

RENDEZVOUS

Nr 1 april 2002

Extra!
Handbok ProgSäk på 3,5"
diskett medföljer detta nr



Innehåll

Ordföranden har ordet	3
Första kursen med Tofs i USA.....	3
Årets mikroprojekt i arbetsgrupperna	4
Inbjudan till introduktion i programvarusäkerhet och H ProgSäk	5
Att arbeta med krav – referat från "Journal of Systems Engineering"	6
Nya CMMI med integrerade mognadsmodeller nu ute på allvar	10
Ada NYTT via Ada Information Clearinghouse	13
Mer Ada-Nytt	15
Kalender	16

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 FOI.

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 tillvägagångssättet vid utveckling och vidmakthållande av programsystem. Implicit i detta ligger givetvis att använda processer skall tillförsäkra de resulterande produkterna efterfrågade egenskaper. Produktens egenskaper 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
christopher.bengtsson@telia.com
Carlsson Ingemar, adjungerad
ingemar.carlsson@mbox2.swipnet.se
Ekman Mats, Saab AB
mats.ekman@saab.se
Hallén Johan, FMV
johan.hallen@fmv.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@hisafe.se

Ag Teknik

Lars Asplund, Mälardalens Högskola
lars.asplund@mdh.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 kan nu glädjas åt det resultat vi lyckades åstadkomma under 2001 trots att viss turbulens rådde kring det administrativa stöd som vi skall ha. Med en välvilja från alla parter så löste sig det mesta.

Jag vill nämna några av de projekt som genomfördes och sedan resulterade i seminarier.

Metodikgruppens projekt "drömverktyget" som granskade olika leverantörer i samband med ett seminarium.

Höstseminariet som gick under temat Arkitektur och som var besökt till sista stol. Efter detta seminarium producerade SESAM en CD i begränsad upplaga med 1000 sidor kring arkitektur. Temat var så intressant och låg helt rätt i tiden varför vi under 2002 troligen följer upp temat med ny verksamhet.

Idag är redan verksamheten för 2002 i full gång. Vad jag kan se så har vi både fått in nya medlemmar och igång verksamhet där vi under 2001 hade vissa problem. Jag ser därför fram emot detta verksamhetsår. Allt tyder på att vi kommer att ha nya områden att ta hand om samt förvalta vissa tidigare startade projekt. Därför tror jag att 2002 kommer att bli ett bra år för SESAM.

Claes Wadsten
Ordförande i SESAM

Tfn. dir : 013-23 16 52
Mobil: 070-6000271

E-post: claes.wadsten@aerotechtelub.se

Första kursen med Tofs i USA

Bifogat informationsblad avser en kurs, som planeras av Tofs amerikanska samarbetspartner Spreadsheetworld i Las Vegas i oktober.

Vi på Tofs är glada för att ha etablerat detta samarbete eftersom Spreadsheetworld (www.spreadsheetworld.com) har en mycket lång erfarenhet av att arbeta med Excel-ark och VB-programmering för olika typer av systemtekniska optimeringar. Detta passar speciellt väl ihop med Tofs eftersom programmet numera tillåter att man arbetar med ett antal Excel-ark tillsammans med sin systemstruktur.

I samband med att vi förberedde denna kursen har vi även diskuterat möjligheten att erbjuda den i Europa, kanske Sverige. Problemet är att de reklamutskick, som numera krävs för den här typen av kurser är mycket kostsamma för småföretag som Spreadsheetworld och Tofs.

Därför lägger jag ut denna begränsade information till Sesams medlemmar för att pejla om det finns ett intresse för denna kurs i Sverige. Om så är fallet bör vi kunna erbjuda den här ganska snart efter kursen i Las Vegas.

Om någon är intresserad hör av Er till mig eller kontakta Dr. Mincer (tmincer@spreadsheetworld.com) . Alla frågor är välkomna och jag berättar gärna också om den nya versionen av Tofs stöder standarden 15288 för systemarbete.

Ingmar Ögren
iog@toolforsystems.com
0176-54580

PS: Vi håller just på att slutföra intern provning av Tofs i version 02. Om någon är intresserad av programvaran för betaprovnings, hör av Er.

Årets mikroprojekt i arbetsgrupperna

Vid Ag Teknik senaste möte på Mälardalens högskola i Västerås den 14/3 beslutades om två mikroprojekt:

Användning av Freeware

SaabTech Systems
Saab Bofors Dynamics
Aerotech Telub

Säkerhetskritiska system med speciell inriktning mot kommunikation och blandning av high/low safety

Ericsson Microwave
Saab Aerospace
SaabTech Systems
Mälardalens högskola

Andra företag/intressenter som vill ansluta bör snarast anmäla sig till Lars Asplund, lars.asplund@mdh.se.

Nästa möte i gruppen blir den 14 maj.

I Ag Metodik har man kommit överens om följande mikroprojekt:

Komponentbaserad utveckling och provning

HiQ Approve
Kockums

Nya aktiviteter:

- Krav vid upphandling av COTS
- Kravställning på - COTS-leverantörens test/verifieringsaktivitet och
- resultat
- Testning/verifiering vid integration
- Kriterier för när val av COTS eller ej-COTS är det effektivaste valet

Fortsättning på ej avslutade aktiviteter

- Nätsökning - förteckning över adresser
- Begreppsdefinitioner, komponent
- Lista över standards
- Val av nivå för komponentanvändning
- Inventering av komponentmarknaden

Relevant provning

AerotechTelub
Kockums
FOI
Saab Avionics
HiSafe

Utgångspunkter:

- Begrepp, standards – ge lista, sammanfatta.
- Vad skall provas – kvalitet på verifiering kontra kostnad. Vad är tillräckligt bra.
- Verktyg.
- Hur skapar man provfall ur gränssnitts-specifikationer och krav?
- Metoder och processer

Ada-kodgenerering från systemmodeller

Saab Bofors Dynamics

Delpaket:

- Värdera SBD Linköping-studien, komplexitet i applikationsexempel, begriplighet
- Värdera Bridgepoint i förhållande till Rhapsody, komplexitet i applikationsexempel, begriplighet
- Värdera UML mot B, metoderna och notationerna mer än verktygen, skalbarhet

Tidplan

28 febr Definierat mål och avgränsningar
15 mars Skapat arbetspaket och bestämt innehåll
30 april Samlat information
31 maj Disposition till rapport
4 juni Avstämningsmöte
4 juni Detaljplan för rapport och presentation fastställs
30 sept Utkast till rapport
Höstsem Presentation
Början av dec Rapport

Ev ytterligare intresserade deltagare bör ta kontakt med Håkan Edler, edler@hisafe.se.

Inbjudan till introduktion onsdagen den 15 maj 2002 i programvarusäkerhet och H ProgSäk

Bakgrund

H ProgSäk, Handbok för programvara i säkerhetskritiska tillämpningar, utgör FM och FMV:s rekommendationer vid anskaffning av programvara i säkerhetskritiska system. Handboken fastställdes före årsskiftet att tillämpas från och med 2002-03-01.

Vid denna introduktion belyses olika aspekter på programvarusäkerhet (programvarans roll för systemsäkerheten) och H ProgSäk presenteras. Några korta diskussionsuppgifter har lagts in. Seminariet är en något längre variant av tidigare introduktion i mars detta år.

Målgrupp

Seminariet ingår som en del i kompetensutvecklingen av FMV:s personal och är också öppet för deltagare från FM, FOI samt försvarsindustri.

Genomförande

Seminariet genomförs 2002-05-15 kl 09.30 – 16.30.
Plats: FMV, Banérgatan 62, Stockholm: Filmsal C.

Kostnader

Seminariet är avgiftsfritt och deltagarna svarar själva för egna kostnader och tid.

Anmälan

Anmälan sker till Christina Gunnarsson på KC Ledstöd (för externa deltagare namn, företag samt personnummer) via mail, christina.gunnarsson@fmv.se alternativt fax 08-782 5788 senast 2002-05-08. Bekräftelse på deltagande lämnas.

Program

09.30: Kaffe
09.45: Inledning (*Chief Scientist K-G Lövstrand*)
10.00: Programvarusäkerhet – vad är det? (*Inga-Lill Bratteby-Ribbing, Strategisk specialist*)
10.45: Diskussionsuppgift 1-2: Programvarusäkerhet, Återanvändning
11.15: Tillförlitlighet och fel i programvara (*I-L B-R*)
11.45: Diskussionsuppgift 3: Programvaruspecifika egenskaper
12.00: Lunch
13.00: Programvarans roll för säkerhetskritiska system (*I-L B-R*)
13.10: Diskussionsuppgift 4: Arkitektur
13.20: Vad innehåller H ProgSäk? -Kravaspekter (*I-L B-R*)
14.00: Diskussionsuppgift 5: Språk - OS
14.10: H ProgSäk: Informativa avsnitt (*I-L B-R*)
14.20: Diskussionsuppgift 6: Felträd - Riskreduktion
14.45: Kaffe
15.00: Standarder (*I-L B-R*)
15.30: Hur skall H ProgSäk användas? (*I-L B-R*)
15.50: Diskussionsuppgift 7 : Inför säkerhetskritiska projekt
16.10: Framtid, frågor och diskussion (*I-L B-R*)
16.30: Avslutning

FÖRSVARETS MATERIELVERK

Per Magnus Wicén
C KC Ledstöd

Att arbeta med krav – referat från ”Journal of Systems Engineering”

Ingmar Ögren, iog@toolforsystems.com

I volym 4 No 3 2001 av Systems Engineering Journal fanns två intressanta artiklar om kravarbete:

- ”Att använda ett kravhanteringsverktyg för att hantera verifiering” av Hamann och Oort från universitet i Delft, respektive Fokker Space i Nederländerna
- ”Kompletthet hos krav: absolut eller relativt” av Ronald S. Carson från Boeing och Tony Shell från DarkLake Synetics i England.

Eftersom jag arbetar med modelleringsverktyget Tofs, som bl. a. hanterar krav så var detta mycket intressant och jag skrev en kommentar ”Kravhantering – en verktygskonstruktörs syn”, som har accepterats av Journal för införande i år. Den är upplagd så att den diskuterar ”krav på kravhantering” i de båda första artiklarna och kommenterar de olika kraven från min erfarenhet av kravarbete och verktygsbygge.

Referat av Haaman och Oort (H&O)

Bakgrunden till artikeln är kravarbetet vid Fokker Space där man har gått från enkla kalkylbladsverktyg till specialkonstruerade kravhanteringsverktyg. Ingen av lösningarna har varit tillfredsställande, speciellt inte när man kommit till verifiering. Erfarenheten leder till frågan om bristen på acceptabla verktyg beror av brist på efterfrågan eller på att det helt enkelt inte är möjligt att konstruera ett acceptabelt verktyg.

Processen, för kravarbete och verifieringsstyrning vid Fokker Space beskrivs sedan:

- Analysera systemkrav
- Fördela, härled och definiera krav på delsystem
- Tilldela verifieringsmetoder
- Planera och specificera verifiering
- Genomför verifiering.

Diskussionen fortsätter sedan med en detaljerad diskussion av att arbeta med kalkylblad, Oracle databas, MS Access databas och ett kommersiellt tillgängligt kravhanteringsverktyg. Inget av verktygen befanns duga för företags krav:

Specialbyggda verktyg kräver för mycket expertkunskap inom utvecklingsgruppen (databaser) eller ger för mycket utrymme för katastrofala mänskliga fel (kalkylblad), men har fördelen att de är väl anpassade till den använda utvecklingsprocessen. Kom-

mersiellt tillgängliga verktyg är enkla att använda, men kräver avsevärt extra arbete för att få att stämma med den egna utvecklingsprocessen ofta med knökiga resultat.

H&O konkretiserar ett antal krav på verktyg, återkommer till dessa nedan.

Referat av Carson & Shell (C&S)

Man utgår från en tidigare artikel av Shell som hävdar att det inte går att bevisa ”kompletthet” hos krav. Man börjar med att diskutera ”Problem statement” (problemdefinition) och finner att det inte går att bevisa att detta är komplett och korrekt.

Sedan följer en diskussion om vad som faktiskt går att göra med ett formellt arbetsätt när man utgår från en problemdefinition för att fånga upp en uppsättning krav och visa att kraven hänger ihop och inte innehåller motsägelser.

Vad som inte går att göra formellt, eller på annat sätt, är att bevisa att den ursprungliga uppgiftsbeskrivningen (problemdefinitionen) är komplett och korrekt. Detta leder ofta till att man föregriper lösningar och skriver krav inriktade mot en specifik lösning, något som kan leda till märkliga ”självreferenser”.

Slutsatsen är att det viktiga i systemarbete är att tolka ”mänskliga behov” och ”problemdefinitioner” i en formell uppsättning krav – men inte glömma den kvarstående osäkerheten vad gäller kravens kompletthet.

Referat av eget bidrag (Ö)

Mitt bidrag, med anledning av de två artiklarna började med att ställa frågan om man som verktygstillverkare skall försöka tillgodose alla krav man hittar eller försöka en "egen vision" av hur kravhantering skall gå till. Svaret var att man i första hand skall "följa visionen", men upprätthålla en levande dialog med alla intresserade i ämnet. Det är så vi alla lär oss!

Kanske kan vi börja med en diskussion på e-mail? Börja gärna med en kommentar till iog@toolforsystems.com!

Bidraget innehöll sedan dels en kommenterad genomgång av kraven, dels en kort redogörelse för den "vision" vi utgick från med verktyget Tofs.

Krav och kommentarer

Spårning av krav (H&O)

Spåra mellan enskilda krav uppåt och nedåt. Det innefattar kravnummer, nyckelord, text, version, status, motivering mm.) Detta är kravspårningsmatrisen.

Jag är tveksam till detta krav dels eftersom det förutsätter en lösning(matrisen), dels för att ger intrycket att kraven "skall ha ett eget liv", oberoende av själva systemet man arbetar med. Jag skulle hellre formulera: Verktyget skall göra det möjligt att koppla krav, dels till varandra, dels till konstruktionsobjekt (som attribut).

Sortering (H&O)

Sortering av krav för att hitta missade källor, för att hitta krav per verifieringsattribut etc.

Vad detta krav egentligen handlar om (tycker jag) är att man hela tiden behöver ha koll hur giltig den aktuella kravbilden är. Sortering kan vara en hjälp, men även andra tekniker kan behövas.

Producera rapport (HO)

Producera rapporter, som ger en överblick av alla krav med status och relationer, liksom delmängder med krav per attribut.

Korrekt, men jag tycker inte att det räcker utan att vi även behöver kravens kopplingar till och fördelning på konstruktionsobjekt så inte kravarbetet blir ett självändamål.

Förändringsindikering (H&O)

Det skall gå att indikera alla förändringar i krav eller attribut till krav.

Korrekt, men mest viktigt vad gäller "originalkrav", lite mindre viktigt för "härledda krav".

Tillgänglighet (H&O)

Kravverktyget skall vara tillgängligt "read-only" för hela utvecklingsgruppen.

Kravet avspeglar en toppstyrd organisation, med en person ansvarig för kraven. I en mera demokratisk organisation tycker jag nog att alla behöver tillgång till verktyget med rätt att uppdatera krav inom det egna ansvarsområdet.

Generera kravspecifikation (H&O)

Det skall gå att generera kravspeciar direkt från kravhanteringsverktyget.

Riktigt, men jag tycker man bör få tolka "direkt" så att verktyget får ansluta till en standard textbehandlare (annars blir det dyrt)

Verifieringsreferenser (H&O)

Anslut verifieringsreferenser till varje krav genom kravspårningsmatrisen.

Korrekt krav, men kravspårningsmatrisen kanske inte är den enda lösningen. Se nedan för alternativet "Krav/prov-matris".

En vision med tre nivåer och två sidor

Kraven från Haaman och Oort tillsammans med konstaterandet från Carson & Shell att fullständighet hos krav är en omöjlighet har orsakat en del fundering. Resultatet för min del är en vision om en kravhantering med tre nivåer och två sidor.

Först måste man gör rent hus med ett par missuppfattningar:

En missuppfattning är att kravarbete kan göras isolerat från konstruktion och verifiering. Det går inte eftersom det är omöjligt att konstruera utan att lära nytt och eftersom en del av det nya man lärt kommer att påverka kraven.

En annan är att kraven är lika med uppsättning nedskrivna krav med nummer, rubriker etc, som fastställs tidigt i ett projekt. Det räcker inte eftersom man måste formulera kraven olika, beroende på var man är i

projektet och eftersom kravbilden måste vara "levande" när kunskapen växer.

Min första slutsats blir alltså att problemen som Haaman och Oort redovisar helt enkelt beror på att själva idén med "kravhanteringsverktyg" är omöjlig eftersom kravhanteringen inte kan isoleras, och att vi snarare behöver systemmodelleringsverktyg med förmåga att hantera krav, verifieringskriterier och verifieringsresultat.

För att åskådliggöra tanken med de "tre nivåerna" kan man se ett projekt som en växande planta i Figur 1. Man kan då se tre nivåer för kravarbetet:

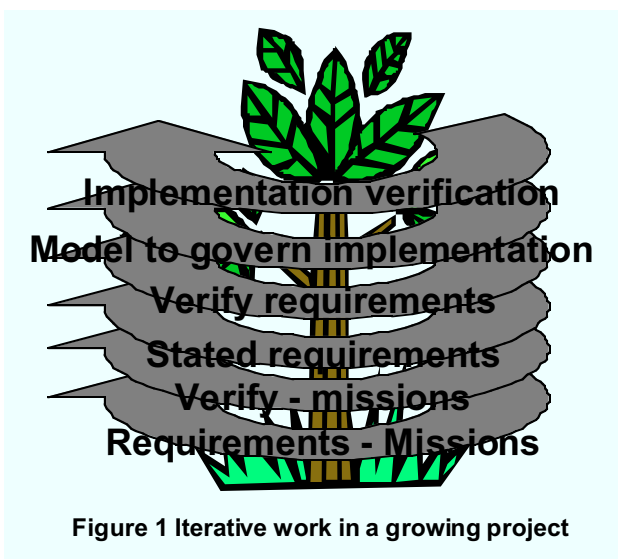


Figure 1 Iterative work in a growing project

1. Uppgiftsnivån

Det är lätt bli överens om att det är svårt att bygga rätt system om inte alla berörda är överens om vad uppgiften för systemet är. Alltså är första nivån för kravhantering att klarlägga och definiera systemets uppgift(er).

Om vi t. ex. skall bygga ett övervaknings-system kan den övergripande uppgiften vara "Skapa situationsförståelse" och en underuppgift vara "Hämta data om aktuell situation".

Notera att identifiering av underuppgifter kan vara nödvändigt för att skapa förståelse hos berörda intressenter, samtidigt som det innebär en första ansats till konstruktion – vi behöver verifiering redan här, dvs vi kan inte arbeta med "rent kravarbete"!

2. Nästa nivå när alla är överens om uppgifterna är "Numrerade nedskrivna krav", i fallet övervakningssystem t. ex.:

Nummer: 2013

Rubrik: Identifiering av övervakningsresurser

Text: Varje övervakningsresurs skall kunna presenteras på en karta med sin typ, lokalisering, täckning och status. Härlett från krav 2002.

Attribut: Härlett, Prestanda.

En uppsättning numrerade krav duger normalt som underlag för ett utvecklingssavtal även om man måste räkna med att antalet krav växer under arbetets gång när man lägger till härledda krav i samband med konstruktionsarbetet.

3. Den tredje nivån representeras av detaljkonstruktionen, som ger krav på implementeringen. Kraven kan här uttryckas t. ex. i formaliserad text (pseudokod) och/eller som UML-diagram.

Detta om nivåer för krav. De två sidorna handlar om krav contra verifiering eftersom ett krav inte är till mycket nytta om det inte kan verifieras. De tre nivåerna för krav motsvaras av nivåer även för verifiering:

1. Uppgifter kan verifieras mot scenarios som definierar typiska fall för fullgörande av uppgiften.
2. Numrerade krav verifieras mot provfall, tillsammans med andra krav och mot uppgifter.
3. Krav uttryckta i konstruktionsbeskrivningar kan verifieras mot provfall. Här är det en bra idé att fördela både krav och provfall till konstruktionsobjekt eftersom detta gör det möjligt att prova per objekt och undvika integrationsprov av typen "big bang". Figur 2 visar ett exempel på en prov/krav-matris för ett konstruktionsobjekt i verktyget Tofs. Här syns även en problembeskrivning, som används bl. a. för att uppdatera kravbilderna på ett ordnat sätt.

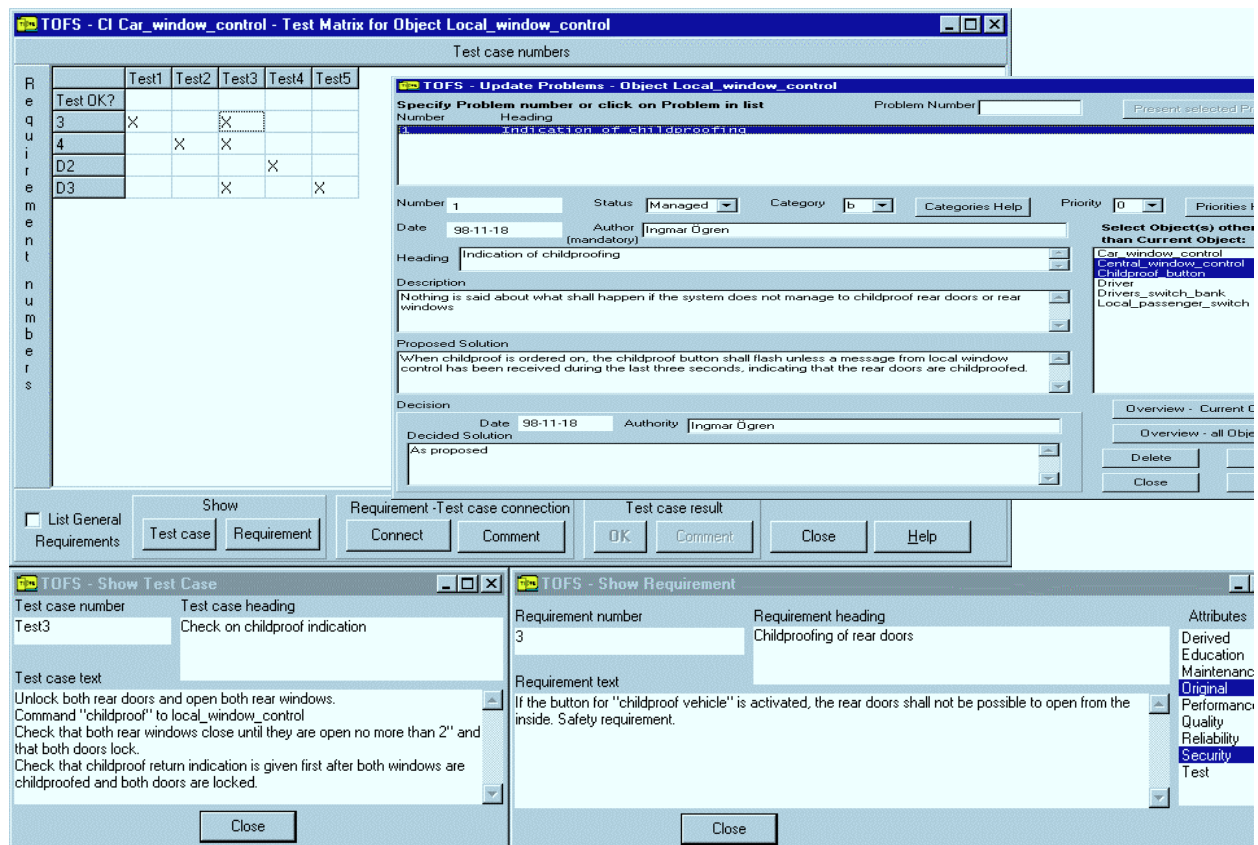


Figure 2 Requirements/test case matrix

Slutsatser

Diskussionen som utgick från de två artiklarna i "SE Journal" har givit värdefulla indata och följande slutsatser:

- "Kompleta krav" är något som knappast existerar. Bättre är att arbeta med "tillräckliga krav", där detaljnivån för vad som är tillräckligt ökar när ett projekt går framåt.
- Svårigheterna, som upplevts när man försökt använda ett "kravhanteringsverktyg" för att hantera även verifiering synes väsentligen bero på att kravarbete, konstruktion och verifiering är tre proces-

ser, som måste vara aktiva tillsammans genom ett utvecklingprojekt. Problemen med att hantera "rena kravhanteringsverktyg" blir då knappast någon överraskning.

- Om man accepterar de tre nivåerna (Uppgift, Numrerade krav och Detaljkonstruktion) och de två sidorna (Krav och Verifiering) får man ett underlag som bör duga för att bygga verktyg, som duger för att hantera krav tillsammans med verifiering.

Nya CMMI med integrerade mognadsmodeller nu ute på allvar

De flesta inom programvarubranschen känner säkert till "CMM for Software", SW-CMM, den första av en rad av "mognadsmodeller", som togs fram av Software Engineering Institute (SEI) i början på 90-talet på uppdrag av Department of Defense (DOD).

Mognadsmodeller är ett hjälpmedel att använda som ledning vid utveckling och förbättring av viktiga verksamhetsprocesser inom myndigheter och företag. Modellerna innehåller de väsentliga beståndsdelarna i effektiva processer inom olika aktuella områden. De bakomliggande idéerna till vad som bör ingå i effektiva processer kommer från märkespersoner som Crosby ("Quality is Free!"), Deming, Juran och Humphrey.

Mognadsmodeller definerar alltså krav på bra processer (arbetssätt, förfaranden), men de föreskriver inte hur de skall utföras. Modellerna skall kunna användas som underlag att ange en "road-map" för att förbättra de processer man har i företaget etc, utifrån det läge (den mognadsnivå) de befinner sig i dag, till där man vill vara vid någon tidpunkt framöver. För att konstatera var man befinner sig görs en utvärdering – assessment eller appraisal – mot modellen, kanske med hjälp av en utomstående expert.

I SW-CMM finns ju fem mognadsnivåer – Levels – 1. Initial, 2. Repeatable, 3. Defined, 4. Managed och 5. Optimizing. I den nya CMMI kan det vara litet annorlunda, se nedan. På varje Level finns ett antal Key Process Areas som identifierar vilka verksamheter som ansetts nödvändiga inom den aktuella nivån, och vilka en aktör som vill klättra på mognadsstegen bör införa, och som den värderas mot.

En process i sin tur är en följd av åtgärder för att uppnå ett givet resultat. Capability models kräver att processbeskrivningarna är dokumenterade så att det som verkligen tillämpas "på golvet" är tillgängligt som referens och för att man skall kunna utbilda nya medarbetare.

Efter den första SW-CMM 1991 som alltså avsåg styrning av programvaruutveckling, har CMM:er för en rad olika områden tillkommit, bl a för programvaruanskaffning, systems engineering, integrerad produktut-

veckling, säkerhet etc.

Även andra organisationer inspirerades av CMM-idén och tog fram, själva eller i samarbete med SEI, egna modeller, t ex Electronics Industry Affiliates (f d Association) Systems Engineering Capability Model, EIA/IS 731, och FAAs integrerade CMM, FAA-iCMM (redan 1997).

Användningen av CMM:er har efter hand spritt sig långt utanför försvarsområdet. Här hemma har väl t ex ABB och Ericsson satsat en hel del på att "CMM:a" sig. Och man kunde för flera år sedan läsa i pressen om programvaruföretag i Indien som klättrat ändå upp till den högsta mognadsnivån inom programvara, Level 5.

Inom FMV förekom på 90-talet en diskussion om att FMV inte borde beställa programvarujobb från företag som inom viss tid inte uppnått minst Level 3. Veterligen resulterade detta inte i några tvingande bestämmelser och hur läget är i dag i det avseendet är okänt för referenten. CMM-frågan diskuterades givetvis även i SESAM och kanske har företagen på egen hand jobbat vidare med att förbättra sig och bli "certifierade" enligt lämplig CMM-nivå. Det kan vara intressant att notera att det man säger sig vilja uppnå med tillämpning av CMM är detsamma som formuleras i SESAMs målsättning (från 1988), nämligen att främja tillförlitlighet och effektivitet i utveckling och vidmakthållande av programvarusystem.

Inom DoD och försvarsindustrin i USA fann man tydligen tämligen snabbt att det var ganska opraktiskt och dyrt med CMM:er på olika verksamhetsområden (discipliner) gjorda av olika organisationer på litet olika sätt, särskilt om man skulle tillämpa dem på samma projekt, verksamhet eller produkt. Därav beslutet 1997 av DoD och den amerikanska försvarsindustriföreningen att ta fram en integrerad CMM som skulle utvecklades genom att integrera förfaranden från CMM för programvaruutveckling, programvaruanskaffning, systems engineering och integrerad produktutveckling (IPPD).

Resultatet är The Capability Maturity Model Integrated – *Discipline*, där *Discipline* står för det område (eller body of knowledge)

som modellen täcker in. Den täcker in samma livscyklar som de tidigare modellerna. Hittills finns tre integrerade modeller framme:

- CMMI-SE/SW för systems engineering and software engineering
- CMMI-SE/SW/IPPD för systems engineering, software engineering, och integrated product and process development (IPPD)
- CMMI-SE/SW/IPPD/SS för systems engineering, software engineering, integrated product and process development, och supplier sourcing (ungefär anskaffning).

Utöver mognadsmodellerna har man tagit fram utvärderingsmetoder [Standard CMMISM Appraisal Method for Process Improvement (SCAMPISM)] och utbildningsmaterial m m. Första versionen av CMMI kom ut för litet över ett år sedan och den senaste Version 1.1 (V1.1) i januari i år. Denna kommer inte att ändras de närmaste tre åren, för att ge nödvändig stabilitet så att folk vågar satsa på att använda den.

Systems och Software integreras

Det finns alltså ännu inte någon CMMI-modell enbart för Software Engineering och det verkar inte riktigt klart att en sådan kommer. Man rekommenderar att de som vill täcka in programvara skall välja CMMI-SE/SW, därför att båda modellerna är uppbyggda på samma sätt och lika med undantag för att det är lite olika tyngdpunkt i täckningen beroende på om det handlar om Systems Engineering eller Software Engineering. Förmodligen har man lagt upp modellerna så, därför att med allt mer programvaruintensiva system blir systemaspekterna allt viktigare för dem som sysslar med programvara och programvaruaspekterna allt viktigare för dem som sysslar med system och de båda disciplinerna tenderar att allt mer smälta ihop. Och i den mån de fortfarande skiljer sig åt, bör det vara lätt att relatera dem till varandra. Att integrationen mellan SE och SW är ökande i verkliga projekt, är väl något vi noterar även här hemma. CMMI-SE/SW sägs fö innefatta allt det man lärt sig de senaste tio åren genom användning av SW-CMM, så programvaruaspekterna bör väl vara väl intäckta där.

SW-CMM är fö en av tre källor (source models) till CMMI-sviten tillsammans med EIA 731 och SE-CMM vilka båda avsåg Systems Engineering.

Om man också vill täcka in att verksamheten bedrivs enligt IPPD-principer väljer

man CMMI-SE/SW/IPPD, och om man dessutom har att hantera anskaffningsfrågor (supplier sourcing), väljer man CMMI-SE/SW/IPPD/SS.

Meningen är alltid att modellerna skall anpassas (bli "tailored") till de förhållanden som gäller för det sammanhang de skall tillämpas i, men man varnar också för oförsiktiga anpassningar p g a det ofta höga inbördes beroendet mellan olika förfaranden i modellerna.

Efter hand kommer förmodligen en hel skara modeller för olika "bodies of knowledge" att komma fram, men där de alla har ett gemensamt ramverk, gemensamma komponenter, regler och terminologi för hur man konstruerar och använder CMM:er. Därigenom elimineras redundans och inkonsistenser jämfört med om man använt skilda modeller och man sparar tid och pengar på utbildning, utvärdering etc. En annan fördel kan vara att gemenskapen uppmuntrar och underlättar samarbete mellan olika discipliner både betr hur utvärderingar skall gå till och betr uppläggning av processkartor och utveckling. Många processproblem beror på dåliga kontaktytor mellan olika grupper/enheter inom organisationen. Detta bör kunna undvikas/mildras genom tillgång till ett gemensamt, integrerat ramverk. Vidare har man sett till att få konsistens med ISO 15504 (Software Process Improvement and Capability dTermination Model (SPICE)).

Andra nyheter i CMMI

Det finns naturligtvis utöver integrationsaspekterna en del skillnader mellan uppläggningsen av CMMI och de tidigare modellerna.

Två modellrepresentationer, kontinuerlig eller stegvis

I CMMI kan man välja mellan två s k modellrepresentationer Continuous och Staged (\approx stegvis), eller välja båda. Den kontinuerliga är tänkt att användas om man vill rikta in sig på förbättringar bara i vissa processområden (Process capabilities), i princip oberoende av varandra, medan den stegvisa används om man vill lyfta organisationens mognadsnivå samtidigt för alla processer inom ett visst kunskapsområde (ex programvaruutveckling) jämt över hela organisationen (Organizational maturity). Den kontinuerliga representationen medger alltså att organisationen kan förbättra olika processer olika snabbt, beroende t ex på var man bedömer det mest angeläget, vilket säkert ökar benägenheten att ge sig in på

CMM-området. Orsaken till dessa två varianter är att man vill underlätta för dem som tillämpat EIA 731 (Systems Engineering) som är processinriktad och SW-CMM (Software Engineering) som är organisationsinriktad.

Modellernas beståndsdelar

De ingående komponenterna i den kontinuerliga och den stegvisa modellen är desamma: process areas, specific goals, specific practices, generic goals, generic practices, typical work products, subpractices, notes, discipline amplifications, generic practice elaborations, och references.

I den stegvisa representationen är process areas fördelade på fem stycken mognadsnivåer vilket anger vilka process areas som måste finnas implementerade för att man skall uppnå en viss av de fem definierade mognadsnivåerna - Level: 1. Initial, 2. Repeatable, 3. Defined, 4. Managed och 5. Optimizing.

I den kontinuerliga representationen används som *organisationsprinciper* för modellkomponenterna sex stycken *capability levels* (vilka anger en rekommenderad ordningsföljd för förbättringar inom en process area), *capability profiles* (som är listor över processområden med tillhörande capability levels som hjälper organisationen att följa förbättringsarbetet per processområde), *target staging* (en följd av target profiles som beskriver den förbättringsväg som man vill följa) och *equivalent staging* (en översättning om någon skulle önska det, enligt särskilda regler, av en uppnådd kapabilitetsprofil till en mognadsnivå).

Den kontinuerliga representationen fördelar process areas på fyra processkategorier (Process Management, Project Management, Engineering och Support) med fem till åtta process areas var och anger kapabilitetsnivåer för processförbättringar inom varje process area. T ex ingår i kategorin Engineering följande process areas: Requirements Management, Requirements Development, Technical Solution, Product Integration, Verification och Validation.

De sex kapabilitetsnivåerna i den kontinuerliga representationen är: 0. Incomplete, 1. Performed, 2. Managed, 3. Defined, 4. Quantitatively Managed, 5. Optimizing.

Assessment blir Appraisal

I CMMI har man övergått till att kalla "utvärderingar" Appraisal och inte som i CMM

Assessment. Detta är väl litet av finlir med terminologin, men sägs återspegla ett verkligt skifte från en "discovery" ansats i CMM till nu en "verify and validate" ansats. Enligt utförda försök skall detta påtagligt minska utvärderingstiderna, vilket varit ett av problemen med CMM-användningen.

Man har ökat flexibiliteten i utvärderingsomfattningen genom att definiera tre nivåer av appraisals. Class A är den fullständiga utvärderingen som skall uppfylla alla krav i utvärderingsspecifikationen Appraisal Requirements for CMMI (ARC) och utförs enligt Standard CMMI Appraisal Method for Process Improvement (SCAMPI_{SM}). SCAMPI skall ge "benchmark quality" värderingar relativt CMMI-modeller. Class B och C behöver endast uppfylla delar av ARC. Vad man väljer beror på syftet och omfattningen av utvärderingen, bl a hur officiell (externt gångbar) den måste vara. Det finns uppskattningar av insatsbehovet för dessa utvärderingar om för Class A 4-10 personer (under kanske två veckor), för Class B 1-6 personer (en vecka) och Class C 1-2 personer (några timmar), plus en ledare. Class A utvärderare måste vara auktoriserade av SEI och är de enda som kan utföra en "officiellt giltig" värdering.

Konsistens med övriga standarder

Det finns ett antal standarder av olika status och räckvidd som berör samma område som CMMI. Exempel är ISO 9000-2000, den förut nämnda ISO/IEC 15504 (SPICE), den blivande ISO/IEC 15288 Life Cycle Processes, ISO/IEC 12207, mer "inhemska" amerikanska standarder som EIA 731, FAA-iCMM, förutom de tidigare SEI-CMM:erna. Det finns alltså potentiella konsistens-, kompatibilitets- och kanske även konkurrensproblem för CMMI.

Ett definitivt krav på CMMI är att den skall vara konsistent och kompatibel med ISO/IEC 15504 (Software Process Improvement and Capability determination Model (SPICE) när denna blir fastställd (troligen 2003) och detta har man hanterat på olika sätt, bl a genom att CMMI-personer även varit inblandade i berörda ISO/IEC-verksamheter.

I en artikel i CrossTalks februarinummer om CMMI uttrycker en av författarna förståelse att de som redan gått igenom ekluten och arbetar efter, eller håller på att införa någon av de andra standarderna och nu skulle vara tvungna att gå över till CMMI, kan gripas av uppgivenhet. Dock framhålls att dessa andra standarder, t ex ISO 9000 för det mesta

föreskriver ungefär samma eller identiska saker, varför att föra in en ny modell bör ses som mappning och anpassning snarare än att börja om på nytt.

Har CMMI någon plats i svensk försvarssektor?

Detta kunde kanske vara ett ämne för diskussion i kommande Rendezvouser.

Ovanstående redogörelse är baserad på studier av dokument från Software Engineering Institute (SEI) och artiklar i CrossTalks februarinummer 2002, vars huvudtema var CMMI.

En mängd material om och kring The Capability Maturity Model®(CMM®) IntegrationSM (CMMISM), inkl de olika modellerna och annan relaterad dokumentation kan laddas hem från <www.sei.cmu.edu/cmmi>

CrossTalk kan laddas ner från www.stsc.hill.af.mil/crosstalk/crosstalk.html

Anm: CMM and Capability Maturity Model are registered in the U.S. Patent and Trademark Office.

CMM Integration, CMMI, and SCAMPI are service marks of Carnegie Mellon University.

(En personlig not: Den bok av Crosby, på den tiden ansvarig för kvalitetsfrågor i ITTs koncernledning, som var en av inspirationskällorna för CMM, kom ut i USA 1979. Det var på den tid referenten ännu var inblandad i att ta fram verkliga system. På något sätt fick jag tag på boken direkt den kom ut, läste den och beställde 10 ex av den som jag skickade som julklapp till VD:arna för våra största leverantörer (inom flyg- och elektronikindustrin). Intressant var att se reaktionerna från mottagarna; en av dem ringde t ex omedelbart tillbaka och frågade om jag hade något att anmärka på kvaliteten hos deras produkter. Detta bokköp på skattebetalarnas bekostnad tror jag var en mycket kostnadseffektiv investering. Crosby tillhandahöll t ex väl sifferunderbyggda exempel på vad man tjänade t ex i form av mindre re-work, returer etc (med alla sina följdkostnader) på att satsa på kvalitetshöjande åtgärder innan leverans (t ex komponentscreening och inbränning); han visade verkligen även för våra leverantörer, då mest aktuellt var serieutrustningar till JA 37, att "Quality is Free!". Tyvärr var jag oförsiktig nog att låna ut mitt eget ex av boken till någon på verket; det har fortfarande inte kommit tillbaka.)

I Carlsson

Ada NYTT via Ada Information Clearinghouse

<http://www.adaic.org/>

För erfarna programmerare som vill bekanta sig med Ada; bok gratis på nätet

Ada Distilled: An Introduction to Ada Programming Features for Experienced Computer Programmers

Författaren, Richard Riehle på AdaWorks Software Engineering, välkänd Ada-förespråkare i USA, har gjort denna bok gratis tillgänglig på nätet.

Han karaktäriserar boken som att vara riktad till erfarna programmerare som vill lära sig Ada på programmeringsnivån. Den är inte en "... for Dummies" bok och inte heller är den avsedd för programvarukonstruktion. I stället fokuserar den på vissa nyckelgenskaper hos Ada-språket, med kodade

exempel, som är väsentliga för att komma igång som Ada-programmerare.

Riehle konstaterar att Ada är ett rikt och flexibelt språk som används för att konstruera stora programvarusystem. Boken lägger tyngdpunkt på syntax, styrstrukturer, underprogramregler och hur-att-koda aspekter snarare än konstruktionsfrågor. Betr sådana hänvisar han till flera bra böcker som behandlar programkonstruktion. Ada Distilled är inte heltäckande för Ada-språket, utan kan

ses som en "quick start" bok som gör det möjligt för den erfarna programmeraren att snabbt och enkelt komma in i Ada.

Innehållsförteckningen:

1. WHAT IS ADA DISTILLED?
2. SUMMARY OF LANGUAGE
3. TYPES AND THE TYPE MODEL
4. CONTROL STRUCTURES FOR ALGORITHMS
5. ACCESS TYPES (POINTERS)
6. SUBPROGRAMS
7. PACKAGE DESIGN
8. CHILD LIBRARY UNITS
9. OBJECT-ORIENTED PROGRAMMING WITH PACKAGES
10. USING STANDARD LIBRARIES

11. EXCEPTION MANAGEMENT
 12. GENERIC COMPONENTS
 13. NEW NAMES FROM OLD ONES
 14. CONCURRENCY WITH TASKING
- A. ANNEXES, APPENDICES AND STANDARD LIBRARIES
- ANNEX K (INFORMATIVE): LANGUAGE-DEFINED ATTRIBUTES
- ANNEX L PRAGMAS - LANGUAGE-DEFINED COMPILER DIRECTIVES
- B. WINDOWS 95 AND NT CONSOLE PACKAGE
- C. BIBLIOGRAPHY

Ada Distilled är på ca 100 sidor och kan laddas ner via <http://www.adaic.org/free/freebook.html>

För nybörjare: Utgången bok om objekt-orienterad programmering med Ada 95 nu fri på nätet

Ada 95: The Craft of Object-Oriented Programming

Det börjar finnas ett antal exempel på att några år gamla böcker som gått ur trycket börjar bli gratis tillgängliga på eller via nätet. En sådan bok med fortfarande full aktualitet finns t ex också med på SESAM 2001 CDn, nämligen Engineering Complex Systems with Models and Objects av David Oliver, Timothy P. Kelliher och James G. Keegan, Jr.

Det fungerar tydligen så i förlagsvärlden att när en bok går ur trycket, kan copyrighten återgå helt till författarna, som kan göra vad de vill med materialet.

John English har på detta sätt under 2001 på nätet lagt ut sin bok Ada 95: The Craft of Object-Oriented Programming, vilken ursprungligen utgavs av Prentice Hall 1996.

I det här fallet kan man antingen läsa boken i html-format på nätet, eller ladda hem den som html-fil och läsa den på skärmen hemmavid. I html-utgåvorna, varav den senaste är från februari 2002, har English dessutom rättat ett antal fel som fanns i de tryckta utgåvorna av boken. Att vem som helst får tillgång till boken via nätet betyder dock inte att man får göra vad som helst med den, men för personligt bruk är det OK. För annan spridning fodras oftast tillstånd av författarna, vilket SESAM t ex fick för Systems Engineering boken på SESAM CDn.

I sitt företal förklarar English på ett övertygande sätt varför han valt att tala om hantverk (craft) när det gäller programutveckling och inte konst eller vetenskap, vilket annars ofta förekommer. Bara det resonemanget är värt en titt.

Bokens uppläggning är "exempeldriven", med ett begränsat antal exempel som successivt förbättras och förfinas genom att han lägger till "bells and whistles" efterhand. De två största exemplen är i sina slutversioner inte mer än ca 1000 rader, men illustrerar ändå de underhållsproblem man kan stöta på och hur noggrann konstruktion kan lindra dessa.

English bok, i motsats till Riehles som riktade sig till erfarna programmerare, är helt inriktad på en nybörjare som skall lära sig programmera med Ada som första språk.

Boken har tre huvuddelar; den första behandlar fundamenta betr Ada-programmering, den andra abstrakta datatyper och den tredje handlar om hur man använder Ada 95s objekt-orienterade egenskaper för att utveckla utvidgningsbara program. Dessutom finns det fyra appendix; det första ger en informell sammanfattning av Ada 95 syntax, den andra ger specifikationerna för de standardpaket som används i texten, det tredje anger språkdefinierade attribut och det fjärde listar de paket som utvecklats i boken.

Boken kan läsas på <http://www.adaic.org/docs/craft/html/contents.htm> och laddas ner från <http://www.it.bton.ac.uk/staff/je/adacraft/bookhtml.zip> (i zip format för Windows) eller från <http://www.it.bton.ac.uk/staff/je/adacraft/bookhtml.tar.gz> (gzippad tarball för Unix system). Varje distribution innehåller också de fullständiga exemplen ur boken både för Windows (adacraft.zip) och för Unix (adacraft.tar.gz).

The Consolidated Ada Reference Manual

Denna manual från Springer Verlag kombinerar Ada 95 standarden (ISO/IEC 8652:1995(E)) från 1995 och de över 200 ändringar, rättelser och klarlägganden som finns i Technical Corrigendum 1 (ISO/IEC 8652:1995/Cor.1:2001(E)) daterad June 2001 och som nu är en del av den officiella Ada-standarderna.

Eftersom ISO har valt att inte själva lägga ihop Corrigendum med den ursprungliga standarden till ett dokument, har Springer för att underlätta för läsarna att inte behöva hantera två dokument jämsides, gett ut en

”konsoliderad” manual. Men den är alltså inte ett officiellt ISO-dokument, även om man har gjort sitt bästa för att komma så nära ett sådant dokument som möjligt, om det hade funnits.

The Consolidated Ada Reference Manual kan beställas från Springer-Verlag på www.springer.de, eller i bokhandeln.

Den kan också gratis laddas ner i pdf- och html-format från Ada Information Clearinghouse på <http://www.adaic.org/standards/95lrm/RM.pdf> (eller.html)

Mer Ada-Nytt

För dem som har svårt att glömma den första, mindre lyckade Ariane 5-uppskjutningen i mitten på 90-talet

BURLINGTON, MASS (MARCH 8, 2002)—The Ada Resource Association (ARA) announced another Ada success story today with the Ariane 511’s launch of the Envisat 1 within the one-second tolerance allowed for the Sun-synchronized spacecraft. As the biggest satellite ever built in Europe, Envisat 1 also boasts being the largest environment survey satellite ever launched.

“The extraordinary accuracy of the Ariane’s launch once again validates Ada for realtime use,” said S. Tucker Taft, President of the ARA.

With flight control systems in Ada, the Ariane 511 for the first time succeeded in launching a “polar” satellite; i.e. towards the North instead of the traditional easterly direction. The Envisat 1 also depends on many Ada realtime software subsystems, including the radar altimeter.

The Envi sat mission plan set a series of very specific and narrow windows for the launch and orbit. For example, the time for the Kourou, French Guiana, launch was 22:07 and 59 seconds, and Ada made sure that Flight 145 lifted off at one second before 22:08.

Also, fuel-saving algorithms in Ada allowed the engine to turn off when the Envisat 1 reached its precise orbit three seconds before the scheduled time. It began orbiting within the right 100 meters where tolerance was 7.5 kilometers. As a result, the 26-meter long (when deployed), 8211-kilogram rocket carries a minimum of fuel.

For the next five years, the satellite will record data on how humans are affecting Earth’s health, including through patterns of land use and its effect on soil moisture and fertility; the quantity and size of ice flows; and the amount of ozone and other chemicals in the atmosphere observed on the planet’s “limb” or edge.

Developed in a European Space Agency program, the satellite will be operated from ESA’s European Space Operations Center (ESOC) in Darmstadt, Germany. Europe’s Astrium led an industry consortium of 50 companies to produce the Envisat spacecraft.

For more information on Ada and the Envisat, please write to

Ann Brandon, Communications Director,
Ada Resource Assoc. abrandon@sover.net

Kalender

2002-04-25	SESAM Rådsmöte	
2002-04-25	Ada i Sveriges vårseminarium	Ta gärna en titt på vår websajt
2002-06-12	SESAM VU	
2002-08-15	Höstsem och Rådsmöte	
2002-10-03	SESAM VU	http://sesam.tranet.fmv.se
2002-10-22	SESAM Tutorial kurs	
2002-10-23	SESAM Höstsem	där finns konferenser för hela året uppräknade
2002-10-24	SESAM redovisn + Rådsmöte	
2002-11-28	SESAM VU	

Konferenser utomlands

020501-03	3rd IFIP Working Conference on Infrastructures for Virtual Enterprises, Algarve
020506-08	2002 10th International Symposium on Hardware/Software Co-Design, CODES'02, Stanley Hotel, Estes Park, CO http://www.codesign-symposium.org
020506-08	IFIP / SEC 2002: 17th International Conference on Information Security, Kairo
020512-15	PADS '02: ACM/IEEE/SCS 16th Workshop on Parallel and Distributed Simulation, Arlington , VA
020513-15	MS '02: IASTED International Conference on Modelling and Simulation, Marina del Rey , CA
020515-18	3rd International Conference on Integrated Formal Methods 2002 (IFM'2002). Åbo
020519-24	XML Europe 2002 Conference & Exposition, Barcelona
020519-25	International Conference on Software Engineering (ICSE'2002) Orlando, FL, USA OBS flyttat från Buenos Aires, Argentina
020526-29	3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP'2002) Alghero, Sardinien
020527-31	14th Conference on Advanced Information Systems Engineering (CAiSE'02), Toronto, Kanada
020604-07	2002 IEEE 8th International Software Metrics Symposium, Ottawa, Kanada http://www.software-metrics.org
020606-08	ECIS'02: The 10th European Conference on Information Systems Gdansk
020609-12	7th European Conference on Software Quality Helsingfors
020610-14	16th European Conference on Object-Oriented Programming (ECOOP'2002), Málaga
020610-14	DAC '02: 39th Design Automation Conference New Orleans , Louisiana
020613-14	PPPJ '02: Principles and Practice of Programming in Java Dublin
020617-21	7th International Conference on Reliable Software Technologies - Ada-Europe 2002, Wien
020619-21	ACM SIGPLAN Joint Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES'02) and Software and Compilers for Embedded Systems (SCOPES'02), Berlin

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
-----------------------------	--	---