

# RENDEZVOUS

NYHETSBLAD FÖR SESAM  
Försvarssektorns Adaintressenters Användargrupp för Software Engineering

*Nr 1 april 1999*

## *Innehåll*

Ordföranden har ordet.....	3
Nya arbetsgrupperna fick flygande start .....	4
Rivstart och roadmap för nya .....	5
SESAM/Ag/Teknik .....	5
Forskning och Teknikutveckling för Anpassning (FoTA) - aktuellt läge .....	6
Handbok för Programvara i säkerhetskritiska tillämpningar (H ProdSäk) .....	7
A Comparison of the Concurrency and Real-Time Features of Ada 95 and Java .....	7
Experiences from systems engineering education.....	8
AWESOME-skivan .....	12
Kalender .....	12
Bidrag till efterlyses .....	12

# 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 härfor 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. 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 Björn Källberg, CelsiusTech Systems, tel 08-580 84813.

### VU

Bengtsson Christopher, FMV  
chben@tranet.fmv.se

Brandt Roger, FMV

robra@fmv.se

Eckfeldt Sune, Enator Telub AB

sune.eckfeldt@enator.se

Johansson Billy, CelsiusTech Electronics AB

bijo@celsiustech.se

Källberg Björn, CelsiusTech Systems AB

bjkae@celsiustech.se

Arbetet utförs i ett två arbetsgrupper:

Ag Metodik

Håkan Edler, CTH/Datorteknik

edler@ce.chalmers.se

Ag Teknik

Torbjörn Andreasson,

Ericsson Microwave Systems AB

torbjorn.andreasson@emw.ericsson.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

Anna Kåsjö, FMV:INFOSYST, 115 88 STOCKHOLM

Tel: 08-782 6745, Fax: 08-66 77 392

Epost: alkas@tranet.fmv.se

# Ordföranden har ordet

## ”Hur e’ leget?”

är ju en vanlig hälsningsfras i Stockholm, liksom det vanliga svaret att allt är bra. Stämmer detta också på Sesamfronten och Adafronten?. Jag skall försöka ge några motiverade synpunkter.

På SESAM-fronten har det praktiska arbetet i de nya arbetsgrupperna under ledning av Håkan Edler och Torbjörn Andreasson lyckats rivstarta på ett utmärkt sätt. Ett flertal s.k. mikroprojekt har definierats, och arbetsgruppsmedlemmarna har börjat arbeta i dem. Resultaten av mikroprojekten resulterar ju dels i rapporter, och kommer också att presenteras på Sesamseminariet den 21-22 oktober. Du har väl markerat tiden i din almanacka?.

Om det praktiska arbetet går snabbt, så gäller det inte den administrativa, formella sidan, som draggar efter. Alla intressenter har inte ännu skrivit på det nya avtalet. Inom det verkställande utskottet arbetar vi hårt på att få detta klart till det kommande rådsmötet.

Vad gäller på Adafronten i stort? Som alla kan se är ju inte Ada det största språket, och kommer nog inte heller att bli det. Det är givetvis synd, men inte heller mer än så. Mest synd är det om dem, som ännu inte kan nyttja fördelarna som Ada ger. Ett språk behöver inte heller vara störst. Vad som behövs är att det är tillräckligt stort. Tillräckligt stort för att kunna ge näring åt ett antal kompilatorleverantörer. Och Ada har definitivt denna storlek. På de stora plattformarna, Windows och Unix, finns det ett flertal utmärkta och billiga kompilatorer, och även för realtids-

plattformar finns det tillgång till bra kompilatorer, fler än för de flesta andra språk.

Och de egenskaper som Ada har, och som främjar god Software Engineering, börjar att uppskattas i allt vidare kretsar, även sådana som inte använder Ada. Säkerhet är den sådan egenskap, speciellt viktig inom militärområdet. Andra Adaegenskaper har också börjat sprida sig till andra språk. Inbyggda realtidsprimitiver, generic (templates), kontroll över namnrymden, modularisering är några exempel, där Ada varit och är ledande.

Även de andra Adaorganisationerna har en hälsosam verksamhet. Ada i Sveriges årsseminarium i samarbete med Sesam den 22 april är en hållpunkt. Och glöm inte Ada Europes konferens i Santander på Spaniens västkust den 7-11 juni.

Ett exempel på Adas och Sesams ställning visas av en av de nyare medlemmarna i Sesams råd. När han tillträdde trodde han att Ada var något som knappast användes, och att Sesam borde snarast avvecklas. Vid närmare undersökning visade det sig, att hans bolag hade flera pågående Adaprojekt än någonsin, och hans bedömning av Sesams verksamhet ändrades till att bli mycket positiv. Han är numer en av de mest aktiva inom Sesam.

Så svaret på inledningsfrågan är ett övervägt:

## *Läget är bra!*

Björn Källberg

# Nya arbetsgrupperna fick flygande start

Den 2 februari hade de två nya arbetsgrupperna Metodik och Teknik sina första, delvis gemensamma, möten vid Ericsson Microwave i Mölndal.

Dessa grupper ersätter de tidigare fem arbetsgrupperna enligt rådets beslut i oktober. Grupperna prövar ett helt nytt arbetssätt som innebär att man definierar ett antal "mikroprojekt" där minst två intressenters representanter samverkar för att lösa en definierad uppgift, t ex att göra en utredning, prova en idé etc. Uppgifterna skall vara aktuella frågeställningar eller problem som intressenterna har och söker svar på i samverkan med andra intressenter.

Varje grupp är tänkt att ha tre obligatoriska gemensamma möten, ett första startmöte där man beslutar vilka miniprojekt man skall driva, ett andra uppföljningsmöte ungefär mitt i mikroprojektet och ett tvådagars seminarium/internat (SESAMS höstseminarium) där resultaten redovisas och diskuteras. Resterande tid enligt avtalet används för hemarbete, kontakter med samarbetspartner och arbetsgruppsordföranden via e-post (eller direkta möten) etc, som fordras för att lösa den gemensamma uppgiften. De allmänna anvisningarna för det nya arbetssättet finns på SESAM hemsida.

Ag Metodik beslutade vid sitt startmöte att sammanföra de 19 st ursprungliga projektförslagen till fyra st, två om V&V: granskningsmetodik resp provningsmetodik, ett om kravhantering och användningsfall samt ett om granskningar. Målen är - mycket ambitiösa - att om möjligt kunna ställa nytta mot kostnad för ett antal tekniker för V&V resp betr kravhantering och granskningar. Vid startmötet deltog ett drygt dussin personer.

Ordf i Ag Metodik är Håkan Edler, edler@ce.chalmers.se och v. ordf. Gustaf Myhrman, gumyh@fmv.se, Försvarets materielverk

Ag Teknik beslutade att ur 6st industriinitierade projektförslag som Mikroprojekt'99 välja att arbeta med frågeställningar kring HLA: HLA -

vad/varför/hur?

HLA - High Level Architecture är som bekant den nya standard för distribuerad simulering som amerikanska försvaret tar fram (med sikte på att göra den till en internationell standard). Även Försvarmakten siktar på att utnyttja HLA.

Dessutom framlades vid startmötet 5 st universitetsrelaterade förslag och idéer som ansågs vara av metodikkarakter; dessa överlämnas till rådet och VU för vidare behandling.

I Ag Teknik startmöte deltog ett femtontal personer.

Ordföranden i Ag Teknik är Torbjörn.Andreasson@emw.ericsson.se, Ericsson Microwave Systems AB och v. ordf. Gunnar Fredriksson, gufre@fmv.se, Försvarets materielverk.

## Förstärkning via projektmedel

Det kan finnas möjlighet att förstärka eller utöka arbetsgruppernas verksamhet genom utnyttjande av SESAMs projektmedel (enl avtalet totalt 500.000kr/år). Rådet beslutar om projektmedlens utnyttjande. Ett moderniserat "ansökningsformulär" för dessa medel är under utarbetande.

Anm: HLA-intresserade kan hitta intressant information på föreningen Modellering och Simulering i Sverige (MoSiS) hemsida <http://www.mosis.a.se>.

## Till den som missat Ag-starten

Den som av någon anledning missat att anmäla sig till någon av arbetsgrupperna bör snarast kontakta Håkan eller Torbjörn för att komma med i något av delprojekten, alternativt för att tillsammans med andra sent anmälda bilda nya delprojekt.

Rådsmedlemmarna som är ansvariga att tillse att intressenterna har avtalsmässig tidsbemannning i arbetsgrupperna bör se över anmälningsläget.

# Rivstart och roadmap för nya SESAM/Ag/Teknik

TORBJÖRN ANDREASSON, ORDFÖRANDE AG/TEKNIK

Historien har gång på gång visat att under de "tubulenta tiderna" så har de största innovationerna. Så mot bakgrund av den turbulens som för tillfället råder inom branschen torde SESAM-arbetet med att samla, skapa och sprida information (S3I) ha fantastisk potential att påverka den framtida försvarsrelaterade systemutvecklingen.

Under året 1999 kommer förutom beslutsmötet Mikroprojekt'99, den 990202, på Ericsson Microwave AB i Mölndal, även följande möten att genomföras

- Uppföljningsmöte Mikroprojekt'99, den 990601
- Internat/seminarie Mikroprojekt'99, den 991021-22

Önskar därmed kandidat för värdskapet till uppföljningsmötet den 990601 - någon frivillig?

Jag antar att Rådet och VU återkommer med detaljer rörande 2-dagars internatet/seminariet.

Vid beslutsmötet den 990202 beslutade samtliga Ag/Teknik-representanter att genomföra ett flertal olika arbetspaket i syfte att studera begreppet HLA.

## **Mikroprojekt '99: HLA - vad/varför/hur?**

För att kunna styra och övervaka stora simuleringar har det amerikanska försvaret tagit fram HLA (High Level Architecture). Sedan tidigare finns en äldre standard kallad DIS (Distributed Interactive Simulation). I USA har man fattat beslut om en fullständig övergång till HLA och man säger till och med 'No Pay - No Play'. Detta innebär att efter 1999-01-01 kommer det inte att läggas ut några fler beställningar på simuleringsprogram som inte använder sig av HLA och efter

2001-01-01 kommer inte sammansatta simuleringar inom försvaret att förekomma annat än under styrningar av HLA-program. Det svenska försvaret följer den amerikanska utveckling och ser också ett införande av HLA som en trolig väg.

Mikroprojektet syftar till utreda och sammanställa frågor, svar, tips och tricks för att underlätta förståelse och ett eventuellt införande av HLA i konkreta försvarsrelaterade projekt. Läs mer

<http://sesam.tranet.fmv.se/Teknik/index.htm>

## **Utdata**

1. Draft av rapporten tillgänglig vid uppföljningsmötet den 990601.
2. Muntlig och skriftlig rapport/redovisning vid det försvarsindustrigemensamma SESAM-seminariet den 21-22/10-1999.
3. Artikel (skriftliga rapporten) publiceras i SESAM-bladet Rendezvous.

## **Arbetsinsats**

En person per deltagande SESAM-företag/organisation.

- 1 arbetsdag - beslutsmöte mikroprojekt den 9/2-1999
- 1 arbetsdag - uppföljningsmöte under månads-skiftet maj/juni-1999
- 2 arbetsdgr - SESAM-seminarie/internat den 21-22/10-1999
- 6 arbetsdgr - att disponera för studier

## **Delleveranser**

1. Den 15/3-99: Avstämning genom utbyte av delrapporter
2. Den 15/5-99: Sammanställning av utkast till rapport

# Forskning och Teknikutveckling för Anpassning (FoTA) - aktuellt läge

Enligt Bror-Arne Ersson, FMV sammanhållande för FoT-programmet för anpassning, har FMV nu skickat kundbeställning (KB) från Försvarsmakten på alla programvaruförslagen i FoTA-programmet, utom ett (P. 7). Dessutom har ett förslag (P. 2) fallit från på grund av att förslagsställaren vid industrin inte kunnat driva det vidare.

Det enda projekt som i skrivande stund är beställt från FMV och pågår är MANA-projektet (P. 10) vid Uppsala Universitet. Det projekt som enligt ovan, FMV ännu inte har offererat och således inte har KB på, är COTS-produkter i militära lednings- och informationssystem (P. 7). Det väntas dock bli offererat från FMV i dagarna.

För övriga projekt hänger det nu främst på FMVs handläggare och resp projektförslags industrisammanhållande när verksamheten kommer igång.

Här följer en lista på alla förslagen med uppgift om handläggare på FMV resp industrins sammanhållande.

[P. 2. Verktyg och komponentutveckling med ASIS (Ada Semantic Interface Specification). Har utgått.]

## **P. 3. COTS och objektorientering som bas för konstruktion av realtidsapplikationer**

FMV handläggare:

Gustaf Myhrman, UVsystS,  
gumyh@fmv.se

Industrins kontaktpersoner:

Torbjorn.Andreasson@emw.ericsson.se,  
Lennart.Bie@emw.ericsson.se

## **P. 4. Patterns och komponentåteranvändning**

FMV handläggare:

Christopher Bengtsson INFOSYST,  
chben@fmv.se

Industrins kontaktperson:

Jonas Jerfsten, jonasj@st.se

## **P. 7. COTS-produkter I militära lednings- och informationssystem**

FMV handläggare:

Göran Kihlström, ElektroT,  
gokih@mv.se

Industrins kontaktperson:

Bengt Gustafsson,  
begu@celsiustech.se

## **P. 9. Experimentell verifiering av feltolerans**

FMV handläggare:

Gunnar Fredriksson, FlygE,  
gufre@fmv.se

Industrins kontaktperson:

Håkan Edler, edler@hisafe.se

## **P. 10. Projekt MANA - Ett run-time system för säkerhetskritiska komplexa system**

(Kompletteringsprojekt till NUTEK Komplexa Tekniska Systems projekt "A Run-Time System for Safety Critical Complex Systems")

FMV handläggare:

Gunnar Fredriksson, FlygE,  
gufre@fmv.se

Industrins kontaktperson:

Lars Asplund,  
asplund@Minsk.DoCS.UU.SE

## **P. 11. Formalisering, analys och hantering av krav på säkerhetskritiska system**

FMV handläggare:

Per-Henrik Persson, VapenP,  
phper@fmv.se

Industrins kontaktperson: Göran Anger,

anger@L4i.se

## **P. 12. Överföring till industrin av programvaruteknik för säkerhetskritiska system**

FMV handläggare:

Inga-Lill Bratteby-Ribbing, InfosystT

Industrins kontaktperson:

Dag.Folkesson@saab.se

P.12 är ett litet annorlunda projekt eftersom det är avsett att interagera med andra projekt som behandlar frågor som kan ha bäring på utveckling av säkerhetskritiska programvara, i första hand P. 9, P. 10 och P. 11, men förmodligen har även andra projekt sådana inslag. P.12 är särskilt intressant eftersom det kan ge tillfälle att få belysa en del av anvisningarna i FMV:s färskare remiss-

utgåva av en handbok för anskaffning av säkerhetskritiska system med programvara (se notis på annan plats) och därför torde ett brett deltagande från företagen vara önskvärt.

Tidsmässigt behöver väl de andra projekten ett visst försprång i förhållande till P. 12, men det kan ändå vara bra att börja förbereda det projektet så snart som möjligt.

---

## Handbok för Programvara i säkerhetskritiska tillämpningar (H ProdSäk)

H ProdSäk ger riktlinjer för hur systemsäkerhetsverksamheten skall tillämpas på programvara vid anskaffning av säkerhetskritiska system. Handboken är inte inriktad mot någon viss applikationsdomän. Den listar generella säkerhetskrav m a p programvarans produkter samt de personer, processer och produktionsmiljö som medverkat vid framställningen. Den ger också anvisningar för hur säkerhetskrav specifika för applikationsdomän och projekt kan härledas.

En remissutgåva färdigställdes 99-02-28. Denna är en vidareutveckling av en förstudieutgåva från 97-09-30 och har bla försetts med nya och omformulerade krav, vilka numrerats och kritikalitetsgraderats. Även refererade säkerhetsstandarder och övriga informativa avsnitt har uppdaterats.

Under nästa etapp kommer granskningskommentarer inkomna senast 99-04-23 att ligga till grund för ytterligare kompletteringar. Avsikten är att några projekt efter en introduktion i handboken skall få prova nästa version.

Den som är intresserad av utgåva 2.0 kan vända sig till Anette Marke, FMV:InfosystT, tel 08-782 6620, anmar@fmv.se.

Synpunkter och förslag på provprojekt (det senare till 99-03-30) kan lämnas direkt till mig.

Inga-Lill Bratteby-Ribbing, FMV  
(ilbra@fmv.se)

---

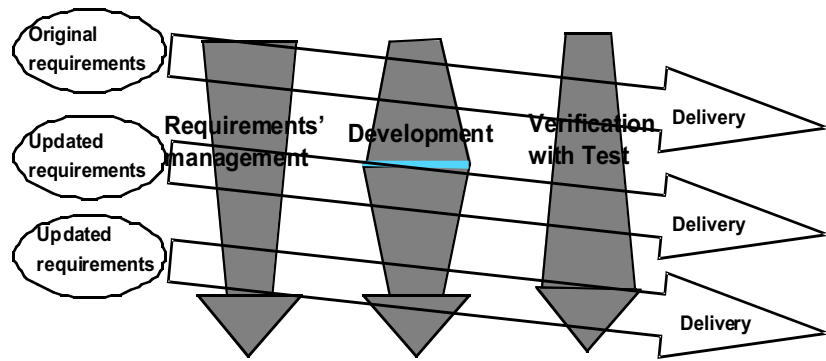
## A Comparison of the Concurrency and Real-Time Features of Ada 95 and Java

För dem som inte har tillgång till Ada User Journal, kan nämnas att dess januarinummer innehåller en intressant jämförelse mellan det stöd Ada resp. Java ger för hantering av (potentiellt) parallella processer och realtid. Förutom att ge en allmän översikt över Javas egenskaper och förklara hur de skiljer sig från Adas, beskriver Ben Brosgols artikel detaljerat egenskaperna hos Adas "tasks" resp. Javas "threads" och ger en mängd jämförande programexempel.

Ref. Ada User Journal. The journal for the international Ada community. Volume 19 Number 4 January 1999. ISBN 1381-8551. Produced by Ada Language UK Ltd and Ada-Europe.

# Experiences from systems engineering education

Ingmar Ogren, Tofs corporation,  
iog@toolforsystems.com,  
www.toolforsystems.com



Concurrent processes for progressive systems engineering

**It is now more than ten years** since I started teaching systems engineering courses. However, in the beginning, I believed I was teaching software engineering, but then things started to happen:

- I saw that the students preferred “humanized” examples and also that it was possible to use the software engineering tools to describe formalized human behavior.
- I found that you cannot really work with the software without considering its context. This meant adding a part to the course: how you understand the software’s representation and understanding of its environment.
- Even if the course examples were small, I started to understand that the real problem was not in the coding, but in system understanding, when system volume

and complexity grew beyond what was understandable to one person at one time.

The result was that the software engineering course grew into a systems engineering course. This gave mixed reactions: Some student were nice and told me that this is real systems engineering while others just asked: “shouldn’t we start coding soon?”

**The three-day limit is a problem.** Industrial courses are normally limited to three days and in this time the teacher is expected to give a basic understanding of systems engineering as well as working knowledge of a systems engineering tool. I found that this situation leads to a risk that the student’s get a number of “information pieces”, soon forgotten since these are not very useful without the general understanding.

However, some means to create a meaningful three-day course are:

- Make them understand that the understanding problem is central.
- Make them understand that systems engineering is a team effort (Integrated Product Teams)
- Use a small, but not trivial example (no more than one page of specification)
- Use an intuitive tool, which complies with Microsoft Office.

**Make them understand that new development is rare and that requirements, design and verification must be managed together.**

In the software and systems engineering words, most methods and tools presuppose that each development project starts anew with a fresh set of requirements. Even if such projects do exist, they are like unicorns: everyone knows what it looks like, but few have met one.

The conclusion is that it is



important to teach methods, which support system update, rather than new development. Further, people used to believe in separate timed phases for requirements management, design and verification of systems. We now know that this does not work, for reasons such as:

- You need to verify the requirements as soon as they are defined.
- During analysis and design, knowledge will grow and cause changed requirements.
- During verification, problems will surface and cause changes in requirements and design.

The conclusion is that, although it may not be obvious to all students, you should try to find methods and tools to support concurrent systems engineering as outlined in the above figure.

**Declare your basic modeling principle** and make them understand it.

There are several modeling principles, which are useful for systems engineering work. The simplest is the block schema principle, which lies behind “Structured Analysis”. Another, which supports component reuse is “Inheritance-based object orientation”, where you may use UML class diagrams (UML = Unified Model Language by Rational Software).

However, my favorite is to model according to principle of

“compositive object orientation”, since this principle gives clear dependencies among the objects in the system and gives you a system structure, which clarifies how the different objects in a system contribute to completion of the system’s missions.

Even if all of these principles are useful for systems engineering, it will be confusing if you mix the different principles in a course. Decide on your favorite and stick to it!

**Make them understand the process of building an information repository.**

Since systems engineering mainly concerns building a common understanding among multiple stakeholders, concerned with a system, documentation is extremely important. In modern systems engineering this understanding can be based on a central computer-stored model of the system, from which you can draw the necessary information to create different documents.

It is further important to make the students understand that efficient systems engineering cannot be achieved through just writing documents, but that you need to build, review, cross-check and update the information structure, which then becomes a “backbone” of the systems engineering project.

To build this basic understanding, it helps if you can let the students use a software tool to store the information structure.

**Similarities and differences between systems and software engineering.**

Since software engineering is a much larger profession than systems engineering and since there are similarities, it is not unusual that software professionals “slide into” systems engineering. This is not too bad, since there are important similarities, for example:

- Both software and systems engineering are very much concerned with the problem of mastering complexity.
- Several methods, originated in software engineering, such as reviews, formalization, etc. are applicable to systems engineering.
- Management of interfaces and concepts are of crucial importance both in software and systems engineering.
- Efficient software engineering as well as systems engineering, requires that multiple concurrent processes be established.

However there are also differences, which can cause conflicts and misunderstandings such as:

- Software engineers tend to believe that a “system” is only a set of cooperating software modules. For a system engineer this is nonsense since some hardware and most often some operator actions will be necessary

to make the software do something useful

- The view of operators being “outside” is expressed in the software community’s fascination with “use cases”, putting the operator outside the software system, while a systems engineer normally prefers to concentrate on the system’s missions.

The conclusion is that there are fundamental similarities and some subtle differences between software and systems engineering. Since many of the students in a systems engineering course have a software engineering background, it is important to elicit the similarities and differences.

### **Concepts are important.**

Unless you know what you are talking about, you will have difficulties to make yourself understood and you may even get confused yourself. Systems engineering will often require new concepts, which are not very widely known or understood.

The traditional software engineering solution is to work with “data dictionaries”. This however is of little help to systems engineering, since the traditional data dictionary does not do very much besides naming the data items used with an informal description.

What you need is a type catalogue, where each concept is formalized, as a type, with its possible values and possible operations.

Further, some concepts may not be suited to formalize as types. These should be put into a glossary in order to build a common understanding through a project.

**Standards are helpful**, but beware of using them untailed. Standards are good servants, but horrible masters. A standard becomes a master when you accept the whole standard uncritically and allow it to govern your work into completion of every DID (Data Item Description). You will then risk slipping of schedules, unnecessary work, write-only documentation and bad reputation for the standard used.

If, on the other hand, you view the standards concerned as checklists, you can take advantage of the efforts put into the standard and use just these parts, which are relevant for your project.

If you manage to make your students understand this principle for using standards, they will be able to make standards their useful servants.

### **Teams are important.**

The traditional teaching situation includes the situation where each student works on his or her own to complete personal work, which is then accepted or rejected by you as a teacher. This is dangerous practice for systems engineering, since it may lead the students to believe that systems engineering can be completed single-handed. Systems engineering must be a team effort, except for trivial

systems, since the number of disciplines involved, combined with system volume and complexity goes beyond what a single individual can manage.

The conclusion is that you must create an educational situation, where the students work together in “Integrated product teams” and get a good understanding of the necessity to work together to get an acceptable product.

A special case of cooperation is the teaching of reviews, since it is not always a nice situation to have your product inspected and gone through. To avoid making the students nervous in this situation, I use the following principle for a review exercise:

- 1) The students inspect a “teacher’s” solution of the course example.
- 2) I make a viewgraph copy of the inspection material and select a few students for the roles in a review.
- 3) We do a short walk-through meeting in front of the class around the overhead projector.

This simple exercise makes the students eager to find errors without feeling awkward through having their own immature efforts inspected. It also allows me to comment on any “misbehavior” by the students in their roles as inspector, review leader and secretary.

**Tooling is an issue** of crucial importance in systems engineering education.

It is possible to teach systems engineering without a software tool. However this is not advisable, since it is important that the students learn that systems engineering tools are just that and are to be used by systems engineers, not secretaries. When selecting a tool cost is an issue, since you want to keep the course cost down. Some other important points are:

- The tool must be intuitive and fun to use or they won't use it.
- The tools should support concurrent processes for requirements management, design and verification.
- The tool should comply with commonly known principles for Office tools to avoid that most of the course time goes into making the students understand the tool.

**System safety is another issue.** Inexperienced students tend to believe:

- Software will always work as specified.
- The hardware will work forever.
- Operators will behave as expected by the system and software designers.

Experience teaches us that none of this is true and that the result can be system failures, which may be serious in critical systems.

The conclusion is that systems engineering education should include methods to manage criticality issues such as formalism and fault-tolerance.

**Students are different**, and some of them will never become systems engineers.

When you present for example a formal description of operator behavior to the students as a state chart or a piece of pseudo code, you will get different reactions, such as:

- That is code, leave it to the keyboard monkeys.
- You mean that you can define this operator role formally. That is neat!

My conclusion is that the first person may well do useful work, but rarely become a real systems engineer while the second person is a good candidate to become a real systems engineer.

## AWESOME-skivan

Genom en mycket hård spurt av de inblandade, främst Jesper Andersson och hans kompisar på högskolan i Växjö, men också många andra, hölls med ett par dagars marginal (den reviderade) målsättningen att få ut denna CD-ROM i tid till Ada-Europe-konferensen den 8 juni. Skivan blev till slut mycket innehållsrik och en bra exponent också för programvarukunskan i försvarssektorn. Huvudändamålet var dock att få fram ett utbildningspaket och en kvalificerad gratis utvecklingsmiljö på PC, inkl två Ada95-kompilatorer och en Ada/Java bytecode kompilator. Alla SESAM-medlemmar skall ha fått sitt eget ex av skivan. Ca 2000 ex har spritts till SESAM-intressenterna och en del andra intresserade, 500 ex sprider AiS till vissa utvalda gymnasieskolor.

Restlagret uppgår f n till ca 500 ex som sekretariatet gärna vill bli av med till något nyttigt ändamål. Spekulanter bör kontakta Anna Kåsjö.

## Bidrag till Rendezvous och SESAM hemsida efterlyses

Varken Rendezvous eller hemsidan blir bättre än vad medlemmarna gör den till. Håll ögon och öron öppna för intressanta händelser i eller nära SESAM-världen och skicka in en notis till sekretariatet, [alkas@tranet.fmv.se](mailto:alkas@tranet.fmv.se). När det är aktuellt att kunna publicera rapporter eller beskrivningar från projekt som Ni är inblandade i, glöm då inte våra egna medier.

<http://sesam.tranet.fmv.se>

### Kalender

- |                |   |
|----------------|---|
| 1999-04-27     | SESAM Rådsmöte, FMV, TrV rum K401, Stockholm  |
| 1999-05-02--06 | Software Technology Conference , "Software and Systems for the Next Millennium", Salt Lake City, Utah |
| 1999-05-16--22 | ICSE'99 - 1999 International Conference on Software Engineering, Los Angeles, CA                      |
| 1999-06-07--11 | Ada-Europe '99 Conference, Santander, Spanien   |