

KONGSBERG

# SIMflash

Kongsberg Defence & Aerospace - Simulation & Training - UTGAVE 2/2008



## Besøk oss på stand #3710 I/ITSEC 2008

Kongsberg Defence & Aerospace - Simulation & Training skal i år være utstillere på I/ITSEC i Orlando fra 1. til 4. desember. Vi skal representere KONGSBERG sammen med Kongsberg Maritime som har lang tradisjon som utstillere på I/ITSEC med fokus på sine brosimulatorer for større fartøyer.

Kongsberg Defence & Aerospace - Simulation & Training vil ha fokus på vår PROTECTOR Training System på I/ITSEC. Vi har hatt stor internasjonal suksess med vår Skyte- og rettesimulator for PROTECTOR Remote Weapon Station og har nå videreutviklet systemet til å inneholde e-læring og interaktive verktøy for vedlikeholdstrening.

Vårt nye treningssystem er det mest komplette og nyvinnende som finnes av elektronisk opplæring på markedet og vi vil gjennomføre mange demonstrasjoner i løpet av messen.

Vi vil oppfordre våre gamle og potensielt nye kunder til å ta kontakt med oss i forkant av messen for å sette av tid til demonstrasjon.

Send en e-post til

[christian.urheim@kongsberg.com](mailto:christian.urheim@kongsberg.com)  
så avtaler vi tid og agenda.

# Komponentbaserte programvareprodukter

Å bygge et system basert på gjenbruk av ferdige byggeklosser er opplagt et billigere alternativ enn å lage byggeklossene om igjen hver gang. Utfordringen er å lage byggeklossene første gang slik at de kan benyttes til å bygge forskjellige systemer – og gjerne systemer en enda ikke har forutsett at en skal lage.

Dag Gravningsbråten – 2008-10-06

Billig gjenbruk av en kloss forutsetter at en ikke må bruke masse tid på å rive løs den klossen en ønsker å gjenbruke fra andre klosser. Klossene må være "løst" koblet (Legoklosser er lettere å gjenbruke enn faststøpte Lecablokker). Det krever at klossene har veldefinerte grensesnitt mot hverandre. Et rammeverk er ofte nødvendig for å sikre at klossene kan lages på en riktig og billig måte. Å lage Legoklosser er ikke håndarbeid. Et rammeverk i form av maskiner og støpeformer kreves i produktfremtakingen.

Byggeklosser for gjenbruk kan være av forskjellige størrelser. Hele systemer kan gjenbrukes ved for eksempel å koble de sammen i en HLA føderasjon. I denne artikkelen fokuseres det på mindre byggeklosser for å bygge programvareapplikasjoner, som for eksempel å tegne symbol for en vogn i et kart, eller format som kartkoordinater vises med. Men også løs kobling mellom større komponenter i systemer med distribuert prosessering belyses.

## Component Based Development

Komponentbasert programvareutvikling (Component Based Development - CBD) har vært en retning innen programvareutvikling i lang tid. Den har stadig fornyet seg i teorier og teknologi som skal muliggjøre hensiktsene på en effektiv måte. Service-oriented Architecture (SOA) kan sees som en for tiden populær representant for disse teknologiene. Noen av de overordnede egenskaper til en programvarekomponent i CBD er:

- Stabile grensesnitt
- Kan benyttes i mange sammenhenger
- Kan benyttes sammen med andre komponenter
- Innkapslet; innholdet kan ikke undersøkes gjennom grensesnittene

Dette er sterke krav og særlig er det vanskelig å tenke seg fremtidige måter å benytte komponenter på. Resultatet kan bli at komponenten kopieres og modifiseres, og en ender opp med flere versjoner av en funksjonalitet som må konfigurasjonsstyres og vedlikeholdes. Dette er ikke kostnadseffektivt i det lange løp. Det kan være lønnsomt for et prosjekt, men ikke i en produktsammenheng.

## Utfordringen

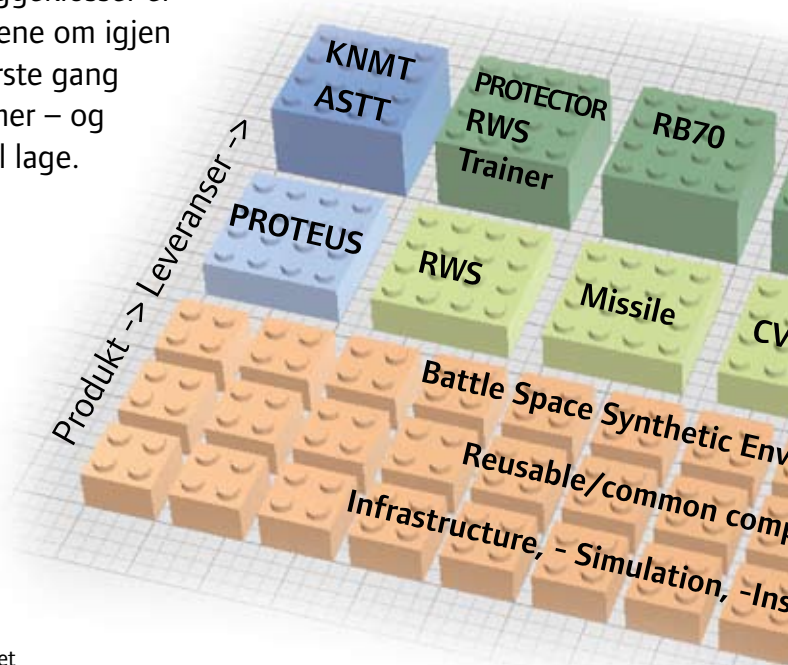
Det hevdes at for å lykkes med CBD så må applikasjonen som skal lages være meget godt definert før kodingen starter. I dagens programvare utviklingsmiljøer er smidige (agile) prosesser populære. Utgangspunktet for disse er at det er nærmest umulig å definere et helt system før en starter implementering. I smidige prosesser foretrekkes mindre tid til produktspesifikasjon og initiell design. Fokus er på korte intervaller mellom prototyper som evalueres sammen med interessenter for å sikre at en kommer fram til et riktig produkt. Smidige prosesser hevdes å resultere i billigere og bedre produkter fordi det er mindre kostnadskrevene å gjøre endringene tidlig enn å rydde opp til slutt. Men er det da en motsigelse mellom å benytte smidige prosesser og komponentbasert utvikling?

## Howdan lykkes?

Programvarekomponentene bør være så "små" som mulig. Små klosser er lettere å gjenbruke enn store. En bør være nøye med å definere komponentene godt før de implementeres og tas i bruk. Særlig er det viktig at komponentenes grensesnitt defineres godt slik at komponentene kan bygges ut på sikt uten at bakoverkompatibilitet ødelegges. Dette krever at en har en definert strategi, gjerne støttet av et rammeverk, for hvordan komponenter skal kobles sammen. Utvidelse av komponentenes innmat, sammenstilling av komponentene til et større system, og uttesting av komponenter før de frigis for bruk i flere sammenhenger kan med fordel følge prinsippene til smidige prosesser.

## Hva gjør vi?

I våre simulatorer benytter vi smidige prosesser i utvikling av våre produktkomponenter, og i prosjekter hvor alle interessenter er enige om dette. Vår komponentbaserte utvikling baserer



seg på .NET teknologi som bidrar med teknologi som gjør det lettere å beholde grensesnittene stabile mens komponentene bygges ut. Videre baserer vi oss på løst koblede komponenter – dvs. komponenter som tåler at «samarbeidende komponenter» kan byttes ut eller til og med mangle.

Vi har to måter å "løskoble" på:

1. En som benytter et egenutviklet rammeverk for å bygge applikasjoner. Dette applikasjonsrammeverket bistår med en lettere måte å laste inn og konfigurere samarbeidende komponenter innenfor en applikasjon. Applikasjonen kan endres ved eksekvering (plug-and-play). Dette gjør at systemene raskt kan tilpasses kundens behov ved å benytte programvarekomponenter på en skalerbar og fleksibel måte.
2. Den andre måten som benyttes er for dynamisk datautveksling mellom komponenter. Komponentene kan kjøre på samme eller forskjellige datamaskiner i et nettverk. Datautveksling mellom komponentene baserer seg på publisering av og abonnering på objekter og interaksjoner slik som i HLA. En objekt modell (slik som FOM i HLA) definerer dataene som utveksles.

# PROTEUS

“Løst koblede byggeklosser = Forutsigbar gjennomføring med lav risiko!”

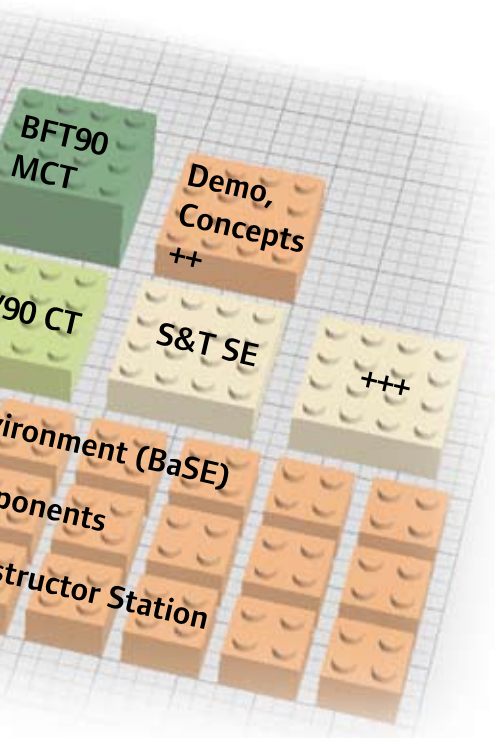


PROTEUS er en av de tidlige greske gudene fra havet, sønn av Poseidon. PROTEUS endrer sin egen form etter behov, er havets vismann og gjeter for alle havets dyr. Inspirert av den greske guden overfører KONGSBERG PROTEUS teknologien mange av disse egenskapene til trenerverden ved å være konfigurert til vekslende behov og omgivelser, samt som veileder og mentor for studentene.

Med den rikholdige samling løst koblede byggeklosser etter Lego-prinsippet, som KONGSBERG sin PROTEUS teknologi består av, har vi de siste årene gjennomført flere leveranseprosjekter ved gjenbruk av store og små klosser. Prosjektene leveres til rett tid, med rett kvalitet og til pris som avtalt med kunden, samt at de legger tilbake oppgraderte og nye klosser i samlingen.

Eksempler på dette er prosjektene for å levere Nansen Klassen Fregatt Trener (FT) til Sjøforsvaret (se link under) og Instruktørstasjon til ny Minejakt Sonar Simulator (MCMS) fra THALES Underwater Systems.

Kjernen i PROTEUS teknologien hos vår viktigste kunde på sjøsiden, Sjøforsvarets Skolesenter KNM Tordenskjold i Bergen, er PROTEUS Action Speed Tactical Trainer (ASTT)-funksjonaliteten. PROTEUS ASTT er vårt produkt for prosedyre-, kommunikasjon-, rolle- og samhandlings-trening for Sjøbaserte kamporganisasjoner, og tilbyr generiske operasjonsrom og fartøyer som kan samtrene internt i operasjonsrommet, og også flere fartøyer imellom.

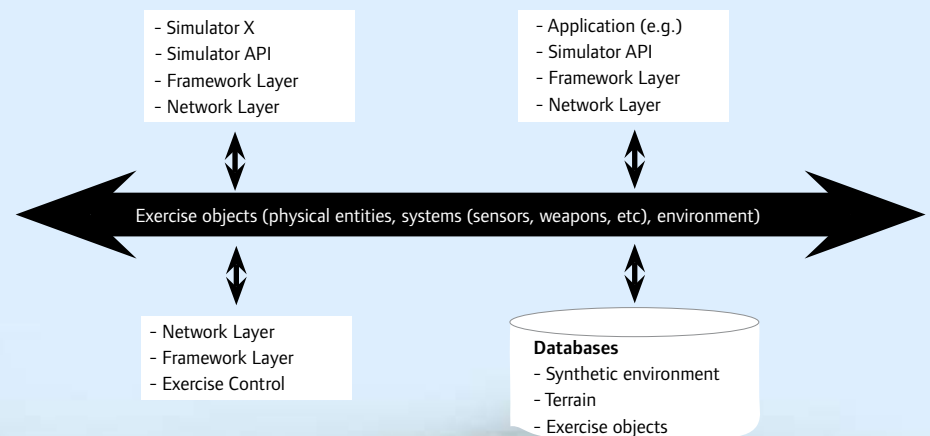


Applikasjonsrammeverket muliggjør fleksibel gjenbruk av små programvarekomponenter i mange sammenhenger, sammen med plugins som ivaretar særbehovene i forskjellige systemer. Løs kobling mellom komponenter i et distribuert system ivaretas av et HLA basert rammeverk for dynamisk datautveksling.

**Til slutt tilbake til spørsmålet om det er en motsigelse mellom å benytte smidige prosesser og komponentbasert utvikling?**

**Det er vel heller ja takk – begge deler.**

Så lenge en har fokus på godt definerte grensesnitt og en strategi for sammenkobling som støttes av gode rammeverk, da kan fordelene ved å benytte smidige prosesser utnyttes.



Ferdig Simulator X, Application



Link

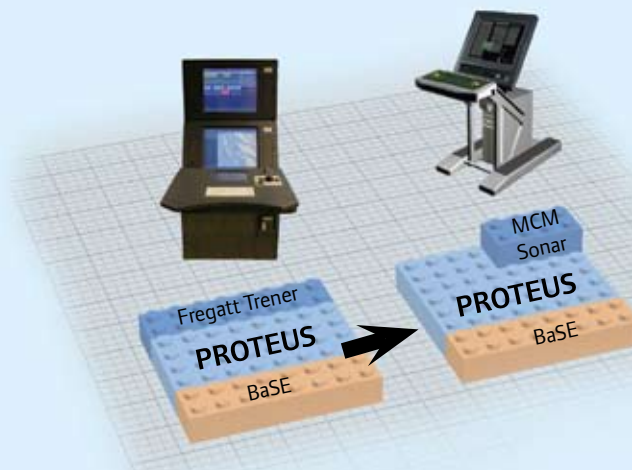
[www.mil.no/fregatter/start/article.jhtml?articleID=108730](http://www.mil.no/fregatter/start/article.jhtml?articleID=108730)

## PROTEUS simulorteknologien

### kan beskrives som:

- Ett nettverk og rammeverk som tillater spillobjektene (fartøyer, våpen, sensorer) og de syntetiske omgivelsene å kommunisere med de aktuelle simulatorene og applikasjonene (studentpanelene).
- Spillkontroll for start, pause og stopp av spillet, endring av spillet underveis, samt planlegging av nye spill (Instruktøren)
- Database som holder alle modeller av spillobjektene, syntetisk miljø og spillene (scenarioene)
- Simulatorene for hvert spillobjekt, som abonnerer på relevante opplysninger fra rammeverket.
- Applikasjoner for hvert spillobjekt, som mottar informasjon fra sin simulator over rammeverket. Applikasjonene er det studentene ser som paneler

Med andre ord meget STORE byggeklosser. Ved å koble disse løst har det vært mulig å benytte klossene fra PROTEUS simulorteknologien for å integrere eksterne systemer som FT og MCMS sømløst inn i PROTEUS. De største byggeklossene gjenbrukt i disse prosjektene dekker selvfølgelig all infrastrukturen som nettverk, rammeverket, basis instruktørstasjon, databaser osv. Ved at de samme byggeklossene benyttes og utvides betyr det at alle leveransene er fullstendig integrert i PROTEUS, og at de også følger med i utviklingen etter hvert som PROTEUS oppgraderes og utvides.



Begge de nevnte prosjektene har også utviklet nye store og små byggeklosser og MCMS som det siste prosjektet har igjen gjenbrukt (og forbedret) byggeklosser fra FT.

### Fregatt Trener:

- Bruk av studentpaneler for kommunikasjon med eksterne systemer åpner for samtrening med andre enheter.
- Utviklet KMessageFrame komponent for fleksibel meldingsbehandling mot ulike kommunikasjonsteknologier og typer.
- Utviklet basis Interfacepanel med XML logger, for instruktørkontroll av kommunikasjon og status på interfascene mot de eksterne systemene.

### Instruktørstasjon til Minejakt Sonar Simulator:

- Kraftige utvidelser på minefunksjonalitet

og mer detaljerte scenariobeskrivelser som bunntyper, vegetasjon osv.

- Gjenbruk av KMessageFrame komponent med utvidelser på meldingstørrelse.
- Ny generell komponent for å la instruktøren kontrollere studentkontrollerte enheter.
- Rollestyrt konfigurasjon av instruktørstasjon.

Det har i tillegg vært gjort mye gjenbruk av de litt mindre byggeklossene.

**Fregatt Trener og Minejakt Sonar Simulator er gode eksempler på at vi ved å designe systemer som løst koblede Legoklosser, og å gjenbruke disse, klarer å levere med lav risiko for kunden og oss selv.**



USA er et av de viktigste markedene for PROTECTOR og vi har hatt stor suksess i mange år. August 2007 fikk vi CROWS II kontrakten med levering av opptil 6500 våpenstasjoner. Dette er med på befeste KONGSBERG's posisjon som verdensledende produsent av våpenstyringssystemer. KONGSBERG har også stor suksess med leveranser av simulatorsystemer for PROTECTOR, både i og utenfor USA. I tillegg til skyte-

simulator har vi videreutviklet systemet til å inneholde e-læring og interaktive verktøy for vedlikeholdstrening.

Interessen og pågangen på messen var stor, spesielt fra US Army, men også andre internasjonale forsvar. Neste AUSA utstilling hvor hele PROTECTOR familien er representert er AUSA Winter Symposium, FL 25-27. februar 09.

# AUSA

6-8.oktober 2008 gikk AUSA Annual Meeting av stabelen i Washington D.C., USA. KONGSBERG stilte ut flere modeller fra sin familie av PROTECTOR Remote Weapon Stations, inkludert PROTECTOR Training Systems.

