke har interessenter som vil vedlikeholde den bør det ikke gjøres.

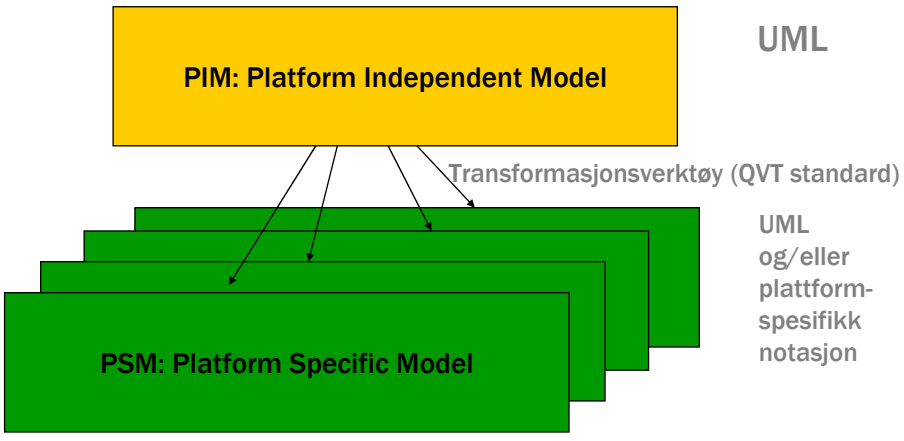
esito

Retningslinjer for smidig modellering

- Finn hvilke modeller som skal brukes. Ha konkrete begrunnelser for hvorfor disse er nyttige.
- Finn hvilke modeller som skal vedlikeholdes. Ha gode grunner for at de bør vedlikeholdes
- Ha modellstandarder på samme måte som kodestandarder
- Modeller må vedlikeholdes av noen som har en interesse i at de er vedlikeholdt.

esito

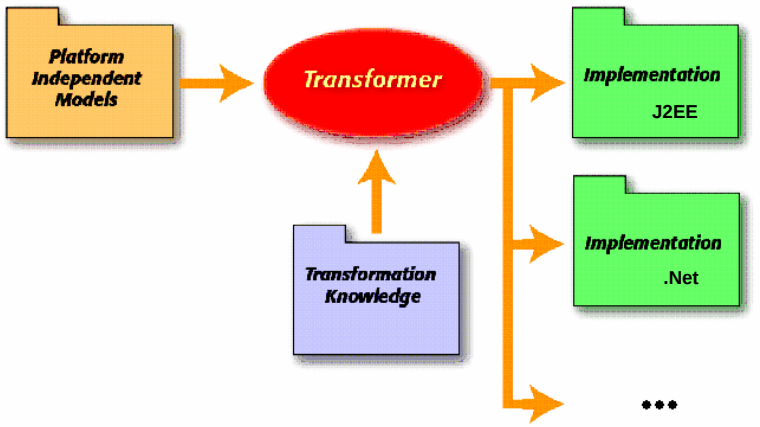
MDA – Model Driven Architecture



Plattformer: Web Services, ebXML, J2EE/EJB, CORBA, MS .Net, ...

esito

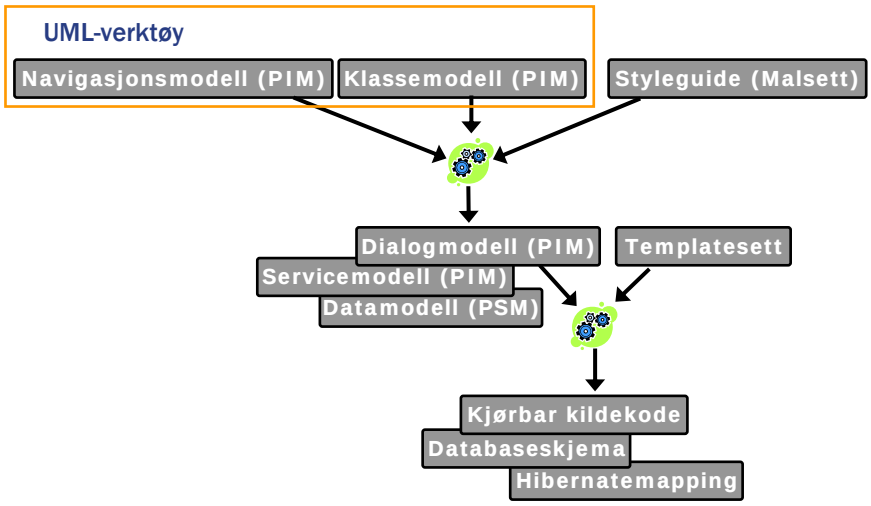
Overblikk over MDA



- Bruk plattformuavhengige modeller (PIM) som spesifikasjon
- Transformasjon til plattformavhengige modeller (PSM) ved hjelp av verktøy

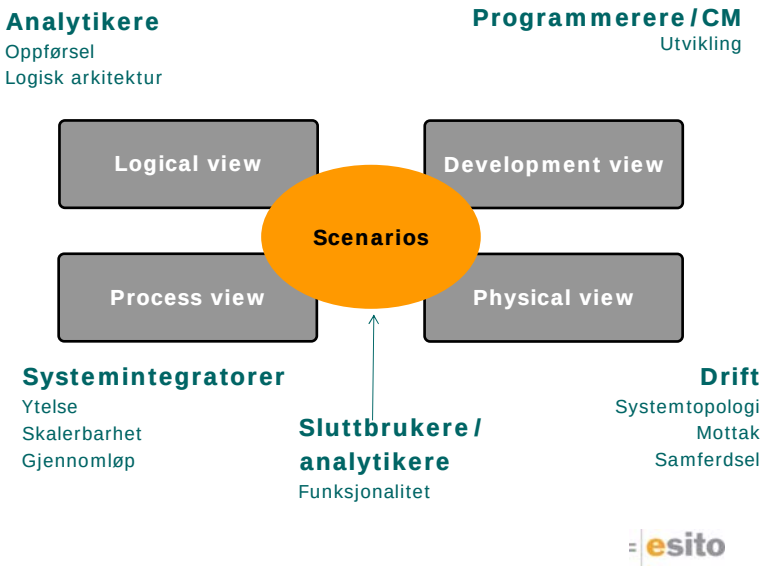
esito

Genova og MDA



esito

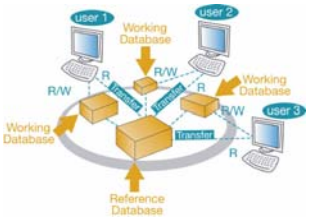
4+1 Architecture view



esito

Arkitektur

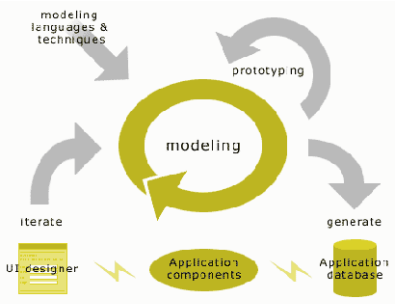
- Arkitektur er en prosjektsubjektiv løsning på de viktige avgjørelsene (for prosjektet)
  - Lagdeling
  - Interaksjon
  - Integrasjon
  - Logisk inndeling
  - Funksjonell inndeling
- God arkitektur gir fleksibelt system
  - Nye klientkanaler
  - Nye integrasjonspunkter
  - Andre tredjepartsverktøy
- Det finnes mønstre (patterns) for mange aspekter ved arkitektur
  - Ansvarsfordeling
  - Innkapsling
  - Lav kopling



esito

Faser i et systems livssyklus

- Vi deler et systems livssyklus inn i seks faser:
  - Idé
  - Utdyping
  - Konstruksjon
  - Overlevering
  - Vedlikehold
  - Avvikling



esito

esito

Idéfasen

Idé

- Dette er ofte bestillers fase.
- Systemgrenser avklares
- Virksomhetsmodellering
- Kravinnhenting og spesifikasjon
- Innledende overordnet arkitektur
- Rammevilkår (Tidsfrister, budsjettrammer osv)
- Ender gjerne opp i en anbudsforespørsel

esito

esito

Utdypingsfasen

Prosjektetablering

- Prosjektstandarder for
  - Kode
  - Modellering
  - Rapportering
  - GUI
  - Navngiving
- Prosjektplan
- Prosjektinfrastruktur
  - Maskinvare
  - Programvare
  - Organisering
- Lokaler
- Rollefordeling

esito

Innledende analyse

- Arkitekturbeskrivelse
- Integrasjon med andre systemer
- Gjenbruk av eksisterende systemer og/eller delsystemer

esito

Modellering

- Modellering er en viktig aktivitet for analyse og design
- Noen modeller brukes til analyse og vedlikeholdes ikke
  - Vedlikeholdes dersom de brukes til annet i tillegg til analyse
- Noen modeller er grunnlag for kodegenerering
  - Disse raffineres til et høyt detaljeringsnivå
  - Holdes kontinuerlig ved like i hele systemets livssyklus

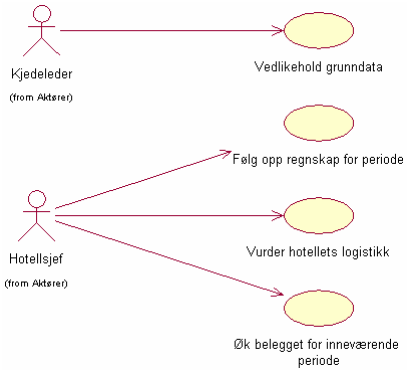
esito

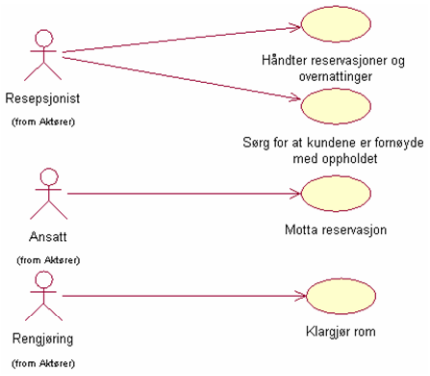
Hvilke diagrammer brukes og til hva

- Use case-diagrammer (UML)
  - Funksjonell beskrivelse
  - Virksomhetsmodellering
- Sekvensdiagrammer (UML)
  - Virksomhetsmodellering
  - Integrasjon
  - Kompliserte kallsekvenser
- Aktivitetsdiagrammer (UML)
  - Virksomhetsmodellering
  - Prosessmodellering
- Klassediagrammer (UML)
  - Navigasjonsmodellering
  - Domenemodellering
  - Programdesign
- Objektseleksjoner (Genova)
  - Tjenestegenerering
  - Grunnlag for dialogmodeller
- Dialogmodeller (Genova)
  - Dialogutforming
  - Dialoggenerering

esito

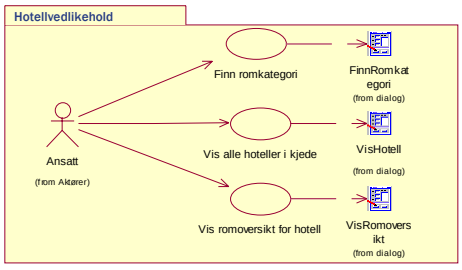
Virksomhetsmodell som use case

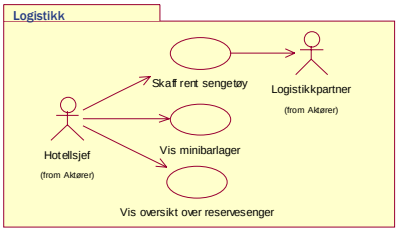




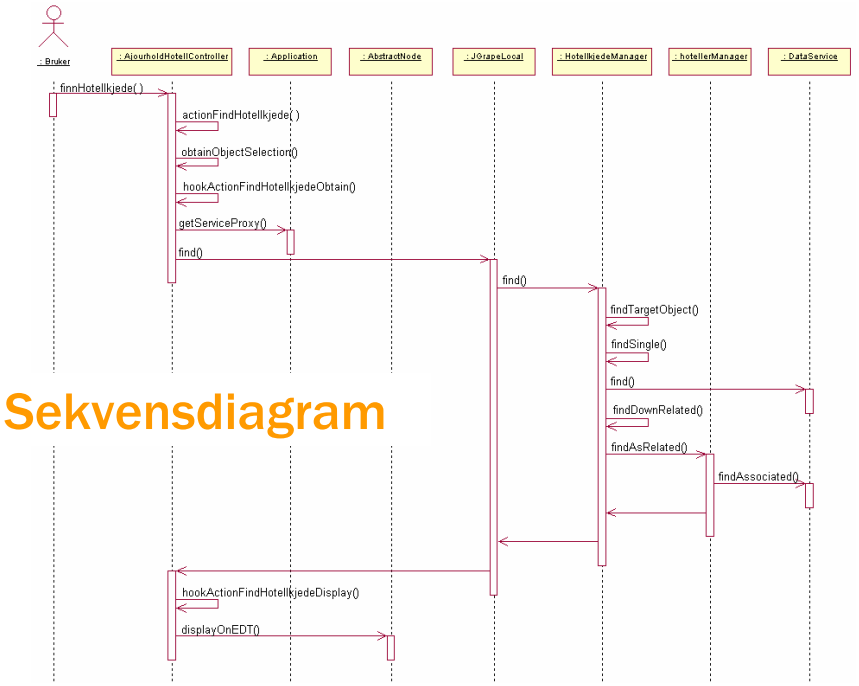
esito

Detaljerte use case



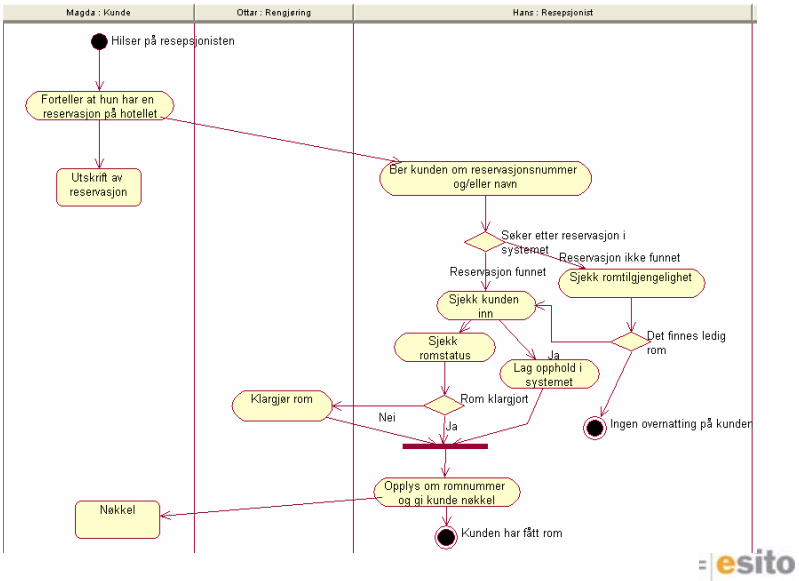


esito

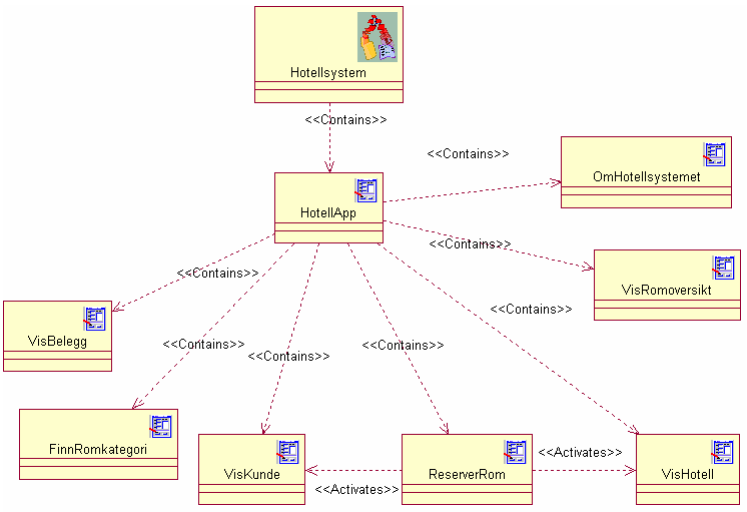


Sekvensdiagram

Aktivitetsdiagram

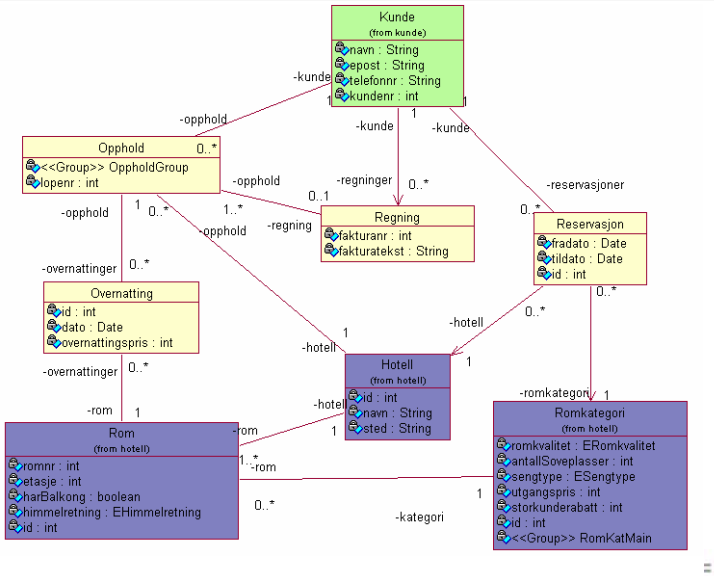


Navigasjonsmodell



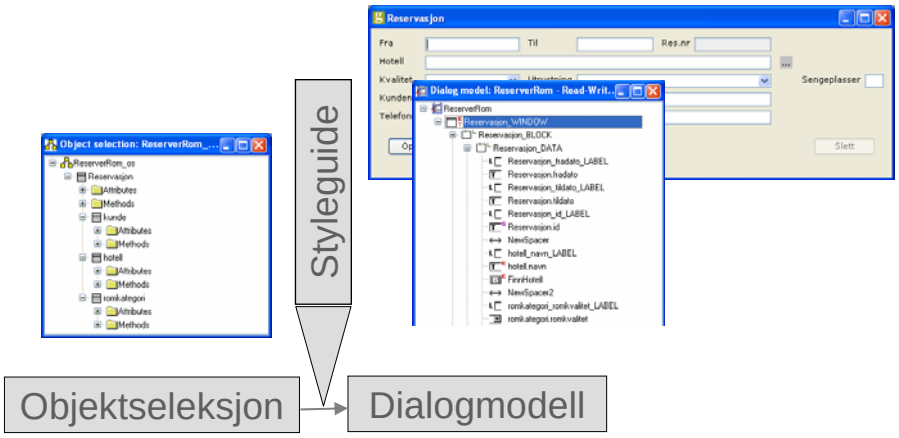
esito

Domenemodell - Opphold



esito

Genova – Fra objektseleksjon til dialogmodell



esito

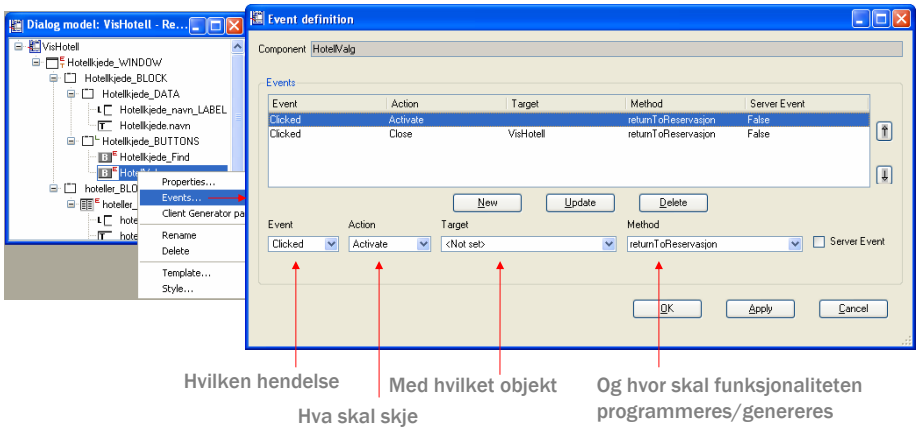
Dialogutforming

- Tegne og fortelle
  - Skisse for struktur og datainnhold
  - Diskuteres med kunden
- Detaljere datainnhold
  - Berike domenemodell med informasjon for dialogene (f eks Title)
  - Utarbeide objektseleksjon
    - velge klasser og assosiasjoner/roller
  - Ekskludere uinteressant innhold (attributter)
- Utseende
  - Style guide
    - farger, fonter,
  - Layout
- Oppførsel
  - Bestemme hvilke tjenester som skal brukes
  - Knytte tjenester til hendelser
- Detaljere modellene
  - Dialogutforming detaljerer domenemodellen
  - Oppdater use case med utfyllende informasjon og dialogavhengigheter

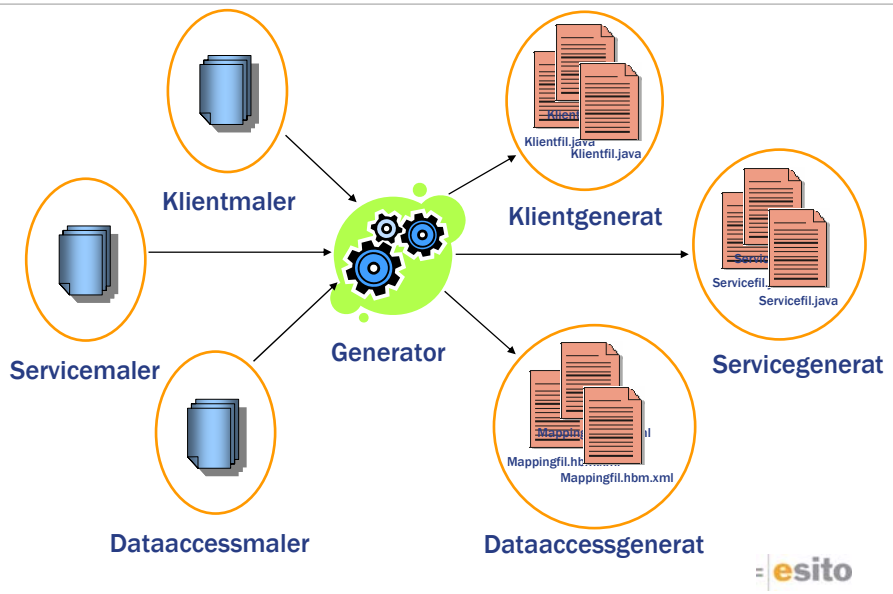
esito

Konstruksjonsfasen

Hendelsesdefinisjoner

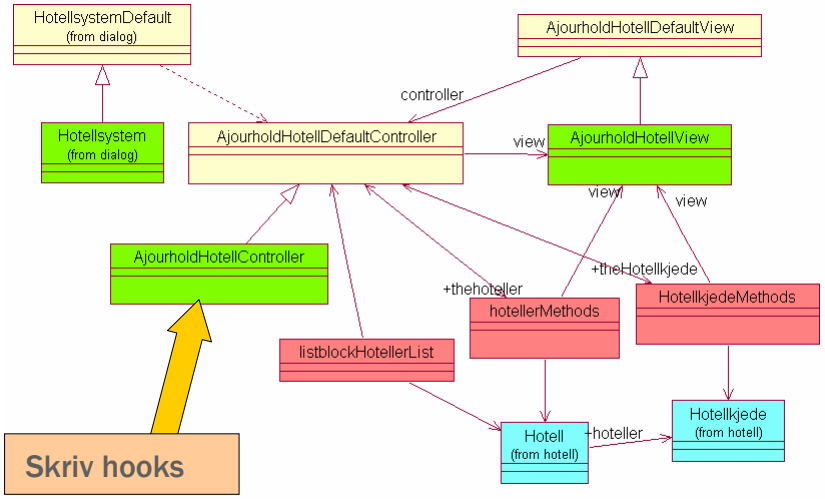


Generering av kildekode



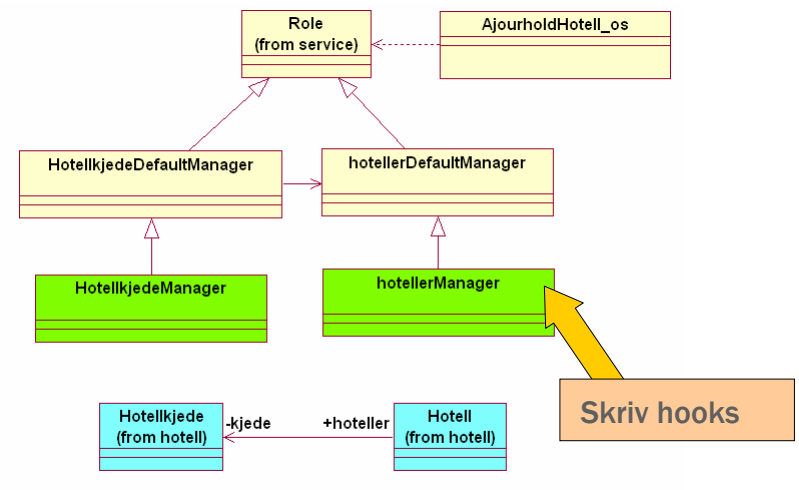
esito

Klassemodell for generert kode



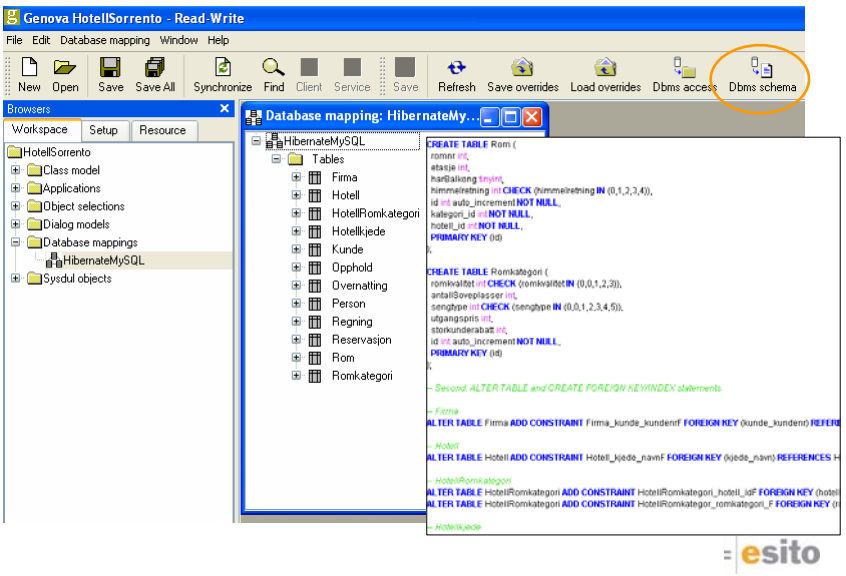
esito

Klassediagram service



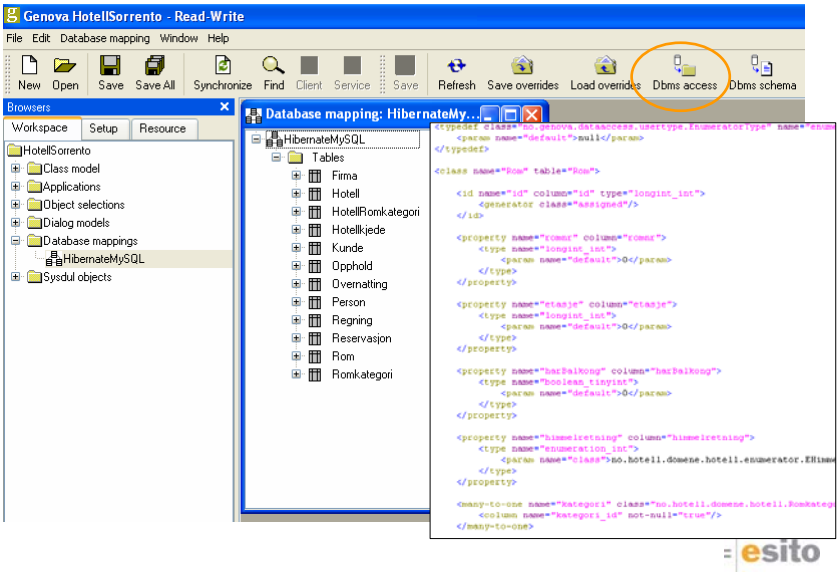
esito

Genova – Generering av database



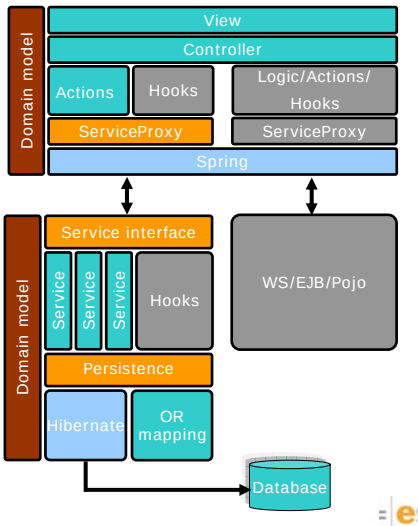
esito

Genova – Generering av mappinger



Applikasjonsarkitektur - Java

- **Generert av Genova**
- **Genova rammeverk**
- **3. part rammeverk**
- **Generert av Rose**
- **Egen kode**



Programmering

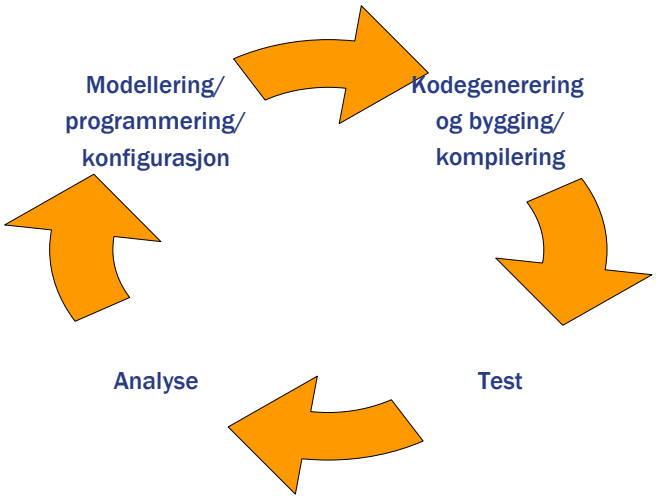
Vi programmerer det vi ikke kan generere, konfigurere eller gjenbruke

- Forretningslogikk (Hooks)
- Integrasjon
- Sikkerhet
- Arbeidsflyt
- Varsling
- Andre tidsbestemte jobber
- Automatiske tester

Til tross for geniale kodegenereringsverktøy og flotte konfigurerbare systemer, må det alltid programmeres. Heldigvis.

esito

Utviklingssyklusen



esito

Leveranse

- Bygging og pakking av systemet
- Installasjonsbeskrivelser
- Systemtest i testmiljø
- Overlevering til driftsmiljø

esito

esito

Overlevering og vedlikehold

Overlevering og vedlikehold

- Ved overlevering pakkes systemet for drift
  - Brukerdokumentasjon
  - Installasjonsbeskrivelse
  - Systemdokumentasjon
- Rutiner for forvaltning trer i kraft
- Vedlikehold viktig i levende system – 60%-90% av kostnadene for systemet
- Levende modeller gir trygg og god oversikt over systemet som skal vedlikeholdes
- TVINN overlevert i 1988. Vedlikeholdt av to utviklere siden

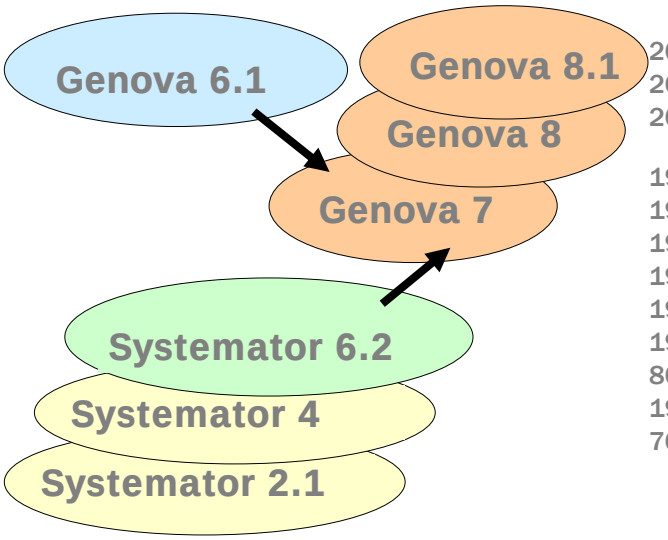
esito

Vedlikehold og modernisering

- Faste rammer og modellkunnskap peker raskt ut hvor endringer og nyutvikling skal gjøres
- Høyt abstraksjonsnivå gjør at man raskt identifiserer hvor noe skal gjøres. Eksempler på lavere nivåer er
  - Java/C
  - Assembler
- Kan generere systemet på nytt ved oppdatering, for eksempel
  - Bytte databasesystem (fra DB2 til Oracle)
  - Oppgradere operativsystem (fra HPUX 9.5 til 10.2)
  - Bytte operativsystem (fra HPUX til Solaris)
  - Bytte kjøremiljø (fra Sysdul til Java)
- Løsninger fra 1995 kan vedlikeholdes i Genova 8.1

esito

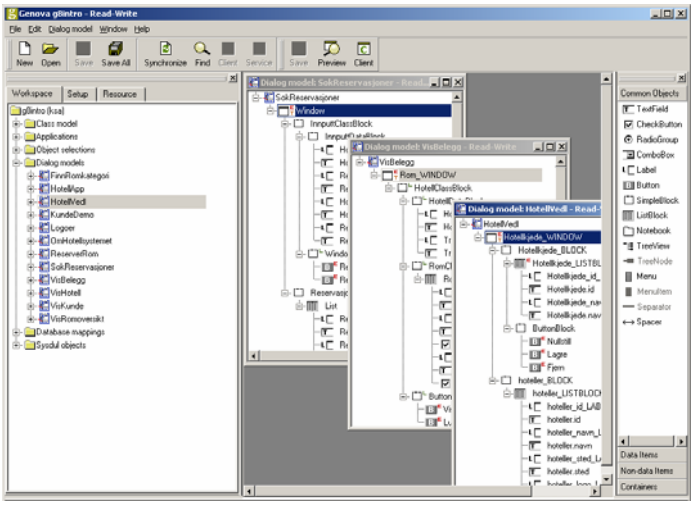
Historikk



2006: Genova 8.1
Service designer
2005: Esito
2004: Genova 8
2000: Genova 7
med sysdul
1998: Genova 6
1997: Genera
1996: Windows NT
1995: Systemator 5
1992: GUI
1989: Unix
80 åra: NSB, TAD
1980: Sysdeco
70 åra: forskning
Frode Aschim

esito

Demonstrasjon



esito

Oppsummering

- Modellbasert utvikling gir vedlikeholdbare og utvidbare systemer
- På relativt kort tid kan man lage kjørende system
- Mye av den koden som er trøblete (og kjedelig å lage) er generert
- Det er forsvinnende få feil i generert kode
- Forretningslogikken må fremdeles skrives

esito