



Försvarets Historiska Telesamlingar
Marinen och Flygvapnet



2016-02-15

Philips i Järfälla

Från lokal tillverkare av apparater till systemleverantör på världsarenan

En industrihistorisk tillbakablick från åren 1960 till 2000
med tonvikt på programvaruutvecklingen

Bertil Eklund med flera

M 01/2016



Omslagets framsida:

Philips industrifastighet i Järfälla i slutet av 1980-talet

Omslagets baksida:

Medverkande vid framtagningen av detta dokument.

På bilden står i övre raden från vänster:

Ingemar Carlsson

Thomas Jungeföldt

Bengt Olofsson

Lars Fransson

Bertil Eklund

Sven Bidö

Per-Anders Nilsson

Ulf Olsson

Hans Gustafson

Stellan Borg

Nedre raden från vänster:

Malte Jönson

Göran Granberg

Tomas Ahlberg

Hans Ljungqvist

Göran Pettersson

Redaktionskommitté:

Göran Granberg, Bertil Eklund (f.d. Philips)

Malte Jönson, Bengt Olofsson (f.d. FMV)

Bilder och fotografier kommer i huvudsak från Marinen eller Philips

Förord

Inför Försvarets Historiska Telesamlingar 30-årsjubileum 2014-09-25 sammanställdes dokumentet *Försvarelektronik från svenska leverantörer Företagen – produkterna – samhällsnyttan* som översiktligt redovisar de större elektronikföretagen och deras produkter. I dessa produkter kom datorer och programvara att i allt högre grad svara för funktionaliteten – och för den totala livstidskostnaden. Dator- och programutvecklingen, utvecklingen av programmeringsmetoder och hjälpmedel liksom kompetensuppbyggnaden i företagen fick dock inte den framträdande roll i sammanställningen som man kunde önska sig.

Målet för FHT verksamhet är dels att dokumentera utrustningar, apparater och system som avvecklats och dels medverka vid urvalet av vilka som ska bevaras vid museer. Hitintills har dock dokumenteringen av program och programproduktion skett i väldigt begränsad omfattning och i stort har ingen programvara (främst källkod) till de avvecklade systemen bevarats (arkiverats vid Krigsarkivet eller de stora museerna).

För att inte värdefull kunskap om den stora utvecklingen inom dator- och programvaruområdet för marina ledningssystem ska falla i glömska tog vi kontakt med Tomas Ahlberg och Göran Granberg, båda tidigare vid Philips, för att om möjligt få hjälp med "historieskrivningen". Det visade sig att intresset hos f.d. Philipsanställda att dokumentera Philips Elekt-

ronikindustrier AB (och de efterföljande företagskonstellationernas) industrihistoria var stort. Ett antal personer, som tidigare och i betydande grad lett eller medverkat i utvecklingsarbetet, har på ett förnämligt sätt lämnat bidrag inom sina respektive verksamhetsområden. Denna typ av sammanställning hade svårigen kunnat göras utan medverkan från f.d. anställda.

Vi hoppas och tror att detta dokument kommer att vara en mycket intressant läsning för dem som varit med om denna utveckling eller följt den från sidan och som kanske inte vetat om eller nu glömt bort vilka svårigheter som förelegat, vilka problem man ställts inför och vilka avgörande beslut som tagits och som lett till den framgångsrika utveckling som företaget haft. Vi hoppas också att detta dokument kan vara till nytta när man längre fram i tiden vill undersöka hur datorer och programvaror för realtidsapplikationer utvecklades under perioden från 1960 fram till 2000 och även att dokumentet kan vara till nytta inom Saab för att visa vad som tidigare producerats och som nu utgör grund för vidareutvecklingen.

Vi tackar alla skribenter och andra medverkande som på olika sätt hjälpt till med att ta fram detta dokument. Ett speciellt tack till Bertil Eklund som svarat för huvuddelen av arbetet med att foga samman alla bidrag till en enhetlig skrift.

Malte Jönson
medlem FHT maringrupp

Bengt Olofsson
medlem FHT marin- och flyggrupp

Philips i Järfälla.
**Från lokal tillverkare av apparater
till systemleverantör på världsarenan**

En industrihistorisk tillbakablick från åren 1960 till 2000
med tonvikt på programvaruutvecklingen

Innehållsförteckning

1. Översikt samt andra aspekter än de rent tekniska	8
Översiktlig historik	8
Ekonomi-, marknads- och kompetensutvecklingmässiga aspekter	13
2. Analogimaskiner och övergången till de första datorbaserade systemen.	14
Analogimaskiner	14
Första projektet med datorbaserade system	14
Inför övergången till datoriserade ledningssystem	15
3. Första generationens datoriserade system. 9LV200 Mk2	16
Systemarkitektur.....	16
Datorer och datorutveckling	17
Operativsystem.....	18
Diagnostikprogramvara.....	18
Programutvecklingshjälpmedel	18
Programspråk	19
4. Utvecklingen till 9LV200 Mk2.5.....	20
Systemarkitektur.....	20
Datorer och datorutveckling	20
Operativsystem.....	20
Diagnostikprogramvara.....	20
Programutvecklingshjälpmedel	20
Programspråk	20
5. Stina: Philips blir ledningssystemleverantör	22
Arvet.....	22
Arkitekturen.....	22
Erfarenheterna	23
6. Det stora steget: Fartygssystem 2000 (9LV Mk3). Arkitekturaspecter	25
Arvet.....	25
Utmaningen	25
Fundamentala egenskaper	26
Erfarenheter	28
Addendum. Underlag och historik avseende beslutet om Ada.....	29
7. Fartygssystem byggda på BS2000/FS2000 (9LV Mk 3)	32
Systemarkitektur.....	32

Datorer och datorutveckling	33
Operativsystem.....	34
Diagnostik.....	34
Programutvecklingshjälpmedel	34
Programmeringsspråk.....	35
Utnyttjande	35
Spin-offs	38
8. Vidareutveckling till 9LV Mk3E	39
Systemarkitektur.....	39
Datorer och datorutveckling	39
Operativsystem.....	39
Diagnostik.....	40
Programutvecklingshjälpmedel	40
Programvara	40
Användning.....	40
Teknisk utveckling av fartygssystemfamiljen	42
9. Ledningssystem för Flygvapnet, STRIC och Strics	43
Bakgrund	43
Initial utveckling av STRIC baserad på BS2000	44
Övergång till COTS	45
Simulatorsystemet Strics.....	48
Vidmakthållande och Vidareutvecklingsprojektet VASS.....	49
10. Datorutveckling på maskinvarusidan.....	51
11. Teknisk databehandling/stödsystem	56
Inledning.....	56
1950- och 1960-talet.....	56
1970-talet	57
1980-talet	60
12. Förkortningslista	62

1. Översikt samt andra aspekter än de rent tekniska

(Författare: Bertil Eklund m.fl.)

Detta dokument redovisar hur dåvarande PTAB, Philips Teleindustri (senare Philips Elektronikindustrier PEAB, Bofors Electronics BEAB, Nobeltech och CelsiusTech innan det blev Saab fr.o.m. 2000-07-01) utvecklades från en tillverkare av mindre system och apparater på 1960-talet till i huvudsak den svenska marinen, till att på 1990-talet bli en teknikledande systemleverantör på världsmarknaden. Bland företagets olika verksamheter är det endast systemsidan, d.v.s. produktområdet med lednings- och elledningssystem, som redovisas här och inom detta bara de projekt som skapade stora teknik- och kunskapslyft. Gemensamt och centralt för de flesta av dessa projekt är att de omfattade utveckling av alltmer komplexa programvarusystem. Många andra lyckade projekt skulle också kunna beskrivas, men de låg antingen inom andra produktområden eller utgjorde inte samma tekniska utmaningar.

Den svenska hemmamarknaden var fram till början av 1990-talet relativt skyddad genom att kunden ofta föreskrev eller siktade mot nationella särlösningar. Däremot fanns det ett flertal konkurrerande inhemska företag. På elledningssidan var PEAB ganska ensamt och ohotat, medan stridslednings- och datakommunikationsfunktionalitet även utvecklades av Stansaab (som sedermera blev såväl Datasaab som Ericssons H-division). En hård konkurrens och understundom rivalitet fanns mellan företagen fram t.o.m. FMV:s upphandling av STRIC. Dåvarande Bofors Electronics, BEAB, fick beställningen och inkorporerade därpå Ericssons H-division, varvid ytterligare spetskompetens inom bl.a. sensordatafusion tillfördes företaget.

Länge fanns även en hård konkurrens inom spaningsradarsegmentet mellan Ericssons

Mölnaldsdivision och PEAB. Efter Ericssons framgångsrika Girafferadar och FMV:s val av Ericssons artillerilokaliseringsradar ARTHUR, lämnade PEAB området med långräckviddig spaningsradar. Däremot bibehölls och vidareutvecklades följeradarn med Ku-bands hoppfrekvensmagnetron på fartygssiktet. Under samma period introducerades den "viskande" närspaningsradarn PILOT.

Översiktlig historik

PTAB började sin verksamhet i början av 1950-talet med bl.a. leverans av torpedledningssystem till svenska marinen. Under 1960-talets första hälft vann PTAB ett par centrala upphandlingar, vilka möjliggjorde att företaget kunde göra ett kompetenslyft och etablera sig som den ledande leverantören av elledningssystem till den svenska marinen. Samtidigt skapades också en standardisering av elektroniken i s.k. B-moduler, vilket väsentligt ökade styrningen mot återanvändbarhet. Upphandlingarna omfattade sjöfrontselledningen Arte 719 till kustartilleriet, samt ytmåls- och luftvärnsledningssystemet Arte 722 till flottan. Försvarsmakten utgjorde PTAB:s dominerande kund, men viss export skedde även, främst till de nordiska länderna.

Projekten baserades på ett eller flera av följande centrala teknik- och kompetensområden:

- Radar med hoppfrekvensmagnetron för högre störhållfasthet och bättre noggrannhet
- Fartygssikte med hydraulmotorer, vilket skapade större snabbhet och högre precision
- Analogmaskiner med modulariserade elektronikkort, där 400 Hz räknepänningar med hjälp av resolverar hanterade omräkningar mellan polära och cartesiska ko-

ordinatsystem. Dessutom fanns olika typer av vippor och register varigenom additioner, t.ex. uppdatering av position med avseende på hastighet, kunde genomföras

- Integration av olika produkter till sammanhängande totalsystem. Ingående produkter kom främst från det egna företags områden med kompetenstorn enligt ovanstående punkter. Produkter från andra leverantörer bestod mest av komponenter och apparater på detaljnivå

Eldledningarna var förvånansvärt noggranna och kunderna nöjda, v.v. se kapitel 2. Dock fanns ett behov av att tillföra vissa basala stridsledningsfunktioner, vilket var svårt att åstadkomma med analogimaskiner. Det framryckande teknikkonceptet med datorer skulle kunna åstadkomma önskvärt resultat samt möjligen också utökad återanvändning och en kostnadsreduktion relativt analogimaskiner. En fascinerande utveckling vidtog, vilken förde PTAB till att bli en teknikledande aktör – inte bara i Sverige utan på hela världsarenan. Vidareutvecklingen, som huvudsakligen berör dator- och programvaruutveckling, kan grovt indelas i nedanstående faser, i vilka produktportföljen samtidigt utvidgades:

- Senare hälften av 1970-talet: Teknik- och kompetenssprång genom införandet av datorbaserade marina eldledningssystem för sjöfronts- och luftvärnsartilleri
- Första hälften av 1980-talet: Vidareutveckling av marina ledningssystem med stridslednings-, datakommunikations- och eldledningsfunktionalitet
- Senare hälften av 1980-talet och första åren på 1990-talet: Tekniksprång i form av distribuerade och strikt modulariserade programvarusystem baserade på Ada, vilket möjliggjorde utökad återanvändning, spårbarhet och kvalitet
- 1990-talet: Utökad produktområde med luftförsvarssystemet STRIC. Övergång till

att använda kommersiellt tillgängliga maskin- och programvaruprodukter som datorplattformar för målmaskiner

Datorerna på 1960- och 1970-talen användes nästan uteslutande till ADB, med hålkort som in- och utmatningsmedium, utan några som helst krav på realtidsprestanda. Beräkningarna fick ta den tid de tog! I PTABs fall var scenariot helt annorlunda där ett kraftigt rullande och stampande fartyg skulle kunna styra ut sikte och kanon mot ett luftmål med hög hastighet, samt att därefter följa målet med hög noggrannhet.

Företagets målbild för fartygssystemet var att kunna uppnå ovanstående krävande realtidsprestanda och samtidigt kunna komplettera systemet med vissa grundläggande stridsledningsfunktioner. Ett av FMV och PTAB gemensamt initierat testprojekt i form av SELMA, Simulator Eldledning Marinen, tillkom för att utröna om den datorbaserade vägen var framkomlig. Utfallet blev så positivt att FMV i mitten av 1970-talet beställde närmare ett 30-tal datorbaserade system som tillsammans skulle komplettera de analoga systemen Arte 719 och Arte 722.

Som framgår av kapitel 3 lyckades PTAB att framgångsrikt leverera de datorbaserade systemen Arte 724 till kustartilleriet och Arte 726 till flottan. Systemen hade i stort sett samma funktionalitet som de analoga föregångarna, dock med skillnaden att Arte 726 även hade en stridsledningsfunktionalitet som saknades i Arte 722. En hel del vedermödor krävdes för att uppnå det framgångsrika resultatet, bl.a. ett omfattande övertidsarbete för att möta överoptimistiska programvaruutvecklingskalkyler och därtill hörande tidplaner. Det verkliga tekniklyftet var emellertid att lyckas hantera realtidsproblematiken i det fartygsbaserade Arte 726. Grunden var ett egenutvecklat realsystemoperativ (RTM), vilket framsynt byggde på parallella processer

och intelligent avbrottsantering. PTAB etablerade sig därvid som framgångsrik leverantör av datorbaserade eldledningssystem med kompletterande stridsledningsfunktionalitet, till skillnad från flertalet konkurrenter vilka fruktlöst kämpade med att få monolitiska programvarusystem att klara realtidskraven.

Som kuriosas kan nämnas att Arte 726-eldledningen indirekt fick största beröm av den högste chefen för det sovjetiska militärområdet på andra sidan Östersjön. Vid ett av de sällsynta och tämligen stela utbytena besökte denne sin svenska kollega i slutet av 1970-talet. Skjutning med Arte 726 och Bofors 57-millimeterskanon förevisades. Som vanligt var skjutresultatet utomordentligt. Den sovjetiske generalen dristade sig då till att påstå att allt var arrangerat med hjälp av radiostyrda explosioner i målet. Ett bättre beröm för Arte 726 är nog svårt att uppnå!

Bemanningsmässigt bestod respektive projekt bara av ett par eller några få medarbetare. Kostnadseffektiviteten var stor men även sårbarheten. Som tur var personalomsättningen närmast obefintlig, sannolikt beroende på det starka engagemang och delaktighet som medarbetarna kände. I vanligtvis gemensamma luncher skapades en lärande organisation samt en närhet till sektionsledningen. Någon ur företagsledningen brukade veckovis vara närvarande för att följa upp framsteg och eventuella problem; en fantastisk möjlighet att få synas för unga medarbetare.

I perioden efter Arte 724 och Arte 726 fortsatte datorsystem liksom tillhörande maskin- och programvarusystem att utvecklas i en rasande takt, v.v. se kapitel 4 om 9LV200 Mk2.5! Fortfarande fanns det i systemen en central huvuddator, men för att klara belastningen infördes också ett antal slavprocessorer. I takt med att stridsledningsfunktionerna utökades blev antalet programvaruutvecklare allt flera. Ifrån att ha tillverkat det mesta "in house"

drev den civila utvecklingen på att alltmer integrera externa produkter i PTABs system. Integration av programvarusystem blev en nyckelkompetens – här skulle många olika "konstnärer" och oliktankande enas mot ett gemensamt mål. Att kunna förena egenskaper som innovativ förmåga med följsamhet mot regler och bestämmelser samt en med kunden ensad och gemensam kravbild är inte alla medarbetare förunnat!

De tekniska framstegen hos 9LV200, vilket numera de fartygsbaserade systemen kallades, uppmärksammades av ett flertal fartygsvarv, vilka bidrog till att PEAB (som bytt namn från PTAB) lyckades att exportera 9LV200 till ett antal länder i Asien. Företaget hade tagit en inbrott på världsmarknaden, vilket krävde känsla att hantera det nya läget liksom den allt argare konkurrensen från de stora drakarna. (I parallell lyckades PEAB:s produktområde motmedel med bedriften att under god lönsamhet utvidga marknaden till att även inkludera USA, UK, Tyskland och Frankrike!).

Genom FMV:s beställning på Stina Sverige vidgade PEAB produktportföljen inom systemsidan till att även omfatta landbaserade sjöbevakningscentraler.

I takt med att datoriseringen utvecklades, möjliggjordes en inflation av krav på nya funktioner som skulle realiseras med hjälp av programvara. En utländsk kund till 9LV200 imponerades av Stina Sverige, föreslog och krävde ett ledningssystem för kustförsvaret med flera olika typer av centraler, v.v. se kapitel 5. Denna exportversion, Stina Export, fick en väsentligt utökad funktionalitet och projektet blev därför mycket mera programvaruintensivt än de dittillsvarande. Fler än dussintalet programvaru- och testingenjörer erfordrades. Programvaruutveckling visade sig vara svårbedömd då uppskattning av återstående insats (integration, verifiering, rättningsomgångar) ofta grovt underskattades. Bedömd utveck-

lingsinsats och tidplan för Stina Export visade sig bägge vara överoptimistiska. Detta medförde en tung ekonomisk belastning på företaget, men efter fullbordad och godkänd leverans, också ett teknik- och kompetenslyft vad gäller funktionalitet avseende stridslednings- samt datakommunikationssystem.

I upphandlingen av en provcentral för MASIK (Marinens stridsledning i krig) kunde PEAB till stor del bygga på funktionaliteten i exportversionen av Stina och lyckades därvid etablera sig som leverantör till FMV även inom områdena avancerad stridsledning och datakommunikation.

Datoriseringen hade nu vid början/mitten av 1980-talet fått fullt genomslag, vilket medförde att allt högre ledningsfunktionalitet inklusive stridslednings- och datakommunikationsfunktioner samt integration av spännande "hyllprodukter" förväntades. Bistra erfarenheter från övergången till en högre nivå av programspråk (RTL/2), liksom "explosionen" av stridsledningsfunktionalitet och därtill kopplad utvecklingsinsats i samband med Stina Export, gjorde att framtida teknikstrategi noga övervägdes.

Företaget hade vid denna tidpunkt i sin produktportfölj alla delar till ett komplett marint eldlednings- och stridsledningssystem. Kravbilderna ökade dock successivt och därtill kopplade krav på allt större utvecklingsinsatser. Behovet av återanvändning accentuerades alltmer för att skapa ekonomisk konkurrenskraft. I mitten av 1980-talet lyckades företaget att uppnå beställningar på inalles inemot 20-talet (med optioner 30-talet) komplexa fartygsbaserade lednings- och eldledningssystem från Sverige, Danmark och Finland. Efter mycket övervägande och i samklang med tekniktrenden valde PEAB:s företagsledning att helhjärtat satsa på Ada, ett programspråk, fast ännu mycket mera en strukturerad och kvalitetsmässigt högtstående metod för att skapa

spårbarhet och återanvändning. Arkitekturmässigt togs det stora steget till distribuerade system, där ett flertal processorer arbetar samtidigt och oberoende. Principiellt finns då ingen övre gräns för programvarusystemens storlek. Kapitel 6 beskriver överväganden och aktiviteter för att skapa återanvändbarhet, spårbarhet, hög kvalitet och hantering av disparata kravbilder liksom valet av Ada. Beslutet att använda Ada togs sent efter projektstarten efter samråd med beställarna, v.v. se addendum till kapitel 6. Viktiga potentiella kunder krävde Ada samtidigt som företaget inledde sin största satsning någonsin i systemområdet baserat på ett föråldrat programspråk. Det sena skiftet blev möjligt då ett omfattande avtal om leverans av utvecklingshjälpmedel och stöd kunde träffas med Rational, ett företag i Silicon Valley, som besatt unika kunskaper och produkter inom området. Strategiskt medförde detta investeringsbeslut ett antal fördelar:

- Satsningen på den nya systemgenerationen framtidsäkrades
- PEAB kunde på allvar ta upp konkurrensen om ett deltagande i STRILC-90. Avgörande för framtiden inom ledningssystemområdet
- PEAB kunde på allvar ta upp konkurrensen om ANZAC ship för Australien och Nya Zeeland
- Ökad konkurrenskraft genom ökad återanvändning av mjukvara och lägre livstidskostnader för kunderna

Kravbilderna var förvånansvärt olika i de tre samtida projekten från tre nordiska länder. T.ex. skilde sig danska StandardFlex och svenska Kkv Göteborg arkitekturmässigt på några väsentliga punkter:

- StandardFlex operatörsplatser placerades uppe på bryggan och utformades därför i raster-scan-teknik ("TV-skärm" och därigenom även färg!) för att klara visning i dagsljusmiljö. Kkv Göteborgs operatörs-

platser återfinns i stridsledningscentralen med dämpad belysning inne i fartyget, varför den dittillsvarande random-scan-tekniken ("radarskärm") kunde användas

- Kravet på StandardFlex att kunna utföra en antal olika roller gjorde att C3-systemet blev som en spindel i ett nät. Bearbetning av alla insignaler från sensorer eller utsignaler till dessa skedde integrerat i C3-systemet. T.ex. var sonarens operatörsplats integrerad i C3-systemet. FMV:s strävan var i Kkv Göteborg tvärtemot att flytta bearbetning så långt ut mot de olika sensorerna/vapnen som var möjligt

Kapitel 7 beskriver de olika fartygsprojekt i 9LV Mk3-familjen som baserades på denna utveckling. En hektisk och understundom frustrerande utvecklingsperiod med fler än hundra utvecklare vidtog. Företaget hade verkligen "spänt bågen" och lyckades efter stora vedermödor att leverera det första systemet till Danmark i slutet av 1989. Därefter skedde motsvarande leveranser till Sverige och Finland i rask takt.

De lyckade leveranserna skapade genklang både bland slutanvändare och varv. Marknaden utökades markant. Som kronan på verket erhöll PEAB en beställning från Australien och Nya Zeeland på lednings- och eldlednings-system till ANZAC-fregatterna.

Den rasande snabba tekniska utvecklingen på den civila sidan kom till nytta då Mk3-familjen vidareutvecklades till Mk3E. Kapitel 8 redovisar skillnader liksom några projekt inom Mk3E-familjen.

PEAB:s fokus hade under många år varit marina system. Såväl eldlednings- som stridslednings- och datakommunikationsfunktionalitet behärskades till fullo. Denna kompetens kvalificerade företaget, vilket bytt namn till Bofors Electronics, att i konkurrens med Ericssons H-division seriöst offerera det svenska luftstrids-

ledningssystemet STRIC. Företaget gick segrande ur striden om denna för företaget avgörande upphandling, v.v. se kapitel 9. Beslutet innebar även att Ericssons H-division sedermera uppköptes och inkorporerades i Bofors Electronics. Kompetens på stora ledningssystem erhöles då definitivt, liksom världsledande spetskompetenser inom bland annat sensordatafusion.

Vid realiseringen av STRIC ändrade FMV kravbild till att enbart omfatta fasta centraler. Därvid kunde kommersiella plattformar användas, i STRIC:s fall till IBM med AIX som operativsystem. Basset systemet från fartygssystemen kunde utan större problem konverteras till att även passa för STRIC. Tekniklyftet till att använda kommersiella plattformar utnyttjades därefter även för fartygssystemen.

En mycket intressant produktportfölj med stor försäljningspotential hade uppnåtts, både på marin- och luftförvarssidan. Dessutom utbyttes och återanvändes såväl kunskap som produkter mellan marin- och luftförvarssidorna. Företaget hade blivit en teknikledande systemleverantör på världsmarknaden! Den stora utvecklingsinsats som möjliggjort detta medförde att företaget bytte ägare och namn under 1990-talet (från Bofors Electronics till Nobeltech till CelsiusTech) innan det slutligen inkorporerades i Saabkoncernen den 1 juli år 2000.

Parallellt med framstegen på programvarusidan pågick (och pågår) en motsvarande fascinerande utveckling på maskinvarusidan, v.v. se kapitel 10. Utöver den makalösa ökningen av prestanda och sänkningen av priser på datorprodukter, medförde utvecklingen bl.a. dels att man-maskin-gränsytorna blev mycket bättre samtidigt som företaget köpte allt större delsystem på den kommersiella marknaden istället för att tillverka dem själv.

Stödsystem och system för teknisk databehandling framgår av kapitel 11.

En lista över använda förkortningar återfinns i kapitel 12.

Ekonomi-, marknads- och kompetensutvecklingsmässiga aspekter

Fartygssystem 2000 blev teknikledande på världsmarknaden i början av 1990-talet, men ekonomiskt blev det inte samma succé. I och med Sovjetunionens fall inträdde "den stora freden", varvid marknaden imploderade. En välsignelse för oss som världsmedborgare, men företagsekonomiskt negativt för dåvarande PEAB.

Marknadssidan på dåvarande PEAB var underdimensionerad jämfört med konkurrenterna. En betrodd och mångårig medarbetare (Bengt-Olof Ås) beskrev det som att PEAB:s spindelnät avseende exportsidan bara hade en tråd, men likafullt lyckades vi då och då. Strategin var nog dels att närma oss utvalda tekniskt och kravmässigt kompetenta slutkunder, dels att liera oss med ett fåtal inflytelserika varv. Teknik- och funktionalitetsförsprång var "salient feature", liksom nöjda referenskunder på hemmamarknaden.

PEAB:s framgång på fartygssystemsidan byggde på att kunna ligga på framkant med kundnyttan genom att förutse inte bara dagens behov utan även framtidens. Detta hade aldrig kunnat åstadkommas om inte företaget haft

ett antal kompetenta, krävande och framsynt förutseende kunder. Deras tekniska och funktionella krav var inte sällan besvärliga att uppnå och orsakade både ekonomiska överdrag och senarelagda tidplaner. Kunderna var pådrivande men visade samtidigt tålmod. Resultatet blev slutligen nöjda kunder och ett företag som kunde marknadsföra produkter på teknikens och kundnyttans framkant. Utan det nära samarbetet med dessa kvalificerade kunder, hade PEAB:s fartygssystem aldrig nått en sådan framgång. En kompetent men krävande, ibland lite besvärlig kund på hemmamarknaden, som framsynt både ser till dagens och morgondagens behov är en förutsättning för att bli tekniskt ledande på världsmarknaden! En sådan kund skapar en lyhördhet, "spänst" och nödvändig förändringsbarhet hos företaget.

En tydlig och för alla inblandade (såväl för kundrepresentanter som för medarbetare hos leverantören) stabil och ensad kravbild utgör en förutsättning för framgångsrika leveranser av komplex programvara i stora projekt. Självfallet måste ändringar i kravbilden godtas, men annat än i nödfall endast till någon efterföljande release. Tidigare fanns inte sällan en bristande mognad hos såväl kund- som leverantörssida att mindre ändringar i programvaran kunde införas lite hur som helst. Dessa ändringar ad hoc är dock ofta förödande genom att den ensade mål- och kravbilden för kommande release eroderar.

2. Analogmaskiner och övergången till de första datorbaserade systemen.

(Författare: Stellan Borg med stöd från Bertil Eklund och Hans Gustafson)

Analogmaskiner

Under 1960-talets senare hälft levererade Philips Teleindustrier (PTAB) Arte 719 Mareld för kustartilleriet och Arte 722 Rakel för fartygssystem. Bägge var radarbaserade eldledningssystem med i huvudsak analoga beräkningar, vilka utfördes via 400 Hz räknespänningar i resolvar. Med hjälp av dessa gjordes omräkningar emellan polära och cartesiska koordinatsystem. Dessutom fanns olika typer av vippor och register varigenom additioner, t.ex. uppdatering av position med avseende på hastighet, kunde genomföras.

Beräkningarna var förvånansvärt noggranna och bidrog till att Philips Teleindustri, PTAB, inte tog steget in i datoråldern förrän ganska

sent. En nyckelkompetens var färdigheten (konst snarare än hantverk!) att förstå styrka och svagheter med de analoga räknemaskinerna, samt att med hjälp av en hel uppsättning olika trimpotentiometrar åstadkomma en dittills inte uppnådd noggrannhet i beräkningarna.

Radar utgjorde huvudsensor för såväl Arte 719 (kombinerad spaning och följning Track While Scan) som Arte 722 (i både spaningsradar och siktesradar). Radarn baserades på PTABs patenterade hoppfrekvensmagnetron, vilken både gav väsentligt högre störhållfasthet såväl som ökad följnoggrannhet. Radarekot ligger mer stabilt och "vandrar" inte över målet som vid fastfrekvens.



Arte 719

Första projektet med datorbaserade system

PTABs första skarpa projektet med ett datorbaserat system utgjordes av en simulator till Arte 722, SELMA – Simulator ELdledning MA-rinen. SELMA levererades till Berga örlogsskolor sommaren 1973.

Något egentligt utvecklingsstöd fanns inte. Det var upp till var och en att skapa spårbarhet och kvalitet.

Beräkningarna och styrningen av systemet gjordes av en central minidator (General Automation GA SPC-16) som valdes för att den hade en modern struktur och var en av de snabbaste som fanns att tillgå.

Programutvecklingsmiljön var primitiv med hålremsläsare och stans. Som konsol och utskrift användes en Teletype, som var långsam (10 tecken per sekund) och bullrig. En radskrivare tillkom vilket underlättade arbetet avsevärt. Små rättningar gjordes ibland till och med handstans!

All programvara skrevs i assembler och testades ut med enkla hjälpmedel, huvudsakligen med användning av ett GA-program BUS, som kunde sätta s.k. "traps" som stannade programmet vid vissa ställen och villkor där undersökningar om bearbetningsresultat kunde kontrolleras. Oftast skrevs små specialprogram för att spara ("trace") delresultat.

Operativsystemet, kallat RTM, var egenutvecklat på PTAB med vissa idéer tagna från ett realtidsexekutiv (RTX) som GA hade gjort för processtyrning. Vi baserade programstrukturen på begreppet parallella samtidiga processer med prioritetsnivåer och interruptstyrning. Kommunikation mellan processerna gjordes med meddelanden och gemensamma dataareor. Tanken var att man skulle kunna utveckla funktionerna i totalsystemet som naturligt begripliga och i högsta möjliga omfattning oberoende delprogram för att undvika att göra ett stort monolitiskt stelt slutprogram.

Utvecklingen hade hela tiden målet effektivitet och snabbhet i sikte. Det gällde ju funktioner som skulle reagera i millisekundsområdet.

Styrningen av hårdvaran i anläggningen som radardisplay och operatörskonsoler gjordes med hjälp av I/O-instruktioner till ett antal specialutvecklade interfacekort.

Sammanfattningsvis var systemet tidigt och uppbyggt enligt principer som "låg på dåtidens framkant". Projektet blev vår inkörsport till

datorstyrda system och kom att bli vårt sätt att bygga upp system för hela den kommande 9LV200-generationen. Naturligtvis överfördes och implementerades lösningarna då på Philips P800-serie av datorer.

Inför övergången till datoriserade ledningssystem

I mitten av 1970-talet var synen på programvara en annan än idag. Hårdvaran i form av CPU och minne var både långsam och dyr. Programvaran, vars utveckling skulle visa sig vara kraftigt underskattad i flertalet projekt, sågs mer en service som på bästa sätt skulle se till att den dyra hårdvaran utnyttjades maximalt.

Vid tiden för Mk2-generationen var antalet medarbetare engagerade i programvaruutveckling färre än tio. Det var en liten grupp med god sammanhållning och lite av pionjäranda och med en väl fungerande intern informations-spridning. Gruppen brukade äta lunch gemensamt, varvid många nyttiga erfarenheter och kunskaper utbyttes över lunchbordet. Då gruppen så småningom växte kraftigt och översteg 12 personer räckte inte längre ett lunchbord, varvid det informella informationsutbytet ej längre blev lika omfattande.

Genom att programvaran i senare projekt svällde och därmed behovet av ytterligare medarbetare i projekten, ökades radikalt behoven av overhead i form av ökad projektstyrning inklusive interninformation samt kalkyl-, utvecklings-, verifierings- och dokumentationsstöd. Vi var i detta skede lyckligt omedvetna om vad som skulle komma att krävas i de stora projekten.

3. Första generationens datoriserade system. 9LV200 Mk2

(Författare: Hans Gustafson med stöd från Bertil Eklund och Stellan Borg)

9LV200 Mk2 var den första generationen datorstyrda marina elledningssystem som utvecklades av Philips Elektronikindustrier AB (PEAB). Den plattform som utvecklades, och som kom att kallas Mk2, användes för de svenska systemen Arte 724 (för ERSTA-batterierna) och Arte 726 (för patrullbåtar av

Huginklassen). Dessa system levererades under perioden 1976-1981. Samma plattform användes också för Arte 726B/C (minfartyget Carlskrona) och Arte 726D (jagaren Halland, senare flyttad till minfartyget Älvsborg). Även Arte 725 KALLE, kustartilleriets elledning för luftmål, baserades på samma plattform.



Arte 726 på patrullbåt

Det kan noteras att ett antal Arte 726-system senare, i samband med att patrullbåtarna utrangerades, kom att modifieras till Arte 726F, varpå de flyttades till Koster-klassen där de fortfarande (2015) är operativa.

Arte 726B/C på Carlskrona uppgraderades med interface för ett nytt ledningssystem inför Operation Atalanta i Adenviken 2011.

Mer än 35 år efter leverans används alltså fortfarande ett antal Mk2-system.

9LV200 Mk2 kom också att bli en stor exportframgång, både som fartygssystem och som KA-system, till flera kunder i såväl Europa som i Asien.

Systemarkitektur

Strukturen för Arte 724/726 var densamma som för deras respektive föregångare Arte 719/722, men nu var uppgiften att datorisera upplägget. Systemen byggdes upp centraliserat av två datorer - en huvuddator och en bildator.

Bild datorn (bildgeneratoren) ombesörjde symbolritning och den realtidskritiska triggningen av B-indikatorns startsväp (noggrannhet motsvarande mindre än halva tiden mellan två radarsväp).

Huvuddatorn hanterade alla övriga funktioner och beräkningar. För de fartygsbaserade sy-

stemmen var realtidsproblematiken omfattande. Ett fartyg skall ju även i svår sjöhävning med stor noggrannhet kunna följa ett snabbt luftmål med hjälp av fartygssiktet och samtidigt styra ut sin kanon mot detta.

Systemens realtidsklocka var 1024 Hz och systemet hade även en sekundär klocka på 256 Hz.

CPU-kapacitet och minne var två mycket begränsande faktorer och stor vikt fick läggas vid effektiv kodning och införande av hårdvarustöd för bl.a. trigonometriska begränsningar. T.ex. kunde ballistikberäkningen i Arte 726 inte genomföras som en ren beräkning av en ekvation utan man fick använda en modell med tabellslagning för att kunna genomföra beräkningarna tillräckligt snabbt. Pjäsvinklar matades ut på synchroformat (analogt) med 256 Hz (finvinklar) och 64 Hz (grovvinklar). Timingen styrdes då av den sekundära realtidsklockan. Pjäsvinklarna var hela tiden kompenserade för målposition, målkurs, målfart, vinddata, parallax, rull, rullhastighet, stamp, stamphastighet, egen kurs och egen kurshastighet.

Radarmålföljningen hanterades i huvuddatorn (målfiltrering m.m.) med hjälp av data från videokorrelatorn i bilddatorn.

Eftersom ERSTA-batterierna var fasta behövde Arte 724 inte, till skillnad från sina fartygsbaserade system, kompensera för egna rörelser i form av rull och stamp mm. Däremot måste långa skjuttider och skjutavstånd, liksom stora ingående parallaxer kunna hanteras. Eldledningsdata från Arte 724 styrde pjäserna, vilka kunde vara placerade på stort avstånd från huvuddatorn (normalt på andra öar), varför det fanns behov av separata särskilda pjäsdatorer. För att inte spridning i sida och längd ska blandas ihop visar Arte 724 nedslagsobservationer både i observations- och skjutriktning (sett från eldledningen respektive

batteripunkten). Dessutom visas nedslagsposition både relativt målföljarens framräknade målposition i nedslagsögonblicket respektive relativt den position + hastighetsvektor * skjuttid som riktades mot i avfyringsögonblicket.

Arte 724 var förberedd för att kunna lämna måldata till andra batterier, liksom för att kunna ta emot måldata från andra eldledningssystem i området.

Datorer och datorutveckling

Processorerna som valdes för 9LV 200 Mk2 var P857M och P852M inom familjen Philips P800. Benämningen på de här processorerna vid den tiden var minidatorer. Dessa processorer var utvecklade av Philips-CTI i Fontenay Aux Roses i utkanten av Paris. Korten hade TTL-kretsar och var anpassade för att sitta i 19"-rackar (och fick därför monteras i datorskåpens bakplan). Det var byteadresserade, 16 bitars CPU:er, med 14 programmerbara register och med kapacitet för upp till 32 kiloords minne på separata minneskort. En extra minneskontrollenhet gjorde det möjligt att genom s.k. bankswitching växla mellan minneskort och då utnyttja upp till 72 kiloords minne. Stöd för flyttalsberäkning saknades.

Kommunikationen mellan CPU och periferenheter hanterades av en I/O-bus.

P800-familjen var, trots att den var okända för de flesta, en av de vanligaste CPU-familjerna i Sverige runt 1980. Den satt i Postens och nästan samtliga bankers bankterminalsystem. PC:n var då ännu inte uppfunnen, det fanns få datorer i samhället och definitivt inga i hemmen. Men eftersom denna CPU-familj fanns på i stort sett samtliga bank- och postkontor så blev den per automatik "mycket vanlig".

För bl.a. pjäsdatorerna i ERSTA-forten, användes en mindre processor, PACE (Processing And Control Element) från National Semicon-

ductor, som släpptes ut 1974. Detta var den första 16 bitars mikrodatoren på marknaden.

Programvaran lagrades på en bandkasset av typ DC300 (magnetband som var 300 fot långa) och lästes in vid systemstart. Med åren visade det sig att just systemladdningen var en akilleshäla. Banden blev med tiden oläsliga eller så gick kassetbandspelarens känsliga delar sönder och kunde till slut inte repareras.

Särskilt för Arte724 var detta kritiskt. Utbildningen på dessa system skedde i huvudsak på skolanläggningar. Systemen, utplacerade i ERSTA-batterier, användes sällan och kanske bara vart fjärde år vid krigsförbandsövningar. Att efter fyra år försöka starta ett sådant system var svårt, bl.a. för att banden kunde ha klibbat ihop. Man införde omständiga procedurer att med jämna mellanrum besöka batterierna bara för att starta systemen så att bandkassetterna fick "gymnastiseras", men detta hjälpte bara delvis.

Lösningen blev att FMV, för Arte 724 och Arte 726, beställde att bandspelare och minneskort skulle ersättas med ett nytt minneskort som laddades från PROM. Systemstarten blev därmed säker och snabbare. Genom att de nya minnekapslarna var både mindre och snabbare kunde systemen dessutom hantera mer minne och fick mer kapacitet.

Operativsystem

Grunden för operativsystemet för de här systemen var en vidareutveckling vid PTAB av de idéer som togs fram i SELMA (se kapitel 2). Det kallades RTM (Real Time Monitor) och implementerades för P800-datorerna och den nya generationen hårdvara, bl.a. realtidsklockan. Fortfarande var snabbhet och effektivitet det viktigaste för att kunna uppfylla beräkningskraven i den krävande realtidsmiljön.

Programarkitekturen byggde på parallella samtidiga processer som arbetade på olika prioritetnivåer och kunde avbryta varandra

efter prioritet och behov. Kravet på effektivitet ledde till att mycket arbete gjordes för att hålla nere programstorleken. Då talade vi om kilobytes (kB). Man kan jämföra med dagens läge där ingen behöver backa för att en liten hjälpfunktion blir flera megabytes (MB) stor.

All systemprogramvara i en enskild dator länkades till slut ihop till en laddmodul.

Diagnostikprogramvara

För test av hårdvara utvecklades speciella testprogram, s.k. funktionskontrollprogram (FK). Då test skulle utföras laddades avsett testprogram in i systemet och testen utfördes i ett antal steg. Fel indikerades med felkoder som sedan beskrevs i underhållsdokumentationen tillsammans med vilka åtgärder som kunde vidtas.

Testprogrammen utvecklades i huvudsak för att kunna testa kretskorten efter tillverkningen. Samma program, men med ett subset av teststeg, blev sedan ett FK-program i de levererade systemen.

Programutvecklingshjälpmedel

Utvecklingsmiljön var initialt enkel och baserad på arbetsstationer med en Standard P857M CPU med diskar av typ X1210 och senare X1215. Hjälpmidlen var texteditor, assemblerare och länkare.

En användare i taget kunde använda en sådan arbetsstation. Om flera personer behövde använda denna fick de stå i kö och göra uppdateringar en åt gången.

Utvecklingsmiljön blev snabbt en trång sektor. Flera produkter utvecklades parallellt och upp till ca 25 medarbetare behövde samsas om två arbetsstationer. Personalen tvingades omfördela sina arbetstider samtidigt som ett strikt bokningssystem behövde tillämpas. I takt med att verksamheten växte så blev ju detta i praktiken en ohållbar situation.

Då VAX11/780 från Digital Equipment introducerades 1980, inköptes därför en sådan för att flytta hela utvecklingsmiljön från P857M till VAX. Korsverktyg utvecklades för assemblerare och länkare och nya verktyg tillkom såsom simulatorer för debuggning av kod utan att använda målmaskiner.

Det visade sig snart att en VAX11 inte räckte för det ständigt ökande antalet utvecklare och snabbt införskaffades ytterligare ett system.

Som debugger på målmaskin användes BUS, ett stand-aloneprogram, "placerat mellan" CPU HW och applikationsprogrammet. Användarinterfacet var med dagens mått att anse som mycket primitivt. Ett fåtal kommandon fanns, systemet kunde startas, stoppas och stegas, man kunde läsa och skriva i minnet och i registren och man kunde sätta "trappar" i

programmet så att systemet stoppade då datorn skulle exekvera "trappen". Inmatning av adresser och data skedde med "hex-kod", d.v.s. man läste och skrev i minne med hexadecimala tecken. Exempelvis skrevs/lästes talet 1 som /0001, talet 10 som /000A och talet 256 som /0100.

Programspråk

Programspråket i 9LV200 Mk2 var genomgående assembler.

Programspråket skulle vara snabbt för att, dels inte sänka den ganska långsamma CPU:n, dels också möjliggöra minimal minnesåtgång. Valet föll därför på assembler, där programmeraren kan tyda datorinstruktionerna direkt i huvudet från den hexadecimala koden. Något annat alternativt språk fanns egentligen inte vid tidpunkten för Mk2-generationen i mitten av 1970-talet.

4. Utvecklingen till 9LV200 Mk2.5

(Författare: Hans Gustafson)

9LV200 Mk2.5 var den andra generationen datorstyrda marina elledningssystem som utvecklades av Philips Elektronikindustrier. Den plattform som utvecklades för Mk2.5 användes för det svenska systemet Arte 726E (för kustkorvett av typ Stockholm). Samma plattform, något utvecklad, användes senare för det rörliga KA-systemet Arte 727 Karin. 9LV200 Mk2.5 kom också att exporteras till flera kunder i såväl Europa som i Asien.

Systemarkitektur

Systemen byggdes fortfarande upp av två datorenheter; en huvuddatorenhet med CPU tillsammans med ett antal slavprocessorer, samt en bildatorenhet med CPU och en separat extraktordator.

Slavprocessorerna kan sägas ha ingått i samma familj som P800. Programspråket var det samma och det var därför rätt enkelt att flytta över kod som tidigare exekverats i CPU:n.

Slavprocessorerna användes i huvuddatorn för att hantera utmatning av pjäsvinklar, fortfarande på synchroformat (analogt) med 256 Hz (finvinklar) och 64 Hz (grovvinklar) till pjäserna på samma sätt som i Mk2. En slavprocessor hanterade då en pjäs.

Datorer och datorutveckling

Processorn som valdes för 9LV 200 Mk2.5 var P857E inom familjen Philips P800. Efter några år hade P857M vidareutvecklats till P857E av Philips-CTI bl.a. med hjälp av utlånad personal från PEAB. Formatet var nu istället kretskort på Europakortformat. Processorerna var fortfarande byteadresserade, 16 bitars CPU:er, med 14 programmerbara register.

I/O-busen fanns kvar för kommunikation med äldre kretskort, men de centrala CPU-delarna hade nu istället en UPL-bus.

Processorn var något snabbare och hade stöd i mikroprogrammet för trigonometriska beräkningar. Minneshantering hanterades nu istället av en Memory Management Unit (MMU) som kunde hantera upp till 2048 kilobyter.

Processorn fanns också i en mer kraftfull variant under namnet CP2B som var lagd på flera kretskort. Det var denna som användes i Arte 727 Karin.

Programvaran lagrades även här på en bandkassett av typ DC300 (magnetband som var 300 fot långa) och lästes in vid systemstart. Även här visade det sig att just systemladdningen var en akilleshäla. Programvaran blev i de här systemen också större och man fick byta till längre band (450 fot). Samtidigt blev då förstås laddningstiden längre.

Lösningen blev även här nya minneskort som laddades från PROM. I och med att UPL-bus hade införts fick man utveckla ett särskilt kort för Mk2.5.

Operativsystem

Operativsystemet var fortfarande RTM, men anpassat till den nya processorn.

Diagnostikprogramvara

I princip samma typ av diagnostikprogramvara som i Mk2 men vidareutvecklat för en tydligare utpekning av felaktiga enheter.

Programutvecklingshjälpmedel

Då Mk2.5 utvecklades hade utvecklingsmiljön redan flyttats till VAX. I princip samma verktyg kunde användas, dock uppgraderade för den utvidgade funktionalitet som fanns i P857E.

Programspråk

Programspråket i 9LV200 Mk2.5 var i huvudsak assembler. I de ledningssystem som PEAB

utvecklat hade man introducerat högnivåspråket RTL/2 (idag i stort sett bortglömt) som var utvecklat av ICL i Storbritannien. Ett viktigt skäl till att just RTL/2 valdes var att det på marknaden fanns kompilatorer för just P800-familjen.

Det fanns motiv att införa högnivåspråk även i eldledningssystemen. Att programmera i assembler var tidskrävande och att hitta personal med assemblerkompetens var svårt och kostsamt. Beslutet blev att gradvis införa hög-

nivåspråk i Mk2.5. Man valde därför att börja med att införa RTL/2 i programvaran för funktionsövervakning. Skälet till att man valde just funktionskontrollprogramvaran var att man var mycket medveten om att systemen var realtidskritiska och att RTL/2, om det användes i känsliga delar, kunde påverka systemets realtidsegenskaper negativt. Eftersom funktionskontrollprogramvaran hade lägst prioritet och dessutom inte var realtidskritisk så var valet enkelt.

5. Stina: Philips blir ledningssystemleverantör

(Författare: Ulf Olsson)

I mitten av 70-talet insåg man inom svenska marinen vikten av att bättre kunna följa upp trafik och verksamhet i omgivande vatten och provverksamhet med viss teknisk utrustning startade vid en av örlogsbaserna. Detta ledde till att FMV efter proven gick ut med en anbudsinfordran på materiel för en sjöbevakningscentral till tre svenska leverantörer, SRA (som medverkat i provverksamheten), Datasaab och Philips. Det blev en hård kamp mellan dessa tre där Philips slutligen vann upphandlingen, främst genom att i slutskedet kraftigt sänka priset. Detta var en medveten handling för att få in en fot på ledningssystemområdet som dittills dominerats av Datasaab. Det ledde också till att Philips nu kunde skaffa erfarenhet inom detta område och senare fick leverera anläggningar till ytterligare fyra sjöbevakningscentraler. Philips namn på utrustningen var Stina och detta kom länge att bli synonymt med sjöbevakning. Stina byggdes i huvudsak med Mk2-teknik med civila inslag. Läs mer i FHT-skriften Sjöbevakningscentraler med Stina.

Nästa steg blev Stina Export (9CSI 500) som innebar en väsentlig vidareutveckling från Stina Sverige och från att göra ledningsdelar i fartygsburna elledningssystem till att göra ett komplett, nationstäckande marint ledningssystem. Därmed tog man klivet in i en värld som hade till stora delar andra krav och utmaningar, men det klivet var nödvändigt för att kunna bli en fullödig systemleverantör.

Arvet

Enligt legenden föddes 9CSI-familjen när en 9LV-kund råkade titta in i ett rum där den första Stina Sverige var uppställd. Eftersom den kunden precis hade råkat ut för ett misslyckande i ett ambitiöst projekt med en leverantör som inte höll måttet, så var man ute efter en ersättare. Stina såg attraktiv ut: syn-

tetisk måldatapresentation, arbetsplatser med PPI och textpresentation, civil hårdvara. När dammet – och upphandlingen – hade lagt sig så blev beställningens omfattning mycket mer långtgående, omfattande en ledningscentral (Stina-C), sektorcentraler (Stina-S), regionala centraler (nedskalade Stina –S), avancerade datalänkterminaler som gjorde att även andra enheter kunde förses med ledningsinformation, fartygsburna motsvarigheter till Stina-S och uppgradering av existerande PEAB-fartygssystem. Alla enheter utom Stina-C var fullt mobila. Dessutom tillkom uppgradering av existerande kustradar med autonom målupptäckt och målföljning; förutom sensorinformation från fartygsburna system utgjorde radarn det huvudsakliga sensorsystemet.

Arkitekturen

I grunden var Stina Export byggt på Mk2.5-teknik, med samma byggsätt i hårdvara och mjukvara. I många avseenden togs en hel del steg först i exportversionen av Stina, t.ex. de grundläggande mjukvarukomponenterna. Bland annat var Stina Export först med att lämna RTM och övergå till operativsystemet TOSS, som utvecklats av systerdivisionen PEAB-T. Ett starkt skäl var att landkommunikation över HDLC nu blev ett huvudkrav, och eftersom TOSS var utvecklat för bankapplikationer så var detta en stark sida. Dessutom föreföll det som en klok idé att kunna dela utvecklingskostnader mellan divisionerna. Om det berodde på en verklig skillnad i kravbild (förvisso ganska sant) eller mest vanlig hederlig ingenjörsglädje så dröjde det inte mer än sex månader förrän D- och T-divisionerna hade divergerat så pass att DTOSS hade fötts. I och för sig var DTOSS nu i händerna på ett synnerligen kvalificerat expertteam, så till slut får man konstatera att det vi fick blev en stabil och kapabel systembas.

Människa-maskningränsytan tog ett antal kliv framåt, bl.a. infördes färg-PPI:er (fyrfärgs: grön, gul, orange, röd). Det gällde för utvecklarna att planera noga i vilken ordning sakerna skulle ritas: eftersom färgbyte skedde genom att ändra accelerationsspänningen i röret så fick det inte ske för ofta, annars kunde apparaten fatta eld... Dessutom hade varje arbetsplats tangentbord och textskärm, vilket ledde till att tabelldisplayer och tangentbordskommandon tog sin plats bredvid lägesbilder, rullbollar och funktionsknappar.

Ett formulärbaserat kommunikationssystem ingick, vilket gjorde att även manuella rapporter kunde inlemmas i övrig mer eller mindre automatisk måldatarapportering. Fartygsburna enheter försågs med en störtålig och robust egenutvecklad TDMA-länk.

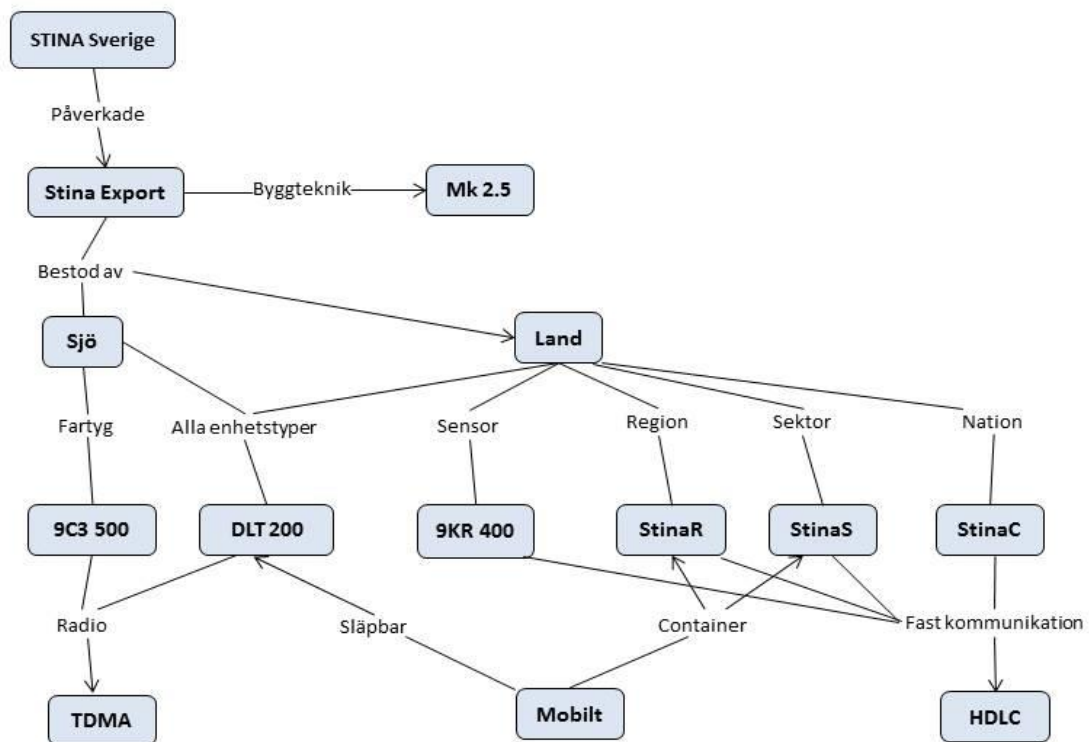
Måldatahanteringen vidareutvecklades från 9LV-systemen. Med tilltagande automatisering av sensorer blev det inte längre självklart att allt som rapporterades var av samma kvalitet: ett automatupptäckt spår kunde upphöra för att en stund senare dyka upp med ett nytt nummer från samma eller andra sensorer. Automatstöttad korrelering och utvärdering blev därför vitala komponenter, och en mycket god skola inför motsvarande funktioner i Mk3.

Erfarenheterna

Som noterats på annan plats i den här skriften så kom Stina i Mk2.5-utförande också till ma-

ringen, i form av provanläggningen MASIK på Berga (som i princip var en Stina S, anpassad för svenska förhållanden). De goda kontakterna med FMV vid utveckling och utprovning av 8000-systemet – vilket var MASIKs huvudsakliga datalänk – gav också praktisk erfarenhet av det systemets möjligheter och egenskaper. Likaledes provades ett antal tekniker ut i Stina Export runt systembyggsätt: principerna med asynkrona meddelandeköer som internkommunikation sattes där och genererade mycket information inför Mk3. Goda erfarenheter vidarebefordrades; andra designbeslut genererade åtminstone nyttiga läxor. Bland annat (se kommentarerna om TOSS och DTOSS ovan) så kunde vi tydligt konstatera att återanvändning och arkitektursammanhållning inte kommer av sig själv; det kräver en målinriktad organisation med tydlig styrning. Vi fick också våra första lärospån i storskalig användning av annat än assembler som programspråk. RTL/2 var enkelt och koncist och gjorde precis det vi begärde. Till exempel visade det sig raskt att Kalmanfiltret (hjärtat i målföljningen) blev av mycket hög kvalitet med fullt acceptabla prestanda.

Det är ganska uppenbart att utan Stina Export hade inte företaget haft den erfarenhet som behövdes för att utveckla produktportföljen till att omfatta även stridsledning och storskalig lägesbildshantering också till svenska marinen.



9CSI systemöversikt

6. Det stora steget: Fartygssystem 2000 (9LV Mk3). Arkitektur aspekter

(Författare: Ulf Olsson)

Fartygssystem 2000 och Bassystem 2000 växte fram ur många källor och många personers erfarenheter: både goda och mindre goda. Man kan notera att även den här gången bekräftades den gamla iakttagelsen att en ny arkitektur tenderar att fokusera på att lösa det som upplevts som problem i den föregående; det man ibland glömmer är att se till att de goda egenskaperna i tidigare generationer inte försvinner i hastigheten. I den här skriften beskrivs många aspekter av hur systemen i 2000-familjen realiserades; detta kapitel går mer in på principerna hur mjukvarusystemet byggdes.

Arvet

Företaget visste sedan Mk2-generationen mycket väl vilka prestandakrav ett fartygsburet eldledningssystem måste möta upp till. Däremot var det klart att i mitten på 80-talet ett distribuerat systembygge var av nöden, vilket ledde till en hel del nya angreppssätt i fråga om mjukvaruarkitektur och realiseringsmetoder. Tidiga steg hade tagits i och med ledningssystemen i Mk2.5-generationen (skrivna i huvudsak i RTL/2; ett exempel var MASIK-anläggningen på Berga) där den grundläggande mekanismen för att låta mjukvarukomponenter utbyta information var asynkrona köer ("mailbox"); komponenterna kunde kommunicera med varandra oberoende av om de låg i samma dator eller ej. Denna egenskap baserade sig på erfarenheter på ett tidigare system (icke-PEAB), där en sen omkonstruktion från fyra till fem datorer på grund av prestandaproblem ledde till omfattande omkonstruktion av programvaran. Alltså hade vi redan från början en grundläggande uppfattning om arkitekturen som baserade sig på många, oberoende program. Men mer om vart detta ledde oss senare.

Utmaningen

Vi fick i princip samtidigt beställningarna för Standardflex 300 (SF300) och Kustkorvett typ Göteborg (se t.ex. kap 1); försäljningschefens dörr pryddes därför några dagar av en skylt med texten "Sold out, gone fishin' ". Systemens grunddesign var tekniskt lika – distribuerade system baserade på dubblerat Ethernet – men hade ett antal intressanta bivillkor som vi nu behövde ta hänsyn till i en gemensam arkitektur. Kustkorvetten var en synnerligen kapabel sensor- och vapenplattform, med extremt avancerade krav på intelligenta insatsoptimeringshjälpmedel, hög systemnoggrannhet och snabb responstid. SF300 å sin sida baserade sig på ett containerkoncept där fartyget bokstavligen kunde gå i hamn en kväll som en patrullbåt beväpnad endast med en 76 mm kanon på fördäck, och komma ut nästa morgon med torpeder, sjömålsrobot och tre ytterligare operatörsplatser. Eller varför inte som ett minjaktsfartyg, komplett med släp-hydrofon? Speciellt den senare rollen ledde till krav på ett byggsätt i glasfiber (där förserieskrovet byggdes av Karlskronavarvet) för att minimera magnetsignaturen. Det hela ledde till intressanta utmaningar för ett eldledningssystem med sensorerna på ett ställe, kanonen på ett annat och skrovet gjort i ett flexande material, speciellt som att designen med standardiserade vapencontainers föreskrev att allt som kunde gå igenom containerväggen var systembussen och kraftförsörjningen.

Samtidigt var det inbyggt i affärsstrategin att systemet skulle byggas för att kunna anpassas till många tillämpningar och många fartygsstorlekar. Eftersom att kustkorvetten och Standardflex redan från början erbjöd en ganska rejäl spridning i kravbild så får man väl i efterhand konstatera att det lade grunden

för möjligheterna till anpassning och vidareutveckling som kunde utnyttjas i senare projekt.

Fundamentala egenskaper

Distribuerbarhet

Initialt var ansatsen att använda ett realtids-operativsystem vid namn DRM som tagits fram i ett samarbete mellan Philips i Holland och ett universitet. Även om projektförse-ningar ledde till att vi så småningom bytte till OS-9 så inspirerades vi till stor del av DRMs koncept med SOMAs (Software Machine) som fritt kunde distribueras över systemets uppsättning HAMAs (Hardware Machine). I någon mening kan man jämföra de idéerna från 80-talet med dagens megatrender runt virtualisering, där mjukvaran också förleds att tro att den är i full kontroll över sin värld. Realiseringen i SS2000 blev dock i termer av många fristående Ada-program (i en stor konfiguration kunde man hitta bortåt 800 program utspridda över 25 noder med flera processorer per nod).

Vår tillämpning av Ada som språk var ett mycket intressant exempel på hur långt man kan nå när man bygger ett tätt samarbete mellan hårdvara, basprogramvara, utvecklingshjälpmedel och applikationsstruktur. Speciellt mycket nytta hade vi av de täta kontakterna med Rational, som vid den tiden var helt fokuserade på Ada-utvecklingssystem, och som därför var bland världens absolut främsta Ada-expert. Ett antal sessioner som pendlade mellan högnivåarkitektur och handfast kodknackande ledde till att vi kunde ta vårt grundkoncept med många individuella Ada-program med kommunikation via asynkrona köer och ge det en synnerligen effektiv implementation. Det som såg ut som ett rendezvous (Adas något ovanliga mekanism för kommunikation mellan processer) för en programmerare var i själva verket snitt mot sänd- och mottagningsidan av köimplementationen i OS-9 som in sin tur – via ett egendesignat

chipset – kunde skriva och läsa direkt i buffertminne på nätverkskortet. Det låter lite komplicerat, men resultatet blev en väldigt elegant lösning med absolut minimala fördröjningar i systemet.

En grundläggande princip (direkt kopplad till mekanismen beskriven ovan) var att varje program designades utifrån en enkel princip: vänta på nästa meddelande, behandla det, gör det som behöver göras (inklusive att skicka nya meddelanden) och vänta på nästa meddelande. Syftet var att undvika ett svåranalyserat och potentiellt förödande problem i distribuerade system: att en process hänger och väntar på svar från en annan, och alltså är förhindrad att ta hand om ny input. Förutom att det leder till att funktionalitet försvinner så är tillståndet svårdiagnostiserat, speciellt som det egentliga problemet kan ligga i en avlägsen del av systemet. Denna designprincip ledde till ett mycket drifttåligt system, och att konsekvenserna av ett fel i en del av systemet kunde begränsas.

Givet att programmen i systemet i huvudsak alltså kommunicerade via namnsatta köer så hade de egentligen inga beroenden till var de exekverade. Även detta upphöjdes till princip i så måtto att det faktiskt inte ens var möjligt för ett program att ta reda på om dess granne (mottagaren av nästa meddelande) fanns i samma maskin eller långt bort i andra änden av nätet. Nyttan med den principen är kanske inte självklar innan man betänker att detta var den fundamentala mekanismen som gjorde det möjligt att ganska fritt möblera om programvarukonfigurationen för olika tillämpningar. Samma princip gjorde det också möjligt att ta hand om nodbortfall: de förlorade tjänsterna kunde (inom ramen för eventuella hårdvarubegränsningar) startas om på andra noder och funktionen kunde återupprättas.

Flexibilitet

Förutom positionsoberoendet som beskrivits ovan så förbereddes programvarukomponenterna för anpassning till olika fartygskonfigurationer genom att dimensionerande konstanter (och andra konfigurationsdata) inte lades in i kod utan lästes in vid programstart i form av parameterfiler. Återigen gjorde samarbetet med Ada-experterna på Rational och deras intima kunskap om vad språket faktiskt tillät att vi kunde använda mekanismer som trots konfigurerbarheten var lika effektiva som om vi hade använt statiska datastrukturer.

En annan metod att höja återanvändbarheten hos programvaran var att allt MMI var utbrutet till särskilda komponenter som lokaliserades till operatörsplatsen. Kommunikationen mellan den aktiva programvaran och presentationsdelarna var via så kallade MMI-objekt: de var en abstrakt representation av vad en funktion gjorde tillgängligt för presentation och hur en operatör kunde påverka funktionen. Detta gjorde att MMI-layout, språkfrågor m.m. lokaliserades helt och hållet till lokal konfiguration i operatörsplatsen. Vi behövde alltså inte riskera att verifierad, stabil programvara uppdaterades sönder som en sidoeffekt av en liten förändring av operatörsgränssytan.

Prestanda

Systemprestanda i eldledningen baserade sig naturligtvis i hög grad på beräkningskapacitet och dito noggrannhet i datorerna. Ännu mer väsentligt var dock att ha full kontroll på konsekvensen av en rörlig plattform under svåra omständigheter, med överföring av vitala data (rull/stamp/kurs, hastighet, vind m.m.) över ett nätverk som i princip inte har förutsägbara prestanda. Efter omfattande teoretiska och praktiska systemförsök visade det sig dock att om vi kunde hålla lastbudgeten på nätet under, säg 15 %, så gjorde statistiken att vi kunde innehålla alla krav på latens och noggrannhet. Ett nödvändigt trick för att hålla nere

meddelandefrekvensen (och därmed nätlasten) var att BS2000 innehöll sofistikerade metoder för att säkerställa ytterst noggrann tids-synk mellan alla noder på nätet. Därigenom kunde vi använda inte bara vinkelinformationen som sådan utan också prediktera till rätt tidpunkt för korrelering med annan sensorinformation baserat på vinkelhastighet och vinkelacceleration. Anekdotiskt hörde vi att vissa konkurrenter aldrig kom på den sortens lösningar, utan ansåg sig tvungna att införa allahanda standardvidriga tillägg och förändringar till Ethernet. Något vi alltså klarade oss undan.

Struktur

Arkitekturen består inte bara av de fundamentala mekanismerna, utan också av de komponenter som vi valde att identifiera. En viktig princip var att abstrahera funktionalitet så att anpassning till en viss radar eller en viss kanon lokaliserades till anpassningskomponenter, medan måldatadistribution gjordes på precis samma sätt för all sensordata och ballistikberäkningar var så gemensamma som möjligt. På det sättet kunde vi bygga upp en stor samling återanvändbara komponenter där anpassningarna till en given fartygsklass uppsättning av sensorer, vapen och kommunikationslänkar kunde lokaliserats till långt ut i systemranden. Totalt sett omfattade komponentuppbådet cirka 200 "systemfunktioner" (grupper av relaterade kodmoduler), i fyra huvudområden (C3, vapen, operatörsplats, "fundamentals").

I ett tidigt skede identifierades ett antal systembärande gränssnitt ("Critical Interfaces") som vart och ett var associerade med en systemfunktion, men som sattes under särskild konfigurationskontroll av den centrala arkitekturgruppen. Avsikten var att säkerställa att de gränssytor som skrevs av få men användes av många skulle skyddas mot förändringar som skulle resultera i omfattande merarbete i form av konsekvensändringar. Denna princip ledde naturligtvis till att det var vitalt att dessa gränssytor definierades på ett sätt som kunde

antas överleva under lång tid. Lösningen blev att inte automatiskt låta den funktionsansvarige/a designa sitt eget interface; risken då är att man mer eller mindre medvetet låter interna designval lysa igenom i interfacedesig-nen. Istället ska interfacet vara en så koncep-tuellt ren representation som möjligt av den funktion som erbjuds. Alltså lät vi de tilltänkta användarna (d.v.s. utvecklare av komponenter högre upp i beroendekedjan) provskri-va kod mot föreslagna gränssytor och bedöma om det var lätt att göra rätt och svårt att göra fel (en grundläggande egenskap hos Ada, i kontrast mot en hel del andra språk). Om de inte var nöjda så itererade vi processen till dess att vi kommit fram till något vi trodde hade god chans att överleva långsiktigt.

Som hjälp att definiera välformade moduler använde vi oss av objektorientering (OO), i den relativt råa form metoden då hade (före UML). Vår huvudsakliga guru i det avseendet var Grady Booch, som i egenskap av Rational-person och väl förtrogen med Ada 83's möjlig-heter och begränsningar såg till att vi använde grundtankarna i OO på ett sätt som ledde till god Ada-kod, ett synsätt som inte nödvändigt-vis var renlärt men som gav goda och prak-tiskt användbara resultat.

Konsistens och arkitekturell stabilitet

Ett återkommande tema är alltså att valet av Ada och den nära relationen till Rational ledde till ett antal goda egenskaper hos arkitektu-ren. Till exempel så hade Rational vid det laget infört begreppet subsystem, vilket tog Adas separation av specifikation och implementa-tion ett steg vidare. Det visade sig passa väl-digt väl för att hantera definitioner av våra systemfunktioner på ett effektivt sätt. Intres-sant nog var det dock en besvärande egenskap som verkligen såg till att vi höll ordning på våra stabila gränssytor. Rational R1000 var nämligen en mycket ovanlig maskin i det att den egentligen alltid var mitt inne i en kompi-lering: det som lagrades var den abstrakta

interna representationen, garnerad med det som behövdes för att generera det kodaren hade skrivit (inklusive kommentarer), liksom genererad objektкод. Fördelen var att det var fysiskt omöjligt att låta objektkodens divergera från källkoden (ändra en kodrad så försvinner den gamla objektoden!), nackdelen var att om man råkade ändra en kodrad i en specifikation som hela systemet var beroende av, så kunde det ta många timmar att kompilera om. Alltså var det vitalt att inskräpa att vissa saker ändrade man inte på utan att tänka igenom konsekvenserna flera gånger. Sedermera bytte Rational princip i och med övergången till Apex, och kompileringstiderna blev mer han-terbara. Men nyttan med långsiktigt stabila gränssytor har systemet mått bra av ändå.

Ada var dessutom byggt för att isolera pro-grammeraren från den underliggande hårdva-ran och operativsystemet så mycket som möj-ligt. Detta var grundstenen i det som gjorde det möjligt att med – trots allt – begränsad insats flytta över arkitekturen till en serie plattformar med radikalt olika teknikbas: från Motorola 68K/OS-9 via RS6000/AIX till In-tel/Windows NT och vidare framåt i tiden. De förändringar som senare gjorts har i hög grad varit inom MMI-området, där tekniken gjort framsteg som vi bara anade konturerna av i mitten på 80-talet.

Erfarenheter

Som sades i början av det här kapitlet så var de grundläggande arkitekturbesluten väl för-ankrade i erfarenheter från tidigare systemge-nerationer och en gedigen kunskap om till-lämpningsområdet. På motsvarande sätt gav arbetet med BS2000 och SS2000 alla oss som deltog insikter som förhoppningsvis bidragit till att forma utvecklingen av direkta och indi-rekta efterföljare. Till exempel så lärde vi oss efterhand att beslutet att tidigt bygga upp en stark integrationsorganisation var nyttigt för att poängtera vikten av integration och test, men samtidigt så lärde det oss den hårda vä-

gen att vattentäta skott mellan utvecklande och integrerande organisationer inte fungerar. Mogna komponenter kan – och ska – hanteras som ren integration, men nya komponenter ska åtföljas av sina konstruktörer in i integration, åtminstone initialt. Det ger inlärningseffekter i bägge riktningar: integratörerna behöver gissa mindre, och konstruktörerna får känna på hur deras produkter betar sig i verkligheten.

För systemarkitekterna blev det tillfälle att dra intressanta slutsatser när systemen började nå full skala i testanläggningarna. När vi väl började mäta på nätlaster (som nämnts ovan så var detta av vital betydelse för systemegenkaperna) så visade det sig att de lastbidrag som vi varit oroade för och därför ägnat stor uppmärksamhet i designfasen låg mycket nära beräknade nivåer. Totalnivåerna var dock flera gånger högre än väntat: det visade sig vara en effekt av att vi grovt underskattat lastbidragen från MMI-relaterad trafik. Tack vare den ändringsvänliga grundkonstruktionen var dock lösningen enkel: vissa MMI-flöden ändrades till att genereras lokalt i operatörsplatsen istället för över nätet. Nödvändig information fanns redan, och onödig replikering kunde undvikas. Läxan var att inte ta design för givet: även till synes okontroversiella arkitekturbeslut måste verifieras med mätningar. Och även arkitekter måste då och då ner i labbet...

En ytterligare läxa av mer filosofisk natur var hur vi så småningom förändrade hanteringen av systemfunktioner. Man kan frestas att tillämpa samma regler för alla komponenter med avseende på hur de konfigurationshanteras; det är det naturliga beteendet om man tillämpar metoder och regelverk okritiskt. Med tiden blev det dock tydligt att för en stor del av systemet så blev implementationsmönstren tämligen likartade: speciellt inom området Fundamentals fanns det väldigt få skäl att förändra konfigurationen mellan projekten. Alltså förändrades storleken på vad vi

hanterade som återanvändbara komponenter från att ha varit enstaka systemfunktioner till att bli en blandning av förintegrerade aggregat (med radikalt sänkta integrationskostnader per projekt som följd) och individuella komponenter (exempelvis sensoranpassningar). Det vill säga: vad som började som en arkitektur för återanvändning övergick med tiden till en arkitektur för systembygge med återanvändning. En naturlig utveckling, som dock sällan syns i metodhandböckerna.

Som avrundning kan man konstatera att optimisterna bland oss hoppades på att SS2000 skulle kunna vara en bas för aktiv utveckling i 15 år och kanske underhållas i 10 år till. Nu, trettio år senare, kan vi konstatera att det verkar ha lyckats. En del aspekter har förändrats över tiden, men ännu vet ingen hur gammalt ett Fartygssystem kan bli!

Addendum. Underlag och historik avseende beslutet om Ada

I detta addendum redovisar dåvarande VD Sven Bidö företagsledningens syn på hur beslutet om övergång till programspråket Ada fattades.

Hur fattades beslutet om övergång till Ada i fartygsprojektet

I samband med FHT insamling av materiel och data rörande utvecklingen av mjukvara på PEAB har flera ställt frågan hur det kom sig att PEAB var så tidiga att använda Ada och hur och av vem de avgörande besluten fattades. Strategiskt hamnade företaget i en svår situation när kraven på Ada började accentueras samtidigt med att utvecklingen av det största programvarusystemet i företagets historia skulle startas upp i ett föråldrat programspråk. Som VD var jag djupt involverad i beslutsprocessen och naturligtvis ytterst ansvarig för de beslut som fattades. Många var inblandade i beslutet men jag vill i synnerhet nämna Göran Granberg med övergripande tekniskt ansvar, Jaak Urmi ansvarig för utvecklingen av Bas-

system 2000 samt Roland Fors ansvarig för utvecklingen av programvara för fartygssystemen.

Bakgrund

I slutet av 1986 erhöll PEAB beställningar på en ny generation fartygssystem från såväl den svenska som den danska marinen. Fartygssystem tillhörde företagets kärnområden och beställningarna var avgörande för företagets framtid inom området. Båda typerna av system innebar på olika sätt stora krav på ökade prestanda och flexibilitet jämfört med tidigare. Det svenska fartyget skulle kunna uppträda i flera roller med utökat beslutsstöd och ledningsfunktioner. Det danska StandardFlex bestod av många enheter som skulle kunna ombestyras med olika funktionalitet efter behov. Kunden ställde explicita krav på ett distribuerat system med standardiserade operatörsplatser. Systemen upphandlades i konkurrens. Marknaden för denna generation av system uppskattades till ett 40-tal med ledning av tidigare erfarenheter. Initialkostnaderna fördelades i enlighet med denna nyckel och beställningarna innebar därmed en stor egensatsning av företaget för att säkra beställningarna.

Programvara

Systemen projekterades för att programmeras i RTL/2, ett högnivåspråk välkänt på företaget och använt i den tidigare generationen system. Man kände väl till både fördelar och nackdelar med språket men de nya systemen var väsentligt större och mera krävande och ansvariga planerade för rejäla satsningar på utbildning och utveckling av nya hjälpmedel för att bemästra situationen.

Samtidigt pågick inom företaget tidiga studier av STRILC 90. Inom ramen för ett exportprojekt hade PEAB byggt upp en kompetens inom ledningssystem som var i stort sett okänd av FMV. Många medarbetare hade också kunskaper om det STRIL-system som skulle ersät-

tas. Det stod från början klart att det skulle ställas krav på Ada som programspråk och det pågick tidiga undersökningar om konsekvenserna av en sådan utveckling. **I början av 1987 inträffade ett antal händelser som ställde valet av programspråk för fartygssystemen på sin spets:**

- FMV deklarerade att man skulle ställa krav på Ada för alla framtida system
- Specifikation och anbudsfrågan på AN-ZAC ship presenterades och tilldrog sig stort intresse från marknaden. Det var ett osedvanligt stort projekt och tolv fartyg skulle preliminärt byggas. Ett första steg var att välja fartygsdesign och välja det varv som skulle få bygga fartygen. Med tanke på Australiens nära kopplingar till USA var det troligen så att man skulle kunna komma att kräva Ada

Arbetet med utveckling av mjukvara för fartygssystemen var under uppstartning men bestod mest av grundläggande specifikationer av generell karaktär oberoende av programspråk. Vi beslöt att undersöka om det fortfarande var möjligt att använda Ada som programspråk istället för RTL/2 som riskerade att bli en återvändsgränd. Förutsättningen för ett skifte var att skaffa stöd och hjälp för en flygande start. USA var ledande och pådrivande inom området och ett naturligt första steg är att undersöka om där fanns stöd och hjälpsystem för Ada-utveckling tillgängligt på marknaden.

Magnavox som tillhörde Philips fick bli inkörsporten för en rundresa i USA. Det visade sig att Magnavox liksom flera andra företag inom området hade pilotprojekt med Ada för olika försvarsgrenar. Sekretessen medförde att det var omöjligt att få något grepp om några detaljer. Hjälpsystem och utvecklingsstöd av traditionell karaktär finns internt i företagen på diverse plattformar från stordator ned till PC. Magnavox använde en källare full med

DEC 20 som utvecklingsmiljö och för daglig kompilering för att testa och integrera ny kod. IBM har verktyg på stordator men har inga intressanta utvecklingsverktyg kommersiellt tillgängliga. En medarbetare på IBM tipsar om Rational dit vi redan är på väg.

Rational, ett nystartat företag i Silicon Valley, visar sig ha den kompetens och den utrustning vi behöver. De verkar ha löst en del av de problem med t.ex. kompilering som vi hört talas om under vår resa. Deras chefsideolog Grady Booch har skrivit en bok om Ada utveckling och Rational har utvecklat hjälpmedel för Ada enligt hans intentioner. Grady blev för övrigt sedermera – med Ivar Jacobson och James Rumbaugh – grundare av Unified Modeling Language (UML), vilket numera är grundvalen till det mesta inom systemmodellering. Rationals princip är att programvaruutvecklingen är hårt styrd av hjälpsystem i en specialbyggd dator vilket jag personligen tycker verkar bra, även om vi måste köpa en del hårdvara¹. De är dessutom intresserade av ett avancerat referensprojekt inom det militära området. Företaget ger ett gott intryck och är beredda att ge oss ett långtgående stöd under vår uppstartningsperiod. Vi bestämmer oss omgående för att gå vidare med dem för att definiera omfattning, kostnader och villkor för ett samarbete med målsättningen att få fram ett beslutsunderlag för övergång till Ada i fartygssystemen.

Beslut

Det fanns två tungt vägande skäl för PEAB att välja Ada för fartygssystemen. Vi riskerade att investera storleksordningen 200 000 timmar i mjukvara i system som blev svårsålda på grund av de kvalificerade kundernas krav på Ada. Vi måste med andra ord försvara vår

investering. Det andra tungt vägande skälet var att en tidig uppbyggnad av Ada-kompetens öppnade vägen för att på allvar konkurrera om STRILC 90. För kunderna var det naturligtvis intressant med en bättre strukturerad och återanvändbar mjukvara med hänsyn till livstidskostnaderna.

Ett samarbete med Rational enligt de offerter vi erhållit lovade att ge oss det lyft på metodsidan och den flygande start vi behövde. En omstart Ada innebar en del förlorad tid i projektet. Framför allt låg kustkorvetten på den kritiska linjen och vi befarade att förseningen kunde bli upp till ett år. Specifikationsarbetet tillsammans med kunden var också allvarligt försenat vilket inte förbättrade situationen. Skulle vi gå den här vägen krävdes ett snabbt och informellt beslut på högsta nivå. Under tecknad lade fram problematiken för chefen för huvudavdelningen för marinmateriel på FMV, Ola Backman och för den tekniske chefen på SMK Kaj Gregersen. **Efter interna konsultationer gav kunderna klartecken för förändringarna i projektet och FMV medgav också en senareläggning av kustkorvetten med 6 månader.**

¹ Man kan notera att utvecklingen senare löste det här problemet genom den explosiva utvecklingen av processorprestanda som gjorde att industrin – och Rational likaväl som Philips – kunde övergå till kommersiell hårdvara.

7. Fartygssystem byggda på BS2000/FS2000 (9LV Mk 3)

(Författare: Göran Pettersson)

Systemarkitektur

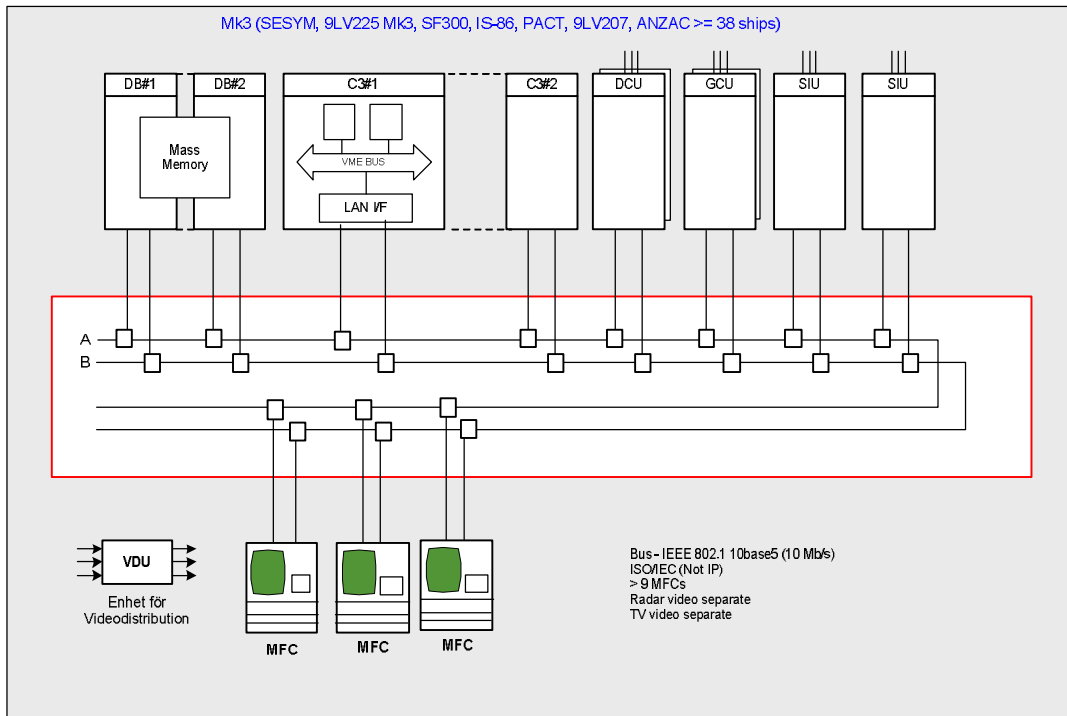
Systemen i generation Mk3 var i många avseenden vidareutvecklade relativt den föregående generationen (Mk2/Mk2.5). För de nya system som projekterades i mitten på 1980-talet krävdes andra lösningar. Kravbilden inkluderade aspekter som:

- Integration av fler sensorer och vapensystem
- Modulära nätverksbaserade system med utlokalisering av elektronik i containers
- Utökad redundans
- Multifunktionskonsoler med multipla skärmar
- Möjlighet till man-maskingränsytor för olika språk

Kravbilden på denna arkitektur med ett fartygs-LAN för ledningssystemet var starkt influerat av behoven för de modulbaserade fartygssystem som byggdes av europeiska varv (SF300 i Danmark och MEKO-konceptet från Blohm+Voss). Många delar av den nya generationen hade undersökts i studier, men när dåvarande Philips Elektronikindustrier inom ett fåtal månader fick tre projekt, varav åtminstone två kraftigt kravdrivande i ny teknik, ställdes allt på sin spets. För att överhuvudtaget kunna hantera tre (senare fyra) parallella nyutvecklingsprojekt ändrades processen för

framtagning radikalt från en i huvudsak projektorienterad utveckling till utveckling av en systemfamilj. För att säkerställa investeringarna i den nya generationen tillkom ytterligare förändringar jämfört med ursprungligt antagande. Till slut kan man säga att flertalet elektronikheter nyutvecklades med undantag av en del I/O kort för speciella tillämpningar. Arkitekturens grund utgjordes av ett nätverk med multi-funktionskonsoler (MFC) och ett antal redundanta noder för ledning och datalagring. Till detta anslöts noder för anpassning mot och styrning/övervakning av yttre enheter som navigeringsutrustning, sikten, robot, torpeder, pjäser, kommunikationsutrustning och så vidare. Yttre enheter anslöts med det som fanns för dessa, normalt antingen via resolver/synkron, serielinor eller i ett fåtal fall förbindelser av typ bussar eller nätverk, allt beroende på om ansluten materiel var av äldre eller nyare typ. En grundläggande egenskap hos den nya generationen var att den var mycket mer flexibel än den gamla och kunde användas både för uppgraderingar och för nykonstruktion. En skiss på hur ett ledningssystem kunde se ut ges nedan.

En ny höggradigt komponentbaserad programvaruarkitektur med generella transportmekanismer baserade på asynkron meddelandekommunikation med egenutvecklad "middleware" infördes.



Mk3 med anslutna enheter

Datorer och datorutveckling

En av de största skillnaderna mot system av generation 9LVMk2 var övergången till ett höggradigt distribuerat och nätverksbaserat system. I och med detta blev nätverket en fundamental komponent i systemet som på sikt kom att bli mer betydelsefull än enskilda datorer.

Nyheter var många:

- Nätverk av typ IEEE 802.3 10Base5 "Thicknet" med kabelredundans och tidssynkronisering infördes. Antalet nätverksnoder kunde röra sig om ett 20-tal och nätverksdiametern var 1 (dvs alla noder kan direkt nå alla andra noder utan att behöva gå omvägen via någon annan nod) vilket innebär korta fördröjningar i själva nätet. Kommunikationsprotokoll från ISO (Manufacturing Automation Protocol - MAP) användes till en början
- Alla centrala delar av systemet dubblerades för att få redundans

- Ny maskinvara infördes i form av skåp (CAB-85) bussystem (VME) och utbytesenheter (så kallade V-moduler)
- Processorarkitektur där Motorola MC68 ersatte föregående generation från Philips.
- De flesta gränssytor fick egna programvarumoduler
- Multifunktionskonsoler med scriptbaserad konfiguration och programmerbara knappar på s.k. Touch Input Displays infördes
- Gränssytan mot grafikenheter standardiserades till VDI

De flesta nya utbytesenheterna var kortenhet och inkluderade:

- Generella Datorkort
- Grafikprocessorer
- Grafikenheter för TV-skärmar
- Nätverkskort (Vad som i dagens terminologi skulle kallas NIC)
- Bussomvandlare
- PROM-Minneskort

- Kommunikationskort för seriekommunikation
- Omvandlare för olika analoga signaler

Många av kortenheterna baserades på två egenutvecklade LSI-kretsar. På grund av varie-

rante kundkrav togs ett flertal olika operatörsplatskonfigurationer fram. Ett typexempel på en tidig operatörsplats från ett av dessa system visas i figuren nedan.



Överdel till 2-skärms Multifunktionskonsol

Operativsystem

För den nya generationen system behövdes ett operativsystem till MC68-familjen. Initialt valdes DRM (Distributed Real-time Monitor) från Philipskoncernen. Detta OS var unikt i så mån att det var anpassat just för applikationer ("Software Machines") som exekverade i en distribuerad miljö. Tyvärr blev operativsystemet ständigt försenat och till slut valdes att byta till det mindre och enklare OS/9. Detta krävde dock en blyxtinsats med att ersätta funktionalitet i OS/9 som saknades jämfört med DRM och därmed hålla systembyggare och utvecklare skadeslösa. Därför utvecklades en rad tillägg till OS/9 som exempelvis Adaruntime support, semaforhantering, meddelandeköer och exception-hantering. Slutresultatet fick namnet "OS2000" och ingick som en av de viktigare bitarna i vad som kom att kallas BS2000 (Bassystem 2000).

Diagnostik

För all den nya maskinvaran behövdes ny diagnostik (BIT) av såväl "online-" som "offline-"typ. För att kunna köra "offline" användes en liten teknikerterminal som satt i eller kopplades in direkt på olika skåp.

Programutvecklingshjälpmedel

Miljön för programvaruutveckling genomgick flera faser. Initialt anskaffades en ny avancerad utvecklingsmiljö bestående av utvecklingsmaskiner (Rational R1000) specialgjorda för Ada-utveckling med inkrementell kompilering för Ada. Maskinerna hade intermediär Ada-kod (DIANA) som "maskinkod" och processorns mikrokod exekverade denna direkt. Programmering skedde från textterminaler som anslöts över växlar. Någon back-end för generering av maskinkod för målmaskinerna fanns inte initialt. Maskinerna medgav snabbt

skrivande av kod, men var tyvärr också särdeles långsamma för annan typ av databehandling inklusive kodgenerering för målmaskinerna. Att köra kod och emulera målsystemen i större skala var inte att tänka på. Ett stort antal maskiner anskaffades där olika team byggde programvarukomponenter för systemens olika delar. För att testa den kompilerade koden överförde utvecklare den till målmaskiner i separata testrackar. Efterhand som koden testats ut av utvecklingsteamet och frysts skeppades den över till projektunika maskiner och kompletterades med projektberoende data/parametrar varefter målsystemens maskinkod genererades och överfördes till riktiga referensanläggningar med rätt maskinvara. Detta kunde dock först ske när serie-riktig maskinvara fanns tillgänglig, åtminstone i förserieversion.

R1000-maskinerna var utvecklade av ett relativt litet företag som saknade marknad och ekonomiska resurser för att konkurrera avseende pris/prestanda med den generella datorutvecklingen.

Till senare system byttes därför utvecklingsmiljö avseende programvara från RATIONAL R1000 till APEX på IBM RS6000 som var en betydligt snabbare och mindre kostsam miljö per arbetsplats.

Programmeringsspråk

För applikationsprogrammering användes Ada 83 och därefter senare versioner. Basystemet

med lågnivåprogramvara för diagnostik, drivrutiner och olika tillägg använde framför allt programmeringsspråket C eller C++.

Stora delar av man-maskingränsytan var definierad i form av ett antal snarlika script-språk som interpreterades. Initialt var tanken att generera dessa script från databaser i ett verktyg. Verktöget blev dock aldrig särskilt bra och i stället skrev man-maskinutvecklare scripten direkt i de relativt enkla scriptspråken. Idén med scriptspråken var att dessa skulle definiera relationen mellan data generade/konsumerade av applikationer och vad som presenterades för och matas in av operatörer och bara sköta enklare logik och konverteringar/formatomvandlingar. Helst skulle scripten också kunna laddas och bytas dynamiskt under drift. Det visade sig också att utvecklingen av man-maskingränsytan blev väldigt effektiv genom det specialanpassade "språket". Man skall dock hålla i minnet att man-maskingränsytorna i systemen var ganska enkla för 30 år sedan. Det var mycket text och fasta symboler med ett och endast ett typsnitt och dynamiska ikoner av modernt snitt var inte aktuellt. Apple Macintosh fanns, men Windows var inte uppfunnet.

Utnyttjande

Det svenska system som först utnyttjade den nya generationen var SESYM (SESYM uttyds som Strids- och Eldledningssystem för Yttackfartyg i Marinen) som togs fram för de fyra korvetterna av Göteborgsklass (Kkv 90).



Kustkorvett typ Göteborg

Andra system var ledningssystemen till de danska Flyvefisken (SF300), Thetis (OPV), Nils

Juel (FFG) och ledningssystemet till den finländska Rauma-klassen.



Flyvefisken



Nils Juel



Rauma

Några år senare användes Mk3 till de tidiga releaserna av ledningssystemet till Australiens och Nya Zeelands fregatter av typ MEKO200. Även dessa plattformar var baserade på ett modulkoncept för underhåll och utbyte av sensorer och vapensystem. Detta gjorde att arkitekturen som tagits fram tidigare passade väl. Programmet omfattade totalt 10 fregatter och produktion pågick under många år. Ett av

programmets mer betydelsefulla delar var en omfattande tekniköverföring till Australien. Detta ledde till uppbyggnaden av ett helägt dotterbolag i Australien som i sin tur tog sig an många andra projekt och gjorde delar av ledningssystemen, framför allt sådana som inbegrep amerikansk eller Australien-utvecklad materiel som av sekretesskäl bara med svårighet kunde hanteras i Sverige.



ANZAC (MEKO200)

Det australiska systemet uppgraderades senare till Mk3E i samband med uppgraderingen av

luftförsvar med en Phased Array Radar och uppgradering av LV-robotsystemet till ESSM.

Spin-offs

Det stora tekniksprång som gjordes i samband med till Mk3 har spridit sig till andra domäner än till fartygen och vidareutvecklats/förädlats i egna spår. Exempel på andra sorters system som utgick från samma principiella koncept som Mk3 inkluderar:

- Det svenska luftstridsledningssystemet STRIC där tidiga versioner använde Mk3 som en tidig basplattform för applikationsutveckling innan man definierat sin egen IBM-baserade arkitektur
- Svenska ubåtsprojekt som SESUB 960A till Gotland-klassen

8. Vidareutveckling till 9LV Mk3E

(Författare: Göran Pettersson)

Mk3E utgör nästa steg i den tekniska utvecklingen efter Mk3. Under tidsperioden mellan Mk3 och Mk3E (ca 10 år) skedde en väldig utveckling i den civila domänen:

- Operativsystemet Windows kom fram och ersatte DOS
- Persondatorer och arbetsstationer med ökande prestanda kom fram i en strid ström och blev var mans verktyg både i hemmet och på kontoret
- Internet med IP-protokollen slog igenom på bred front och dominerade marknaden
- Maskinvaran blev allt mer högintegrerad och prisvärd

Systemarkitektur

Från arkitektursynpunkt hade systemen i generation Mk3E en snarlik uppbyggnad som Mk3 men med skillnaden att:

- Processorer byttes från Motorola till processorer av typ Intel®
- Nätverket som förband enheterna fått minst 10 gånger högre kapacitet genom att det uppgraderats till IP-nät. Fysiskt lager av typ 100 Mbit/s Fibre Distributed Data Interface (FDDI) användes till en början, men uppgraderas senare till kopplat LAN av typ IEEE 802.3 – antingen med kapaciteten 100 Mbit/s eller 1 Gbit/s
- Operatörsplatserna fick fönsterhantering
- HMI-verktyg byttes

Man kan säga att den stora nyheten med Mk3E utgjordes av en övergång från egenutvecklade lösningar till kommersiella lösningar inom ramen för ett övergripande koncept som i någon mån satts av Mk3. Det är också betydelsefullt att inse att Mk3E-generationen var den sista hårt sammanhållna egendefinierade arkitekturen. I och med övergången till högnivåspråk, nätverksorientering och lös koppling

till maskinvaran kan maskinvara och programvara till stor del leva separata liv. Från och med Mk3E kan man säga att efterföljande arkitekturer egentligen handlar mer om programvarans arkitektur och att en uppsjö av olika (men snarlika) maskinvarukonstruktioner använts.

Datorer och datorutveckling

Under perioden från det att utvecklingen av Mk3 startade har utvecklingen inom dator och grafikområdet accelererat. Dessutom har den tid som kommersiella datorer av en viss typ finns tillgängliga på marknaden blivit allt kortare. I och med att Mk3E bytte till Intel®-arkitektur på processorelementen har sedan dess många processorvarianter av denna typ använts, både rackmonterade och enkortdatorer för VME eller Compact PCI.

En speciell omständighet under denna period är att flera av den tidigare Mk3-generationens system var planerade att uppgraderas inom ramen för periodiska underhållsprogram med syftet att förbli moderna. Vid en begränsad modernisering måste hänsyn i regel tas till vad som fanns förut och i flera fall har man därför behövt använda speciella varianter av maskinvara.

Operativsystem

I och med att MC68 serien av datorer inte utvecklades i samma takt som alternativen Power och Intel blev det ca 1995 åter aktuellt att byta processorer och därmed blev även frågan om operativsystem aktuell. Företaget hade gjort en hel del arbete med att utvärdera ett byte av operativsystem till UNIX på samma sätt som gjorts inom andra domäner, bland annat porterat marinsystemens Ada-kod till UNIX och definierat en ny operatörsplats. Vid denna tid hade Microsoft tagit fram Windows NT som hade ett personellt och tekniskt arv

från DEC RSX-11 och därmed bedömdes vara betydligt mer anpassat för realtid än tidigare Microsoft-produkter.

För denna generation byttes Operativsystem från OS/9 till Microsoft® Windows® NT. Beslutet påverkades sannolikt av krav från försvarsmakten och bedömningar av marknadsvolymen. Exempelvis föreskrev vissa av försvarsmaktens direktiv specifika versioner av Windows®. Initialt användes Windows® NT 3, men redan före de första leveranserna byttes till Windows® NT 4 och därefter till "Windows® XP Embedded".

Diagnostik

Små principiella skillnader i förhållande till Mk3, men svårigheterna att göra egen online-diagnostik för de allt mer komplexa kommersiella processorkorten blev allt större.

Programutvecklingshjälpmedel

För Mk3E har utvecklingsmiljön flyttats närmare utvecklarna och lagts upp på datorer som i stort sett är ekvivalenta med sådana som används för normalt kontorsarbete. Adaversionerna har bytts från Ada 83 (Mk3) till Ada 95, Ada 05 och till slut Ada 12. Som tidig kompilator för Ada användes AIONIX object Ada och för C-koden MS Visual Studio. Senare byttes kompilator till GNAT/GCC.

Programvara

Med undantag för operatörsplatsen och delar av systemet som är beroende av operativsy-

stemet så har programvaruarkitekturen förändrats förhållandevis lite mellan Mk3 och Mk4. Ett tidigt skifte var dock att byta standard för grafiken från VDI som gällde för MK3 till OpenGL i Mk3E.

Man kan säga att implementationerna och verktygen för att generera programvaran har ändrats, men att övergripande arkitektur för applikationerna ändrats förhållandevis litet. När det gäller man-maskingränsytan har dock ändringarna varit många. Skärmkonfigurationer med flatbildsskärmar, stora knappdisplayer, fönsterhantering, grafik, 3D-presentation, värdskap för programvara från andra leverantörer, användning av webbläsare mm är saker som tillkommit i specifika projekt. Man kan säga att även för programvaran börjar skillnaderna mellan olika generationer bli otydlig och mer beroende på de specifika tilläggskraven i förhållande till basprodukten som krävs på grund av olika förutsättningar, regelverk i olika projekt.

Användning

Mk3E togs fram för korvetter av typ Visby, men fann också användning för ledningssystemen till andra marina svenska projekt som korvett typ Stockholm, Carlskrona/Trossö. Därtill kom systemgenerationen till användning i antal exportprojekt, framförallt för uppgraderingen av ANZAC- fregatterna i projekt ASMD och för användning i de australiska amfibieskeppen av Canberra-klass.



Korvett typ Visby



Korvett typ Stockholm



Canberra-class

Teknisk utveckling av fartygssystemfamiljen

Den tekniska utvecklingen från 9LV Mk2 till dagens fartygssystem kan sammanfattas som:

- Systemen har blivit större och större och fler och fler delsystem integreras, allt fler genom nätverksinkoppling
- Systemen byggs med en distribuerad arkitektur och kan i princip bli hur stora som helst
- Speciell maskinvara ersätts av programvara i mer och mer kraftfulla och krympande datorer från ett fåtal tillverkare av standarddatorer
- Programvaran har blivit alltmer frikopplad från maskinvaran genom lager som abstraherar bort detaljer i maskinvaran
- Nätverk har slagit ut nästan alla andra tekniker för att koppla samman system och delsystem.
- Militära organisationer har tappat sin roll som teknikledare och kommersiella lösningar gäller
- Det område där det skett mest förändring som syns för användaren är förmodligen man-maskingränsytan och troligen kommer denna trend att fortsätta allteftersom nya alltmer avancerade/alternativa metoder för samverkan mellan maskiner och människor forskas fram och slår igenom.
- Tack vare att processorernas kapacitet har ökat så kraftigt kan systemen hantera en helt annan mängd data och/eller betydligt mer komplicerade algoritmer

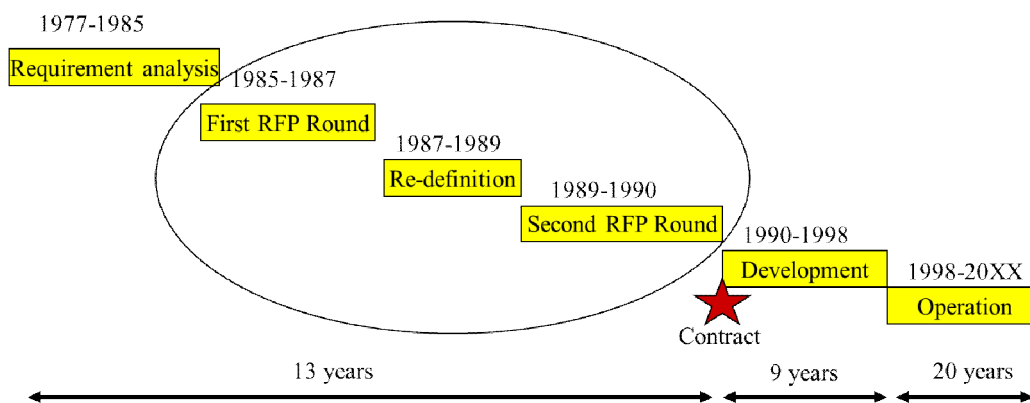
9. Ledningssystem för Flygvapnet, STRIC och Strics

(Författare: Per Anders Nilsson med kompletteringar från Thomas Jungfeldt)

Bakgrund

PEAB hade genom framgångarna med beställningarna av sjöbevakningssystemet Stina för svenska marinen och en stor exportbeställning på ett annat marint ledningssystem i början av 1980-talet (Stina Export) inlett en breddning av verksamheten från elledning till att även omfatta ledningssystem. Flygvapnet hade från slutet av 70-talet fram till 1985 arbetat med en upphandling av ett nytt ledningssystem för Flygvapnet. Orsaken till detta var att man hade stora behov att modifiera sina tidigare ledningssystem benämnt Stril-60 på grund av

planerat införande av främst JAS 39 Gripen och flygburen spaningsradar FSR890. Stril-60 bestod också av flera system från flera generationer – Lfc typ 1, Lfc typ 2 från Marconi, RRG/C/F och simulatorn TAST från Standard Radio/Stansaab samt RIR (RRGC/T) från ett samarbete mellan SRA och Marconi. Avsikten var att ersätta alla dessa system i en ny gemensam upphandling benämnd STRILC90. FMV var intresserad av konkurrens, och både Ericsson (som var dåvarande support för Stril-60) och PEAB (som saknade bakgrund inom ledningssystem för flyget) erbjöds delta i kravanalys och realiseringsstudier.



PEAB genomförde en första realiseringsstudie som redovisades 1985 vilken baserades på VAX minidatorer bestående av ett antal centrala datorer samt datorer i varje operatörsplats. Eftersom det totala systemet var stort krävdes sammanlagt hundratals VAX-datorer – efter den realiserbarhetsstudien blev PEAB mycket uppvaktade av Digital Equipment som producerade VAX-datorerna.

Upphandlingen av STRILC90 inleddes 1985 med en mycket omfattande anbudsspecifikation (mer än en hyllmeter) av ett mycket omfattande system som innehöll både fasta och transportabla system omfattande både "real-tids ledningssystem" med målföljning och stridsledning, "icke real-tids ledningssystem" som idag skulle kallas ett CCIS-system samt

simulatorfunktioner för operatörsutbildning. För att offerera detta system inleddes ett samarbete mellan PEAB och Ericsson (f.d. SRA/Stansaab/Datasaab). Parallellt med detta inleddes PEAB utvecklingen av BS2000 och D85 (Mk3 - ett egenutvecklat system baserat på Motorola 680X0 processor, operativsystem OS2000 baserat på OS9 och programvara skriven i Ada) för bland annat kustkorvett Göteborg. Ericsson höll samtidigt på med utveckling av HDA (ett egenutvecklat system för bland annat större exportprojekt baserat på Motorola 680X0 processor, eget EriOS operativsystem och programvara skriven i EriPascal). Offerten baserade sig på ett göra en gemensam utveckling mot de nya krav som ställdes av FMV på programspråket Ada. Som

operativsystem valdes realtids-UNIX. Offerten 1986 blev dock allt för dyr, och FMV drog tillbaka sin anbudsinfordran och samarbetet med Ericsson upphörde.

Därefter genomfördes en ny realiseringsstudie av PEAB som baserade sig på BS2000 och D85 (Mk3), samtidigt som FMV såg över kravbild och minskade omfattningen av systemet till att bara omfatta "realtidsdelen" av systemet. Parallellt med denna realiseringsstudie startades C3I-labb vid PEAB som främst gick ut på att se i vilken mån "icke realtidssystemet" kunde realiseras med COTS – SUN datorer, metodik med informationsmodellering och automatisk kodgenerering, den av Philips för Struktur 90-projektet utvecklade databasen DREAM-CS5 och SUN's fönstersystem NEWS. I samband med detta företogs också en del prov och försök för att utröna om det var realistiskt att använda COTS (HW & OS) i C2STRIC.

1989 kom en ny anbudsinfordran från FMV som nu benämndes STRIC, där PEAB:s offert baserades på BS2000 maskinvara, utvecklingsverktyg, programvaruarkitektur och en relativt stor del (50 %) återanvänd programvara från bassystem (80 %), operatörsplatsprogramvara (80 %) och viss återanvänd applikationsprogramvara från fartygssystemen (luftbevakning 40 %, kommunikation 20 % och ledningsfunktioner 10 %). Thomson-CSF valdes som underleverantör för multiradarmålföljning. Philips valdes som underleverantör för televäxeln som redan levererats till Försvarmakten (Televäxel 400) medan SATTECH valdes som leverantör av radiokommunikationsfunktionen. I konkurrens med PEAB lämnade Ericssons H-division in en egen offert. STRIC var ett mycket omfattande system som bedömdes kräva uppåt 250 personer. Som en följd av detta köpte PEAB (som nu blivit uppköpt av Nobel Industrier och nu hette BEAB) Ericssons H-division för att säkerställa tillgång till kompetens och resurser för utvecklingsprojektet.

BEAB stod nu med två giltiga offerter mot FMV, och BEAB:s egen offert valdes som lösning, dock ersattes Thomson-CSF målföljning av f.d. Ericssons MultiRadarMålföljning.

Kontraktet förhandlades varvid kravspecifikationen omarbetades och förtydligades för att bättre motsvara den offert som avgivits, och BEAB fick beställningen på STRIC 1990 och projektet påbörjades. Den beställning vi fick omfattade 5 fasta och 5 transportabla anläggningar, och vi föreslog att systemet skulle implementeras i 5 releaser, i stort sett en release per år:

- SR0 – Ett system baserat på BS2000 konfigurerat för STRIC avsett som integrationsplattform för att utvärdera modifieringsbehov och för prestandautvärdering
- SR1, SR2 – Ett system där komplett målföljningskedja och interface mot omgivande simulerade system skulle kunna genomföras, talkommunikationslösning och MMI-principer skulle kunna utvärderas
- SR3 – Ett system som initialt skulle kunna installeras och utprovas på site
- SR4 – Det kompletta systemet installerat på alla siter

Simuleringssystemet som ursprungligen ingick i STRIC90, benämnt Strics, kom sedan att upphandlas och offereras separat under 1992, vilket beskrivs i separat stycke nedan. "Icke realtidssystemet" kom aldrig att upphandlas i den form det ursprungligen var specificerat.

Initial utveckling av STRIC baserad på BS2000

Precis som i offerten började projektet med att bygga upp en testanläggning byggd på BS2000 parallellt med att systemarbetet påbörjades. Vi fick genom denna testanläggning erfarenhet av de större skillnaderna som fanns mellan STRIC och fartygssystem – systemet var minst en magnitud större med avseende på

antal operatörsplatser och servers, antal externa gränssytor där alla var "remote" med dåtidens teknik med modem och krypto, dessutom var fanns andra tillgänglighetskrav eftersom systemet aldrig fick gå ner helt – det krävdes redundans i hela systemet. Återanvändningen bedömdes dock fortfarande hålla baserat på de analyser som skett tidigare vilka skillnader som fanns och behövde modifieras och kompletteras. Marconi i England valdes som samarbetspartner för att portera Multi-RadarMålföljning till Ada, och senare användes även det hälftenägda dotterbolaget SES i Singapore för att utveckla vissa applikationer.

Redan när Bofors Electronics hade fått beställningen på STRIC, startades (1990) mer detaljerade undersökningar som syftade till att ta fram underlag för att bedöma rimligheten i att byta från D85/OS2000 som var offererat till COTS. Initialt kartlades problem relaterade till OS, interprogramkommunikation, operatörsplatsen (grafik och formulär), och utvecklingsmiljö (Ada kompilatorn). På hösten 1990 hade tillräckligt positiva resultat uppnåtts för att ett internt projekt (OLGA) skulle startas med målet att portera STRIC SRO till IBM RS6000/AIX. "OLGA" genomfördes 1990-1991, där SUN och IBM utvärderades som de två potentiella leverantörerna/samarbetsparterna med deras respektive systemfamilj. Baserat på resultaten i OLGA togs under 1991 ett beslut inom NobelTech Systems (som företaget hette vid detta tillfälle) att införa UNIX och kommersiell maskinvara i BS2000, och att STRIC skulle bli det system som införde detta.

I mitten av 1991 valdes IBM som leverantör, och en utvärdering (VOLGA) tillsammans med FMV genomfördes under hösten 1991 för att utreda vilka förändringar som behövdes i STRIC-kontraktet (t.ex. miljö, underhåll och uppgraderingar) samt informera FMV. Resultatet av Volga blev bl. a. att:

- Miljökraven mildrades genom att transportabla system togs bort
- Underhållsfilosofin öndrades med större LRU (Large replaceable units, utbytesenheter) och rullande utbyte av datorer mot likvärdiga och uppgraderingar av operativsystem. FMV satte som mål att maskinvaran skulle bytas ut ungefär var 7:e år

1992 fattade FMV beslut om att förändra kontraktet för att kunna använda sig av kommersiell teknik (COTS). Detta tillsammans med prognoser om leveransförsening utgjorde den s.k. rekonstruktionen av STRIC-kontraktet. Resultatet av denna blev framför allt att COTS från IBM skulle användas, att transportabla centraler utgick och ersattes av 14 fasta centraler samt att en releaseplan överenskoms mellan NobelTech och FMV. Dessutom togs krav på vissa äldre gränssytor bort från kraven på STRIC. Leveransen delades upp i två etapper (1994-1997 respektive 1997-2000), samt att NobelTech skulle leverera en första anläggning för utbildning och utvärdering till Uppsala 1996, före leverans av de första operativa systemen i etapp 1. Genom att de operativa anläggningarna också skulle uppdateras mellan etapp 1 och etapp 2 innebar det totalt 19 anläggningsleveranser.

Övergång till COTS

Eftersom STRIC var baserat på FS2000 (båt i berget) innebar det att stora mängder kod skulle återanvändas från 9LV Mk3. Då 9LV Mk3 var skrivet i Ada och mycket välstrukturerat, kunde porteringen till IBM RS6000/AIX ske genom att vissa delar av koden gjordes om, medan huvuddelen kunde återanvändas ograverad. De förändringar som gjordes var främst:

- Dator D85 byttes mot IBM RS6000
- OS2000 byttes mot AIX
- ISO-protokollet byttes mot UDP/IP och TCP/IP

- Grafik- och formulärpresentationen (GKS & VT100) byttes mot X-Windows och OpenGL
- Utvecklingsmiljön R1000 från Rational byttes mot Apex och OC Systems Ada-kompilator

Ändringarna betydde att cirka 10 % av koden från 9LV Mk3 fick göras om.

Vid övergången till COTS utnyttjades gemensamma utvecklingsresurser för Bassystem och även programvaran hölls samman. När grundutvecklingen var avslutad togs dock några få

personer in i projektet för vidmakthållande varvid BS2000 började divergera något eftersom två olika organisationer skötte det vidare underhållet och modifieringar.

De förändringar som gjorts i arkitekturen utnyttjades sedan av Fartygssystem i samband med övergången till Mk3E då man gick över till PC-teknik (Intel och Windows).

För operatörsplatsen utnyttjades den rullboll och TID (Touch Input Display) som ingick i BS2000 och så här såg operatörsplatsen ut.



STRIC

Systemarkitektur

I stort sett behöll vi BS2000 systemarkitektur vid övergången till COTS.

Vi byggde även fortsättningsvis på ett antal parvis redundanta servers – ett centralt serverpar som också hade det centrala disksystemet och några andra redundanta servers som innehöll applikationer, men som också medgav att extern kommunikation kunde göras redundant.

Vi dubblerade inte externa kommunikationsresurser, utan skapade bara så pass mycket redundans som krävdes – t.ex. byggde radaranslutningar på att en av sex servers kunde falla bort utan att kapaciteten påverkades.

Kommunikationsarkitekturen byggde på att en administratörsapplikation som svarade för resursallokeringen placerades i en redundant konfiguration i en nod som inte innehöll de externa kommunikationskanalerna med dess länkanvändningsprogramvara.

I de första releaserna användes en emulering av en telefon som gränssyta för att styra televäxeln. Dock blev inte prestanda så bra att talkommunikationen fick acceptabla prestanda. Vid en omkonstruktion användes istället ett nytt interface i televäxeln som hade utvecklats för call-centers och kallades CSTA. Med denna förändrade implementation uppnåddes tillräcklig prestanda.

Realtidskraven i STRIC var betydligt lägre, kanske en faktor 10, jämfört med ett fartygssystem, så vi hade inga stora problem med prestanda frånsatt en del skalningsproblem. Ofta fanns det mycket mer av allting (t.ex. antal operatörsplatser, antal sensorer) än i fartygssystemen, vilket krävde justering i implementationen av arkitekturen.

Däremot var grafikprestanda ett problem som vi inte helt kunde lösa förrän i sista releasen. COTS grafik var inte gjord för att hantera ett system som var händelsedrivet från applikationen och med mycket stora mängder mycket dynamiskt data. Grafikprestanda förbättrades för varje release, men behoven växte också för varje release. Vi tvingades använda en extern grafikaccelerator som var ett eget datorsystem innan vi till slut fick ett grafikkort som gav tillräcklig prestanda.

Ett annat problem vi hade som ledde till en del behov av ändrad implementation var att RS6000 byggde på en RISC-arkitektur med Cache-minnen – vissa prestandakritiska delar av programvaran var utvecklad för och utprovad i en CISC-arkitektur, och prestanda blev kanske 100 gånger sämre på grund av denna skillnad.

Datorer och datorutveckling

Valet på datorer föll på datorer ur IBM RS6000-familj. Livslängden på marknaden var ett år eller två, vilket ungefär motsvarade vad det tog att utveckla en release. Redan från början planerades därför projektet för att ersätta datorer i varje systemrelease – även i det omförhandlade kontraktet togs hänsyn till detta, och man planerade för en materielomsättning var sjätte år av de operativa anläggningarna som skulle leva i minst 20 år.

I den första releasen användes RISC-datormodellen 340 (Power2) som i den andra uppdaterades till 370 (Power2).

Inför den första leveransen till operativ anläggning genomförde IBM med Bull som underleverantör utvecklingen av en ny datormodell med SMP-teknik (Symmetrisk Multi Processor) och det blev denna datormodell som valdes. Vi genomförde även en konkurrensupphandling inför denna leverans, en upphandling som vanns av IBM på grund av att Bull inte kunde offerera en grafiklösning som fyllde kraven. IBM kunde dock till rimligt pris klara kraven med hjälp av en extern Grafikprocessor (en burk för varje operatörsplats som var lika stor som datorn). Den dator som valdes för Etapp 1 var G30 (PowerPC 601), och vi låg på framkant och hade tvåsiffrigt serienummer på de första datorerna till vår referensanläggning. Vi hade en del initiala problem med den nya datorn (vi var sannolikt det första företaget utanför IBM och Bull som använde datormodellen) som dock löstes genom att utvecklingspersonal från IBM kom till vår fabrik och personligen gjorde modifieringar i datorerna. Vi upptäckte även fel i deras nya grafikprocessor så att de fick ändra på en integrerad krets som tappade spänning på en del av chipet när vi gjorde vissa uppdateringar av grafiken.

För Etapp 2 upgraderades servers till G40 (PowerPC 604), en nyare modell av G30. Operatörsplatserna med dess grafiklösning byttes ut mot 43P-240 (PowerPC 604) som hade en tillräckligt bra grafiklösning så att vi kunde göra oss av med den externa grafikprocessorn.

Vi hade också en unik insyn i IBM produktplanering, vilket gjorde det möjligt att planera för vilka datorer vi skulle välja för en viss release, och veta när gamla modeller fasades ut – detta var en förutsättning för att använda COTS vid den tidpunkten, då varje leverantör hade sin unika maskinvara och sitt unika operativsystem.

Operativsystem

Genom övergången till COTS gick vi över från OS2000 med OS9 till att använda AIX från IBM som i princip var UNIX-baserat. Eftersom vi behövde ha kvar egenskaper från OS2000 gjordes en portering av OS2000 till den funktionalitet som fanns tillgänglig i AIX. Exempelvis kunde vi inte längre använda den ISO-baserade implementationen i IPC utan fick använda oss av TCP och UDP över IP.

Eftersom vi uppgraderade systemet med nya datorer för varje release var vi också tvungna att följa med i uppgradering av operativsystemet AIX. Vi hade ett synnerligen nära samarbete med IBM i Texas i USA och fick ha informationsutbyte med deras utvecklare – vi fick således en mycket djup insikt i egenskaper i operativsystemet, och fick i viss mån möjlighet att påverka deras implementation där våra krav väsentligt skilde sig mot vad andra kunder hade som krav.

Diagnostikprogramvara

I STRIC var utmaningen främst att systemet som helhet behövde vara redundant, och att man kunde ha systemet i kontinuerlig drift, vilket innebar att reparationer måste kunna genomföras utan att stänga av systemet som helhet.

Genom att vi använde COTS blev det även en annan nivå på underhållet – underhåll på anläggning blev främst att upptäcka och isolera felen till en nod och automatiskt ersätta denna med en redundant enhet. För offline-diagnostik användes den diagnostikprogramvara som tillhandahölls av IBM.

Däremot var tillgänglighetskraven sådana att även fel i omgivande system behövde upptäckas och redundansomkopplas – om exempelvis en nätväxel i telenätet gick sönder behövde detta upptäckas och redundansomkopplas, t.ex. genom att automatiskt koppla upp en ny modemförbindelse i telenätet.

Programutvecklingshjälpmedel

Även efter övergången till COTS utnyttjades Apex från Rational som front-end för utvecklingsmiljön, men en ny back-end för RS6000-datorerna köptes från IBM. Något år efter inköpet från IBM upphörde IBM med support för Ada, varvid utvecklingsmiljön supportades direkt av OC Systems, som tidigare var underleverantör till IBM. Ytterligare något år senare hade OC Systems utvecklingsmiljö förbättrats så mycket att vi kunde gå över till att använda OC Systems front-end och upphörde då att använda Apex. För MMI-utvecklingen utnyttjades ett eget MMI-verktyg som genererade de scripts som erfordrades för operatörsplatsprogramvaran.

Programspråk

Ada, vilket var ett kontraktsskrav, användes i STRIC även efter övergången till COTS.

Simulatorsystemet Strics

Då Strics-utvecklingen inleddes hade redan byte till COTS skett i STRIC-projektet. Strics-projektet baserades i princip på samma arkitektur som STRIC, men en ny arkitektur utvecklades för operatörsplatsprogramvaran där COTS-produkten InterMaphics valdes för grafik och UIMX valdes för knappar, menyer och formulär. Strics utnyttjade även databasen Sybase tillsammans med 4GL-verktyget Sapiens Ideo – alla scenarios/spel som skapades i Strics lagrades i denna databas. Den arkitektur som togs fram för operatörsplatsen i Strics utnyttjades senare i Safir-plattformen för bl.a. SjöC-projektet, men implementerades då på en Windowsplattform.

Syftet med Strics var dels att utnyttja systemet för operatörsutbildning, men också för att utgöra testomgivning för STRIC där scenarios och felsituationer som inte var möjliga att skapa i verkligheten kunde simuleras. Strics fick därför en identisk gränssyta med den gränssyta som STRIC mötte, och hela omvärlden simulerades.

Strics utvecklades i tre releaser – den första levererades 1996, den andra 1997 och den tredje 1999.

Vidmakthållande och Vidareutvecklingsprojektet VASS

Ett av problemen vi hade under utvecklingen av STRIC var att omvärlden till STRIC inte var stabil och att gränssytor ständigt ändrades i omvärlden. Detta var mycket svårt att kombinera med ett fastpriskontrakt, och redan inför leveransen av Etapp 1 behövde modifieringar göras för att det operativa systemet skulle fungera med sin omvärld.

Därför beställdes en "+-release" av STRIC Etapp 1 där nödvändiga ändringar kunde göras utanför grundkontraktet. Detta upprepades sedan för Etapp 2, och i samband med leverans av Etapp 2 omvandlades systemgarantin till ett vidmakthållandeprojekt där ändringar och tillägg fortlöpande kunde genomföras och levereras.

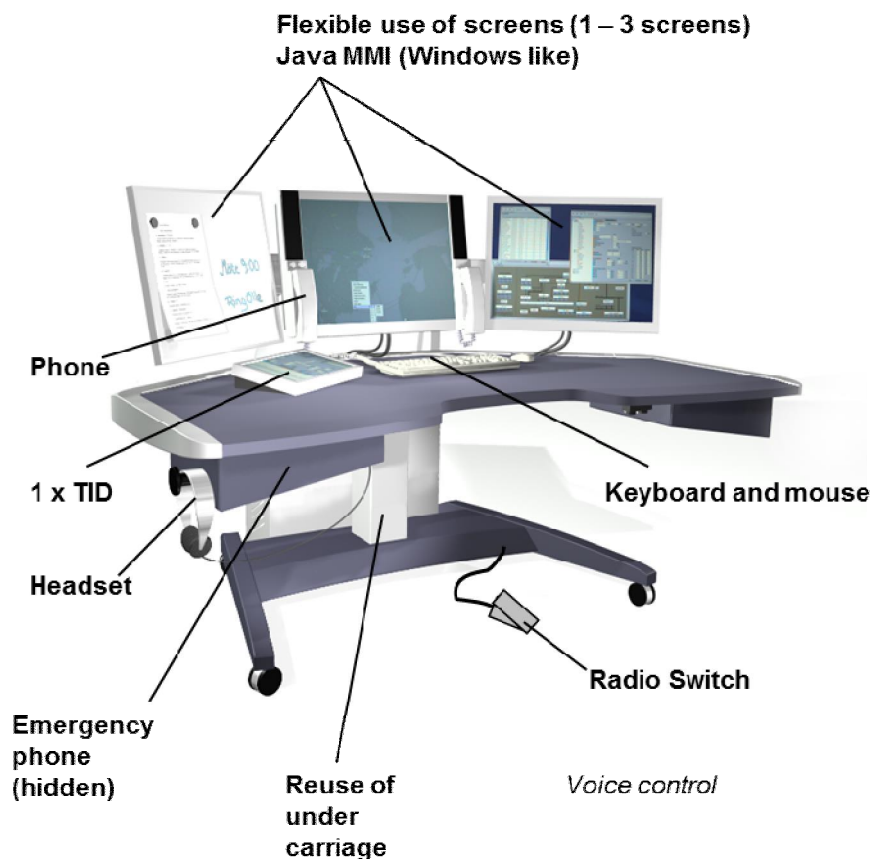
Även Strics påverkades av den instabila omgivningen. Därför behövde "+-releaser" genomföras också för Strics. Det fanns också ett behov av att samordna utvecklingen av Strics mot STRIC för att få till stånd utbildning av nya utgåvor av STRIC.

Så småningom skapades projekt VASS, Vidmakthållande och vidareutveckling av STRIC och Strics, som fick till uppgift att göra nödvändiga modifieringar och tillägg i systemet, samt att genomföra materielomsättningar ungefär vart sjunde år. Detta projekt levererade uppgraderingar av alla operativa STRIC två gånger per år – sommar och jul. Alla operativa system uppgraderades då samtidigt under 2-3 dagar. Detta projekt pågår fortfarande år 2015.

I STRIC-projektet hade en "vattenfallsmodell" tillämpats för utveckling, vilket ledde till mycket långa ledtider. För att möjliggöra halvårsvisa frisläppanden av nya releaser infördes i detta projekt en ny framgångsrik iterativ utvecklingsteknik med kontinuerlig integration med parallell verifiering, och verifiering kunde ske mot verklig omgivning från i fabrik. För integration mot system som inte ännu existerade gjordes integration mot objekt mot simulerade objekt i Strics.

Många stora förändringar gjordes i systemet eftersom hela flygvapnets infrastruktur förändrades. Exempelvis ändrades hela kommunikationsnätet så att idag all extern kommunikation sker över Forsvarsmaktens IP-nät inklusive talkommunikation. Bland de större förändringarna kan nämnas införande av ett sensordistributionsnät, införande av Länk 16, införande av ett nytt system för civil flygtrafikledning, införande av ett nytt IP-baserat radiosystem, införande av IP-telefoni.

En stor materielomsättning genomfördes 2005 – då byttes såväl all servermaskinvara som alla operatörsplatser. Den maskinvara som utnyttjades var fortfarande baserad på IBM RISC-datorer varvid p520 server (Power5+) användes och AIX5L infördes (L stod för Linux och innebar att såväl AIX som Linux kunde exekveras). En hel del basprogramvara skrevs då också om för att minska på mängden egenutvecklad kod, dra nytta av ny funktionalitet i operativsystemet samt för att förbättra underhållsbarheten av programvaran. I samband med denna materielomsättning byttes också den gamla kompilatorn ut mot GNAT/GCC, ett steg som redan tagits i fartygssystemen.



Operatörsplats

Den största förändringen i denna materielomsättning gällde dock operatörsplatsen, där all programvara byttes ut och maskinvaran utgjordes av vanliga Intel-baserade PC-datorer. Grafikprestandan var nu tillräcklig i vanliga kommersiella grafik kort. Initialt utnyttjades PC-datorn APro från IBM, men ersattes sedermera av HP Z10 då minnesbrist samt mer behov av grafikprestanda krävdes p.g.a. införandet av 30" bildskärmar. Programspråket Java mot grafikstandarden OpenGL användes för implementationen, varvid systemet blev ännu mer oberoende av maskinvara och operativsystem. Från början användes JBuilder för utveckling av Java-kod för operatörsplatsen, senare byttes denna utvecklingsmiljö till NetBeans och i senaste uppgradering byttes utvecklingsmiljön till Eclipse.

För fartygssystemen infördes samma operatörsplatsprogramvara senare i Mk4.

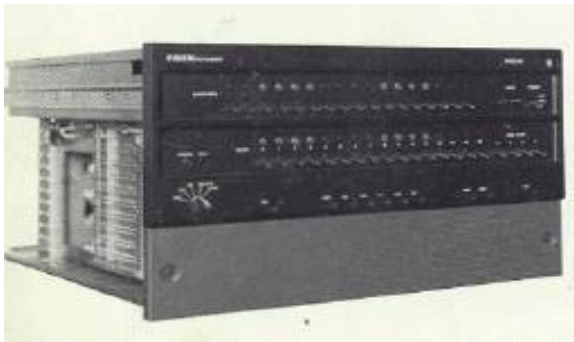
Ett annat område som krävt successiva förbättringar gäller IT-säkerhet, där hoten ökat med tiden och vi blivit mer och mer beroende av TCP/IP. I början var volymen av modem och krypton mer utrymmeskrävande än datorsystemet – idag kräver IT-säkerhetsskydd större volym än datorsystemet.

I alla våra exportsystem baserade på samma programvara som STRIC användes Linux och PC-datorer med Intel processorer redan i samband med införandet av AIX5L. I den andra materielomsättningen infördes samma teknik i STRIC och Strics, varvid IBM Flex System Enterprise med Intel-baserade bladserver utnyttjades.

10. Datorutveckling på maskinvarusidan

(Författare: Hans Ljungqvist)

Introduktion av digitalteknik inleddes på dåvarande PTAB under första halvan av 70-talet. Till de marina systemen införskaffades en minidator P852/P857 från Philipskoncernen. Datorn var en standard 32kwords 16 bitars



Philips P852/P857 minidator

För att kunna använda datorn behövdes en hel del anpassning. Således kom endast datorkärnan bestående av ett stort kretskort till användning i PTABs elledningssystem. Resterade delar såsom minnen och I/O enheter utvecklades av PTAB. Som massminne användes en bandspelare som införskaffades från USA.



Datorskåp med en Philips P857 CPU monterad baktill i skåpet

Som kuriosita kan nämnas att under de första åren så hade programvaruutvecklarna endast tillgång till ett par P800 datorer, varav den ena

minidator framtagen för civilt bruk. Exekveringstiden för en instruktion var från några mikrosekunder till ett antal tiotals mikrosekunder.



Philips P857 CPU kretskort

Till datorn utvecklades ett skåp där datorkärnan på grund av dess storlek monterades i datorskåpets bakstycke. I en utsvingbar rack återfanns minnes- och I/O-kort i dubbelt Europakortsformat med guldpläterade kortkontakter.



användes för textinmatning av assembler. Utifrån datorn fick man en hålremsa som lästes in på den andra datorn som genererade binär-

kod. Tillgång till den kompillerande datorn var starkt begränsad då endast en i taget kunde använda datorn. Som konsekvens infördes ett bokningssystem som tillät programmerarna att boka en kvart i taget!

Under slutet av 70-talet utvecklades en ny generation datorer med hjälp av personal från PTAB hos Philips i Paris.



Den nya datorn byggdes med hjälp av bit slice teknik (AMD 2900) i dubbelt Europakortsformat med standard Europa don vilket avsevärt underlättade integration i PTABs produkter. Datorkärnan bestod av två kortenheter. Memory Management Unit (MMU)- funktionalitet erhöles med ytterligare en kortenhet.



Philips P857 CPU i dubbelt Europakortsformat

Den nya generationen var robustare uppbyggd och mycket mindre och därmed lättare att integrera i PTABs system.

Genom att PTAB deltog i utvecklingen av datorn passade man på att komplettera instruktionssetet med trigonometriska instruktioner för att uppnå snabbare ballistikberäkningar. Dessutom fördubblades adresseringsförmågan till 64 kword, vilket är unikt för en 16-bitars dator.

I början av 80-talet kom nästa generationer av P800. Den första var en vidareutveckling av föregående generation kompletterad med cacheminne. Den andra var en dator baserad på en mikrodator med P800 instruktionsset. Datorerna kallades CP2B respektive P851.

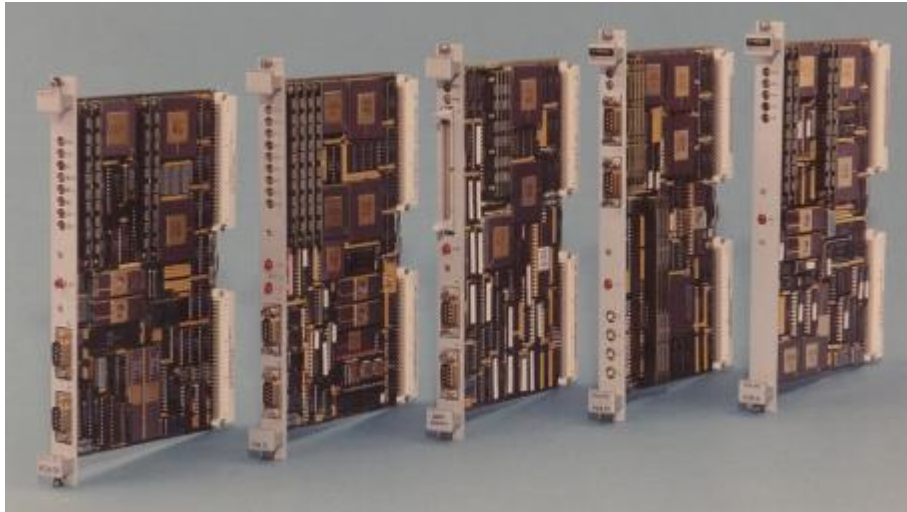
I början av 80-talet började mikrodatorernas prestanda närma sig minidatorerna. Således

gjorde några av oss från PTAB en resa runt USA 1982 för att utvärdera vad som fanns till förfogande. Besök gjordes hos Motorola som höll på att utveckla 68020, Intel som höll på att utveckla I386 och slutligen Digital Equipment som höll på att utveckla MicroVAX. Utöver val av nästa generations dator diskuterades buss och kortprodukter. Besöken var mycket intressanta. Vi fick träffa chefsingenjörerna/chefsarkitekterna/företagsledningarna från de olika företagen under ett par veckor och fick god insyn i deras produkter och planer.

I slutet av första halvan av 80-talet fattade PTAB beslutet att göra gemensam sak med övriga Philips och satsa på VME-bus och 68020. Som en följd av det beslutade PTAB att utveckla en serie av 68020 mikrodatorbaserade VME-kort baserade på en gemensam kär-

na. Det var med dåtidens teknik en stor utmaning. För att få plats på korten krävdes att ett par Gate Array kretsar utvecklades. De kom

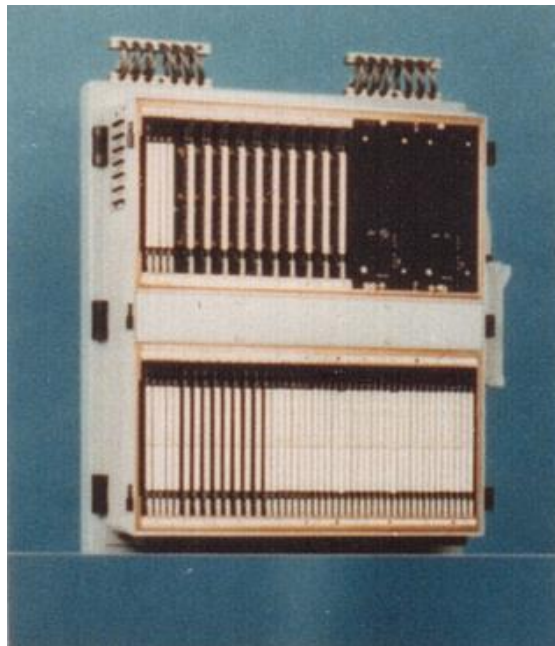
att kallas Dega och Cega där den senare visade sig vara värd sitt namn då ett antal iterationer krävdes för att få den felfri.



Mikrodatorbaserade VME-kort med Motorola 68020 och VME gate array interface

Till den nya generationen datorer utvecklades en ny generation skåp (CAB85) med en integrerad VME-bus rack. Till skillnad från föregående generationers datorer så kunde den nya

generationen dela bussystem och kommunicera via bakplanet, vilket öppnade upp många möjligheter.



Datorskåp

Som kuriosa kan nämnas att ett av datorkorten försågs med en "föregångare" till USB-bus som bl.a. användes för kommunicera med tangent- och displaymoduler.

Vid introduktionen av korten var deras funktionalitet och prestanda så pass bra (dubbla VAX prestanda) så både chiptillverkare och Philips Industridivision hörde av sig för att få

möjlighet att sälja dem på den kommersiella marknaden.

Detta visade sig bli den sista generation av datorkort som PEAB utvecklade. Efter några år

kompletterades utbudet av VME-datorerna med en 68040-baserad dator från Radstone.



VME datorkort med en Motorola 68040 och en 68020 processor

Under första halvan av 90-talet var det dags för nästa generation av datorsystem. Helt klart var att systemen skulle vara COTS baserade. Omfattande utvärderingar gjordes av både datorplattor och val av operativsystem.

För marint bruk valdes att basera datorerna på industriella PC-datorer med ATX moderkort för att kunna tillvarata teknikutvecklingen av

processorer och grafikkort på bästa sätt. Inget lämpligt datorchassi hittades på marknaden så ett eget 19"-liknande chassi togs fram. Konstruktionen kom att användas under närmare 10 år.

Med införande av PC teknik kunde vi som ett av de första företagen, kanske till och med det första, leverera system där både live TV och radar video distribuerades via systemens nätverk (LAN).



Ruggad PC med dubbla processorer på ATX moderkort

11. Teknisk databehandling/stödsystem

(Författare: Lars Fransson)

I detta kapitel ges endast en sammanfattning. Ett fylligt underlag med många bilagor återfinns dels i Försvarets Historiska Telesamlingar i Krigsarkivet (ref 2-10 och ref 12), dels i dokumentet *Försvarelektronik från svenska leverantörer*, FHT 2014-09-25 (ref 1) samt dels hos Patent & Registreringsverket (ref 11-priv.se/spd/8106091-5).

Inledning

Med Teknisk Databehandling avses här helt eller delvis automatiserade system eller verktyg som stöder företagets tekniska produktframtagning. Den viktigaste stödkomponenten är en god metodik. Vissa metoder lämpar sig för automatisering, varvid tid frigöres. Sparad tid kan helt eller delvis utnyttjas för idégranskning, urval och verifiering av teknikalternativ.

En särskild omständighet uppstår när produkterna har inbyggda datorer i liknande teknikkategori som stödsystemen/verktygen. Uppbyggnaden av stödsystem ger då en kompetenshöjning som kan utnyttjas i produktframtagningen. Detta var fallet inom PEAB efter 1975.

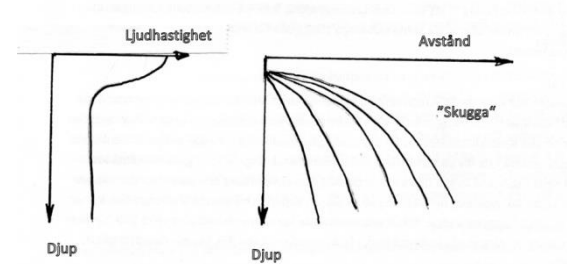
1950- och 1960-talet

Inom dåvarande Philips Teleindustri AB base-rades kundprodukternas beräkningsdelar på analog (400 Hz)teknik och precisionsmekanik. Datorstöd utnyttjades i form av både administrativ och teknisk databehandling. I detta dokument behandlas exempel på den senare.

Ljudbaneberäkning i vatten

Ljudhastigheten varierar med framför allt temperatur och salthalt och mäts enkelt med tillgänglig operativ apparatur. Särskilt i öppet hav varierar hastigheten huvudsakligen i djupled. Vid hastighetsvariationer kröks ljudbanorna och sonarmätningar behöver korrigeras.

För att välja beräkningsmetod studerades typiska djupprofiler, som approximerades med ett antal räta linjer. För ett djupskikt med linjär hastighetsvariation kan ljudbanan beräknas som ett cirkelsegment. Typiska ljudbanor beräknades med BESK. Resultatet blev att tre djupskikt ger tillräcklig noggrannhet.

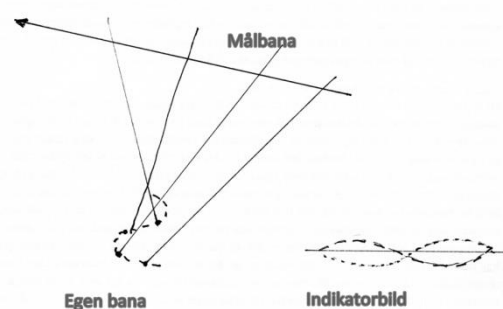


Tidiga produkter blev en elektromekanisk korrektör för AU-raketer och en (analog) ljudbaneindikator.

Målfaktorberäkning

Vid den s.k. AMI-metoden används passiva bäringsmätningar i kombination med kraftiga egenrörelser. Systemoperatören ställer in målfaktorerna för bästa anpassning, alltså "minsta" indikatorbild. Om målet håller konstant kurs och fart under en egenrörelseperiod återstår endast mätbrus i indikatorbilden.

Sedan BESK försetts med katodstråleindikator kunde en systemmodell programmeras och en serie realtidssimuleringar genomföras med en "riktig" kundoperatör.



En tidig produkt fanns i torpedledning TCI 105 för Sjöormen. För att lagra bäringsmätningar krävdes digitalblock. Detta var ett av de första

digitala "intrången" i en fortfarande analog beräkningskultur.

Erfarenheter av BESK

Normal programmeringsteknik var maskinnära men med lättförståelig symbolik. Fiktiva minnesadresser och flyttalsoperationer utgjorde basen för s.k. FA-Flinta-kod. Ett kärnminne på 1000 ord och 1000 operationer/sek kan tyckas löjligt med dagens ögon men gjorde på sin tid stor nytta (jämför med en mänsklig beräkningsförmåga).

Programvaruarkitekturen innehöll en "pseudoackumulator" där mellanresultat passerade och ett symboliskt programminne. Exekvering av ett FA-Flinta-program inleddes med att de symboliska minnesadresserna kompilerades till kärnminnet, varefter det lagrade programmet interpreterades instruktion efter instruktion.

De tidiga tillämpningarna av BESK gav väsentliga tillskott till marinens produktutveckling, samtidigt som PTABs kompetens inom datorområdet började byggas upp.

Långvågsnavigering

Under 1960-talet medverkade PTAB i en navigeringsstudie för AJ 37 (Ref. 1 sid 87). Flera tekniska system skulle samverka med flygplanets centraldator. För att verifiera konceptet framtogs en realtidssimulator för navigeringsfunktionen. PTAB fick disponera en datorprototyp (NSK 3) och genomförde en omfattande programutveckling avseende hyperbelberäkning, automatiskt sändarval och position filter, allt kopplat till en modell av flygplansrörelser, sändarsignaler samt en egenutvecklade långvågsmottagare, baserad på digital frekvenssyntes. PTAB fick härigenom en tidig kunskap om realtidsprogrammering och motsvarande datorarkitektur.

Datorhårdvaran var fortfarande mycket dyr och programmeringen måste vara tids- och kostnadseffektiv (assemblernivå). Flera algo-

ritmer togs fram som blev till nytta i kommande produkter, exempelvis avbrottsarkitektur, tabellhantering, koordinattransformering, digitalfilter, digitalreglering och inbyggd funktionsövervakning/funktionskontroll.

Tidig "Time-Sharing"

Under 60-talet tillkom ett antal leverantörer som erbjöd beräkningstjänster över telefonnätet och fjärrskriftterminaler. Programmeringsspråket var oftast BASIC som är lättanvänt och normalt interpreterande (långsamt). Tjänsterna lämpade sig för större beräkningar med begränsad volym in/utdata. PTAB använde några sådana tjänster för exempelvis antennberäkningar, filterdimensionering och tillförlitlighetsprediktering.

Tidiga kunddatorprojekt

Vid decennieskiftet 1960/70 fanns cirka 10 datorkunniga utvecklingsingenjörer, varav en handfull med realtidserfarenhet. För att utveckla kompetensen studerade företaget de kommersiellt tillgängliga militära datorerna Kongsberg SM-4 och Elliott 920 C i samband med ett arméprojekt.

Vidare studerades (civil) processdator GA SPC-16 för en träningsanläggning till Marinen (SELMA) och en (civil) HP 2116 för enhetsprovning.

1970-talet

Såväl utvecklingen som användningen av datorer och programvara tog fart i både militära och civila tillämpningar. PTAB kunde dra nytta av tidig tillgång till produkter och know-how via Philips internationella kontaktnät. Detta blev en viktig ingrediens i PTAB utveckling av både intern och kundinriktad datoranvändning.

Metallbearbetning

Sedan tidigare fanns en omfattande maskinpark för tillverkning av precisionsmekanik. I och med introduktionen av "digitala" produkter utökades kapaciteten för plåt- och panel-

tillverkning. I takt med teknikutbudet tillkom flera datorstyrda maskiner (Numerical Control).

Ref. 2 avsnitt 2 listar NC-läget 1974.

Kretskortprovning

Ett typiskt kretskort, storlek några kvadratdecimeter, kunde innehålla ett 50-tal komponenter och normalt två förbindningslager. Trots noggranna tillverkningsmetoder gick fel i komponenter, montering och lödning inte att undvika. Då ett färdigt kort representerade ett stort värde kunde det inte kasseras utan måste felsökas, ofta med stor arbetsinsats.

Med ökande komplexitet blev det lönsamt att automatisera kortprovningen. PTAB första autotestsystem levererades av HP (nuvarande Agilent) år 1970.

När digitaltekniken infördes i PTABs produkter ökade behovet av autotest med nya datorer och mätinstrument (Ref. 2 avsnitt 4).

Internationella Philips hade sedan början av 70-talet utvecklat minidatorfamiljen P800. Denna blev senare basen för autotestsatsningen Architecture of Computerized Test Systems (ACTS) som anskaffades av PTAB både för internt och för kundbruk.

Komponentprovning

Med ökande komplexitet och snabbhet blev ankomstkontrollen svårare och mer tidsödande. Dessutom var komponentkvalitén måttlig, typisk felhalt 5 %. Det blev lönsamt att automatisera även ankomstkontrollen.

Prognoser för komponentantal/komplexitet gjordes tillsammans med systerföretagen PTS och PIE (som efter 1975 skulle bli PEAB-T respektive PEAB-I). Genom befintliga kontakter inom Philips fanns tillgång till erfarenheter av aktuella testsystem. En viktig faktor var tillgång till komponenttestprogram (utöver Philipsdivisionen ELCOMA). På grund av tidigt behov av komplexa kretsar placerades den

första LSI-testutrustningen MD154 i PTABs lokaler och blev operativ 1974.

Många av komponentfelen visade sig först vid hög arbetsfrekvens. Mot slutet av 70-talet hade situationen för komponentprovning ytterligare försvårats. Prognoserna för komplexa kretsar, särskilt för PEAB-T, uppgick till minst 0,5 M per år. Vidare hade PEAB-D börjat få behov av testhastigheter över 5 MHz (gräns för MD 154) samtidigt som utrustningen var fullbelagd.

Efter en omfattande utredning anskaffades den marknadsledande utrustningen Fairchild Sentry 7, om förutom god lönsamhet möjliggjorde tillgång till många komponenttestprogram från andra användare. Största volymen LSI-kretsar fanns hos PEAB-T, varför Sentry 7 placerades där. Investeringen omfattade 2,5 Mkr år 1980.

Mönsterkortlayout

Fram till 1974 gjordes alla mönsterkortsmasker manuellt med svart tejp, oftast i skala 2:1. Processen var långsam och arbetskrävande. För åren 1974 och framåt förutsågs en anhopning av nykonstruktioner i samband med den nya generationen digital elledning Arte 724-726. Prognoserna visade ett behov av 20-30 layoutkonstruktörer för PTAB. Samtidigt hade ett antal leverantörer börjat marknadsföra olika datorstyrda hjälpmedel.

I samarbete med olika grupperingar inom Philips kom PTAB fram till två samhörande investeringar gemensamt för (det blivande) PEAB. 1974 anskaffades ett precisionsritbord typ Gerber 740 E och 1975 ett interaktivt grafiksystem typ REDAC, baserat på en minidator från DEC (Digital Equipment Corp). Dessa system var marknadsledande i sina segment. Genom denna investering kunde större delen av PEAB-D mönsterkortvolym klaras av 3-4 operatörer, dessutom med högsta mönsterkvalité och god lönsamhet.

Mot slutet av 70-talet hade konstruktionsvolymen inte avtagit. Dessutom fanns nya och effektivare hjälpmedel. I samband med anskaffning av DEC 20 (se nedan) fick PEAB tillgång till det integrerade systemet BNR-CPS, som bl.a. stöder schemaritning och upptill 25 grafikterminaler (DEC 2060). BNR-CPS anskaffades 1979 (Ref.3).

Dator för digital eldledning

Under 1970 lanserade Philips-CTI (Paris) minidatorn P880, avsedd för processtyrning. PTAB och CTI utredde gemensamt möjligheter till en militär version, kallad RCC 8800, närmast avsedd för JA 37.

Försvarets Materielverk hade siktat på en 24-bit dator (lika AJ-37) och tvekade inför RCC 8800 (16-bit). PTABs offert innebar större andel dubbelordsräkning och motsvarande ändringar jämfört med AJ 37 programsystem.

Som vi vet fick Saab fortsätta med sin 24-bit CK 37.

Under tiden undersökte PTAB andra möjliga leverantörer av 16-bit datorer (16-bit räckte för arméns och marinens behov).

Beträffande militära datorer fanns några typer hos Ferranti, GEC och Sperry. Vidare kontaktades Control Data och Honeywell.

Beträffande civila datorer besöktes NCC -73 i New York, varvid framkom två kandidater till militarisering.

Under 1971 lanserade Philips-CTI en ny minidatorfamilj, initialt benämnd SAGITTAIRE, senare omdöpt i P800-serien. PTAB första tillämpning (P855) blev sjöbevakningssystemet Stina för Marinen (inomhusmiljö).

Kravspecifikationer för PTAB militära minidator sammanställdes under 1972 (Ref.4 och 5).

Vid besök på CTI dec-72 upptäcktes en prototyp av en kommande datorgeneration, be-

nämnd P852. Tillstånd för militariseringsstudier (P852 M) anskaffades och detta blev kärnan i PTAB första digitala eldledning.

MINI-GO

Mini-go var en organisation för informationsutbyte inom Philipskoncernen med tonvikt på P800-tillämpningar.

Under 1977 höll PEAB i ett möte. Utdrag beträffande PEAB-D återfinns i Ref 6.

Produktprogramutveckling

En typisk produkt dator kunde levereras med grundläggande verktyg för editering, kompilering, länkning och filhantering. I PEAB realtidstillämpningar behövdes dessutom hjälpmedel för simulering och felsökning.

I Ref 6 beskrivs några sådana (för P800). Dessutom omnämnes "korsverktyg" arbetande på P800 för mikroprocessorprogramutveckling.

DEC 20

Under 70-talet hade ett större antal stödsystem anskaffats, arbetande på (olika) minidatorer och servicebyråer. En ide växte fram att uppnå en mer ensartad systemmiljö genom en lokal "engineering mainframe" med många terminaler. Utredningen av denna ide tog fart under 1977, varvid inriktningen blev DEC 20, den dåvarande marknadsledaren för (interaktiva) ingenjörstillämpningar.

Det visade sig att giganten IBM hade klumpiga och dyra lösningar för PEAB behov. När detta stod klart, kunde anskaffningen av DEC 20 motiveras efter noggranna förberedelser, bl.a. ett veckolångt studiebesök hos användare i USA. Investeringen omfattade 4 Mkr (1978).

Analog kretsutveckling

Även efter införandet av digital eldledning krävdes utveckling av anpassningar mot en analog omvärld. Ett användbart verktyg var PHILPAC, som var tillgängligt på DEC 20. Användargruppen omfattade ca 10 elektronikkonstruktörer.

Koncernsamarbete Philips

Tidigare omnämnt kontaktnät inom CAD nådde sin kulmen under slutet av 70-talet. Organisationen leddes av gruppen CAD Concern Committee (CCC) under ordförandeskap av tekniske direktören och forskningschefen Dr Pannenburg. CCC hade en liten verkställande ledning kallad Office of System Architects (OSA).

Utveckling av CAD-projekt sköttes av ett antal teknik kommittéer (TC) och arbetsgrupper. En översikt av läget 1978/79 ges i Ref 7.

Gränssytor TDB/ADB

Under "analogskedet" hade TDB och ADB (härmed avses administrativ databehandling) levt i skilda världar med enbart manuell dataöverföring. I och med DEC 20 utökades TDB-stödet, samtidigt som växande kunduppdrag krävde ökad ADB-insats. Inledande studier gjordes därför beträffande lämpliga gränssytor. En viktig gräns var den mellan Utveckling och Materialstyrning. Delresultat framgår av Ref 8.

1980-talet

VAX 11

I början av 80-talet blev DEC VAX/VMS populär för tyngre beräkningstillämpningar. Ett antal system anskaffades av PEAB. VAX/VMS valdes också i PEAB-D anbud på det danska ledningssystemet FOD/CCIS för NATO/Baltaps (Baltic Approaches). Denna beställning gick dock till Thomson (nuvarande Thales).

Egenutveckling av hårdvara avtog och i stället ökade satsningarna i produktprogram. Nya stödsystem för programutveckling växte fram. Läget 1982 beskrivs i ett Mini-go-föredrag (Ref 9). PEAB hade då 4 VAX-system med "korsverktyg" för P800. Detta behandlas vidare i samband med de olika produktgenerationerna (se respektive avsnitt).

AT

Sista fem åren med Philipsnamnet (1983 - 1988) sköttes dator drift och systemstöd av den gemensamma funktionen PEAB-AT. Bland Philipsgemensamma utvecklingsprojekt kan nämnas VLSI-konstruktion, där AT deltog med tre ingenjörer placerade i Malmesbury (EL-COMA England).

VHDL

Inom det Nationella IT4-programmet framtogs en utvecklingsmiljö för det hårdvarubeskrivande högnivåspråket VHDL. Flera digitalkonstruktörer från PEAB deltog i introduktionsutbildning.

Erfarenheter från ett tidigt demonstrationsprojekt redovisas i Ref 10.

Kartpresentation

Försvarsmakten och särskilt armén använder en omfattande kartdokumentation. I början av 1980-talet studerade PEAB möjligheterna för automatisering/datorisering av kartbladshandlingen. Ett uppenbart alternativ var att lagra kartdata i ledningssystemens digitala massminnen. Detta visade sig dock bli för dyrt med dåvarande minnespriser.

Då Philips lanserat videokivan (VLP/Laservision) blev det möjligt att lagra kartblad som videobilder till låg kostnad. Ett systemkoncept utvecklades, där kartbilder visades som bakgrundsinfo tillsammans med förbands- och stridslägesymboler som överlagringsbilder. Anordningen patentskyddades under tiden 1981-1994, se Ref 11.

En tillämpning för artilleriet beskrivs i Ref 12.

Anm. Fortsatt sjunkande minnespriser möjliggjorde under 1990-talet ekonomisk bildlagring på digitalminnen (digitalkameror, digital-TV, GIS-system)

Mekanik

För mekanisk konstruktion anskaffades systemet UNIGRAPHICS, som arbetade på VAX med

upp till fyra arbetsplatser och kunde generera data för tillverkningsberedning. Särskilda programvaror användes för värme- och hållfasthetsberäkningar.

Digitalhårdvara

För utveckling/verifiering och testgenerering anskaffades arbetsstationer från Apollo (senare övertaget av HP).

Gränsyta Utveckling/Materialstyrning

För att hantera artikel- och produktstrukturer utvecklade PEAB systemet CAS, som kunde ge releasedata till materialstyrssystemet PEMAS. CAS arbetade på DEC 20 och ersatte det tidigare P857-baserade stycklistesystemet (Ref 6).

PEMAS arbetade tidigare på IBM/servicebyrå med dataleverans per bud, men fick under 80-talet tillgång till en lokal IBM-dator i Järfälla. Som komplement till CAS anskaffades dokumenthanteringssystemet DOC.REG.

REFERENSER

1. Försvarelektronik från svenska leverantörer, FHT 2014-09-25
2. NC and computer systems used by PTAB, PTAB UA13-204, aug 74
3. CAD/CAM Integration in Professional Electronics, Computers in Industry, 1984
4. Requirement specification for a military minicomputer, PTAB UA4-158, aug 72
5. Environmental specification for a military minicomputer, PTAB UA4-159, sep 72
6. Mini-go sep 77 (utdrag)
7. Koncernsamarbete inom Teknisk Databehandling, PEAB DUD-459, nov 78
8. Samband Materialstyrning/Utveckling, PEAB DUD-428, feb 78
9. Mini-go okt 82 (utdrag)
10. VHDL in practice, VHDL User Group, April 1990, Boston
11. Utläggningsskrift "Presentationsanordning" prv.se/spd/ 8106091-5, ansökan inkom 1981-10-14
12. Produktbroschyr 9FA 400 LINDA, 198505

12. Förkortningslista

9LV	Fartygsbaserat lednings-/eldledningssystem från Saab (f.d. Philips)
Ada	Programspråk
APEX	Oracle Application Express utvecklingsmiljö
ASMD	Ainti-Ship Missile Defence
BEAB	Bofors Elektronikindustrier AB
BESK	Binär elektronisk sekvenskalkylator, en mycket tidig svensk dator
BIT	Built-in-test
BS 2000	Bassystem 2000
CISC	Complex Instruction Set Computer
COTS	Commercial off the shelf, Färdiga produkter på hyllan
D85	Dator 85
DEC	Digital Equipment Corporation
DIANA	Descriptive Intermediate Attributed Notation for Ada
DOS	Diskorienterat operativsystem
DRM	Realtidsoperativsystem från Philips i samarbete med universitet
DTOSS	Operativsystem från Philips, modifierat av dåvarande PEAB
EriOS	Ericssons operativsystem
EriPascal	Eriksson programspråk baserat på Pascal
ESSM	Evolved Sea Sparrow Missile
FMV	Försvarets Materielverk
FS 2000	Fartygssystem 2000
HAMA	Hardware Machine. Kopplat till DRM och SOMA
HDLC	High Level Data Link Controll, Kommunikationsprotokoll
HMI	Human Machine Interface = MMI
IEEE 802.3	Ethernet. Datakommunikationsstandard
ISO	International Standard Organization
Karin	Rörligt KA-batteri 12/80 (KA rörliga invasionsförsvar)
Kkv	Kustkorvett
LAN	Local Area Network, lokalt nätverk
Lfc 1 Lfc 2	Luftförsvarscentral typ 1 resp 2
LRU	Large replaceable units, Utbytesenheter
LSI	Large Scale Integration
MAP	Manufacturing Automation Protocol. Kommunikationsprotokoll
Mareld	Marint eldledningssystem
MASIK	Marinens stridsledning i krig
MEKO	Containerbaserat fartygskoncept från Blohm+Voss, som möjliggör att fartyget snabbt kan byta mellan olika roller
MFC	Multi-Function Console
MMI	Man-Maskin-interface
NIC	Nätverkskort
OO	Objektorientering
OS	Operativsystem
OS-9	Operativsystem
PEAB	Philips Elektronikindustrier AB
PIE	Philips Industrielektronik
PROM	Programmable Read-Only Memory
PTAB	Philips Teleindustrier AB
PTS	Philips Terminalsystem
Rakel	Arte 722. Radarbaserad kanoneldledning

RISC	Reduced Instruction Set Computer
RIR	Rörligt indikatorrum för radarstation
RRGC/F (Rgc/F)	Fast Radargruppcentral
RRGC/T (Rgc/T)	Transportabel radargruppcentral
RTL/2	Real Time Language, programmeringsspråk
SESUB	Strids- och eldledningssystem för ubåt
SESYM	Strid- och eldledningssystem för ytattack i marinen. Lednings- system för Kkv typ Göteborg
SF300	StandardFlex 300. Danska Flyvefiskerklassen med MEKO-koncept
SOMA	Software Machine. Kopplat till DRM och HAMA
SS2000, FS2000	Ship System 2000, Systemkoncept
Stina	Sjöbevakningssystem
STRIC	Stridsledningscentral för flygvapnet
Strics	Stridsledningssimulator
Stril-60	Luftbevaknings- och stridsledningssystem modell 60
TDMA	Time Division Multiple Access. Kommunikationsprotokoll
VASS	Vidmakthållande och vidareutveckling av STRILC och Strics
VDI	Virtual Device Interface

Ett systemföretag i Järfälla
var redo att delar beställa
från olika håll
men alltid med koll
på att vara INTERNATIONELLA

(Lars Fransson / Malte Jönson)



Deltagare vid framtagningen av dokumentet

Philips i Järfälla.

Från lokal tillverkare av apparater till systemleverantör på världsarenan

En industrihistorisk tillbakablick från åren 1960 till 2000 med tonvikt på programvaruutvecklingen

FHT M 01/2016