NYHETSBLAD FÖR SESAM
Försvarssektorns Adaintressenters Användargrupp för Software Engineering

RENDEZVOUS

Nr 3 december 2001

Innehåll

Försvarssektorns Adaintressenters Användargrupp för Software Engineering

SESAM

Vad är SESAM?

SESAM har tillkommit för att organisera och stimulera samarbete och samverkan inom programvaruområdet mellan försvarsindustrin, FMV och FOA.

Det avtalsfästa syftet med SESAM är ”att genom organiserat samarbete mellan användargruppens medlemmar främja tillförlitlighet och effektivitet i utveckling och vidmakthållande av programvarusystem i Ada inom försvarssektorn”. Inom ramen härför skall SESAM även anpassa, profilera och förnya sin verksamhet med hänsyn till ändrade tekniska och andra omständigheter av betydelse för intresseområdet.

Följande kommer att ske under den närmaste 2-3-årsperioden.

1. SESAM skall allmänt verka för att sprida information om faktorer som påverkar möjligheterna till tillförlitlig och effektiv utveckling och vidmakthållande av programvarusystem. Särskilt skall härvid Adas betydelse i sammanhanget klargöras.

2. SESAM skall i sin verksamhet fortlöpande bevaka möjligheterna att samla, skapa och sprida information om objektiva mät- och andra resultat och erfarenheter vunna vid användning av ”software engineering”-principer och Ada.

3. SESAM behandlar tillvägagångsättet vid utveckling och vidmakthållande av programsystem. Implicit i detta ligger givetvis att använda processer skall tillförsäkra de resulterande produkterna efterfrågade egenskaper. Produktegenskaper som påverkas av processerna är därför av primärt intresse att bevaka i SESAMs verksamhet.

4. SESAM skall i sin verksamhet fästa stor vikt vid att underlätta samexistens mellan Ada-program och programvara skriven i andra språk. Speciellt skall aspekter vid användning av COTS beaktas.

5. SESAM skall där så är möjligt sätta konkretiserade och mätbara mål för sin verksamhet under avgränsade tidsperioder.

SESAM styrs av ett Råd med representanter för gruppens medlemmar. Rådet har till sin hjälp ett Verkställande Utskott (VU) och ett sekretariat.

Rådets ordförande är Claes Wadsten, AerotechTelub, tel 013-231652 .

VU

Andersson Tommy, Ericsson Microwave Systems AB
tommy.andersson@emw.ericsson.se
Bengtsson Christopher, FMV
chben@fmv.se
Carlsson Ingemar, adjungerad
ingemar.carlsson@mbox2.swipnet.se
Ekman Mats, Saab Aerospace AB
mats.ekman@saab.se
Johansson Billy, CelsiusTech Electronics AB
bijo@infomatics.saab.se
Merkell Curt, Saab Bofors Dynamics AB
curt.merkell@dynamics.saab.se
Wadsten Claes, AerotechTelub AB
claes.wadsten@aerotechtelub.se

Arbetet utförs i två arbetsgrupper:

Ag Metodik
Håkan Edler, CTH/Datorteknik
edler@ce.chalmers.se
Ag Teknik
Lars Asplund, Uppsala Universitet
asplund@docs.uu.se

Vilka kan vara med i SESAM?

Medlemmarna i SESAM är svenska företag, organisationer och myndigheter (förvaltningar, utbildningsinstitutioner etc) med anknytning till försvarssektorn. Medlemmarna indelas i följande kategorier

- ordinarie medlemmar
- arbetsgruppsmedlemmar
- informationsmedlemmar.

Enskild person kan endast komma ifråga som informationsmedlem.

Inträde i SESAM

För samtliga medlemskategorier gäller att inträde beslutas av Rådet.

För inträde som ordinarie- och arbetsgruppsmedlem krävs status som leverantör till FMV. Dessutom krävs en skriftlig förbindelse att uppfylla åtagande som ordinarie- och arbetsgruppsmedlem.

För inträde som informationsmedlem (erhåller endast informationsbladet) krävs status som leverantör till FMV eller status som myndighet inom totalförsvaret. Rådet kan emellertid anta annan part som informationsmedlem.

För ansökan om medlemskap i SESAM vänd er till sekretariatet.

SESAM-Sekretariatet
AerotechTelub AB
c/o Kåsjös Kontor
Ytterspåret 14
187 54 TÄBY

Ordföranden har ordet

Vi början nu närmar oss slutet av år 2001. Året började inte på det sätt jag önskade vilket jag berört i tidigare ledare. De problem vi hade då har dock trots att det tog viss tid rättats till. Trots att mycket energi fick läggas ned på formaliteter så kan vi idag se tillbaka på att vi även i år kan visa upp verksamheter som varit till nytta för SESAM:s medlemmar.

Vi har påbörjat ett antal intressanta ”mikroprojekt” som jag hoppas kommer att slutföras under nästa år.
Jag vill speciellt nämna det mikroprojekt vi kallat ”drömverktyget” som också resulterade ett seminarie med inbjudna verktygsleverantörer. Genom för- och efterarbete kring seminariet har vi fått en bedömning av olika utvecklingsverktyg. Resultaten var så pass intressanta och skapade ytterligare frågor så VU hoppas att detta projekt kan fortsätta även under 2002 dock med nya direktiv.
Kanske kan SESAM vara en organisation som lyckas få igenom förbättringar hos de olika verktygen för programutveckling. I SESAM ingår de flesta stora företagen som utvecklar programvara åt svenska försvaret.

Förutom detta speciella seminarie, genomfördes också det traditionella höstseminariet. Ämnet för året var Arkitektur i System av System, förr och nu. Seminariet var välbesökt in till sista plats vilket gladde mig.
Som ett resultat från detta seminare tar SESAM också fram en CD i begränsad upplaga med underlag från seminariet kompletterat med artiklar och dokumentation i ämnet. CD:n omfattar cirka 1000 sidor text och innehåller flera artiklar från utländska kända personer i ämnet. Då många av dessa ställer upp ideellt så har vi lovat att upplagan är begränsad.
Delar av detta material finns med i detta nummer av RENDEZVOUS.
Jag vill här passa på att lyfta fram det arbete ”våran” Ingmar Carlsson lagt ned för att detta skall bli förverkligat.

Jag vill också framföra mitt tack till vårt sekretariat, VU, medlemmar och vissa personer på FMV och Försvarsmakten som gjort det möjligt att även 2001 fått ett innehåll i SESAM:s historia.

Vi lämnar nu 2001 bakom oss och kommer igen under 2002 med nya friska tag efter en välbehövlig julvila.

God jul och gott nytt år på Er alla

Med hälsningar

Claes Wadsten
Ordf. SESAM

Gunnar Lindqvists diskussionsinledning från SESAM-konferensen

ARKITEKTUR för MILITÄRA LEDNINGSSYSTEM

Ämnet för dagen är ju Arkitektur då, nu och sedan. I inledningen till den kommande diskussionen vill jag koncentrera mig på ledningssystemen inom försvaret.

Den vedertagna betydelsen av ordet *arkitektur* torde vara byggnadskonst. (Arkitekt kan härledas från de två grekiska orden arki och tekton som betyder förnämlig respektive timmerman.) Till följd av att språken inte längre tycks utvecklas i samma takt som behovet av att beskriva nya upptäckter, fenomen och produkter tvingas vi att använda gamla ord på sätt som inte avsågs från början. Det gör språket oklart och leder ofta till missförstånd. Exempel på sådana ord är system, design, nätverk, patterns etc. Egentligen borde man kunna ”kvittera ut” ett helt nytt ord från någon global språkinstans när ett behov bevisligen uppstått. Men vi får nog tyvärr leva med dessa språkliga problem även i fortsättningen

Från att bara betyda byggnadskonst har ordet arkitektur inom datatekniken kommit att först betyda en dators inre uppbyggnad, såväl hård- som mjukvara, för att nu mer övergå till att omfatta hela strukturen av ett system där datorer ingår i mer eller mindre omfattning. Det aktuella systemet kan vara alltifrån ett mycket överordnat sådant till ett mindre delsystem inom ett större system. Arkitektur i vidare bemärkelse är således inte bara IT-system och än mindre bara mjukvara.

Vad är då ett *militärt ledningssystem*? Jag skulle vilja definiera det som ett system vars uppgifter är att skapa och överföra information om motståndarens och mina egna förbands tillstånd, bearbeta och sammanställa information från olika källor, presentera information för beslutsfattare, ge dessa stöd för beslut, överföra order till verkställande enheter såsom informationskällor, kommunikationssystemens ledning och främst till ledare av olika luft-, sjö, och lantstridskrafter.

Ledningssystem finns på olika nivåer alltifrån högkvarterets strategiska, operativa nivå ned till taktisk ledning av enskilda förband och enheter. Som exempel på den operativa nivån kan ledningen av luftförsvaret från ett flygkommando gälla. Ledning från luftförsvarets stridsledningscentraler, STRIC och från kustkorvetterna samt ledning inom ett jaktstridsförband kan exemplifiera ledningssystem på den taktiska nivån liksom det system en förare använder sig av i ett Gripenflygplan. Om vi i fortsättningen kommer att kunna samarbeta med Nato-förband tillkommer ytterligare en nivå.

Det totala ledningssystemet består alltså av olika ledningssystem på olika nivåer. Alla dessa måste kommunicera med varandra medelst kommunikationssystem. Systemen kan vara mer eller mindre beroende av varandra. Ledningssystemen kan också vara integrerade i olika stridsförband, med sina arkitekturer, på olika nivåer. I kommande system kommer möjligheterna till information vara mycket stor; naturligtvis under förutsättning av robusta kommunikationssystem. Det gäller dock fortfarande att utnyttja tekniken på rätt sätt, bland annat genom att fastställa vem som har rätt till information och vem som har rätt att utfärda order till olika förband med hänsyn till erforderlig delegering av ansvar. Detta är inte ett tekniskt problem utan snarare ett organisatoriskt.

Man kan skilja på ledningssystem på övergripande nivå där besluten oftast inte behöver fattas sekundsnabbt och stridsledningssystem som arbetar mer eller mindre i realtime. I det senare fallet kan ledningssystemet snarast karakteriseras som ett servosystem. Ett exempel på detta är luftförsvar med jaktflygplan. Länkarna i servokedjan utgörs av motståndarflygplanen, mark- och flygburna radarstationer, stridsledningscentralen med sina datorer för beräknande av optimala flygbanor samt jaktflygplanen. När så jaktflygplanen närmar sig målen bildas en ny servokedja bestående av mål, jaktflygplanets radar, dess styrsystem etc.

I Sverige måste Flygvapnet anses ha haft en ledande roll vid uppbyggnad av ledningssystem. Även med internationella mått har man varit på framkanten. Följande exempel kan nämnas: Flygplan J35 var antagligen det första datastridsledda jaktflygplanet i världen. Det var STRIL60 med sina datorer och datalänken upp till flygplanen som möjliggjorde detta. AJ37 var det andra krigsflygplanet i världen som fick en dator

inbyggd i sitt avioniksystem. Ordet dator var inte uppfunnit på den tiden, dvs tidigt 1960-tal. AJ37's dator fick därför benämningen central kalkylator CK37. Den har använts ända in i våra dagar. JA37 modifierades tidigt med en sk jaktlänk, innebärande att inte bara STRIL kunde skicka data till jaktflygplanen utan att dessa kunde sända data ner till STRIL och sinsemellan, vilket var ett stort steg framåt avseende jaktflygplanens effektivitet. Försvarets telenät byggdes också upp från 1950-talet och framåt i samarbete med Televerket och består av olika radio-, radiolänk- och kabelförbindelser i vad som nu så populärt kallas nätverk.

Även beträffande administrativa datasystem var Flygvapnet tidigt ute, till exempel reservdelsförsörjningssystemet.

Beträffande försvarets olika sk stabsledningssystem får man nog säga att födslovåndorna varit stora. Det har nog inte så mycket varit tekniska problem utan problem med ledning av framtagningen av dessa ledningssystem.

Men i det stora hela måste man anse att många effektiva system skapats till rimliga kostnader inom vårt försvar. Orsaken till detta var tillgången på kvalificerad personal både vid industrin och i förvaltningen samt en långsiktande forskning och framsyn i övrigt.

Men nu gäller det framtiden. Vi måste bygga vidare på existerande system, men givetvis också skapa nya och därvid utnyttja ny teknik på ett rationellt sätt för att förändra vårt försvar så att det även kan möta nytillkomna hot.

Varje hot kräver en viss typ av försvar sammansatt av olika stridsförband (eller motsvarande). Dessa måste i sin tur bestå av personal, organisation och materielsystem. Dessa förband kan ofta utnyttjas i flera hotsituationer. Det ekonomipolitiska läget i landet innebär att vi inte anses ha råd att möta alla hotsituationer med tillräcklig kvantitet av försvarsresurser. En prioritering baserad på bedömd sannolikhet för olika hot måste göras. Exempelvis anser man nu att risken av en stort upplagd invasion mot vårt land är låg. Just nu, borde man tillägga. Om denna prioritering visar sig felaktig tror de styrande politikerna att det finns tid att rusta upp. Det fanns det inte förra gången.

Av säkerhetspolitiska skäl bör förändringar ske inom vårt försvar. Det är nog alla överens om. Men hur det skall ske finns det delade meningar om, även om regering, ÖB och högkvarteret gett ut riktlinjer.

Vad är då det nya? Till följd av Sovjetunionens och Warszawapaktens upplösning och det utökade säkerhetspolitiska samarbetet inom EU har tre väsentliga ändringar skett:

* Dels har det säkerhetspolitiska läget förbättrats i vår närhet, men latenta risker för konflikter finns dock kvar.
* Dels har det nu blivit möjligt att samverka med andra demokratiska nationers försvarsorganisationer på ett helt annat och omfattande sätt än som var förenligt med vår utrikespolitik under det kalla kriget. Det är mycket positivt och bidrar till vår säkerhet och till bevarande av freden i vår närzon. Men det bör dock påpekas att vi inte tagit steget fullt ut mot ett solidariskt ansvar, utan fortfarande står utanför genom vår alliansfria politik.
* Dels innebär denna samverkan en viss integration med exempelvis Nato's ledningssystem och kommunikationssystem samt krav på sk interoperability mellan våra förband och materielsystem och Nato's motsvarande enheter.

Ovanpå detta kommer att nya hot, som förut betraktades mer hypotetiska, nyligen har visat sig högst reella.

Även om vi på många områden, särskilt materielområden, raskt skulle vilja skrota befintliga system och ständigt omsätta dem med nya måste vi inse att detta varken effektivitetsmässigt, ekonomix˘t eller praktiskt är vare sig optimalt eller möjligt. Vi har ett antal materielsystem och ledningssystem som är effektiva i flera hotsituationer. I vissa fall krävs nya. *Men vi måste alltid leva med det faktum att försvarets materiel alltid kommer att förnyas successivt.* Även de nya måste konstrueras så att de kan samverka med existerande system. Detta är ett oavvisligt krav och ett mycket svårt krav att ta hänsyn till vid nykonstruktioner. En annan försvårande faktor är att materielen måste kunna underhållas och i vissa fall uppgraderas under flera decennier, detta trots IT-branschens tendens till generationsväxlingar med kortare tidsintervall.

Dagsläget är att vi beträffande verkanssystem (flygplan , stridsvagnar etc) och stridsledningssystem har ett ganska bra läge, i varje fall kvalitativt. Däremot krävs en snar förnyelse inom stabsledningssystemen. Kommunikationssystemen, de som förr kallades sambandssystem, måste byggas på och framförallt göras mer stryktåliga.

Försvaret skall alltså genomgå en ganska stor omstrukturering, eller för att tala med dagens språk: Arkitekturen måste ändras. Det erfordras en bättre balans mellan verkansenheter, stridsförbanden å ena sidan och informations- och ledningssystemen å andra sidan. Detta måste förberedas genom optimeringsstudier. Dessa studier bör baseras på hotbilden, dvs olika konfliktsituationer och uppgifter för försvaret, möjlig teknisk utveckling samt ekonomiska ramar. Det mest osäkra här är naturligtvis hotbilden. Vi kan ställas inför konfliktsituationer i vår närhet, dvs hot mot oss eller våra grannar, där solidaritet krävs av oss, eller mer i periferin på större avstånd eller t.o.m inom vårt land.

Studierna måste bedrivas *top-down på en övergripande nivå* för att få balans mellan försvarets olika delar, fördelat på de olika tänkbara konfliktsituationerna. Men samtidigt måste idéer angående nya materielsystem skapas, grundade på den möjliga *tekniska utvecklingen*. Det måste ske en iteration mellan de övergripande systemstudierna och studierna av materielsystem. Några exempel: Luftförsvarets resurser och kostnader måste vägas mot fjärrstridsförbandens. Inom luftförsvaret måste balans skapas mellan markbundna och luftburna radarstationer, stridsledningscentraler, jaktflygplanprestanda, vapenprestanda samt luftvärnssystem. En komplikation vid avvägningen, fast operativt en stor fördel, är att flera materielsystem är användbara för olika försvarsuppgifter. Ett fall är flygplan 39 Gripen, som ingår både i luftförsvaret och fjärrstridsförbanden

Det finns olika verktyg för att arbeta med detta arkitekturarbete. Modellering och simulering har använts ända sedan 1950-talet inom försvaret. Datorutvecklingen har betytt att simuleringar med tiden har kunnat gjorts mer realistiska och omfattande. Goda resultat har erhållits på materielsystemnivå. Redan för JA37 användes hybridsimuleringar av jaktuppdrag där Stril60 eller en simulator för detta system kopplades ihop med JA37-flygplan eller dess systemsimulator. På mer övergripande nivåer såsom det militära försvaret eller totalförsvaret blir modellerna mer svårhanterliga. De tar lång tid att utveckla, för att inte tala om att verifiera. Modellbyggarna ställs inför valet att åstadkomma mindre relevanta modeller eller att arbeta mycket långsiktigt och kostsamt. Det har varit och kommer att vara en svår avvägning även i framtiden. Inte desto mindre är modellering och simulering ett av de kraftigaste verktyg vi har för optimering på alla nivåer.

Inom det svenska försvaret har alltså optimeringsstudier bedrivits i ganska stor omfattning. Ledningen av dem har inte alltid varit rationell. Framförallt har man inte haft någon enhet som kontinuerligt kunnat ägna sig åt detta arbete och därvid kunnat samla på sig successivt ökad erfarenhet.

Men simulering kan utnyttjas inte bara under studier av framtida system utan även under ett systems hela livslängd. När man väl valt ett system, definierat i stora drag, måste man optimera dess ingående delsystem mot varandra samt specificera mätbara kontraktskrav. Simulering utnyttjas också vid utprovning och modifieringar, taktisk optimering, utbildning och i framtiden förhoppningsvis även som beslutsstöd.

Rent administrativt kan man strukturera arbetet med ny arkitektur enligt det mönster som nu skapas inom USA's försvar. Flera styrande dokument har framkommit såsom JTA (Joint Technical Architecture), C4ISR (Command, Control, Communication, Computation, Surveillance, Reconnaisance), TRM (Technical Reference Modell), ADL (Architecture Description Language) m fl. Här delar man in försvarsuppgifterna i olika domäner och underdomäner. Inom dessa och i vissa fall mellan dem förväntar man sig göra avvägda strukturer, arkitekturer. Systemen antages byggas upp i noder med definierade samfunktioner. Gränsskikt inom ett system mellan noderna och inom noderna samt till samverkande system förväntas definieras på ett standardiserat sätt inte minst när det gäller mjukvaran.

Detta är väl egentligen inget nytt sätt, utan mer ett sätt att bättre ordna sitt arbete. På lägre nivåer kan dock viktiga standarder komma fram som kan bli styrande även för oss, inte minst om vi beaktar kraven på interoperability med Nato-förbanden.

Den nya tekniken ger onekligen stora möjligheter. Bättre möjligheter för att kunna utnyttja information från flera datakällor finns nu, datafusion. Vi kan bygga kommunikationssystem med mycket stor kapacitet för att överföra information.

Säkerheten måste dock beaktas bättre i fortsättningen. Det krävs åtgärder på alla nivåer mot intrång, störning, förstöring (data och hårdvara), vilseledning etc. Det kanske verkar tjatigt att påpeka detta, men erfarenheten visar att nya metoder att störa och

förstöra ständigt utvecklas. Vi måste skaffa mer robusta system och införa en högre grad av redundans. Personligen tycker jag att problemet delvis också är av organisatorisk karaktär. Teknisk kan informationsflödet öka enormt, men borde vi inte i stället försöka begränsa en del krav?

Ett annat område som jag uppfattar som bekymmersamt är möjligheten till *beslutsstöd.* Det har pratats mycket om detta och articifiell intelligens. Men få användbara resultat har kommit fram. Eftersom den mänskliga hjärnan inte mätbart ändrat sig på flera tusen år borde en trängd beslutsfattare verkligen få bättre stöttning i sina avgöranden när han nu riskerar att formligen översköljas med information.

Vilka resurser krävs det då? För att kunna skapa och vidmakthålla ledningssystem, och även annan materiel, på ett effektivt sätt inom försvaret krävs alltså att grundstrukturen, arkitekturen kan specificeras tillräckligt bra med hänsyn till hotbild, tekniska möjligheter, kostnader, samverkan med andra system och försvarsmakter. För att sedan skapa systemen erfordras skicklig personal vid vår industri men även inom förvaltningen och försvaret i övrigt. Man får hoppas att åtminstone IT-delen av vår försvarsindustri fortfarande kan upprätthålla en tillräcklig kapacitet för att till stor del utveckla och producera dessa viktiga delar av vårt försvar. IT-industrin finansieras ju till största delen av civila kunder varför den inte blir så beroende av militära kunder som den övriga försvarsindustrin.

Utökad samverkan med andra nationers försvar och minskad kapacitet hos vår försvarsindustri medför att vi måste samarbete med andra länders industri vid framtagning av försvarsmateriel. Frågan om när samarbete är mer lönsamt än egen tillverkning eller direktköp blir nu mer aktuell. Samarbete har sina för- och nackdelar för försvaret och för vår industri. Det är ej möjligt att vidare utveckla detta spörsmål här. Frågan är mycket komplicerad.

Under många gånger har relationerna mellan försvarets förvaltningsmyndigheter och försvarsindustrin diskuterats. Pendeln har svängt mellan ett starkt ömsesidigt beroende med långsiktiga anskaffningsplaner till att endast de gällande kortsiktiga beställningarna varit bindande för parterna. Vid utformningen av de olika arkitekturerna på de olika nivåerna måste ekonomiska och andra kommersiella hänsyn tas vid utformningen och specificerandet av nya system. För att en leverantör skall kunna ges ansvar för en viss produkt, hårdvara likväl som mjukvara, måste hans åtagande kunna definieras och resultaten verifieras. Att arbeta i sk nätverk synes praktiskt när det gäller forskning. Men det verkar inte särskilt realistiskt för utveckling och tillverkning, utan mer traditionella hierarkiska organisationer blir nog nödvändigt även i fortsättningen om prestanda, tid och kostnader skall innehållas.

Inom försvaret vill man nu lägga mer vikt vid informations- och ledningssystem än förut, vilket är rätt. Märk dock att det fortfarande gäller att ha rätt balans mellan dessa och de sk verkanssystemen. Hur stora medel finns då tillgängliga inom en viss framtid? Det borde på något sätt, åtminstone grovt, antydas för aktuell industri vilken storleksordning det rör sig om.

Slutledning: Arkitektur, strukturering behövs på alla systemnivåer inom försvaret inklusive ledningssystemen. Struktureringen och optimeringen av försvaret är en viktig utgångspunkt för utformning av ledningssystem.

En evolutionär uppbyggnad av ledningssystemen är nödvändig på grund av den successiva materielomsättningen inom försvaret. Försvarssystem inom IT-sektorn har en längre livslängd än vad som åtminstone tills vidare gäller den civila marknaden.

Den tekniska utvecklingen skapar stora möjligheter. Ledningssystemen måste dock byggas med stor vikt på att de skall vara robusta och stryktåliga. Härvid är också organisatorisk strukturering av informationen (behörighet, begränsningar etc) viktig.

Olika former av beslutsstöd bör om möjligt skapas.

Nya krav ställs på samverkan med andra nationers försvar, vilket påverkar utformningen av våra försvarsmaterielsystem.

Utveckling och tillverkning av ledningssystem bör åtminstone till viss del ske inom landet.

Speciell uppmärksamhet bör ägnas åt de erforderliga personella resurserna inom industrin och försvaret.

Gunnar Lindqvist

SIGAda 2001, Bloomington MN, 30 okt- 4 nov

Den årliga SIGAda-konferensen hölls i år i Bloomington, strax utanför Minneapolis i Minnesota. Självfallet lade händelserna den 11 september en viss sordin på arrangemanget, men konferensen hölls ändå som planerat. Antalet deltagare såg till en början lovande ut, +34% i förhållande till förra årets föranmälningar fram till början av september, men därefter kom inga fler nyanmälningar, däremot åtskilliga avbokningar så att den slutliga deltagarförteckningen omfattade 99 namn (mot c:a 150 året före). Endast 6 deltagare kom från Europa.

Thunderbird Hotel & Convention Center ligger mycket strategiskt placerat, bara några få miles från den internationella flygplatsen och ett kvarter från ”Mall of America”, givetvis USA’s största! Det genomgående temat vad gäller inredningen på hotellet är indiankultur, som det finns mycket av i Minnesota. Det svenska arvet i delstaten är däremot inte särskilt tydligt.

Söndagen och måndagen av konferensveckan ägnades åt olika tutorials, och den egentliga konferensen startade först på tisdagen med en Keynote Address av Scott Edgerton från United Defense L.P. som talade om ”Architecture-based Software Development on the Crusader Program”. Detta utvecklingsprogram bygger på en stridsvagnsplattform som kan ha många olika konfigurationer. Ett mått på framgången var att 80% av programvaran för diagnostik och prognostisering kunde återanvändas.

Peter Amey från Praxis Critical Systems hade sedan två föreläsningar, ”A Language for Systems not Just Software” och ”Logic v/s Magic”. Praxis är ju det företag som marknadsför SPARK, ett formellt kommentarspråk till Ada, ursprungligen utvecklat för statisk analys och formell verifiering. Genom en utvidgning av syntaxen för dessa formella kommentarer (annotations) har man gjort det möjligt att beskriva hela system och inte bara implementera programvara.
Ref: http://www.sparkada.com/

Nedanstående korta referat omfattar endast de presentationer jag själv var närvarande vid, eftersom många sessioner avhandlades i parallella salar. Första sessionen handlade om återanvändning och produktarkitekturer. Joel Sherrill från On-Line Application Research Corp. presenterade ett framgångsrikt projekt. Man hade samordnat programvara för skrivare, kopiatorer och faxar så att en betydande återanvändningsgrad hade uppnåtts.

Därefter var det ett svenskt inslag, då undertecknad presenterade ”Ship System 2000, a Stable Architecture under Continuous Evolution” – en rapport från mer än 15 års arbete med återanvändning av Ada-program.

Kenneth L. Ehresman från U.S. Navy presenterade sitt ”Electronic Maneuvering Board and Dead Reconing Tracer Decision Aid for the Officer of the Deck”. Utgående från beslutsloopen: Observe – Orient – Decide – Act, presenterade han en design baserad på UML och Use-Case diagram som var implementerad med GtkAda och GNAT. Därmed kunde man visa att systemet var fullt portabelt mellan Linux Redhat 7.0 och Windows 2000.

Onsdagen började med en spännande och underhållande föreläsning betitlad ”Confessions of an Academic Ada Zealot” som Martin C. Carlisle bjöd på. Hans berättelse handlade om hur han har gått från en initialt skeptisk inställning till Ada, via fasen då han imponerades av språkets kvaliteter, fram till nutid då han är en av de sista och kanske den mest fanatiske Ada-förespråkaren på U.S. Air Force Academy. Därefter visade han hur man programmerar Legorotbotar i Ada under rubriken ”Teaching Computer Science with Robotics Using Ada/ Mindstorms 2.0”. Ett mycket uppskattat och engagerande sätt att lära studenterna realtids programmering i Ada med omedelbar feed-back.
Ref.: http://www.usafa.af.mil/dfcs/adamindstorms.htm

”Mr. Ada95”, dvs. Tucker Taft, höll ett föredrag om ”Using Ada95 in a Compiler Course”. De byggstenar som kom med Ada95 (t.ex. tagged types) har ju gjort att Ada mycket väl fungerar för att skriva kompilatorer. Och Tucker har tillämpat detta med framgång i sina kompilator-kurser.
”Fixing Software Before it Breaks” var titeln på Tucker’s följande Keynote Address. Där lade han fram sin tes: ”Alla runtime-fel går att ta förebygga och ta hand om under kompileringen!” Men även om detta är något

av en utopi, (eftersom det kräver en mycket mer rigorös kontroll av syntax och semantik) har ju Ada kommit betydligt längre på den vägen än de C-baserade språken. En praktiskt hanterbar avvägning blir sannolikt den naturliga kompromissen med nuvarande teknologinivå, men i framtiden kanske...

Torsdagen inleddes utdelandet av Ada-statyetter till medarbetarna i SIGAda. Därefter följde två paneldebatter: "Ada Experiences" och "The Making of ISO/IEC 8652: Ada 2005". Den i mitt tycke mest intressanta diskussionen var vad som kommer ut av nästa Ada-uppdatering. Panelendebatten leddes av Erhard Plöderer och panelen bestod av Randy Brukardt (RR Software), Alan Burns (Univ. of York), Pascal Leroy (Rational, France), Jim Moore (Mitre Corp.) och Tucker Taft (AverCom Corp.) Inom ISO/IEC JTC1/SC22 WG9 arbetar nu en Ada Rapporteur Group (ARG) som är i färd med att definiera nästa Ada-standard. En trevlig detalj som borde uppskattas av alla C++ programmerare är att man förhoppningsvis kommer att kunna skriva procedur-anrop på "objekt-stil", dvs. med Object.Method(Params); som alternativ till Ada-stilen med Method(Object, Params); Detta kallade Tucker lite nedlåtande för "syntaktiskt socker".

Varje morgon såg vi konferensdeltagare fram emot ett nytt nummer av den intressanta tidningen Ada-WOW (Window On the World) som Hal Hart och Ann Brandon jobbade hårt med varje natt. Där kunde vi läsa om föregående dags händelser på ett informativt och inspirerande sätt, illustrerat med bilder från konferensen.

Utställare under konferensen var Aonix, Rational, Top Graph'X, Green Hill, Daimler Chrysler AG SW, Praxis och Ada Core Technologies, totalt 8 st. Dessvärre blev två ytterligare anmälda utställare förhindrade att delta på grund av oroligheterna som följde på terrorattackerna i New York och Washington DC.

Under en SIGAda-vecka försiggår, förutom utbildningar och föreläsningar, även ett antal kommittémöten, styrelsemöten och BOF (Bird-Of-a-Feather)-sessions, där man kunde snappa upp de senaste trenderna inom Ada-området. Fast det mest betydelsefulla är nog ändå de personliga kontakterna som knyts under konferensen. För mig var det även ett glatt återseende av en gammal bekant som jag inte träffat på 9 år! En SIGAda-konferens är väl värd att åka på, och jag fick i alla fall nytt hopp om en framtid med Ada 2005.
Mer info, liksom dokumentationen till de flesta föredragen, finner du givetvis på hemsidan: http://www.acm.org/sigada/conf/sigada2001/

Rei Stråhle
SaabTech Systems AB
S:t Olofsgatan 9 A
753 21 Uppsala

Tel: +46 8 5808 7124
Fax: +46 8 5808 7260
Email: rest@systems.saab.se

Inga-Lill Bratteby-Ribbing, FMV, återkommer här med en redogörelse för det aktuella läget i FOTA P12 och berörda aktiviteter.

FoTA P12: Överföring till industrin av programvaruteknik för säkerhetskritiska system

Ett riksdagsbeslut togs 1996 om särskilda medel för ett Forsknings och Teknikutvecklingsprogram avsett att stärka svensk industris förmåga att stödja ett anpassningsförsvar (FoTA). Ett antal teknikutvecklingsprojekt inom system- och programvaruteknik definierades 1998 med avsikt att insatser och resultat skulle komma flera företag tillgodo. För att underlätta utbytet startades 1999 ytterligare ett projekt, FoTA P12, med uppgift dels att interagera med de huvudprojekt inom FoTA, som behandlar frågor av intresse vid utveckling av säkerhetskritisk programvara, dels att vara diskussionspartner i FMV:s arbete med Handbok för Programvara i säkerhetskritiska tillämpningar (H ProgSäk):

- **FoTA P3:** COTS och objektorientering som bas för konstruktion av realtids-applikationer.
- **FoTA P4:** Patterns och komponent-återanvändning.
- **FoTA P7:** COTS-produkter i militära lednings- och informationssystem
- **FoTA P9:** Experimentell verifiering av feltolerans.
- **FoTA P10:** Projekt MANA – Ett run-time system för säkerhetskritiska komplexa system (kompletteringsprojekt till NUTEK Komplexa Tekniska System "A Run-Time System for Safety Critical Complex Systems").
- **FoTA P11:** Formalisering, analys och hantering av krav på säkerhetskritiska system
- **H ProgSäk**: Handbok för programvara i säkerhetskritiska tillämpningar.

Dessa projekt befinner sig i olika stadier:

- **P3** har beviljats förlängning till 1 juli 2002. Mätningar av konstruktionsfall implementerade i Java och C++ i WNT och Linuxmiljö är under avslutning på EMW, SaabTech och Kockums (Ada och VxWorks har utgått i o m att Saab lämnat projektet). Nya konstruktionsfall inriktade mot RMA, DBA och LedsystT:s behov håller på att tas fram för mätningar i distribuerade nätverk.
- **P4** med deltagare från AerotechTelub, Chalmers, EMW, S&T, Växjö univ samt Colorado Techn. Univ. (prof. Bo Sandén) är under avslutning. Ett antal studier, seminarie- och kursdagar har genomförts. En egen hemsida har upprättats (www.st.se/patterns). Slutrapportering ges den 6 december kl 9.30-13.00 (prel. lokal: FMV, Filmsal C, se P4:s hemsida).
- **P7** är ett redan avslutat studieprojekt avrapporterad vid SESAM:s oktober-möte år 2000.
- **P9** är f.n. vilande inför den sista 1-årsetappen. Första etappens fyra arbetspaket med HiSafe:s uppbyggnad av en experimentmiljö och prov av felinjiceringsteknik på programvara i MACS-datorn hos EMW och Saab har utförts med givande resultat. P12:s teknikprov på två vitt skilda tillämpningar inom S&T och AerotechTelub har också gett nya, positiva erfarenheter. Inför etapp 2 övervägs därför två alternativ: att fortsätta enl. ursprunglig plan (dvs att stärka robustheten hos svaga delar och köra om felinjiceringarna), eller att utvidga intressentgruppen för att efter generalisering av tekniken använda den på nya tillämpningar. Beräknat avslut för P9 (givet start av etapp 2) är dec. 2002.
- **P10** är avslutad för FoTA:s del. En spinoff av projektet är planer på att realisera Ravenscar-kärnan i ett "Safety-chip".
- **P11** har ny slutmilstolpe 2002-10-30 efter ominriktning mot inbyggda system och finansiering via ett antal vapenuppdrag (dvs ej genom FoTA, dock uppges P12 vara föreslagen som fortsatt referensgrupp). Deltagare är SBD samt Industrilogik.
- **H ProgSäk**, är nufastställd av FM, väntar på FM-beslut inför tryckning och översättning.
- **P12** har m.a.a. ovanstående begärt förlängning till 31 dec 2002.

Inom FoTA P12 har sju medlemmar från industrin aktivt följt dessa projekt. Arbetet har bl a bestått i rapportgranskningar, inbjudan till och deltagande i seminarier, bildandet av en hemsida för information inom P12 och övriga projekt via kontaktpersoner för dessa (www.aerotech.ffv.se/fota_p12/). Möjlighet att genomföra kortare prov av tekniker från dessa har också erbjudits. Fyra medlemsföretag har därvid valt att prova P9 på egna applikationer, samt att m h a H ProgSäk utvärdera företagets metodik och stöd vid utveckling av säkerhetskritiska system.

Medlemmarna inom P12 har funnit arbetet i projektet givande och uttryckt en förhoppning om att denna modell för samarbete och informationsspridning skall kunna tillämpas även efter P12:s avslutande. En avrapportering av erfarenheter inom projektet kommer att göras både skriftligt och i form av ett avslutningsseminarium under 2002. En av frågeställningarna inför avslut är om och hur P12:s hemsida skall göras mer allmänt åtkomlig.

Inga-Lill Bratteby-Ribbing
AOL FoTA P12,
ilbra@fmv.se,
tel 018-12 02 63

FoTA p4 klart ”Designmönster & återanvändning”

FoTA Projekt 4 slutrapporterades vid en genomgång på FMV den 6 december. Följande redogörelse har klippts ihop ur projektets slutrapport.

Syftet med projektet var att skapa intresse och kunskap om det viktiga området ”återanvändning och design patterns” genom att initiera och stimulera till en kunskapsuppbyggnad inom området hos försvaret, försvarsindustrin och högskolan samt att demonstrera tillämpbarheten av återanvändning och design patterns.

Designmönster (på engelska ”design patterns”) uppfattades inledningsvis av många som ett av de modebegrepp som ständigt tycks avlösa varandra inom databehandling med större eller mindre anspråk på att vara den slutliga lösningen till alla problem. I själva verket symboliserar begreppet ett paradigmskifte där man går från ostrukturerad återanvändning av primitiva komponenter till en strukturerad (”ingenjörsmässig”) återanvändning av erfarenheter på en högre systemnivå. För den breda publiken blev uttrycket allmänt känt nästan över en natt i samband med att boken Design Patterns (Gamma m. fl.) kom ut 1995 (egentligen summerar boken flera års arbete inom ramen för OOPSLA – den ledande konferensen inom objektorienterad systemutveckling). Författarna, Erich Gamma, Richard Helm, Ralph Johnson och John Vlissides, brukar kallas ”de fyras gäng” (”Gang of Four”, ”GoF”). Trots den störtflod av andra böcker och artiklar som har följt är det de 23 mönstren i Design Patterns som ständigt nämns och har blivit en ge mensam vokabulär i branschen. Snart sagt varenda presentation av en programvarudesign innehåller referenser till några av dessa mönster. Detta är det kanske bästa beviset för deras praktiska användbarhet och mera bestående värde. Som en följd av det gensvar som boken fick finns nu mönster av alla möjliga slag. Inom programvaruutveckling finns förutom designmönster också arkitekturmönster, analysmönster, mönster för komponentgränssnitt osv. På ett allmännare plan finns också organisationsmönster, processmönster, osv. Mönstren i Design Patterns har också anpassats till andra programmeringsspråk än C++ som till exempel Ada (Heaney, 2001).

Designmönster är praktiska lösningar till problem som ofta uppstår i verkligheten (Appleton, 2001). De ska klart visa hur och när man ska ansätta en viss lösningsstruktur. Lösningarna själva ska också förekomma I praktiken. Det räcker alltså inte att de fungerat någon enstaka gång. Mönster är inte otestade teorier, nya uppfinningar eller abstrakta principer. Den som samlar och dokumenterar mönster letar upp vad som redan förekommer istället för att skapa nya och originella algoritmer och metoder, vilket skulle vara mer meriterande akademiskt. Det handlar med andra ord om ingenjörskonst och återbruk av beprövade lösningar genom som generalisering och strukturering – man identifierar återkommande mönster och formaliserar eller stiliserar dem så att de kan användas igen i liknande problem (Monroe m. fl., 1997). Som mönsternedtecknare kan man antingen reflektera över system man själv byggt, studera system byggda av andra eller intervjua systembyggare. Man talar ibland om dessa angreppssätt som ”introspective”, ”artifactual” respektive ”sociological” (Kerth & Cunningham, 1997).

I litteraturen förekommer flera definitioner av mönster. Brad Appleton (2001) anger:

> ”En vanligt förekommande standardlösning som visat sig praktiskt hållbar.”
>
> ”En (fack)litterär form för att beskriva experterfarenheter.”
>
> ”En namngiven insikt som förmedlar det väsentliga av en testad lösning till ett problem i ett visst sammanhang med konkurrerande krafter”

Arbetet i FoTA-projektet har i huvudsak bedrivits i form av examensarbeten på magisternivå vid de medverkande företagen och institutionerna. Utöver examensarbeten har litteratur-studier, konferensbesök, seminarier samt fördjupningsstudier inom några områden bidragit till kunskapsuppbyggnaden kring mönster. Arbetet inom projektet har koordinerats av en styrgrupp med representanter för de deltagande företagen och institutionerna, nämligen, AerotechTelub, Ericsson Microwave och Sjöland & Thyselius Datakonsulter (även projektlederi), Chalmers (Kent Pettersson), Colorado Technical University (genom Bo Sandén) samt Växjö universitet (Ulf Cederling).

Flera lyckade examensarbeten och liknande projekt har genomförts inom projektets ram vilket innebär att ett antal studenter och andra har kommit i närkontakt med mönsteridéerna. Arbetena har också visat på fall där mönster inte varit särskilt användbara, som

fortsättning på sidan 17

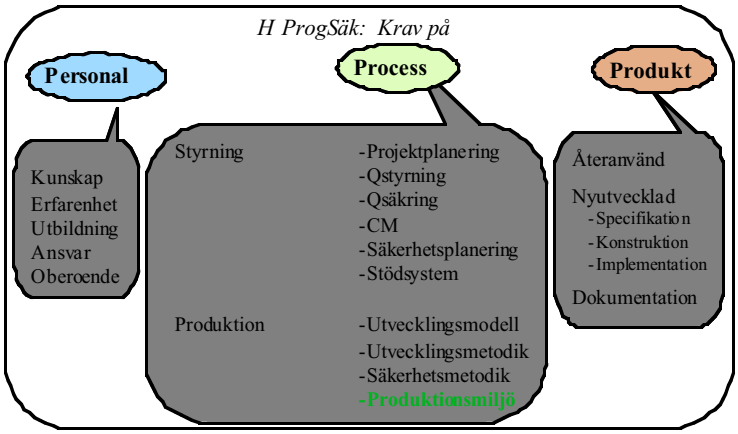
Handboken om programvarusäkerhet fastställd av FM

H ProgSäk 2001

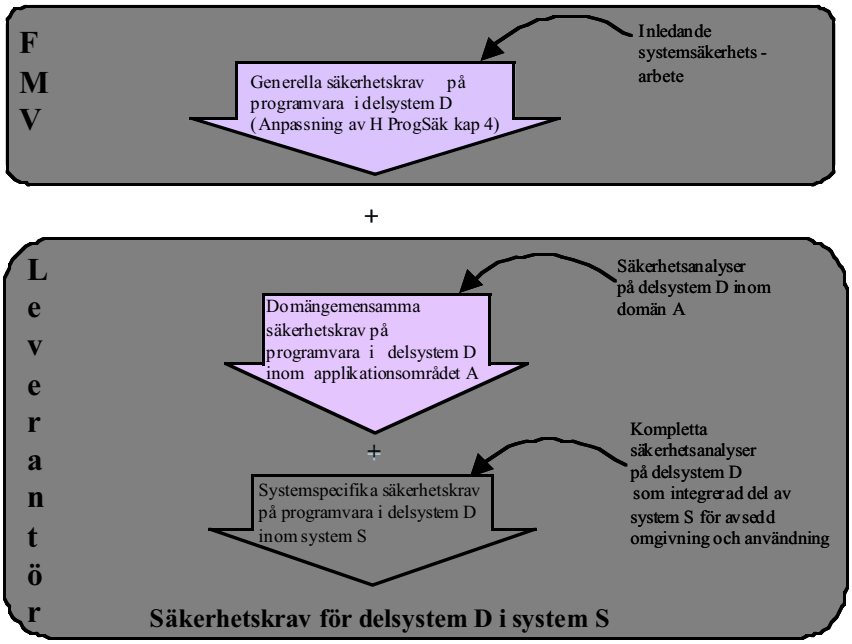
Handbok för Programvara i säkerhetskritiska tillämpningar, M7762-000531, innehåller Försvarsmaktens och Försvarets materielverks rekommendationer för anskaffning av programvara i säkerhetskritiska system.

Handboken, som fastställdes i december 2001, kommer att göras tillgänglig via Försvarets bok och blankettförråd samt på FMV:s hemsida, www.fmv.se: Aktuell information.

H ProgSäk ger generella säkerhetskrav m a p programvaran som produkt samt personal, process och produktionsmiljö aktiva under hela programvarucykeln. Kraven är numrerade och kritikalitetsgraderade -strängare krav ställs på delar av högre kritikalitet.



Handboken är avsedd att anpassas till enskilt projekt / applikation och ger därför vägledning i hur säkerhetskrav specifika för applikationsdomän och projekt kan härledas. Detta innebär både begränsningar och tillägg m a p de generella säkerhetskraven.



En stor del av handbokens innehåll är informativt. Förutom användarhandledningar och säkerhetsmetodik ingår aktuella referenser, begrepp, checklistor, problembeskrivningar samt referat av publikationer inom ämnesområdet.

Bakgrund

FMV fattade 1994 beslut om att tillämpa en ”försvarsstandard för säkerhetskritisk elektronik och programvara”. Arbetet med att färdigställa en handbok för säkerhetskritisk programvara (H ProgSäk) startade 1997. H ProgSäk har tagits fram etappvis genom successiva arbetsversioner, som granskats och provats inom och utom materielverket.

Publikationsansvarig

Inga-Lill Bratteby-Ribbing
FMV:KC Ledstöd
ilbra@fmv.se

Felinjicering för provning av programvara

Håkan Edler, HiSafe AB

När man provar programvara är det viktigt, att man hittar provdata för både normala och onormala situationer. Funktioner såväl som begränsningar skall provas. Felinjicering i anropen av tjänsterna hos en programenhet är ett sätt att provocera onormal användning. Vid felinjicering får man ofta ett mycket stort antal provfall. Ett automatiskt verktyg för generering, exekvering och loggning av provfall är nödvändigt. Inom FoTA P9 har vi utvecklat en metod och ett verktyg för automatisk provning av programenheter.

Bakgrund

Pålitliga datorbaserade system är i allmänhet noggrant konstruerade och byggda samt omsorgsfullt utprovade. Fel uppträder därför sällan i dem. För att verifiera konstruktion och validera pålitlighet måste man därför använda någon form av forcerad provning. Att artificiellt injicera fel i systemet är ett sådant sätt.

Institutionen för datorteknik vid Chalmers tekniska högskola har lång tradition i forskning på pålitliga datorbaserade system. Den sträcker sig tillbaka till sjuttiotalet, då de första datorerna för rymdfart konstruerades och byggdes. Idag har laboratoriet för pålitliga datorsystem vid institutionen ett grundmurat gott internationellt rykte och räknas till de främsta i världen inom området. Forskningen gällde till en början maskinvara och man arbetade med metoder att experimentellt validera vissa aspekter av pålitlighet hos datorelektronik. En viktig del var att bygga utrustning, där man på ett reproducerbart sätt kunde injicera artificiella fel i ett system och mäta effekten av felen. Sedan början av nittiotalet finns en grupp inom laboratoriet, som studerar metoder att bygga pålitlig programvara och att validera dess egenskaper. Främst har man studerat algoritmer och metoder att konstruera program robusta respektive feltoleranta. Ett robust program tål fel utan att skada sin omgivning, medan ett feltolerant program tål fel och ger ändå rätt service. Man har också byggt system för att experimentellt mäta egenskaper hos program genom att injicera fel och registrera programmets beteende.

Vid institutionen har man använt tre olika metoder för felinjicering:

- Injicering av fel i källtext - fault injection.
- Injicering av fel i anrop av subrutiner - fault injection.
- Injicering av feltillstånd - error injection.

Baserat på erfarenheterna från Chalmers har HiSafe i FoTA projekt P9 tillämpat den andra metoden för att pröva dess användbarhet i industriell skala.

Felinjicering

Felinjicering är ett sätt att verifiera konstruktioner och validera pålitlighet.

Vid injicering i källtext vill man efterlikna programfel. Man bör då man ha en uppfattning om hur felen ser ut och sedan föra in ett lämpligt urval i källtexten till programsystemet. För varje fel kompilerar man programmen och får ett stort antal varianter, ett för varje injicerat fel. Varje variant kör man sedan med varierande indata valda ur en lämplig användningsprofil.

Vid injicering av feltillstånd vill man oftast efterlikna maskinvarufel. Man stoppar ett exekverande program, förändrar instruktioner eller data i programmet och fortsätter sedan exekveringen med lämpliga indata. Om injicerade fel skall efterlikna verkliga programfel bör man även här ha en uppfattning om hur felen ser ut och därtill kunskap om hur de manifesterar sig i ett exekverande program.

Det vanligaste sättet att injicera feltillstånd är ”bit-flip”. Man kör systemet och stoppar exekveringen vid någon vald tidpunkt. Där förändrar man innehållet i register eller minne och fortsätter sedan exekveringen. Metoden simulerar väl temporära maskinvarufel men knappast programvarufel. På grund av sin enkelhet är det dock den metod, som används mest av andra forskare.

Felinjiceringförsök ger en stor mängd körningar, då resultaten ofta utvärderas statistiskt. Därför måste man bygga automatiska provbänkar.

Vid institutionen för datorteknik har en provbänk byggts upp för att studera fel i programvara och effekten av mekanismer för robusthet och feltolerans i programvara. Den undersöker varianter av program, där fel introducerats i källtexten till program. Alla varianter kompileras och provbänken laddar och kör dessa automatiskt enligt ett givet schema.

Data från körningarna loggas och analyseras sedan i efterhand. Effekten av fel i programvaran studeras på systemnivå, då man loggar systemets felyttringar.

I forskningen på feltoleranta system har man i allmänhet studerat felens effekter på systemen i sin helhet, alltså felyttringarna på systemnivå. Idag talar man mycket om komponentbaserad systemutveckling och man kan klart se en trend mot ökad användning av komponenter vid systembygge. Komponentbaserad utveckling är troligen den bästa möjligheten för att höja produktiviteten i processen och tillförlitligheten i produkterna vid utvecklingen av datorbaserade system. Kan man prova feltoleransen i komponenterna i stället för i hela systemet? Den metod som ligger närmast till att använda är då beteendeprovning, black-box testing, och injicera fel i anrop av programenheter.

Felinjicering på gränssnitt

Beteendeprovning används för att verifiera ett system mot både funktionskrav och övriga krav. Man provar dess funktioner för att verifiera att de överensstämmer med specifikationerna. Vid beteendeprovning kör man systemet utan kunskap om dess interna uppbyggnad, bara indata och utdata är kända. Man väljer en mängd provdata, kör systemet och observerar resultatet. Ett sätt att välja provdata är att använda ekvivalensklasser och göra gränsvärdesanalys. Genom att välja ogiltiga värden som provdata provar man även en moduls robusthet i vissa avseenden. Sett ur systemets synvinkel gör man felinjicering i systemet.

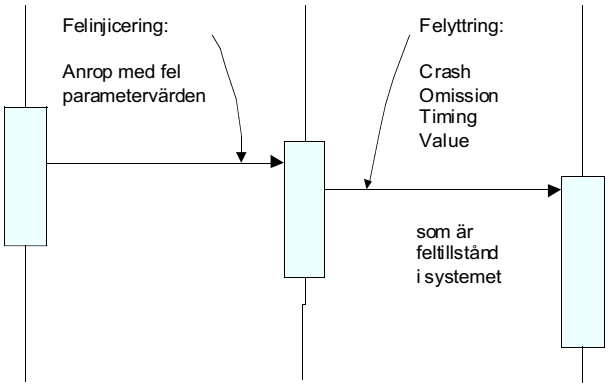


Fig 1. Felinjicering i anrop av ett objekt kan ge felbeteende hos objektet och ett feltillstånd i systemet.

Beteendeprovning av en modul med systematisk variation av provdata vid anrop av modulens funktioner bör göras i en automatisk provbänk. Den bör logga provdata och resultat, så att analys av prov kan göras i efterhand. En sådan logg kan också användas för regressionsprov av en modul efter ändringar. Vid regressionsprovet låter man provbänken köra alla provfall enligt ett tidigare prov och jämföra den nya loggen med den tidigare.

I FoTA P9 arbetar vi med att utveckla metoder att verifiera och validera system genom mätningar på gränssnitten mellan komponenterna i systemen.

Den generella uppbyggnaden av en provbänk

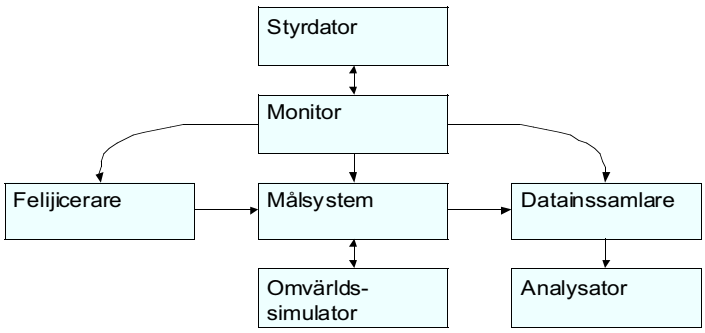


Fig. 2.
En provbänk för automatisk provning med felinjicering.

Provbänken är kärnan i experimentmiljön. Den studerade programvaran körs i målsystemet. Eftersom man i allmänhet inte kan köra tillämpningen i sin normala omgivning, måste dess omvärld simuleras. Omvärldssimulatorn genererar insignaler för tillämpningen och reagerar på ett realistiskt sätt på tillämpningens utsignaler. Såväl värden på signalerna, deras inbördes beroenden och deras tidsförhållanden skall vara autentiska.

En monitor övervakar och styr detaljerna i experimenten. Den läser instruktioner från styrdatorn och ger tillbaka tillståndsinformation om experimentens fortskridande. Monitorn laddar ner program i målsystemet och startar dem. Den läser ner instruktioner om fel som skall injiceras, startar felinjiceraren och initierar datainsamlaren. Monitorn eller målsystemet initierar omvärldsimulatorn.

Datainsamlaren läser data från målsystemet och loggar dem. Mätningar sker ofta i realtidssystem med snabba förlopp. Då kan datainsamlaren generellt sett inte göra annat under experimenten än logga. Insamlade data lagras och analyseras i efterhand.

Styrdatorn ger gränssnittet mot operatörer. Den används för att definiera försöksuppsättningar och försökssekvenser, vilket sker i enkla dialoger. För ett provobjekt beskriver man signaturen på de tjänster, som skall provas. De kan vara proceduranrop och det kan vara meddelande i textform. För varje ingående term ger man en formell värdemängd, som ofta motsvarar en datatyp i programmet och dess giltiga värden. Vidare ger man ett provmönster, som är underlag för de enskilda provfallen. Det kan inkludera programavsnitt som behövs före och efter det provande anropet. Ett prov definieras sedan med provmönster och en provvärdemängd, vilket för varje term är en vald delmängd av den formella värdemängden kompletterad med ett antal ogiltiga värden.

Under ett prov genererar styrdatorn alla provfall ur den givna informationen. För samtliga kombinationer av provvärden för ingående termer skapar den de provande anropen med kompletterande programavsnitt före och efter och skickar till monitorn.

Undersökning av ett RTOS

Inom forskningsprojektet på Chalmers har en experimentmiljö har byggts upp, där ett realtidoperativsystem kan provas under realistiska förhållanden. Felinjicering sker i dessa försök med en generell anropare för systemanropen till RTOS och som är länkad som en tillämpning till detta. Anroparen körs som en process under operativsystemet. Den läser en kommandosträng, avkodar systemanrop samt läser och konverterar sedan parametervärdena enligt anropets datatyper. Ett antal anrop kan ligga i strängen, vilket behövs om man skall kunna sätta provobjektet i ett givet tillstånd före det provande anropet.

Styrdatorns information är i detta fall signaturerna för alla systemanrop, beskrivning av alla ingående datatyper och deras formella värdemängder. Provmönstren ger följden av systemanrop, som omger det provande anropet.

Leverantören av operativsystemet hade tidigare ett antal provfall för varje systemanrop, som var för sig kompilerades, länkades, laddades och kördes vid provningar. Nu kan man istället automatiskt generera och köra stora mängder provfall vid behov.

Försök med procedurer ur ett runtime bibliotek

Erfarenheterna från provbänken för realtidsoperativsystemet användes inom FoTA P9 för att bygga motsvarande provbänk för en stor tillämpning inom Försvaret. I den provas enskilda procedurer och proven måste därför genereras i källtext. Definitionsdelen av styrdatorns funktioner är som i den generella provbänken. Signaturer, värdemängder och provmönster skapas i dialog med datorn.

Vid genereringen av provfallen skapar styrdatorn alla varianterna av provande anrop genom systematisk kombination av provvärden för alla termer. För varje provfall lägger den till ett prefix och ett suffix enligt det valda provmönstret. Den genererade källtexten skickas till en annan dator, som har monitorns roll. Den kompilerar, länkar och laddar programmet, varefter den initierar exekveringen, övervakar den och sparar resultatet. Till provmönstret hör därför också direktiv till monitorn för dess operationer.

För en provad procedur fanns fyra provfall tidigare, två med giltiga anrop och två med ogiltiga anrop. I den första körningen i provbänken genererades 2 200 provfall med värden valda slumpvis ur de formella värdemängderna och alla provfall verifierade provobjektets korrekta funktion. Med provvärden valda med hjälp av ekvivalenspartitioner och gränsvärdesanalys skapades 3 328 provfall med värden som skulle provocera proceduren. I proven kraschade såväl den provade proceduren som systemet i ett antal fall. Detta var i och för sig helt väntat, då proceduren är konstruerad för höga prestanda och inte för robusthet. Proven i detta fall verifierade alltså inte bara funktionerna utan också begränsningarna hos proceduren.

Etapp 2 av FoTA P9

I den första etappen av FoTA P9 har vi byggt upp en provbänk, analyserat ett antal provobjekt och gjort felmodeller för dem. De har varit underlag för de prov som sedan körts. I etapp 2 skall enligt ursprungliga planer metoden för felinjicering på gränssnitt provas för robusthet och feltolerans. Provobjekten från etapp 1 byggs robusta. Därefter görs samma prov som i etapp 1 och förbättringarna mäts. Sedan byggs programmen feltoleranta. Proven görs än en gång och förbättringarna mäts.

Den metod för felinjicering i gränssnitt till programvara, som använts i FoTA P9, har visat sig användbar och flexibel. Med en tämligen enkel beskrivning av ett provobjekt kan ett verktyg skapa tusentals provfall och exekvera dem. Metod och verktyg har med gott resultat använts i fler tillämpningar än de ovan nämnda.

Den planerade fortsättningen av FoTA P9 skulle istället kunna vara en breddning av

tillämpningarna till andra områden och fler intressenter än vad som ursprungligen planerats. Vi är öppna för förslag, så kontakta mig om intresse finns. Den största användningen för metod och verktyg har man för att verifiera funktioner och begränsningar hos en programenhet. Att visa robusthet och feltolerans är en del av detta.

Sammanfattning

Vid provning av programvara bör man inte slumpmässigt välja provdata för normala och onormala fall. Försöken med proceduren visar detta klart. Ekvivalenspartitionering och gränsvärdesanalys är ett sätt att hitta bra provdata, som kan provocera ett provobjekt. Det finns andra metoder, som inte tagits upp här. För att effektivt prova en programenhet kan man använda ett generellt verktyg för definition och generering av provfall och bygga en anpassning till den aktuella programenheten. Gör man detta redan under analys och konstruktion kan man vinna mycket i produktivitet och kvalitet.

Bibliografi

Hermansson, S. och Sinclair, J., *Etablering av experimentmiljö*, Rapport inom FoTA P9, HiSafe, juni 2000.
Edler, H., *Felinjicering och felmodeller*, Rapport inom FoTA P9, HiSafe, juni 2000.

Torbjörn Andreasson, EMW, var på höstens Simulation Interoperability Workshop och konstaterade att hur man bygger återanvändbara komponenter blivit en allt aktuellare fråga även i HLA-sammanhang.

Reflektioner från SIW Fall 2001

Höstens upplaga av konferensen ”Simulation Interoperability Workshop” (SIW Fall 2001) började bra: nya tutorials, högt tempo och stor uppslutning till nya arbetsgruppen Base Object Model Working Session (BOM-WS). Tyvärr avstannade konferensen redan andra arbetsdagen då attentatet mot WTC genomfördes. Därefter fick många av delegaterna annat att tänka på.

Personligen tycker jag att förvånansvärt många presentationer/diskussioner fortfarande fokuserar på metoder för ”data alignment”, dvs mekanismer för konvertering av textmeddelanden mellan databasorieterade simuleringar .

BOM-gruppens arbete initierades av FOM-gruppen (Federation Object Model) för något år sedan då man insåg att det är svårt att skapa återanvändbara referens-FOMar. Detta har BOM-gruppen tagit fasta på och försöker istället identifiera små/primitiva byggblock (modeller) för återanvändning vid design av FOMar.

BOM-gruppen har inspirerats av följande koncept:

- Stödja analys- och designarbetet genom identifiering och återanvändning av goda mönster (Design Patterns)
- Stödja realisering av konkreta simulatorer genom återanvändning av prefabericerade komponenter (klasser, metoder, metadata i form av krav, konceptuella modeller, etc)

Begreppet *”Design Patterns”* introducerades i boken med samma namn av gruppen *”The Gang of Four (GOF)”*. Den är egentligen en sammanställning av det arbete som bedrevs inom ramen för OOPSLA i början på 1990-talet (Object-Oriented Programming, Systems, Language, and Applications, http://oopsla.acm.org/). BOM-gruppen har på liknande sätt identifierat mönster inom området distribuerad och löst kopplad simulering.

A Pattern is ”an idea that has been useful in one practical context and will probably be useful in others” (Martin Fowler, OO-orakel).

BOM-gruppen har hittills identifierat/definierat följande BOM-typer:

- Interaction BOMs
- Trigger BOMs

och följande BOM-kategorier:

- Interface (IF) BOMs
- Encapsulated (ECAP) BOMs.

Arbetet i BOM-gruppen lär fortsätta och förväntas efter hand leda fram till fler återanvändbara BOMar.

En personlig reflektion/kommentarär att BOM-gruppens arbete har goda möjligheter att leda fram till en användbar bygglåda med primitiva och återanvändbara komponenter, inte bara för simuleringsändamål. Konceptet har potential att stödja LedsystT-visionen ”att på kort tid sätta samman skräddarsydda och situationsanpassade system, så kallade (SitSystar)”.

ISO/IEC 15288: A New Powerful Standard for Systems and their Life Cycles

A new standard entitled ISO/IEC 15288 System Life Cycle Processes is in the final stages of development and is scheduled to become an International Standard during the 4Q of 2002.

It is the first standard to take a holistic view of human made systems composed of hardware, software, and human elements, as well as combinations thereof. The standard is built around a very few generic, but powerful concepts that can be reapplied to all instances of systems (including systems of systems). The concepts when applied collectively provide a very useful and practical characterization of a system and its life cycle (from conceptualization to retirement). The standard describes how systems are decomposed into constituent system elements at multiple levels as well the relationship between the system of (current) interest and other systems upon which the system is dependent called enabling systems.

While the scope and contents of the systems for which life cycles can be created and managed is virtually unlimited, the standard is quite small due to the fact that the generic concepts are reapplied in formulating life cycles (one system at a time).

The processes of the standard are organized into four categories that represent the major viewpoints in respect to a system; namely, Enterprise, Agreement, Project, and Technical processes. The entire standard is about 60 pages and the processes to be applied during the life cycle are specified in about 35 pages.

Dr. Harold (Bud) Lawson leads the Swedish delegation that has actively participated in the development of 15288. In addition, Dr. Lawson was elected by the participating countries to be the architect of the standard. In this role, he has been a key designer of the few, but powerful concepts of the standard.

It is expected that 15288 will play a vital supporting role to organizations that seek ISO 9001:2000 certification. The processes of the standard provide a useful logical set that can be applied in defining product realization processes that are to be used in transforming customer requirements to customer satisfaction.

Further information on the standard is available at www.15288.com or bud@lawson.se

---

fortsättning från sidan 11

till exempel när man kommunicerar med icke-specialister. Det har också visat sig svårt att dokumentera ett redan existerande system med hjälp av mönster. Projektet har också lett till en spridning av intresset för mönster inom de deltagande organisationerna med bland annat studiecirkelverksamhet som resultat.

Rapporten innehåller förslag till fortsatt verksamhet. Bland annat kan studierna och examensarbetena utökas till att omfatta mönster för annat än programvarudesign, som till exempel analys-, organisations- och processmönster. I kompetenshöjande syfte rekommenderas också deltagande i mönsterkonferenser, exempelvis VikingPLoP, som hålles i Skandinavien.

Den som vill veta mer om ämnet och projektets resultat kan studera SESAMs CDROM, SESAM 2001, som distribueras med detta nr av Rendezvous, Där finns FoTA P4 slutrapport och två av bilagorna till denna., "Software Patterns" av Åsman & Engene samt "Beskrivning av Arkitekturer med Mönster. Ett exempel" av Kent Petersson. Det går givetvis också att ta direktkontakt med t ex projektledaren Mattias Larsson på Sjöland & Thyselius (Mattias.Larsson@st.se) som kan förmedla vidare kontakter.

Dessutom, ett rent sammanträffande, finns på CDn även Matthew Heaneys tutorial "Implementing Design Patterns in Ada95; Tips, Tricks, and Idioms".

Lyssnare och noterare på slutredovisningen var I Carlsson.

SESAM 2001- välmatad CDROM

Med detta nummer av Rendezvous följer CD-skivan SESAM 2001. Den innehåller förutom föredrag och presentationer från höstens seminarium, också en hel del material av bakgrunds- och referenskaraktär. Skivan är avsedd för SESAM-medlemmar och deltagare i seminariet.

Seminariematerialet är något uppdaterat och kompletterat i förhållande till vad som föredrogs vid seminariet.

I bakgrunds- och referensmaterialdelen har sammanställts material med stor bredd som i olika avseenden har beröring med arkitektur, systembyggnad och nätverksförsvar. Många världsledande personligheter inom dessa områden har varit mycket tillmötesgående med att bidra med material. Urvalet är dock inte på något sätt heltäckande eller representerar SESAMs uppfattning om vad som är rätt eller fel, men det ger förhoppningsvis läsaren/betraktaren ökad förståelse för sammanhangen och de olika synsätt som kan förekomma.

I urvalet ingår bl a referenser som gavs under seminariet och vilka kan tänkas besvara frågor från åhörarna som p g a tidsbrist inte hanns med att behandlas då.

Liten vägledning och några påpekanden betr innehållet:

SYSTEMS ARCHITECTURE

Avsnittet inleds med John Zachmans artikel *Enterprise Architecture* som bl a bygger på hans grundläggande artikel från 1987 i IBM Systems Journal, "A Framework for Information Systems Architecture". Därefter en presentation av Mark Maier om *Architectures as a Metaphor for Complex System Engineering*. Kathie Sowell redogör i sin artikel *The C4ISR Architecture Framework: History, Status and Plans for Evolution* för arbetet med att bredda detta ramverk, ursprungligen tillkommet för ledningssystemområdet, till att kunna gälla som grund i alla typer av amerikanska försvarssystem som omfattas av Joint Technical Architecture. Kathie meddelade i anslutning till att hon ställde artikeln till SESAMs förfogande, att arbetsgruppen nyligen presenterat sitt förslag, men att det kan ta avsevärd tid innan ett fastställt ramverk (under nytt namn) föreligger.

David Oliver, välkänd inom Systems Engineering kretsar har med sina två medförfattare ställt hela sin bok *Engineering Complex Systems with Models and Objects* till förfogande här. Två KTH-forskare, D.J. Chen och Martin Törngren, beskriver i en färsk artikel *Towards a framework for architecting mechatronics software systems*, sitt arbete med ett arkitekturramverk för Mekatronikområdet, något som bör vara av intresse för många SESAM-medlemmar. Vidare innehåller avsnittet ett par artiklar av SESAM-veteranen Ingmar Ögren om olika aspekter av modellbaserad Systems Engineering, *On principles for model-based systems engineering* resp *Possible Tailoring of the UML for Systems Engineering Purposes* och en om en möjlig komplettering av JTA; *Clarify the Mission - a necessary addition to the Joint Technical Architecture (JTA)?*

SOFTWARE ARCHITECTURE

Detta omfattande avsnitt inleds med en artikel *Software Architecture: a Roadmap* där David Garlan, en de ledande forskarna inom området, anger sin syn på programvaruarkitekturutvecklingen. Sedan följer två bidrag som refererades till under seminariet, Philippe Kruchtens ofta citerade artikel från 1995 *The 4+1 View of Architecture* och översiktskapitlet *The Business Component Approach* ur Peter Herzums bok Business Components (Wiley and son), vilken är kursbok i Dataföreningens kurs "Certifierad IT-arkitekt". (Vid kontakt med Herzums företag i USA, om tillstånd att inkludera detta material, visade sig Peter Herzum ha svenskt påbrå och tala mycket god svenska.) Därefter kommer en artikel *Adopting Software Product Lines: Approaches, Artefacts and Organization* och presentationsmaterialet för två kurser *Software Product Lines and Software Architecture Design; tutorials* av Jan Bosch, till för något år sedan vid Högskolan i Karlskrona/Ronneby, nu professor vid Universitet i Groningen, Nederländerna. Jan tar bl a exempel från svenska företag utanför försvarssektorn. som han arbetat med, vilka gått in för återanvändning i produktlinjer.

Detta följs av två artiklar av Alexander Ran *ARES Conceptual Framework for Software Architecture* ur en annan arkitektur-

bok, Software Architecture for Product Families Principles and Practice (Addison-Wesley), och *Fundamental Concepts for Practical Software Architecture*, vilken är en kort men klargörande beskrivning av huvuddragen i en kurs (tutorial) baserad på ARES-artikeln som Ran gav vid ESEC-konferensen i september 2001 (och vilken han f ö erbjudit sig komma och ge även i Sverige). Ran identifierar fyra kategorier av väsentliga designbeslut (d v s sådana som är dyrbara att senare ändra) ”concepts, focus, structure, and texture” och beskriver sedan hur de hanteras i hans ramverk.

Sedan tre artiklar om vad arkitetur egentligen handlar om och var man kan gå fel. Först *Separating Fact from Fiction in Software Architecture* av två forskare vid UC Irvine, Nenad Medvidovic (numera vid USC) & Richard N. Taylor. Därefter återkommer Kruchten med en intressant betraktelse om *Common Misconceptions about Software Architecture*, varefter Ran förklarar varför Legokloss paradigmen inte är så hållbar i programvarusammanhang i *Software Isn't Built From Lego Blocks.*

Användning av COTS är något som många numera förespråkar, utan att alltid kanske ha klart för sig att det innebär en helt ny uppsättning omständigheter och problem, som kan visa sig motverka, eller helt eliminera, de fördelar man hoppats vinna. Mark Maybury beskriver en del av dessa frågeställningar i sin OH-presentation *What Rots about COTS: Hidden Risks of Commercial Software.*

Kruchtens anlägger till sist i sitt tredje bidrag *The Tao of the Software Architect* mer långsiktigt filosofiska aspekter på programvaruarkitektens situation.

David Garlan som stod för den inledande artikeln i detta avsnitt, är även ansvarig för The Able Project vid Carnegie Mellon. Från projektets hemsida http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/able/www/index.html citeras:

”Carnegie Mellon University's ABLE Project conducts research leading to an engineering basis for **software architecture**. Components of this research include developing ways to describe and exploit **architectural styles**, providing **tools** for practicing software architects, and creating **formal foundations** for specification and analysis of software architectures and architectural styles. The ABLE Project has developed several **architecture description languages**: Acme, Wright, and Armani. Among the tools currently available are AcmeStudio and AcmeLib, both of which support Acme.”

Som ett smakprov återger vi i *Selected Papers from the ABLE Project*, den fylliga referenslista som projektet upparbetat.

INTERNATIONAL STANDARDS

Här ges beskrivningar av några standarder och pågående standardiseringsaktiviteter som har nära beröring med arkitektur och systembyggnad. D. E. Emery, M. W. Maier, R. F. Hilliard som medverkade i utarbetandet av den rätt nyligen antagna arkitekturbeskrivningsstandarden IEEE 1471, har ställt sitt kursmaterial *Introducing IEEE 1471: Recommended Practice for Architectural Description, tutorial* till förfogande.

Genom tillmötesgående från ISO återger vi ett arbetsdokument från det pågående projektet att ta fram 15288-standarden, som förklarar idén bakom denna standard: *ISO/IEC CD 15288 FCD, Systems Engineering – System Life Cycle Processes, Annex D: Concepts*.

En av grundarna av INCOSE – International Council On Systems Engineering, Jerry Lake, har i *Three concepts; ANSI/ EIA- 632-1998, ISO/ IEC 15288, World Class Systems Engineering* och *A comparison of three concepts* bidragit med ett par jämförelser av 15288 med den existerande ANSI/ EIA- 632-1998 och hans egen World Class Systems Engineering som Jerry bl a lärt ut vid kurser här i Sverige i FMV regi.

Sist i detta avsnitt två OH-presentationer som ställts till förfogande av Torbjörn Holm om några bredare, inom försvarssektorn mycket aktuella, standarder på systemteknikområdet; en översikt *IT Standards for Product Life Cycle Support* av Torbjörn Holm och *An Overview of AP 233. STEP's Systems Engineering Data Exchange Standard* baserad på ett NASA-material.

Ada, Layered Architecture, CORBA, Java, Patterns, OOP

Detta avsnitt är inriktat på användningen av Ada i samband med några mycket aktuella företeelser inom programvaruområdet. Om någon skulle ha blivit invaggad i föreställningar om att Ada hörde det förgångna till och bara platsade i gammaldags inbäddade system, så är det dags för uppvaknande. Vi har fått tillgång till omfattande kursmaterial för flera mycket intressanta tutorials, vilka dock torde förutsätta programmerings-

kunskaper. Men först en artikel av Kelly Spicer, norskättling, som beskriver hur ”gamla” Ada 83 används för att skapa en modern återanvändningsvänlig flerskiktsarkitektur: *A Successful Example of a Layered-Architecture Based Embedded Development with Ada 83 for Standard-Missile Control.*

Därefter OH-bilderna från en kurs med tyvärr rätt blygsamt SESAM-deltagande som Cathy Hrustich höll vid Ada i Sverige konferensen våren 2001: *Real-Time, High Performance and Embedded CORBA for Ada Applications; tutorial.*

Nästa är OH-bilder från en kurs om Ada och Java av Gary Dismukes och Franco Gasperoni: *Developing Ada Applications for the Java Platform with JGNAT, tutorial.*

Slutligen har Matthew Heany bidragit med två tutorials. Den första, i två delar, handlar om tillämpning av högaktuella Patterns i Ada 95: *Implementing Design Patterns in Ada95; Tips, Tricks, and Idioms, Part I : Sequential Patterns* resp *Implementing Design Patterns in Ada95; Tips, Tricks, and Idioms, Part II : Concurrent Patterns.* Matthews andra kursmaterial gäller hur man åstadkommer tillförlitlig objektorienterad programmering (givetvis med Ada): *Reliable Object-oriented Programming*.

ARKITEKTUR & SYSTEMBYGGNAD

Under denna rubrik finns ett antal dokument som har sammanhang med frågan hur man bygger ett nätverksbaserat försvar. En viktig utgångspunkt för detta är kraven på den s k Demonstrator 2005, vilken är ett led i att utveckla, prova och verifiera en delmängd av FM målbild för LedsystT 2010, och hur man avser att utveckla den. Detta framgår av *Demonstrator Nätverksbaserat Försvar 2005.* I ett tidigare dokument*, Arkitektur och förutsättningar för LedsystT*, har FMV redovisat resultat av den första etappen i en långsiktig satsning på att etablera underlag som belyser arkitekturens roll i den kommande utvecklingen av Försvarsmaktens verksamhet. Det arbete som görs där innebär ett utforskande av området arkitektur, dels konceptuellt, dels i syfte att lära av andras erfarenheter.

Ett par år längre tillbaka i tiden ligger en FMV-rapport om *Infrastrukturbaserad IS-försörjning,* som mot bakgrunden av erfarenheter av utveckling av informationssystem inom försvaret, föreslår en ansats till hur en systematisk uppbyggnad skulle kunna ske av försvarsgemensamma tillgångar, resurser och tjänster etc vilka efterhand kunde nyttjas vid utveckling, vidmakthållande och drift av verksamheten och dess informationsförsörjning. Denna rapport utarbetades i samband med FMVs deltagande i ett WEAG-projekt om förbättring av anskaffningsprocessen för försvarsinformationssystem. Två av resultaten från det projektet redovisas i *Guidance on the Use of Progressive Acquisition*, vilken föreslår en evolutionär och inkrementell (progressiv) anskaffningsmodell, och i *Cost-Benefit Justification of Defence Information Systems (DIS)*, som behandlar frågan hur man kan rättfärdiga investeringar i informationssystem, mot bakgrunden av att sådana alltför ofta tidigare visat sig inte hålla vad man ställt i utsikt.

Avsnittet avslutas med redovisning av resultat av fyra forskningsprojekt med bäring även på den nuvarande situationen betr försvarssystemutvecklingen. Det två första har bedrivits inom ramen för NUTEK/VINNOVA nationella forskningsprogram Komplexa Tekniska System. För det nyligen avslutade Projekt DELTA redovisas dels förstudierapporten om *Kontinuerlig behovsstyrd utveckling av komplexa ledningssystem* och dels slutrapporten *DELTA Meta Architecture for Proactive Management of Coordinated Development in Complex Enterprises and Information Systems*. Det andra projektet SEMLA, avslutades 1997 och resultaten redovisade i *Projekt SEMLA – Infrastruktur, arkitektur och metodik för återanvändning* torde fortfarande vara högst relevanta. Därefter följer en doktorsavhandling från Göteborgs Universitet om *Strategisk IT Management* samt utdrag ur en rapport från det nationella IT4-programmet 1993/94 *Projekt DUR; Utvecklingsmetodik för Rationella informationssystem*, som bl a visar att många av de frågeställningar man brottas med i dagens systembyggnad ingalunda är nya. Både SEMLA och DUR dokumenten är sammanställningar av separata rapportdelar från en tid när ordbehandlings- och dokumentbehandlingstekniken inte var så välutvecklad, varför en del ofullkomligheter i deras presentation kan finnas.

MODELLERING OCH SIMULERING

Användning av modellering och simulering anses av många vara en nyckel till framgångsrik utveckling av framtidens komplexa system. Den åsikten förefaller delas av Försvarsmakten, som framgår av *Försvarsmaktens inriktning för Modellering och Simulering (M&S)*. Inriktningen, som var den första för försvaret i sitt slag, fastställdes på våren 2000. HKV meddelar att bearbetning

av dokumentet pågår inför en ny utgåva.

På FMV har man nyligen invigt ett nytt simuleringslabb, SMART-lab och med anledning av detta publicerade FMV Aktuellt några artiklar som visar hur man på FMV arbetar med M&S och hur det nya labbet avses användas. Genom tillmötesgående från FMV, kan vi presentera de tre artiklarna *En SMART satsning*, *Spel inom Försvarsmakten* och *Virtuell prototyp av ett marint fartygssystem*.

En rapport från 1996 *Studie, Användning av Modellering och Simulering för Flygvapenändamål (SAMS/F)* kan fortfarande ha intresse som en sammanfattande analys av M&S-läget för ett helt stridskraftsområde.

I samband med det amerikanska försvarets verksamhet för att utveckla HLA, High Level Architecture, för modellering och simulering, är företaget Pitch i Linköping, med stöd av FMV mycket aktivt. Bl a utvecklar man metoder och verktyg för HLA och ger presentationer vid återkommande workshops och konferenser. Tre sådana presentationer ingår här*: HLA as Conceptual Basis for a Multi-Agent Environment*, *Modeling Component-Based Intelligent Agents in an HLA-Environment using an Agent Development Tool* samt en artikel *pRTI™ 1516 – Rationale and Design* om det ”distribuerade operativsystem”, Run-Time Interface, som Pitch utvecklar för HLA-simulering.

Sist i avsnittet en NATO forskningsrapport om bästa sätt att använda analys och modellering av ledning (C2) vid överväganden i olika sammanhang *NATO Code of Best Practice for C2 Assessment* (tillhandahållen via CCRP, se nedan).

NÄTVERKSFÖRSVAR

Avsnittet inleds med två svenska inlägg, *Command and Control in Network Centric Warfare* och *Network Centric Warfare; ÄR inte något; BLIR vad vi vill.* Därefter ett antal dokument som ställts till förfogande av amerikanska försvarets C4ISR Cooperative Research Program (CCRP), vilket tagit fram en hel del av principerna för nätverksförsvar. Först en allmän presentation *Information Superiority & Network Centric Warfare, OH-presentation,* sedan boken *Network Centric Warfare; Developing and Leveraging Information Superiority*, detaljerat förklarar bakgrund och innehåll i nätverkskonceptet tillämpat i försvarssammanhang. Sedan följer en rapport från DOD till amerikanska kongressen *Network Centric Warfare. Department of Defense Report to Congress. 27 July 2001*, som innehåller utförliga definitioner och beskrivningar av nätverkscentrerat försvar i DOD tappning, samt redogör för vilka åtgärder som är på gång inom DOD på det området. Sist en annan mycket innehållsrik bok *Understanding Information Age Warfare*.

Avsnittet avslutas med tre rapporter om Försvarshögskolans stora ROLF-projekt, Rörlig Operativ LedningsFunktion, som är nära förknippat med uppbyggnaden av ett svenskt nätverksförsvar, *ROLF 2010; Rörlig Operativ LedningsFunktion år 2010* från 1997 och två publikationer på engelska från resp 1998 och 2000 *ROLF 2010; A Mobile Joint Command and Control Concept* och *ROLF 2010 The Way Ahead and The First Step*.

VERKTYGSPROGRAM

Här var avsikten att tillhandahålla demoexemplar av ett par inhemskt utvecklade verktyg för systembyggnad.

Det som finns med är Tofs - (Tool for Systems) för arkitekturell och detaljerad modellering av komplexa system, där programvara samverkar med maskinvara och operatörsroller för att genomföra en eller flera uppgifter; Verktyget installeras på Windowsplattformar från *Installation_Tofs.01V Folder* som innehåller en utvärderingsversion av programvaran Tofs. Detta är den tredje versionen av Tofs, som utvecklats, med stöd av NUTEK, från en tidigare programvara ASP. Programmet har, förutom till svenska myndigheter och företag, levererats till Frankrike, USA, Indien och Storbritannien.

Programmet är komplett och utan begränsningar för 30 dagars användning. De trettio dagarna är arbetsdagar och inte enbart kalenderdagar. Kör ”Setup.exe” för att installera Tofs utvärderingsversion.

Tutorial Folder innehåller ett interaktivt utbildningsprogram för programvaran Tofs, där användaren leds genom ett enkelt exempel som innehåller en uppgift (värma mat i mikrougn), en operatör (kocken) samt programvara och maskinvara i själva mikrougnen. Programmet tar upp flertalet uppgifter som stöds av Tofs som kravarbete, konstruktion, provberedning, problemhantering, dokumentation etc. Kör ”Setup.exe” för att installera utbildningsprogrammet.

Den nya versionen av Pitch HLA-verktyg, hann tyvärr inte med på skivan, men den kommer att mycket snart finnas tillgänglig för nedladdning i demoversion från www.pitch.se.

I Carlsson

SESAMs höstseminarium 2001

Höstsemniariet ”ARKITEKTURER DÅ, NU OCH SEDAN - från vision i retoriken till integration uti praktiken” ägde rum på Täby Park Hotel den 24 oktober och lockade ett 70-tal deltagare. Föredragsmaterialet kommer att finnas att hämta ner från hemsidan och återfinns också på CD-skivan SESAM 2001, se särskild artikel.

Seminariet avsåg att lyfta fram arkitekturfrågornas betydelse vid utformning av försvarssystem.

Meningen var att inför de stora förändringar i försvarets uppgifter och därmed i försvarssystemens utformning som är på gång, försöka

- samla arkitekturrelaterade erfarenheter och lärdomar av bortåt femtio års svensk systemutveckling och se hur de kan utnyttjas i den nya situationen
- ge en uppfattning om vilka ”arkitekturdrivande” förutsättningar som gäller för det framtida försvaret och med hänsyn till dessa ge några exempel på hänsyn som behöver tas vid utformning av arkitekturansatser
- samt slutligen belysa hur framtida arkitekturarbete med de nya förutsättningarna kunde bedrivas

Programmet bestod av fyra huvuddelar:

- ARKITEKTUR - SYFTE, BEGREPP OCH PRINCIPER, TRENDER
- ARKITEKTURER UNDER 50 ÅR; ERFARENHETER OCH LÄRDOMAR
- DET FRAMTIDA NÄTVERKSFÖRSVARET – ARKITEKTURUTMANINGAR
- HUR BEDRIVA FRAMTIDA ARKITEKTURARBETE - EN FRÅGA OM SAMARBETE

Seminariet avslutades med en diskussion om insikter vunna under dagen och hur dessa på bästa sätt skulle kunna utnyttjas i det fortsatta arkitekturarbetet. Den inleddes av förre chefen för Flygmaterielförvaltningen m m Gunnar Lindquist. Hans manus återges i sin helhet på annan plats i detta nr.

På grund av middagslokalen och bordsarrangemangen fick denna diskussion föras bordsvis, men det verkade inte nämnvärt dämpat åsiktsutbytet. Dock gjorde detta det svårt att samlat referera diskussionen, vilket vi alltså avstår från här.

Tyvärr missades en viktig del av målgruppen för seminariet p g a av en olycklig kollision med ett av FM/FMV samtidigt anordnat möte för den personal som är närmast inblandade i arbetet med att ta fram den nya försvarssystemarkitekturen. Förhoppningsvis kan en del av SESAM-seminariets faktiskt mycket intressanta erfarenheter och lärdomar, ändå på något sätt tränga upp i den kretsen.

Du besöker väl vår websajt?
http://sesam.tranet.fmv.se

SESAM-Sekretariatet: AerotechTelub AB
c/o Kåsjös Kontor
Ytterspåret 14
187 54 TÄBY

Telefon: 08-510 51866
Telefax: 08-510 51932
GSM: 070-716 9702
E-post: alkas@tranet.fmv.se