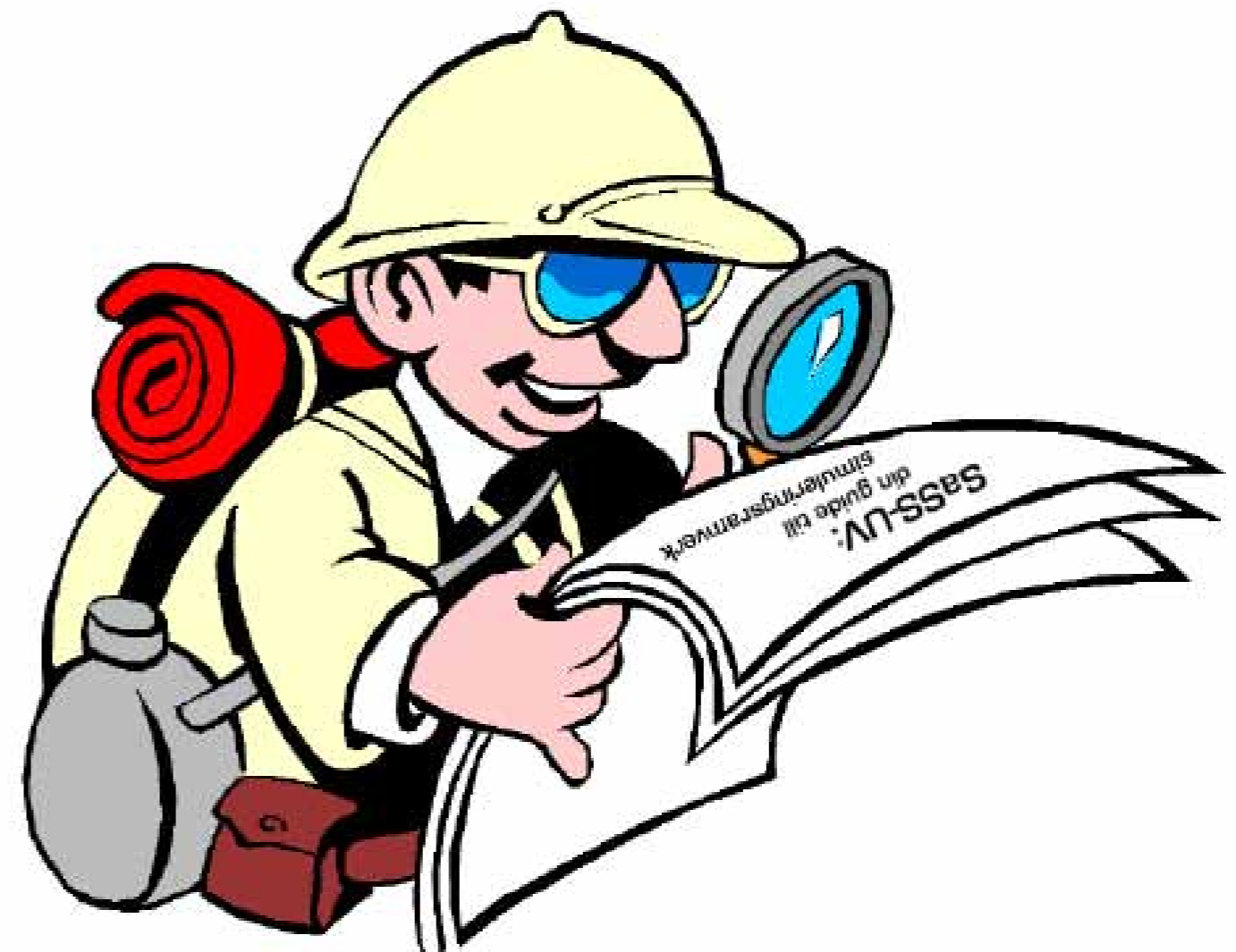


GÖRAN BERGSTRÖM, JOHAN PELO, MATTIAS KARLSSON,
VAHID MOJTAMED,



FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1350 anställda varav ungefär 950 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömningen av olika typer av hot, system för ledning och hantering av kriser, skydd mot hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

Göran Bergström, Johan Pelo, Mattias Karlsson,
Vahid Mojtahed

SaSS-UV, System-av-System Simulering med simuleringsramverk - inriktning UV

Utgivare FOI - Totalförsvarets forskningsinstitut Systemteknik 164 90 Stockholm	Rapportnummer, ISRN FOI-R--1769--SE	Klassificering Underlagsrapport
	Forskningsområde 2. Operationsanalys, modellering och simulering	
	Månad, år December 2005	Projektnummer I60125
	Delområde 21 Modellering och simulering	
	Delområde 2	
Författare/redaktör Göran Bergström Johan Pelo Mattias Karlsson Vahid Mojtahed	Projektledare Vahid Mojtahed	
	Godkänd av Monica Dahlén	
	Uppdragsgivare/kundbeteckning FOI – Avd. för Systemteknik	
	Tekniskt och/eller vetenskapligt ansvarig Vahid Mojtahed	
Rapportens titel SaSS-UV, System-av-System Simulering med simuleringsramverk - inriktning UV		
Sammanfattning FOI har tidigare framgångsrikt utnyttjat simuleringsramverk för att utföra studier för luft- och markstrid. I denna rapport presenteras en undersökning där målsättningen har varit att finna ett simuleringsramverk som även kan hantera undervattensdomänen. För att identifiera potentiella kandidater skickades en förfrågan ut till en stor mängd tänkbara leverantörer. De inkomna svaren med föreslagna produkter utvärderades sedan med bakgrund av en internt framtagen kriterielista, där både affärsmässiga och tekniska aspekter ingick. Efter en utgallring av de verktyg som inte klarade de grundläggande kraven återstod fem leverantörer, TNO, Ternion, Qinetiq, MÅK och CAE, vars produkter genomgick en mer detaljerad utvärdering. Resultatet har sammanställts på en enhetlig form för att möjliggöra rekommendationer av de lämpligaste simuleringsramverken.		
Nyckelord Simuleringsramverk, Simulering, UV		
Övriga bibliografiska uppgifter	Språk Svenska	
ISSN 1650-1942	Antal sidor: 59 s.	
Distribution enligt missiv	Pris: Enligt prislista	

Issuing organization FOI – Swedish Defence Research Agency Systems Technology SE-164 90 Stockholm	Report number, ISRN FOI-R--1769--SE	Report type Base data report
	Programme Areas 2. Operational Research, Modelling and Simulation	
	Month year December 2005	Project no. I60125
	Subcategories 21 Modelling and Simulation	
	Subcategories 2	
Author/s (editor/s) Göran Bergström Johan Pelo Mattias Karlsson Vahid Mojtahed	Project manager Vahid Mojtahed	
	Approved by Monica Dahlén	
	Sponsoring agency FOI – Div. of Systems Technology	
	Scientifically and technically responsible Vahid Mojtahed	
Report title (In translation) SaSS-UW, System of System Simulations with Simulation Frameworks - UW		
Abstract FOI has earlier applied simulation frameworks to study battles in the air and on the ground. This report presents an investigation where the objective has been to find a simulation framework which also can handle simulations in the under water domain. To identify potential candidates a RFI, request for information, was sent to several companies and organisations. An evaluation of the received answers and the suggested products was performed based on a list of criteria, with both technical and commercial aspects. After sorting out the simulation tools which did not meet our basic demands five suppliers remained, TNO, Ternion, Qinetiq, MÄK and CAE. Their respective products were then evaluated in more detail, and the results has been put together to make it possible to recommend the most promising simulation frameworks.		
Keywords Simulation Framwork, Simulation, UW		
Further bibliographic information	Language Swedish	
ISSN 1650-1942	Pages 59 p.	
	Price acc. to pricelist	

En kort översikt av rapporten

SaSS-UV står för System-av-System Simuleringar med simuleringsramverk - inriktning Under Vatten. Men vad innebär det egentligen att använda simuleringsramverk för sådana simuleringar i undervattensmiljö? För att få svar på denna fråga så startades under andra halvan av år 2003 det interna strategiska projektet SaSS-UV, vid avdelningen för Systemteknik på FOI.

Syftet med projektet var att undersöka möjligheten att kunna bygga en syntetisk miljö för olika undervattenssimuleringar med hjälp av konceptet simuleringsramverk, genom att:

- Utredda om det är tekniskt möjligt att använda simuleringsramverk för UV-området.
- Belysa de viktigaste aspekterna inför ett anskaffande av ett simuleringsramverk.
- Göra en grundläggande undersökning av existerande ramverk som kan användas för undervattenssimulering.

Denna rapport beskriver resultatet av detta samarbetsprojekt som har skett mellan de tre institutionerna Systemmodellering, Signalbehandling och Numerisk vågutbredning och analys vid avdelningen Systemteknik.

Målet med projektet har varit att enhetligt sammanställa hur väl de mest lovande produkterna uppfyller våra krav för att möjliggöra rekommendationer för lämpliga framtida simuleringsramverk. Ett övergripande mål med satsningen var att resultaten och rekommendationerna från projektet sedan skulle kunna användas av andra projekt som har behov av att genomföra simuleringar i UV-området. Framförallt projekt som behöver anskaffa ny simuleringsprogramvara. En bedömning av hur en organisation bäst kan dra nytta av de verktyg som utvärderats ligger därför utöver målsättningen, men diskuteras kortfattat i rapporten.

Vad är då ett simuleringsramverk? Ett sådant verktyg ska förse användaren med ett antal tjänster som är gemensamma för alla simuleringar och en mängd bra att ha saker, som en utvecklingsmiljö, ett antal standardverktyg, en uppsättning av exempelmodeller, o s v. Ett simuleringsramverk är därför ett komplext verktyg och kan uppfattas som en infrastruktur för M&S, som kan bedömas utifrån ett flertal perspektiv. Att etablera användandet av ett simuleringsramverk som bör klara av såväl olika problemdomäner (mark, luft, sjö, undervatten) som olika tillämpningar är en svår uppgift. För att lyckas krävs god kännedom om behov, såväl tekniska som organisatoriska. En alltför hög grad av specialisering gör visserligen det komplexa verktyget lättanvänt, men medför ofta oönskade begränsningar.

Inför ett sådant arbete så behöver den problemdomän som är föremål för M&S ses över. Undervattensmiljön med akustisk och elektromagnetisk signalpropagering är komplex, och det är inte självklart att alla fysikaliska fenomen och egenskaper som inverkar kan, eller ens bör beaktas i sådana simuleringar. Utöver de fysikaliska egenskaperna för omgivning och plattformar som är föremål för intresse, behöver också beteenden och uppgifter, som i verkligheten utförs av människor, samt situationer beskrivas och modelleras. Utöver dessa domänspecifika egenskaper så finns det ofta ett antal ”randvillkor” som i praktiken kommer att slå in som styrande krav vid anskaffning av ett simuleringsramverk. Dessa kan t ex vara tidsramar, ekonomiska förutsättningar, användningsområde, lämplig detaljeringsgrad, tekniska resurser eller tillgängliga kompetenser. I vissa fall kan något av dessa randvillkor ensamt avgöra valet av produkten.

Det tillvägagångssätt som valdes av oss i SaSS-UV var att inleda arbetet med en allmän informationsinhämtande fas, följt av en utvärderingsfas. Aktiviteter som genomförts är:

- Intern inventering av kunskaper på FOI om simuleringsverktyg för UV-området.
- Omvärldsbevakning av konferenser, utställningar och tidskrifter.
- Utskick av förfrågan till de leverantörer som identifierats efter mer information om deras respektive produkter.
- Kandidater som inte bedömdes kunna uppfylla våra grundläggande krav valdes bort.

- Djupare analys av fem leverantörer samt deras respektive produkter, där resultatet av undersökningen beskrivs i denna rapport.

Tidigt under arbetet med att granska de aktuella simuleringsverktyg som utgjorde fokus för projektarbetet framstod behovet av utvärderingskriterier som en nödvändig grund för att göra en rättvis och strukturerad bedömning av simuleringsramverkens för och nackdelar. Utvärderingskriterierna skulle möjliggöra en ensad bedömning av de olika verktygen och säkerställa att alla viktiga aspekter analyserades vid utvärderingarna. En sådan kriterielista indelad i affärsmässiga, arkitekturmässiga och applikationsmässiga aspekter har skapats och kommer också att beskrivas i rapporten. Vidare avrundas rapporten med ett antal rekommendationer och förslag till fortsatt arbete.

1.	Inledning	9
1.1.	Bakgrund.....	9
1.2.	Syfte	9
1.3.	Mål	10
1.4.	Metod	10
1.5.	Läsanvisning	10
2.	En tidig pilotstudie	12
2.1.	Syfte med försöket	12
2.2.	Undervattensmodeller och JEPE.....	12
2.3.	Simuleringsramverk och Flames.....	13
2.4.	Pilotscenario.....	13
2.5.	Försök - Integrering av JEPE och Flames.....	15
2.6.	Beteendemodell och analysverktyg	16
2.7.	Lärdomar från pilotförsöket.....	17
3.	Tillvägagångssätt för identifiering av potentiella simuleringsramverk	18
3.1.	Intervjuer på hemmaplan	18
3.2.	Omvärldsbevakning	18
3.3.	Identifiering av potentiella leverantörer och begäran av information om deras produkter.....	19
4.	Utvärderingskriterier	20
4.1.	Affärsmässiga aspekter	20
4.2.	Arkitekturmässiga aspekter.....	22
4.3.	Applikationsmässiga aspekter.....	24
5.	Fem valda kandidater	27
5.1.	Inledning	27
5.2.	TNO och en kortfattad beskrivning av MOSES	27
5.3.	Ternion och en kortfattad beskrivning av FLAMES	28
5.4.	QinetiQ och en kortfattad beskrivning av ODIN.....	28
5.5.	MÄK och en kortfattad beskrivning av VR-Forces	29
5.6.	CAE och en kortfattad beskrivning av STRIVE.....	30
5.7.	Detaljbeskrivningar.....	30
6.	Sammanfattning	52
6.1.	Kriterieuppfyllelse	52
6.2.	Verksamhetsnytta.....	53
6.3.	Slutsatser	53
6.4.	Rekommendationer	54
6.5.	Fortsatt arbete.....	54
7.	Referenser	56
	Appendix A – Request For Information (RFI)	57
	Appendix B – Företag till vilka RFI:n skickats	59

FOI-R--1769--SE

1. Inledning

För att kunna studera olika taktiska frågeställningar och prova skilda stridstekniska idéer med hjälp av avancerade scenarier i en simulerad miljö, behövs kraftfulla och komplexa simuleringsverktyg. En sådan miljö, simuleringsramverket Flames, har sedan flera år tillbaka använts på institutionen för Systemmodellering på avdelningen för Systemteknik på FOI. Under dessa år har ett antal FOI-projekt med olika mål och syfte utnyttjat detta simuleringsverktyg i olika kombinationer av luft-, mark- och sjöscenarier. På senare år har man också känt ett behov av att undersöka om, och i så fall hur, man kan använda konceptet simuleringsramverk även för undervattensområdet.

Med anledning av detta identifierade behov, startades det interna strategiska projektet SaSS-UV på avdelningen för Systemteknik på FOI, under andra halvan av år 2003. SaSS-UV står för "System-av-System Simulering med simuleringsramverk - inriktning Under Vatten" och syftar till att undersöka möjligheten att kunna bygga en syntetisk miljö för olika undervattenssimuleringar m h a konceptet simuleringsramverk. Ett simuleringsramverk ska förse användaren med ett antal tjänster som är gemensamma för alla simuleringar och en mängd bra att ha saker, som en utvecklingsmiljö, ett antal standardverktyg, en uppsättning av exempelmodeller, o s v.

1.1. Bakgrund

Projektet SaSS-UV började sin verksamhet genom att koncentrera sig på ett typfall där möjligheten att kunna integrera en befintlig modell för akustisk sonar i ett simuleringsramverk undersöktes. Typfallet byggdes i det på Systemteknikavdelningens befintliga simuleringsramverk Flames. Flames införskaffades 1997 från företaget Ternion och har sedan dess huvudsakligen använts för simulering av luftstrid, men även för mark- och sjöscenarier. Det arbete som genomfördes visade bland annat på att konceptet med simuleringsramverk är lämpligt för användning inom undervattensområdet. Det positiva provskottet tillsammans med nöjaktiga resultat från ett par andra undersökningar som implementering av styralgorithm för ubåt, motiverade igångsättandet av den större uppgiften; att studera vilka idag existerande simuleringsramverk som är anpassade för undervattensanvändningar, för vilka typer av uppgifter de är lämpliga och hur taktiska undervattenssimuleringar kan utvecklas m h a dem. Vidare var det önskvärt att dessa simuleringsramverk även skulle vara lämpliga för joint-scenarier, d v s kombinationer av UV-, mark- och luftmodeller i samma scenario.

Värt att nämna i sammanhanget är att nästan samtidigt drog FMV igång ett större simuleringsprojekt med fokus på undervattensområdet, UV-stridssimulator-demonstrator. Med tanke på beröringspunkterna med UV-stridsprojektet, bedömdes det lämpligt att göra en analys som kunde svara på frågan, vad det är som skiljer sig respektive är gemensamt för ansatserna SaSS-UV och UV-stridsdemonstratorn. Detta arbete genomfördes under 2004 och resultatet av denna analys har redan dokumenterats i en tidigare FOI-rapport [1]. I rapporten framgår fördelar och nackdelar med att skapa simuleringar i HLA (*High Level Architecture*) vilket gjordes i UV-stridsprojektet, jämfört med att använda ett simuleringsramverk. Trots att de två olika simuleringssätten kvalitativt skiljer sig åt på flera punkter pekades ingen metod ut som generellt bättre än den andra. Istället påvisar rapporten vid vilka tillfällen det ena sättet passar bättre att använda än det andra.

1.2. Syfte

Huvudsyftet med att dra igång projektet SaSS-UV var att undersöka möjligheterna med att utnyttja simuleringsramverk för modellering och simulering av system av system på stridsteknisk, taktisk och operativ nivå, med undervattensmiljön som utgångsdomän. Projektet skulle också ge en helhetsbild av området simuleringsramverk och en uppdatering på det kommersiella utbudet. Frågeställningar som för övrigt skulle beaktas var bl a vilka utvecklingstrender som kan ses inom

simuleringsramverksområdet, hur dessa kan komma att påverka framtida modellutvecklingsprojekt, återanvändning av modeller och om dessa är lämpliga för joint-scenarier. Vidare var ambitionen att där så erfordras skulle framtagna forskningsresultat kunna bidra till modifieringar, förbättringar och eventuellt tillägg till existerande simuleringsramverk.

1.3. Mål

Det viktigaste målet för SaSS-UV var att göra en omvärldsskanning och identifiera vilka simuleringsramverk, både nationellt och internationellt, som existerar idag och som kan användas för UV-området. Vidare skulle dessa verktyg undersökas i syfte att kartlägga vilka typer av uppgifter de är lämpliga för. En av de mest intressanta frågeställningarna var vilka möjligheter som finns inbyggda i dessa simuleringsmiljöer för att snabbt och enkelt kunna bygga taktiska scenarier, köra dem och analysera resultatet. Vidare var det viktigt att ha uppfattning om vilka insatser som krävs för att integrera egenutvecklade modeller och tredjepartsprodukter. Att ha möjlighet att vidareutveckla själva simuleringsramverket, och att få stöd för modellering av mänskligt beteende och beslutsprocesser för taktiska simuleringar, var andra viktiga aspekter som beaktades.

Som slutprodukt för verksamhetsår 2005 skulle projektet leverera ett bedömningsunderlag samt om möjligt en rekommendation av ett simuleringsramverk som ska anskaffas och användas i första hand för UV-området på FOI, men som även ska ha potentialen att användas gemensamt på FOI och i Försvarmakten för luft-, mark-, yt- och UV-modeller.

1.4. Metod

Tillvägagångssättet vi valde för att lösa denna uppgift var att börja med ett inventeringsarbete på hemmaplan genom att samla erfarenheter och göra intervjuer, samtidigt som vi övervakade och deltog i internationella konferenser och utställningar, i syfte att identifiera alla möjliga kandidater. Ett omfattande skanningsarbete drogs igång, och en officiell begäran, RFI – Request For Information (se Appendix A), skickades till de identifierade potentiella leverantörerna. Nästa uppgift var att ta hem tillräckligt mycket material, licenser, dokumentation och övrig information om de mest intressanta simuleringsramverken. Parallellt med detta arbete skapades en kriterielista med vars hjälp dessa verktyg skulle utvärderas, se kapitel 4. Vi försökte gallra bort kandidater som inte klarade de viktigaste kriterierna, utvärderade de mest lovande simuleringsramverken, undersökte i vilken grad de uppfyllde vår kriterielista, samt beskrev dem på ett enhetligt sätt så att det skulle bli lättare att jämföra dem och kunna klargöra deras verksamhetsnytta. Detta skulle utgöra det eftersökta underlaget och underlätta en eventuell rekommendation av det mest lämpliga simuleringsramverket. Denna rapport kommer att beskriva detta arbete och det uppnådda resultatet.

1.5. Läsanvisning

Efter denna inledning kommer vi i kapitel 2 att presentera en pilotstudie och syftet med detta försök som gjordes ganska tidigt i projektet. Vi kommer att presentera det genomförda integrationsarbetet av en hydroakustisk modell (JEPE) i ett simuleringsramverk (Flames), givet ett enkelt scenario. Vi avslutar detta kapitel genom att sammanfatta våra lärdomar från pilotförsöket.

I kapitel 3 beskriver vi vilket tillvägagångssätt vi valde och vilka metoder vi använde för att uppnå målet. Här kommer vi att redovisa såväl vårt inventeringsarbete som vår omvärldsbevakning och resultatet av dessa insatser.

För att möjliggöra en jämförelse mellan olika verktyg och en utvärdering av dessa, så har ett antal kriterier tagits fram. Dessa utvärderingskriterier återges i kapitel 4, med fokus på tre huvudområden: affärsmässiga, arkitekturmässiga samt applikationsmässiga aspekter på ett sådant simuleringsramverk.

I kapitel 5, som för övrigt kan uppfattas som huvudkapitlet i denna rapport, sammanfattar vi

våra erfarenheter av de fem olika produkter som vi tittat närmare på. I samband med varje produktbeskrivning kommer vi i möjligaste mån ge en bedömning på hur dessa produkter uppfyller varje kriterium, beskrivna i kapitel 4. Detta kapitel kan uppfattas lite tungt och på sina håll lite för detaljerat, men läsaren uppmuntras att åtminstone läsa den korta sammanfattningen av varje produkt.

I slutet av rapporten finns ett sammanfattande kapitel, där ett antal viktiga frågeställningar såsom verksamhetsnytta och kriterieuppfyllelse diskuteras. Slutligen avrundar vi rapporten genom att dra några slutsatser och där det är möjligt även ge några rekommendationer.

2. En tidig pilotstudie

SaSS-UV, är en akronym och står för System-av-System Simulering med simuleringsramverk - inriktning UV, men vad innebär det egentligen att använda simuleringsramverk för sådana simuleringar i undervattensmiljö? Det är en fråga, eller snarare en grupp av frågeställningar som behöver besvaras för att kunna nå framgång med konceptet. Frågorna måste dessutom besvaras innan simuleringsverktyg anskaffas och utvecklingsarbetet kan komma igång i större skala.

Framförallt behöver den problemområde som är föremål för M&S ses över. Undervattensmiljön med akustisk och elektromagnetisk signalpropagering är komplex, och det är inte självklart att alla fysikaliska fenomen och egenskaper som inverkar kan, eller ens bör beaktas i sådana simuleringar. Utöver de fysikaliska egenskaperna för omgivning och plattformar som är föremål för intresse, behöver också beteenden och uppgifter, som i verkligheten utförs av människor, samt situationer beskrivas och modelleras.

Simulering av dessa plattformar och beteenden för undervattensområdet är ingen ny företeelse. Även om frågeställningar och detaljeringsnivå skiljer sig åt från tidigare arbeten så finns där ändå ett arv av modeller som är viktigt att även i fortsättningen kunna förvalta. Dessa modeller kan inte byggas om över en natt för att passa i ett nytt verktyg, det måste i stället vara möjligt att anpassa verktyget till att också kunna använda äldre, befintliga modeller av olika slag.

2.1. Syfte med försöket

En metod för att få en tillräckligt djup förståelse för ovan nämnda frågeställningar, är att göra en pilotstudie, där väl valda delar av fysikaliska fenomen och egenskaper, befintliga modeller och kännetecknande beteenden implementeras i ett simuleringsramverk. Produkten av detta arbete blir inte främst den körbara prototypen, utan den kunskap och ökade förståelse som genereras under arbetets gång.

Upplösningen på ingående modeller har därför valts så att de är tillräckligt detaljerade för att vi ska kunna dra nyttiga erfarenheter av arbetet, och inte för att kunna dra så stor nytta som möjligt av den framtagna prototypen.

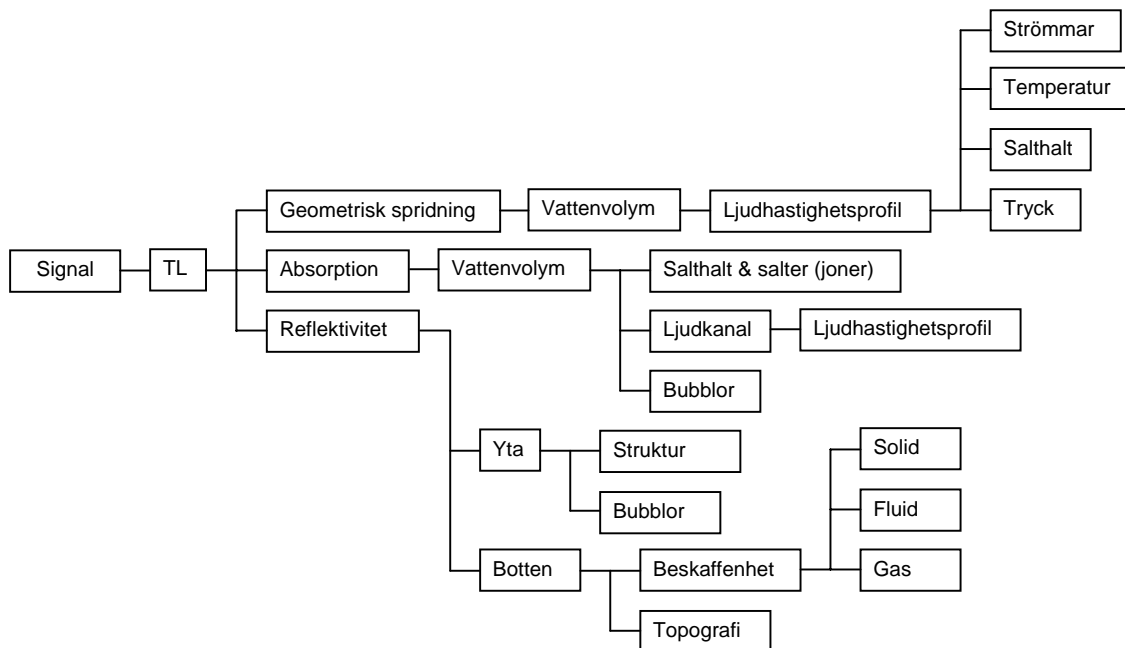
2.2. Undervattensmodeller och JEPE

Valet av modell för integration var inte enbart ett val av simuleringsmodell, utan också ett val mellan hydroakustik och elektromagnetik. Implementering av både akustik och EM bedömdes för omfattande att implementera i en prototyp.

Valet föll på JEPE, som är en hydroakustisk modell, bland annat för att modellen även klarar varierande bottengeometrier. Nlayer, som gör EM-beräkningar, klarar bara planskiktade modeller, en begränsning som inte var önskvärd, med tanke på möjligheten att i ett senare skede kunna öka graden av realism i scenariot.

Valet av JEPE gav också möjligheten att använda intensitetsmodellering framför strålgångsmodellering, en egenskap som var önskvärd av flera skäl, (se figur 2.1). Vid motsvarande scenarier för luftstrid, används radarekvationen, istället för att modellera enskilda radarsvep. Att pröva möjligheterna till strålgångsmodellering hade i sig varit mycket intressant, men ställer också betydligt högre krav på att initialt ha en fungerande omvärldsmoell med hög upplösning.

De beräkningar som görs i JEPE (förutsatt fasta sensorer) är av den karaktären att de kan göras en gång per sensor/mål-par för hela det intressanta området. Resultatet i form av utdatamatrix behöver då hållas associerat med målet av sensorn under simuleringens gång.



Figur 2.1. Akustiska sensorer i UV-miljö. Vid intensitetsmodellering där endast sonarekvationen används beror signalintensiteten på ljudutbredningsdämpningen (TL).

2.3. Simuleringsramverk och Flames

Valet av simuleringsramverk styrdes till stor del av att FLAMES fanns tillgängligt inom FOI, vilket gäller även den kompetens för modellutveckling som behövdes.

Flames har sitt ursprung från M&S av luftstrid, och har under senare år byggts ut till att också kunna hantera scenarier som utspelar sig på marknivå, då verktyget försetts med en vektorbaserad terrängmodell för att kunna modellera vägnät och annan infrastruktur. Undervattensmiljön, med alla fysikaliska fenomen som hör till, existerar inte ens som begrepp i FLAMES. Vattenområden definieras av att området klassas som vatten i omgivningsmodellen, och har en höjd över havet som är noll.

Frånsett undervattensmiljön, fanns dock mycket att hämta från luftstridskoncepten. Framför allt bedömdes konceptet med klasser för plattformar, sensor, signatur och signalbehandlare kunna användas även för akustiska signaturer och sensorer, då radarekvationen har vissa likheter med sonarekvationen. Möjligheterna att modellera beteenden fanns redan, och påverkas inte nämnvärt av att plattformarna flyttar ner under vattnet. Verktyg för visualisering mm bedömdes kunna användas med mindre modifiering allteftersom behov uppstod.

2.4. Pilotscenario

Som vid modellutveckling i många andra sammanhang är det oftast bra att ta fram ett dimensionerande scenario, för att kunna knyta olika utvecklingsinsatser till en mer konkret användning av modellerna. För detta arbete är interaktionen mellan enskild sensor och mål/signatur det viktigaste målet, och scenariot för de inledande simuleringarna blev därför förhållandevis enkelt.

Geografi:

Det område som ska representeras i simuleringen bör avgränsas till ett kvadratisk område med sidan 10 km och som utsträcker sig från havsytan till ett djup av ca 60 m där ett 10 meters sedimentlager och berggrund vid. För att förenkla beräkningar och användande av befintliga

modeller betraktas volymen, inklusive botten, som planskiktad, dvs botten är helt plan och saknar topografi. Eventuella skikt i vattenvolymen betraktas också som plana. I övrigt finns inga geografiska objekt som öar eller kustlinjer i vattenvolymen.

Hydrografi:

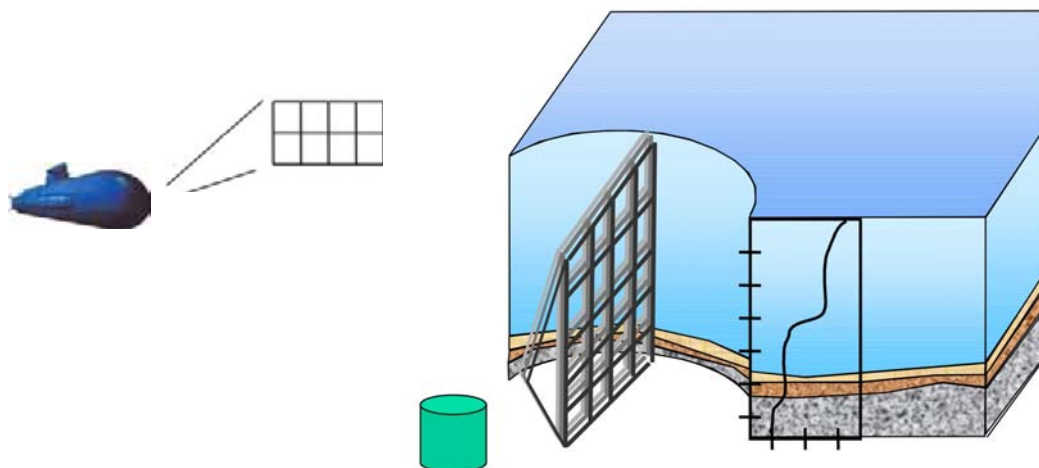
Förutom de rent geografiska avgränsningarna, bör vattenvolymen även ha vissa hydrografiska egenskaper. Dessa bör minst inkludera densitet, ljudhastighetsprofil och eventuellt även absorption. Om man önskar öka detaljeringsnivån, kan vattenvolymen även ges egenskaperna tryck, temperatur och salthalt, ur vilka ljudhastighetsprofilen och absorptionen kan beräknas.

En med vattenvolymen associerad egenskap är bakgrundsbullernivå. Om man definierar ett antal bullerkällor och deras intensitet, skulle man sedan kunna beräkna vilken intensitet dessa har på olika ställen i vattenvolymen. I detta scenario låter vi dock brusnivån vara konstant över hela vattenvolymen.

Hydroakustik:

Själva scenariot består av att en undervattensfarkost av typen miniubåt eller UUV som rör sig längs en förutbestämd kurs genom vattenvolymen. I vattenvolymen finns även en passiv, hydroakustisk sensor som försöker upptäcka undervattensfarkosten.

Undervattensfarkosten har egenskaperna hastighet, kurs, volym (längd, höjd, bredd), position samt signatur. Signaturen förenklas till att endast bestå av ett antal (5) diskreta frekvenser med tillhörande intensitet.



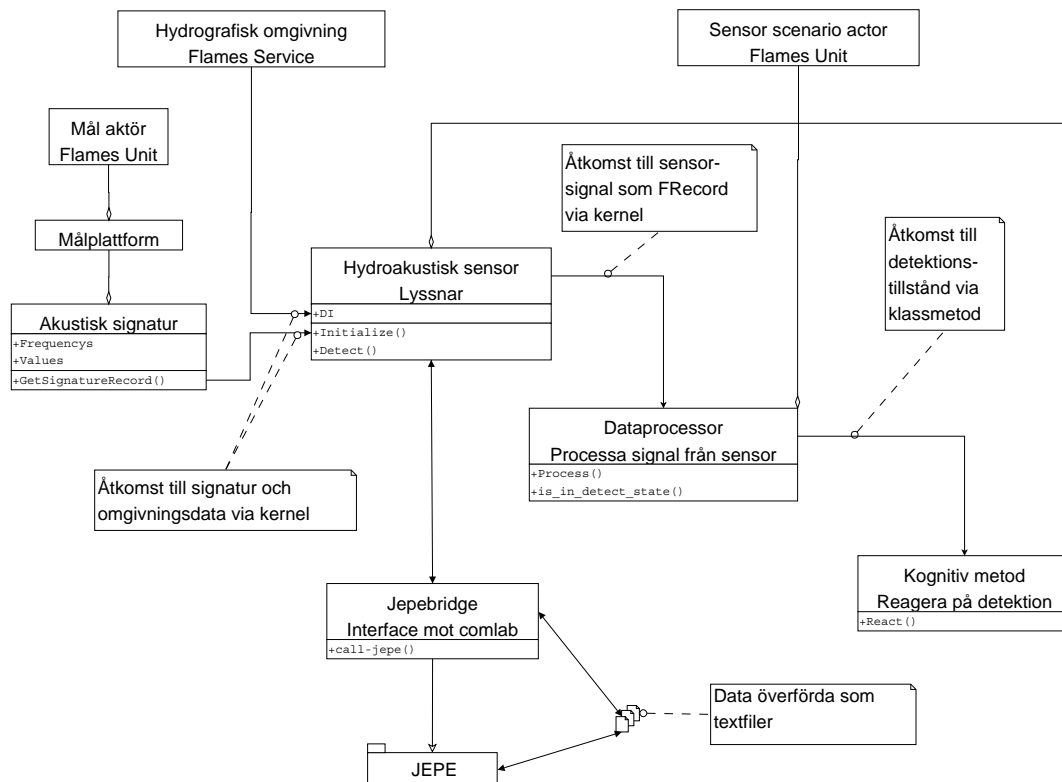
Figur 2.2. Konceptskiss av undervattensmiljö och entiteter. Ubåtens signatur beskrivs med en tabell innehållande fem frekvens- och värdepar. Vattenvolymen och havsbotten beskrivs med ljudhastighetsprofilen. Sensorn är omnidirektionell, och beräknar ett antal tabeller för vågutbredningsförluster, en tabell för varje frekvens. Vid simuleringen användes en planskiktad modell istället för en varierande botten topografi som i figuren.

Sensorn har egenskaperna position och volym (längd, höjd, bredd). Vidare förenklas sensorn till att vara omnidirektionell, vilket gör att den saknar förstärkning i någon riktning. Slutligen ska sensorn kopplas till någon form av detektionsalgoritm. I enklast möjliga fall kan detta utgöras av en egenskap, en detektionströskel, som har ett fixt värde. Om bullret från undervattensfarkosten överstiger detta värde upptäcks farkosten direkt och med 100 % sannolikhet. För att öka detaljeringsgraden och trovärdigheten bör detta modelleras mer ingående.

2.5. Försök - Integrering av JEPE och Flames

JEPE är skrivet i Fortran och går bra att modifiera till en funktion som kan länkas in i FLAMES och anropas från C. Det går också bra att använda comlab som ett fristående program, och passa in- och utdata till och från Flames som textfiler i ett första steg.

För att inkorporera JEPE i FLAMES, har ett antal delmodeller utvecklats. Dessa berör huvudsakligen tre områden: Omgivning, Mål och Sensor. Vidare har en del arbete gjorts för att visualisera detektionen på ett för uppgiften relevant sätt.



Figur 2.3. En principskiss av de olika komponenterna och deras inbördes relation. Pilarna illustrerar dataflödet mellan komponenterna. Den aktör i scenariot som utgör sensorn använder även en dataprocessor samt en kognitiv modell för att reagera på detektion.

Plattformsmodellen utgörs av en modifierad exempelmodell för fartyg, som kan färdas under vatten, bära en akustisk signatur, och positioneras på en negativ altitud, dvs ett djup. En akustisk signatur har implementerats, som i stort kan beskrivas som en tabell med fem frekvens- och värdepar.

Omgivningsdata (ljudhastighetsprofil och bottenbeskaffenhet) har gjorts tillgängliga för sensor och signatur i FLAMES, genom att den hydrografiska omgivningen har implementerats som en tjänst, där delmodeller som sensor och signatur kan fråga omgivningen efter dess egenskaper, framför allt ljudhastighetsprofilen. Däremot har bottenpografien inte tagits i beaktande, utan en planskiktad modell har använts.

Aktören sensor i scenariot är uppbyggd av flera delmodeller: sensor, dataprocessor samt kognitiv modell. Modellen sensor är dels en mappning av JEPE, men också ansvarig för sonarekvationen. Utdatasignalen från sensorn är värden på SNR (*Signal to Noise Ratio*). Som framgår av figur 2.3 var det inte omgivningen, utan sensorn som blev den mest komplexa delen

att implementera. Dels för att gränssnittet mot comlab kräver extra omsorg, men också för att sensorn byggs upp av en detektor, logik för signalbehandling, och en reaktionsmodell.

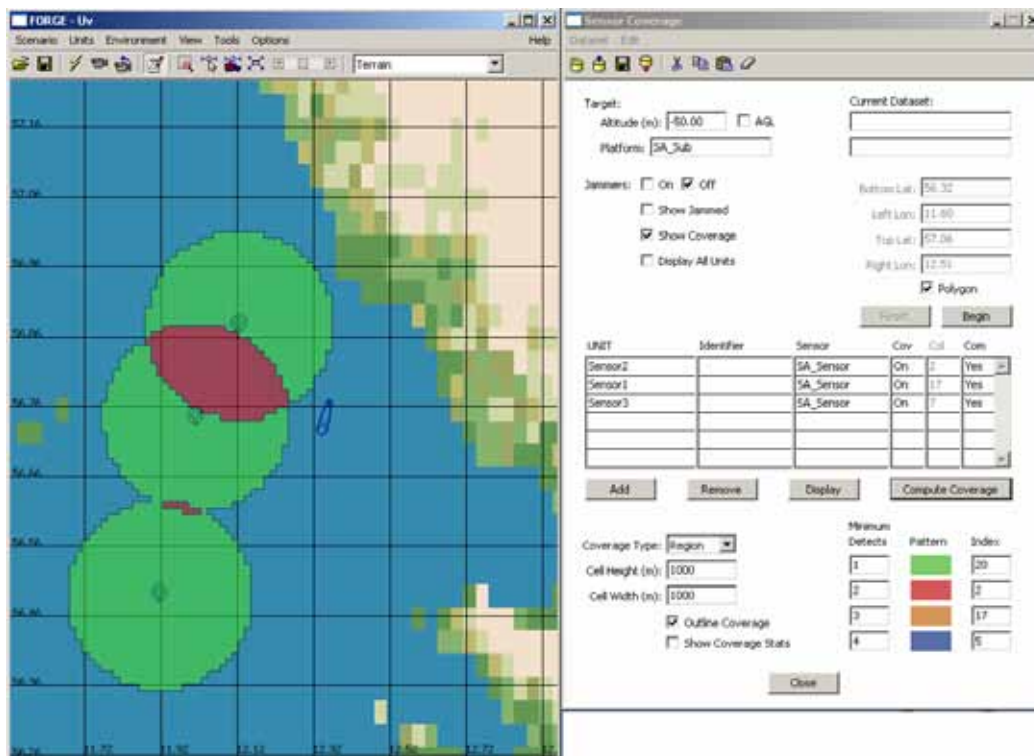
JEPE initieras i början på simuleringen, då ett antal utdatamatriser, en för varje frekvens, från JEPE fås som resultat. I de efterföljande tidsstegen hämtas värden på överföringsförluster (TL) från dessa utdata från JEPE. Ljudhastighetsprofil och brusnivå samt målsignaturens frekvenser är indata till JEPE. Då JEPE i befintlig form är implementerad i Fortran som ett fristående program, har vi för enkelhetens skull valt att använda textfiler för dataöverföring. Detektorn lämnar utdata i form av ett signal-brusförhållande (SNR), som är en interpolering av utdata från JEPE.

Dataprocessorns uppgift är att utifrån sensorns aktuella SNR-värden avgöra detektion eller ej. Detta görs genom att avgöra om summan av sensorns SNR-värden överstiger ett av användaren angivet tröskelvärde. Vidare är dataprocessorn förberedd för att också hantera falsklarm, något som blir intressant om stokastiskt brusnivå införs.

Reaktionsmodellens uppgift är att reagera på en detektion. I detta fall så är reaktionen att visualisera detektionen, vilket åstadkoms genom att skriva data till playback-filen. Visualisering av detektion görs i Flash, och är därför separerad från de ovan nämnda komponenterna, med undantag för de data som infogas i playback-filen.

2.6. Beteendemodell och analysverktyg

Det dimensionerande scenariot i avsnitt 2.4 gav ett fokus på interaktionen mellan en enskild sensor och signatur. Av olika skäl gavs tillfälle att implementera mer funktionalitet än så. Bland annat ett rörelsemönster för ubåtsjaktfartyg, funktionalitet för visualisering av sensortäckning och mindre modifieringar av visualiseringsverktyg.



Figur 2.4. Sensortäckning. Ett av analysverktygen i FLAMES kan användas för att visualisera täckningsområdet för en grupp av sensorer. Upptäckt beräknas mot en given signatur, på en specificerad höjd över havsytan, i det aktuella fallet -50 meter, genom att sensorerna beräknar detektion för ett stort antal olika målpositioner.

De verktyg för analys och visualisering som behövs för undervattensområdet kan skilja sig från de som är brukligt vid luft- och markscenarier. Den rent modellmässiga integrationen med undervattensområdet är bara en del av arbetet. De insatser som krävs för att också anpassa applikationer för analys och visualisering ska inte underskattas, även om de är betydligt mindre komplicerade. Verktöget för att visualisera sensortäckning som visas i figur 2.4 ska inte bara presentera relevanta resultat, utan också vara användarvänligt för andra än modellutvecklaren.

Inom ramen för pilotstudien har också vissa anpassningar av visualiseringsverktygen i FLAMES gjorts. Bland annat har en ny 2D-vy tagits fram som visualiserar aktörernas positioner i djupled. Anpassningar av den standardiserade 2D-vyn från ovan har gjorts för att visualisera när sensorernas signal/brusförhållande överstiger tröskelvärdet för detektion.

2.7. *Lärdomar från pilotförsöket*

Vad kan vi då dra för slutsatser av pilotförsöket? Framför allt finns det här en klar fördel i att använda intensitetsmodeller framför strålgångsberäkningar när likheterna mellan sonar- och radarekvationen är så pass stora. Konceptet med akustiska sensorer skiljer sig dock något från det som kan återfinnas i exempelmodellerna för luftstrid, bland annat är begreppet målspar förrädiskt. Vår implementering med en omnidirektionell sensor ger varken bäring eller avstånd, i stället ges ett signal/brusförhållande (SNR). Implementeringarna av radar bland exempelmodellerna i FLAMES ger utöver bäring och avstånd även en hastighetsvektor, vilket medför att problemet att skilja mellan ekon från olika mål blir betydligt mer komplicerad för vår akustiska sensor.

Arbetet med att integrera JEPE i FLAMES kan i stort sett sammanfattas som att överbrygga de mer fundamentala skillnaderna mellan dessa verktyg. Där JEPE beräknar signalpropagering mellan en enskild sensor och signatur i två dimensioner, är FLAMES ett verktyg för att beräkna utfallet av komplexa scenarier i tre dimensioner. JEPE använder sig av en komplex hydrografisk omgivning, där FLAMES använder en mycket enkel atmosfärmodell.

För pilotstudien har vi implementerat en enkel omgivningsmodell vid sidan av det gängse tillvägagångssättet. För att verktyget ska vara praktiskt användbart behövs dock en bottenografi, som kan användas såväl av ingående modeller för signalpropagering som för visualisering. Hantering av omgivningsmodeller i allmänhet, topografi och dess visualisering i synnerhet, är också en av de egenskaper som görs bäst av ramverket.

3. Tillvägagångssätt för identifiering av potentiella simuleringsramverk

Som beskrivet i inledningen har projektets slutmål varit att leverera ett bedömningsunderlag samt om möjligt en rekommendation av ett simuleringsramverk som ska anskaffas och användas i första hand för UV-området på FOI. För att kunna genomföra detta krävdes en identifiering av ett antal möjliga simuleringsramverkskandidater. I detta kapitel följer en närmare beskrivning av tillvägagångssättet som valdes för att identifiera möjliga leverantörer och produkter.

Den övergripande strategin var att inleda med en allmän informationsinhämtande fas. Fasen bestod dels av en intern inventering av kunskaper på FOI om simuleringsverktyg för UV-området, dels en omvärldsbevakning av konferenser, utställningar och tidsskrifter. Baserat på denna information skulle en mängd tänkbara leverantörer av ett simuleringsramverk identifieras. Slutligen skulle dessa leverantörer tillfrågas efter mer information om deras respektive produkter.

3.1. Intervjuer på hemmaplan

Den första åtgärden som genomfördes i den informationsinhämtande fasen var att intervjua experter inom FOI om deras erfarenheter och kunskaper om simuleringsverktyg för undervattensområdet. Under intervjuerna framkom att det inom FOI finns många modeller som beskriver detaljerade händelser, t ex vågutbredning av olika fält (akustiska, elektromagnetiska, optiska, etc) i vatten. Däremot finns det betydligt färre verktyg på en högre nivå som sammanställer en stor mängd av de framtagna modellerna. Några undantag var de mer omfattande verktygen såsom MUMS (*Modell för Undervattens-Motmedels-Simulering*) [2] och COMBIS (*Combined Maritime Background Information System*) [3]. Inget av dessa verktyg uppfyllde dock de grundläggande kraven som ställs på ett simuleringsramverk. Vidare framkom det att det existerar ett par produkter i Europa som var specialiserade på simuleringar inom UV-området och som skulle kunna benämnas som simuleringsramverk. Det var MOSES som är utvecklat av TNO i Holland och ODIN som är utvecklat av QinetiQ i England. Flera av experterna påpekade även att det finns åtskilliga avancerade simuleringsverktyg på den amerikanska marknaden. En första studie av detta visade på ett enormt utbud inom den amerikanska försvarsmarknaden, både inom försvarsmyndigheten och inom försvarsindustrin. GOTS-produkter (*Government Of The Shelf*) prioriterades dock bort i ett tidigt skede eftersom det var osäkert om det skulle existera ett simuleringsramverk av typen GOTS som skulle vara väsentligt mycket bättre än motsvarande COTS-produkterna (*Commercial Of The Shelf*). Om det skulle finnas skulle det dessutom vara svårt att få tillgång till det verktyget och en support i klass med den som kommersiella företag erbjuder i samband med försäljning av sina COTS-produkter. Inom USA har fokus därför legat på simuleringsramverk av typen COTS.

3.2. Omvärldsbevakning

Parallellt med den interna inventeringen genomfördes en omvärldsbevakning av konferenser, utställningar och tidsskrifter. Tidsskrifterna studerades inte för att i detalj läsa om enskilda produkter, fokus låg istället på att finna potentiella leverantörer. I flera av tidsskrifterna fanns sammanställningar av relevanta simuleringsprodukter. Ett exempel är MS&T [4] där en mängd olika typer av mjukvara för simuleringar presenterades. Bevakningen av konferenser och utställningar utgjordes genom att studera inbjudningar och välja ut de mest relevanta för uppgiften. Dessa konferenser analyserades sedan dels genom att tala med personer som besökt dem, dels genom att studera artiklar från konferensernas *proceedings*. Ett undantag i bevakningsmetodiken var konferensen och utställningen IITSEC'04 (*Interservice/Industry Training, Simulation and Education Conference*) i Orlando december 2004 som även besöktes eftersom det är en av de största och viktigaste konferenserna inom simuleringsområdet. På plats i Florida fanns flera av de

företag FOI tidigare hade haft kontakt med, och tillfälle gavs för att få en samlad bild av deras respektive produkter, och en uppdatering av senaste nytt i deras utveckling. Utöver detta knöts flera nya kontakter med intressanta leverantörer. En sammanställning av deltagande företag och presenterade bidrag på konferensen finns i [5] och de för uppgiften mest intressanta leverantörerna är vidare beskrivna i kapitel 5.

Vid sidan av simuleringsramverken uppmärksammades också andra simuleringsrelaterade produkter på I/ITSEC, som dataspel. En av aktörerna inom detta område var Sonalyst, som marknadsförde Dangerous Water, ett av de senaste dataspelen för undervattensområdet. Spelet är en nedbantad version av ett större simuleringsverktyg som företaget har levererat till US Navy. Arkitekturen på spelet känns väl genomtänkt och det går att införa egna databaser och aktörer i spelet genom att skriva egna tilläggsmoduler, vilket normalt inte brukar vara fallet för kommersiella spel. Någon vidare analys av spel ingick dock inte i denna studie och därför har inte heller Dangerous Water analyserats ytterligare.

3.3. Identifiering av potentiella leverantörer och begäran av information om deras produkter

Baserat på den information som samlades in i den första informationsinhämtande fasen identifierades en mängd tänkbara leverantörer av simuleringsramverk. Med vissa leverantörer fanns väl uppbyggda kontakter medan andra bara var kända till namnet. För att få en mer nyanserad bild av alla leverantörer och deras intresse av att leverera ett simuleringsramverk som uppfyllde våra krav tillfrågades de om deras respektive produkter. Förfrågan som skickades ut var en RFI (*Request For Information*) baserad på kravbilden som ställdes på ramverket och på de tilltänkta användningsområdena. Redan innan arbetet med att sammanställa potentiella leverantörer hade startat existerade det en grundläggande kravbild på simuleringsramverket, denna uppdaterades sedan kontinuerligt fram till dess att RFI:n skickades. RFI:n finns presenterad i appendix A, och alla identifierade leverantörer som RFI:n skickades till är listade i appendix B.

4. Utvärderingskriterier

Tidigt under arbetet med att granska de aktuella simuleringsverktyg som utgjorde fokus för projektarbetet under 2005 framstod behovet av utvärderingskriterier som en nödvändig grund för att göra en rättvis och strukturerad bedömning av simuleringsramverkens för och nackdelar. Utvärderingskriterierna skulle möjliggöra en ensad bedömning av de olika verktygen och säkerställa att alla viktiga aspekter analyserades vid utvärderingarna. Kriterierna viktades dock inte mot varandra, eftersom en sådan viktning förutsatte en betydligt mer detaljerad kännedom om för vilka uppgifter verktyget ska användas till. Av samma anledning ansågs det inte heller lämpligt att genomföra en punktvis jämförelse av de olika verktygen i detta skede. En del kriterier belyser nämligen bara aspekter där vad som är bra respektive dåligt helt beror på tillämpningen. Vidare skulle en sådan punktvis jämförelse endast vara möjlig genom att ha tillgång till lika mycket material (utvecklingslicenser, dokumentation, presentationsmaterial, tillämpningsexempel, etc) om alla valda verktyg, vilket har varit långt ifrån fallet i vårt arbete.

Arbetet med att skapa en lista över lämpliga utvärderingskriterier bygger dels på erfarenheter från tidigare års arbete inom detta projekt och dels på praktiska erfarenheter från ett stort antal forsknings- och utvecklingsprojekt som har utnyttjat simuleringsramverk på FOI under många år. Vidare har vi försökt ta till vara på alla de krav och behov som har visat sig vara gemensamma för flertalet av de stora simuleringsprojekt som har bedrivits på FOI sedan början av 80-talet. Denna samlade erfarenhet har även möjliggjort att vi kunnat undvika olika fallgropar som annars hade varit lätta att falla i.

Listan över utvärderingskriterier är indelad i tre huvudspår – affärsmässiga, arkitekturmässiga och applikationsmässiga aspekter. Varje spår är sedan indelat i ett antal områden under vilka kriterier som berör likartade aspekter har samlats. Utvärderingskriterierna i sig själva är formulerade som punktsatser och till varje punktsats finns en förklarande text. Detta kapitel kan, och i vissa sammanhang rekommenderas det t.o.m. att användas som en fristående biprodukt av detta projekt.

4.1. Affärsmässiga aspekter

Leverantörsbeskrivning

- **Företagets storlek**
Det är av vikt att ha grepp om hur stort företaget som utvecklar produkten är; både totalt sett och i relation till FOI. Med företag avses här såväl privata företag och affärsdrivande statliga bolag som myndigheter. Detta kan ge en indikation om vilken form en eventuell framtida affärsrelation kan få. Ett litet företag är mer sårbart än ett stort och är mer beroende av att dess kunder lägger nya beställningar. Samtidigt kan ett litet företag vara mer lyhört för önskemål från sina kunder. Är företaget mycket stort finns risk att vi blir en lågt prioriterad kund som får stå tillbaka när andra, större kunder har önskemål.
- **Ägarförhållanden för företaget**
Företagets ägarförhållande inverkar på hur vi bör förhålla oss till företaget. Förhållandet till ett privat företag regleras med kontrakt och avtal och har enbart kommersiella aspekter. Förhållandet till andra, utländska myndigheter kommer i grunden även det att styras av avtal och kontrakt, men det finns också möjligheter till mer kollegiala kontakter och informationsutbyten. Kontakter med utländska statliga aktörer kan styras av policybeslut från departement och regering.
- **Företagets syn på kundrelationer**
Detta kan vara svårt att få klarhet i, men är viktigt. Sannolikheten är betydligt större att ett företag som är angeläget om sina kundrelationer fortsätter att uppdatera och vidareutveckla

programvaran. Ett sådant företag torde även vara mer lyhört för återkoppling och önskemål från kunderna.

Produktsupport

- **Kommersiell eller icke-kommersiell produkt**

Kommersiella produkter är på det hela taget enklare att handskas med och förhållandet mellan säljare och köpare finns reglerat i detalj. Om vi köper en kommersiell produkt kan vi t ex förvänta oss support i form av uppdateringar etc av tillverkaren. Skulle produkten vara ofullständig kommer säkerligen uppgraderingar. Är det däremot inte en kommersiell produkt finns risk att vi får en halvdan produkt och att vi inte får ta del av vidareutvecklingar och förbättringar. I värsta fall kan kontakten med tillverkaren upphöra helt.

- **Dokumentation**

För att kunna få ut största möjliga nytta av produkten och för att vi inte ska behöva gå kurser hos tillverkaren om och om igen är en bra dokumentation av största vikt. Denna ska vara lättillgänglig – både vad gäller läsbarhet och användning – och omfatta alla delar av den programvara vi har köpt.

- **Exempelmodeller och –scenarier**

För att underlätta inlärning och fortsatt användning är det högst önskvärt att en rad exempelmodeller och övningsscenarier medföljer den produkt vi köper. En viktig poäng med anskaffandet av ett nytt simuleringsramverk är att detta ska gå att använda för *rapid prototyping* och för att snabbt illustrera nya koncept. I dessa fall spelar tillgängligheten av exempelmodeller stor roll.

- **Kurser**

För att vi snabbt ska nå en tillräcklig kunskapsnivå när det gäller användning av en ny produkt och för att vi ska kunna nyttja denna effektivt krävs formell utbildning, t ex i form av användar- och utvecklarkurser givna av tillverkaren. Därför är det viktigt att veta vilka kurser tillverkaren tillhandahåller för sina kunder och på vilka nivåer dessa kurser ligger.

- **Användargrupp**

Vissa företag har användargrupper där olika användare av samma eller snarlika produkter kan utbyta erfarenheter och stödja varandra. Då vi antagligen kommer att vilja arbeta en hel del med egna modeller och algoritmer skulle en användargrupp vara ett bra stöd. Detta gäller särskilt om det finns andra användare som också arbetar med integration av egna modeller och algoritmer.

- **Eventuella andra användare av produkten**

Om antalet användare är för få så finns det en risk att hamna i en beroendeställning till tillverkaren. Vidare kan andra användare säga en del om styrkor och svagheter hos produkten samt ge en indikation om vilka utvecklingsvägar företaget kommer att fokusera på. Om det exempelvis är många användare med inriktning mot flygstrid är produkten antagligen väl lämpad för detta. Vidare är det troligt att företaget kommer att fokusera på att utveckla dessa delar i första hand. Slutligen, ju fler andra användare som finns, desto större möjlighet att det finns användare som är både villiga att utbyta användbara erfarenheter med oss.

Kostnader och rättigheter

- **Licenskostnader**

Vad skulle det kosta att köpa produkten? Är priset uppdelat i olika typer av licenser, t ex utvecklingslicens och användarlicens. Här avses främst de engångskostnader som uppstår vid ett köp.

- **Underhållskostnader**
Finns det några kostnader som återkommer varje år, t ex årsavgifter eller kostnader för att få ta del av uppdateringar?
- **Långsiktiga kostnader**
Finns det några kostnader på längre sikt som går att förutse? Har produkten t ex några speciella hårdvarukrav som kommer att leda till ökade kostnader när hårdvaran i fråga blir omodern? Tillkommer kompetensutvecklingskostnader i form av kursavgifter eller liknande?
- **Köparens rättigheter**
Finns det några begränsningar i nyttjanderätten av en inköpt produkt? Är det t ex tillåtet att exportera resultat framtagna med hjälp av produkten till andra nationer? Kan vi ta skärmdumpar från simuleringen och använda i våra rapporter? Har vi rätt att ta del av källkoden till produkten?

4.2. Arkitekturmässiga aspekter

Skalbarhet/prestanda

- **Kan verktyget användas för såväl stora (>100 aktörer) som små (<5 aktörer) scenarier?**
Finns det några begränsningar i antalet aktörer eller enheter som simuleringsverktyget kan hantera? Är det fokuserat mot antingen småskaliga eller storskaliga scenarier? Var på skalan stridsteknik-taktik-operationer ligger verktygets huvudfokus?
- **Aggregering och disaggregering av förband och enheter**
I vissa fall kan det vara önskvärt att antingen kunna hantera en grupp aktörer som en helhet snarare än var för sig eller att påverka vissa enheter i en större sammanhållen grupp enskilt. Hur hanteras detta i simuleringen? Är det möjligt att slå samman enskilda aktörer till en grupp? Kan en sammanslagen grupp delas upp i enskilda aktörer? Vad händer när en sammanslagen grupp utsätts för påverkan, t ex får en order eller blir beskjuten?
- **Skalbarhet**
Vad händer när antalet aktörer stiger? Ett verktyg som hanterar få-mot-få-situationer väl kan tappa prestanda avsevärt om antalet aktörer stiger – uppdateringar av spelbordet sker mer sällan, tidshanteringen går dåligt och kommandon tas inte emot.
- **Upplösning**
Finns det några begränsningar vad gäller detaljeringsnivån för modeller och algoritmer? Går det att byta eller ändra modeller och algoritmer så att det går att gå från ett stridstekniskt scenario till ett scenario på operativ nivå utan att resultaten blir felaktiga?

Utbyggbarhet/Anpassning

- **Möjlighet att integrera egna modeller**
Integration av egna modeller kräver att det finns tydliga och väldokumenterade gränssnitt (API:er - *Application Program Interface*). Integration kräver också att data till och från de egna modellerna kan hanteras. Slutligen måste det även gå att stänga av eller ta bort verktygets inbyggda modeller.
- **Komplexitet vid integrering av egna modeller**
Finns det t ex någon funktion för att importera modeller? Måste den egna modellen skrivas om eller förses med någon form av skal.
- **Stöd för att skriva nya modeller**
Vissa simuleringsverktyg har stödfunktioner som hjälper en utvecklare genom att skapa ett skal till en ny modell som innehåller viktig, generell information. Alternativet är att utvecklaren måste skriva en ny modell från grunden.

- **Möjlighet att modifiera befintliga modeller samt komplexitet för detta**
Kan en utvecklare överhuvudtaget förändra de modeller som följer med simuleringsverktyget?
Kräver det avancerade kunskaper i programmering?
- **Stöd för utbyggbarhet**
Är simuleringsverktyget byggt på ett sådant sätt att det enkelt går att byta ut vissa funktioner mot egna varianter? Finns det stöd i dokumentation och gränssnitt (i form av API:er) för att göra detta? Om verktyget är modulärt uppbyggt kan detta underlätta eventuella förändringar.
- **Vilka programmeringsspråk stöds?**
Vilka programmeringsspråk kan användas för att skriva nya modeller och funktioner? Om det går att integrera egna modeller skrivna i andra språk, t ex genom ett skal, vilka språk stöds av denna funktion?

Teknisk plattform

- **Utvecklings- och användarmiljö**
Det är önskvärt att simuleringsverktyget går att använda med standarddatorer som går att köpa som hyllvara och som nyttjar standardprogramvara som passar i den nuvarande IT-miljön vid myndigheten. Om verktyget antingen är plattformsoberoende eller går att använda med fler än ett operativsystem vore det bra.
- **Stöd för parallell exekvering**
För att förbättra prestanda då man t ex vill simulera stora scenarier med många aktörer är det önskvärt att olika delar av verktyget kan exekveras parallellt – antingen på olika processorer i samma dator eller på flera sammankopplade datorer. Vidare vore det också intressant att kunna starta flera instanser av samma del för att t ex koppla flera operatörskonsoler mot samma simuleringskärna.
- **Möjlighet till distribuerade körningar**
I vissa fall är det önskvärt att kunna låta användare som befinner sig på annan plats delta i en simulering. Det kan också vara önskvärt att koppla ihop sitt verktyg med en annan simuleringsmodell. Därför bör verktyget ha stöd för distribuerad simulering, t ex i form av färdiga anslutningsprotokoll. Den för tillfället vanligaste standarden för detta är HLA.
- **Övriga krav i form av tredjepartsprodukter**
Helst bör simuleringsverktyget kunna installeras och användas som det är när det köps in. Om inte är det viktigt att få reda på vilken annan programvara som krävs, till vad den krävs samt om det innebär ytterligare kostnader. Dessutom måste det klargöras om nyttjanderättigheterna påverkas av denna tredjepartsprogramvara.

Andra intressanta tekniska egenskaper

- **Arkitekturens utformning**
Är verktyget modulärt? Hur är de olika funktionerna uppdelade?
- **Särskilda styrkor eller svagheter med utformningen**
Finns det några kända eller uppenbara fördelar och nackdelar med den valda utformningen?
Exempelvis om den är särskilt robust eller resurskrävande.
- **Hantering och lagring av information**
Hur ser informationsflödet ut? Lagras information internt inom verktyget eller används extern lagring, t ex i en databas? Går det att läsa data, loggar, etc under pågående simulering?
- **Öppna eller slutna format**
Vilket format har eventuell utdata- eller loggfiler? Ett vanligt format som kan läsas med i stort sett vilket program som helst, t ex textfiler, eller ett speciellt format, t ex binärt, som kräver en särskild applikation för att läsas?

- **Möjlighet att länka information till och från externa källor**
Går det att läsa in information från externa källor, t ex en databas, en textfil eller en annan applikation? Kan verktyget även exportera information?
- **Stabiliteten (teknisk mognad) hos tekniken/arkitekturen**
Är simuleringsverktyget byggt enligt etablerade standarder? Hur känd är den tekniska lösningen och finns den dokumenterad?
- **Exekvering**
Hur sker exekvering? Går det att exekvera en klump med simuleringar i en följd eller måste varje simuleringsfall startas var för sig? Behöver verktyget nollställas eller korrigeras på annat sätt mellan olika simuleringsfall?
- **Interaktion mellan entiteter**
Hur sker interaktionen mellan olika delar av simuleringsverktyget? Finns det någon form av tidsstyrning eller –kontroll? Hur sker uppdatering av spelbord, aktörer, etc?

4.3. *Applikationsmässiga aspekter*

Funktionalitet

- **Stöd för att konfigurera modeller/aktörer**
Finns det någon funktion i verktyget som gör det möjligt att konfigurera modeller och aktörer, t ex att ändra frekvensområden för en radarmodell eller ändra beteendet hos markstridsförband vid sammanstöt med fienden.
- **Stöd för (efter)analys och återuppspelning**
Går det att spela upp det som händer under exekvering av ett scenario för att vid ett senare tillfälle spela upp inspelningen och se händelseförlopp och dylikt? Finns det några möjligheter att analysera händelser och skeenden i scenariot – t ex om ett visst förband blir beskjutet eller räckvidden för en utvald sensor vid ett givet ögonblick?
- **Förmågan att göra detaljerade loggningar för efteranalyser**
Är det möjligt att spara olika data från scenariot löpande under en exekvering för att sedan analysera dessa? Går det att använda vanligt förekommande program som t ex MS Excel för att analysera loggade data eller krävs det specialprogram?
- **Stöd för avancerad visualisering**
På vilka sätt går det att visualisera det som händer under exekveringen? Går det t ex att visa händelseförlopp i en 3-dimensionell vy?
- **Stöd för batch-körningar**
Går det att förbereda ett stort antal fall (med olika typer av parametervariationer) av samma scenario och sedan exekverar alla i en följd? Går det att lagra resultatet från flera olika scenarier som exekveras i en följd utan att blanda data?
- **Stöd för scenariohantering**
Är det möjligt att med simuleringsverktyget skapa nya scenarier och sedan spara dem? Går det att öppna ett sparad scenario med verktyget för att modifiera det och sedan spara scenariot på nytt?
- **Möjlighet att hantera joint-scenarier**
Går det att exekvera scenarier där enheter från de olika försvarsgrenarna kan interagera och verka på samma arena? Kan t ex fartygsförband lämna måldata till flygplan?

Signatur

- **Modellering av hydroakustiska och elektromagnetiska signaturer**
Kan simuleringsverktyget hantera hydroakustiska och elektromagnetiska signaturer?

- **Modellering av andra typer av signaturer som t ex radarmålyta eller IR-signaturer**
Går det att modellera olika typer av signaturer i simuleringsverktyget? Kan ett flygplan t ex tilldelas en radarmålyta som påverkar radarsensorers förmåga att upptäcka flygplanet?

Sensor

- **Modellering av hydroakustiska och elektromagnetiska sensorer**
Finns det möjlighet att skapa sensorer som kan upptäcka hydroakustiska och elektromagnetiska signaler, t ex de magnetfält som omger fartyg och ubåtar?
- **Färdiga modeller för hydroakustiska och elektromagnetiska sensorer**
Innehåller simuleringsverktyget några färdiga modeller för hydroakustiska eller elektromagnetiska sensorer?
- **Modellering av andra typer av sensorer**
Kan simuleringsverktyget hantera andra typer av sensorer som t ex radar eller IR-sensorer?

Plattform

- **Möjlighet att lägga till nya varianter av befintliga plattformar**
Är det möjligt att ändra och modifiera befintliga plattformsobjekt som redan finns i simuleringsverktyget?
- **Möjlighet att lägga till helt nya typer av plattformar**
Finns det någon möjlighet att skapa helt nya typer av plattformar som inte finns sedan tidigare? Exempel på sådan skulle kunna vara svävare eller VTOL-flygplan (*Vertical Take Off and Landing*).
- **Begränsningar gällande plattformens olika förmågor**
Har verktyget några inbyggda begränsningar för plattformsobjekt? Ett exempel skulle kunna vara att objekt klassade som fartyg inte kan gå över land. Om inte, går det att begränsa rörelsemönster för plattformar på något annat sätt?

Omvärld

- **Omgivningsmodell för undervattensområdet**
Har verktyget någon färdig omgivningsmodell som beskriver undervattensdomänen? Finns det stöd för att beskriva undervattensomgivning och definiera viktiga egenskaper som t ex salthalt och densitet, konduktivitet?
- **Komplexitet vid modifiering av omgivningsmodellen för nya scenarier**
Är det möjligt att ändra viktiga parametrar för omgivningsmodellen direkt i verktyget eller måste detta göras i externa data- eller textfiler?
- **Möjlighet att modifiera andra omvärldsaspekter så som atmosfär/terräng/infrastruktur**
Finns det möjlighet att ändra andra delar av omvärldsmiljön än undervattensdomänen, t ex terräng eller vägar?
- **Möjligheten att importera befintliga mätdata**
Finns det något sätt att nyttja verkliga mätdata i ett scenario, exempelvis uppmätt salthalt i Östersjön?

Beteendemodell

- **Stöd för modellering och simulering av mänskligt beteende**
Hanterar verktyget modellering av mänskligt beteende som t ex konfrontationsregler?

- **Möjlighet att studera beteende på operativ/taktisk/aggregerad nivå**
Kan verktyget hantera beteende även för aggregerade förbandsenheter där ingående enheter kan ha olika beteendemönster?

Tillämpningsområde

- **Applikationens ursprungliga syfte**
Syftet med verktyget brukar ofta påverka val av systemlösning, konfigurerings- och modell-design.
- **Förutsättningar för att kunna använda verktyget för andra områden/syften än det ursprungliga**
Om det är möjligt att få en överblick av modelldesignen för simuleringsverktyget kan det vara möjligt att avgöra dess lämplighet för andra syften än det ursprungliga. Denna möjlighet ökar även om det går att ändra konfigurationen av verktyget.

Kompetenskrav för utvecklare och användare

- **Resurser för utveckling**
Vad krävs för i form av hårdvara, tredjeparts mjukvara och kompetens hos utvecklingsteamet för att det ska gå att utveckla egna modeller m m?
- **Kompetenskrav för användare**
Vilka krav ställs på en användare som önskar nyttja verktyget, t ex skapa egna scenarier och exekvera dessa, med befintliga modeller?

5. Fem valda kandidater

5.1. Inledning

Som tidigare nämnts identifierades ett antal möjliga leverantörer av simuleringsramverk, och de tillfrågades om information om deras respektive produkter. Baserat på de svar som kom in och på information som lämnats via personliga kontakter genomfördes en första gallring. Kandidater som inte bedömdes kunna uppfylla våra grundläggande krav valdes bort. Bland de grundläggande kraven på ett simuleringsramverk finns en simuleringskärna som tillhandahåller hantering av t ex. tid och objekt, att verktyget har en öppen arkitektur där egenutvecklade modeller kan inkluderas, dokumentation för såväl utvecklare som användare och att det är stabil leverantör som står bakom ramverket. Efter detta urval återstod fem leverantörer som bedömdes tillräckligt intressanta för att följa upp närmare. Det var TNO, QinetiQ, CAE, Ternion och MÄK.

För att kunna göra en utförligare studie av leverantörernas produkter behövdes i de flesta fall ytterligare information. Undantaget var Ternion eftersom deras ramverk FLAMES redan har använts under flera år på FOI. I övrigt gjordes olika typer av fördjupade studier. MÄK kompletterade sin presentation på utställningen IITSEC med ett besök på FOI där de gick igenom arkitektur och grundläggande användande av sitt ramverk VR-Forces. TNOs produkt MOSES fanns beskriven i några konferensartiklar, men en tydligare bild av ramverket gavs genom ett besök på TNO i Holland och en efterföljande utvärdering av en demoversion som de skickade i efterhand. QinetiQ anordnade en grundlig demonstration i Weymouth, där deras ramverk ODIN beskrevs i detalj. CAEs följde upp sin presentation av ramverket Strive på IITSEC med att skicka en demoversion av ramverket och en presentation av deras senaste uppdatering Strive Sonar, som är den mest relevanta delen för UV-området.

Nedan följer en sammanfattad (kap 5.2-5.6) och en detaljerad (kap 5.7) sammanställning av erhållen information från respektive leverantör, samt resultat från tester av demoversioner av produkterna. Tester har inte varit möjliga att genomföra med alla kandidater. Där tester ej kunnat genomföras har vi valt att ändå sammanställa den information vi haft tillgänglig, för att ge en så god bild som möjligt av verktygen. Mycket av den information som presenteras här är resultatet av våra egna undersökningar och de slutsatser vi dragit.

5.2. TNO och en kortfattad beskrivning av MOSES

TNO är en holländsk forskningsorganisation som ska agera som kunskapskälla för såväl den offentliga sfären som det privat näringslivet. TNO är en med svenska mått ganska stor organisation med ungefär 5000 anställda som är aktiva inom fem kärnområden – Livskvalitet, Försvar & Säkerhet, Vetenskap & Industri, Bebyggelse & Geovetenskap samt Informations- & Kommunikationsteknik.

Under 2004 hade TNO, inklusive dess olika dotterbolag, en omsättning på 555,8 m€ och en vinst på 4,7 m€

MOSES är ett simuleringsramverk framtaget av TNOs avdelning för Försvar & Säkerhet. Grunden för verktyget är ett antal äldre operationsanalytiska modeller. Syftet med att skapa MOSES var att öka återanvändningen och tillgängligheten av framtagna modeller. Målsättningen har från början varit att MOSES ska kunna användas såväl för modellutveckling som för studieverksamhet, t ex operationsanalys.

MOSES är framtaget för att användas för modellering och simulering av undervattenskrigföring. I huvudsak verkar TNO nyttja ramverket för studier kring skydd mot torpedanfall, minjakt och minsvepning, samt utveckling av koncept för multistatiska sensorer och (SAS) syntetisk apertursonar.

Ramverket kan användas för studier av få-mot-få-situationer och det är möjligt att göra ett stort antal exekveringar i en följd med viss parametervariation mellan varje. Det förefaller inte

möjligt att kunna aggregera eller disaggregera förband i ett scenario. En simulering kan exekveras i realtid, snabbare än realtid eller långsammare. Loggning av data görs via ett observatörsobjekt. I grundutförandet erbjuder ramverket endast visualisering av en simuleringsomgång i två dimensioner, men enligt uppgift ska det gå att ersätta visualiseringskomponenten med ett amerikanskt verktyg, SIMDIS, som ger tredimensionell visualisering.

När det gäller integration och implementering av egna modeller finns det bra stöd för detta, inklusive en funktion som genererar kodstubbar för nya modeller. Det krävs dock utvecklarekompetens i C/C++ och eventuellt även i Java för att kunna göra något mer avancerat. TNO har själva integrerat äldre modeller skrivna i Fortran och Matlab med MOSES. Ramverket verkar ha bra stöd för modellering av undervattensmiljö, yt- och undervattensfarkoster samt av undervattenssensorer och –signaturer.

5.3. Ternion och en kortfattad beskrivning av FLAMES

Ternion är ett litet¹, privatägt² företag med huvudkontoret beläget i Huntsville, Alabama, USA. Företaget är indelat i två olika avdelningar: *Software Products Division* och *Engineering Services Division*. Den förstnämnda avdelningen har ansvaret för utvecklingen av simuleringsprogramvaran FLAMES, och därtill hörande marknadsföring, försäljning, support och träning. Den andra avdelningen tillhandahåller expertis och utveckling av specialiserade modeller och programvara till användare av FLAMES.

Företaget grundades 1989, och har idag ett stort antal försvarsrelaterade organisationer och företag från USA, Europa och Asien som kunder.

FLAMES, eller *FLexible Analysis Modeling and Exercise System* som akronymen står för, är ett kommersiellt simuleringsramverk som visat sig vara väl flexibelt och anpassningsbart. Verket går att använda för olika tillämpningar, såväl taktikutveckling, träning som analytiska studier.

Verket har gott stöd för modellering och simulering av mark-, luft-, och sjöstrid, framförallt på taktisk nivå. Idag ligger fokus på luftstrid eftersom det är det de flesta av Ternions kunder är inriktade mot. En anpassning till markstrid har också gjorts. Stöd för undervattensområdet saknas, men mycket talar för att det är fullt möjligt att anpassa verket även till detta område. Undervattensstillämpningar ställer dock en del nya krav på verketets standardprogramvaror för redigering, analys och visualisering, så viss utveckling behövs.

Även om simuleringar av få-mot-få-situationer är möjliga har ramverket sin styrka i mer taktiska simuleringar med många entiteter. Det är möjligt att göra batch-körningar med viss parametervariation mellan varje körning, men det finns inget inbyggt stöd för aggregering eller disaggregering i ett scenario. En simulering kan exekveras i realtid, snabbare än realtid eller långsammare. Den mängd av data som loggas genom skrivning till resultatfiler kan modifieras efter behov. Visualisering kan ske i såväl 2D som 3D med hjälp av ett medföljande verktyg.

Utveckling av egna modeller görs främst i C eller C++. Det tar en viss inlärningsstid innan en utvecklare kan skriva sina egna modeller även om arbetet underlättas av en mängd befintliga modeller och ett kodgenereringsverktyg. Trots att ramverket saknar stöd för undervattenssimuleringar har lyckade försök med integrera en akustisk undervattenssensor genomförts. Vidare utveckling för undervattensdomänen kommer dock att ställa en del nya krav på ramverket, och bör framförallt ha inverkan på den befintliga klasshierarkin.

5.4. QinetiQ och en kortfattad beskrivning av ODIN

QinetiQ är ett statligt ägt affärsbolag inom området Försvar och Säkerhet. Företaget bildades när den tidigare brittiska forskningsmyndigheten DERA 2001 delades i två delar: en kommersiell och

¹ Uppskattningsvis mellan 20 till 40 anställda, officiell uppgift saknas.

² Alla aktieägare är heltidsanställda inom företaget.

en för forskning. Idag har QinetiQ totalt mer än 9 000 anställda. QinetiQ är en stor aktör inom försvarsrelaterad FoU i Storbritannien och hade 2004 en omsättning på £m 795 och en vinst på £m 39.

ODIN är en händelsestyrd simuleringsmodell för undervattensdomänen, framtagen av dåvarande DERA på uppdrag av Storbritanniens försvarsdepartement (MoD). I samband med delning av DERA tog QinetiQ över ansvaret. Arbetet med ODIN började redan 1996 och syftet vara att skapa ett gemensamt, integrerat simuleringsramverk för UV-stridsdomänen som skulle medge återanvändning av framtagna modeller.

I dagsläget används ODIN främst inom området torpedstrid, vilket inkluderar såväl torpeder som andra plattformar och motmedel. Målsättningen med verktygen är att det ska kunna användas för att stödja såväl materielanskaffning som studieverksamhet. Till stor del nyttjas även ramverket för att studera frågor som effekten av nya lobbildnings- och målföljningsalgoritmer eller för kravformulering för framtida torpedmålsökare (DCL), dvs sådant som normalt sett ligger inom signalbehandlingsområdet.

ODIN är främst fokuserat på att simulera få-mot-få-scenarier. Det är möjligt göra ett stort antal upprepade exekveringar av samma scenario med parametervariation mellan varje. Det verkar inte vara möjligt att kunna aggregera eller disaggregera förbandsenheter i ODIN. Själva exekveringen av ett scenario går väldigt fort och det är därmed möjligt att simulera snabbare än realtid. Vid exekvering av ett scenario loggas resultaten i textfiler. Visualiseringsverktyget Trackplot ger en väldigt rudimentär tvådimensionell bild av simuleringen, men det amerikanska visualiseringsverktyget SIMDIS kan knytas till ODIN för att få en tredimensionell bild av en simuleringsomgång.

Huruvida det finns något stöd för att utveckla och integrera egna modeller i ODIN är något osäkert, främst beroende på att de flesta användarna av ODIN inte utvecklar sina egna modeller utan överlåter detta på QinetiQ. Norska FFI har dock implementerat egna modeller i ODIN, om än med visst besvär. Däremot verkar det inte vara något problem att modifiera befintliga modeller. Detta kräver dock kunskaper i C/C++. Verktyget har ett mycket bra stöd för modellering av undervattensmiljö, yt- och undervattensfarkoster samt undervattenssensorer och signaturer. Fokus ligger i dagsläget på tillämpningar i djupa vatten.

5.5. MÄK och en kortfattad beskrivning av VR-Forces

MÄK är ett litet, privatägt företag, beläget i Cambridge, Massachusetts, USA. Exempel på deras användare och kunder inom den internationella försvarsindustrin är ITT Industries, Boeing, Lockheed Martin, Raytheon, Tenix, Dassault och BAE.

VR-Forces är ett kommersiellt simuleringsramverk som är allmänt känt för sina CGF:er – Computer Generated Forces – d v s ramverket påstås inkludera rätt duktiga aktörer (beteendemässigt). Samtidigt har MÄK lagt ganska mycket kraft på visualisering och att det ska vara lätt att bygga ett scenario. Vi får därför intrycket att fokus ligger på att verktyget ska kunna användas för att snabbt och enkelt implementera ett tänkt scenario och sedan visualisera hur det gick. Typiskt skulle detta kunna användas för att pröva planer och nya koncept utan alltför stora krav på detaljupplösning. Vi har inte gjort några tester och försök med VR-Forces.

I ramverket kan händelser på marken, i luften, på sjön och under vattnet simuleras. Stöd för undervattensplattformar är dock minimalt. Det ingick en mycket enkel ubåtsmodell i VR-Forces. Den är dock inte mer än en förklädd helikoptermodell som inte påverkas av terrängen, vilket bl a leder till att "ubåtar" kan köra rakt igenom undervattensterräng som t ex grynnor. Att det går att använda en exempelmodell för helikopter som ubåt tyder visserligen på att det är ett flexibelt verktyg, men det är också en svaghet att det inte finns en färdig ubåtsmodell.

Ramverket kan både användas för att simulera några få entiteter och större taktiska scenarion med många aktörer. Det finns dessutom stöd för aggregering och disaggregering vilket gör att man kan växla från mer detaljerade studier till att få en övergripande bild av händelseförloppet.

Exekveringarna kan köras i valfri hastighet och det finns möjlighet att köra batch-körningar, med en variation av parametervärden mellan körningarna. För visualiseringen finns det ett starkt stöd, både för illustration av olika enheter och av sensortäckning.

Vid nyutveckling av modeller i ramverket finns det generella grundmodeller att utgå ifrån. Ingen av MÅKs befintliga modeller påverkas dock idag av miljöparametrar. Trots försäkran att detta är lätt att inkludera i en ny modell är det oroväckande. Det faktum att man inte har använt någon modell som påverkas av miljö när simuleringsramverket utvecklades medför att man riskerar att stöta på problem när en sådan modell ska införas. De problem som då kan uppkomma kan visa sig svåra att lösa med nuvarande design av ramverket. För undervattenssimuleringar finns endast visuella plattformar, sensorer och stöd för 3D-visualisering. Funktionaliteten är dock mycket förenklad och det finns ingen undervattensmiljö i ramverket.

5.6. CAE och en kortfattad beskrivning av STRIVE

CAE är ett börsnoterat företag med huvudkontor i Kanada. Företaget bildades 1947 och har numera ca 4800 anställda baserade i 17 olika länder på fem olika kontinenter. Inkomsterna per år ligger på ca en miljard dollar (CAD) varav 90 % kommer ifrån export. Företaget är inriktat både mot militära och civila tillämpningar.

STRIVE är i nuläget resultatet av en pågående investering CAE gör, och bör bedömas främst med avseende på vad STRIVE förväntas vara om ett par år, snarare än den status verktyget har idag. Storleken på den investering CAE gör, och en rad andra faktorer talar för att STRIVE med stor sannolikhet kan bli för den militära simuleringsvärlden en programvara som motsvarar vad Microsoft Office är idag för kontorsapplikationer. Därmed inte sagt att STRIVE skulle vara det självklara valet för infrastruktur/simuleringsprogramvara.

Företaget CAE som står bakom produkten har sedan tidigare haft en tydlig inriktning mot träningsanläggningar, men STRIVE är avsett att bli en flexibel produkt som ska kunna användas i alla olika typer av simuleringar. Ramverket hanterar alla önskvärda domäner, luft, mark, sjö och undervatten. Verktygets modulära uppbyggnad gör dock att man endast behöver införskaffa de delar som man har behov av. Det finns t ex ett stort stöd för undervattenssimuleringar i paketet STRIVE SONAR.

Ramverket hanterar få-mot-få situationer men har även ett utbyggt stöd för taktiska scenarier. Det är oklart om det finns möjligheter att använda aggregering och batch-körningar av scenarier, men det finns stöd för att köra exekveringar i eller snabbare än realtid. Loggningsfunktionen är väl utvecklad och det går t.o.m. att styra vad som ska loggas under pågående simulering. Data från loggningen kan även plottas under simuleringens gång. Visualisering finns för både 2D och 3D. Ramverket har även ett omfattande stöd för visualisering av resultatet från akustiska undervattenssensorer.

Designen på ramverket är gjord för att underlätta integreringen av externa modeller. Det förefaller vara en genomtänkt struktur och det medföljer även riktlinjer för hur nyutveckling ska ske. Stödet för undervattenssimuleringar är mycket bra. Ramverket hanterar undervattensmiljö, yt- och undervattensfarkoster. Det finns även en omfattande modellering av akustiska sensorer och signaturer upplagd i en genomtänkt struktur.

5.7. Detaljbeskrivningar

För att ge en mer utförlig bild av respektive verktyg följer nedan mer detaljerade beskrivningar. Dessa beskrivningar är främst avsedda för personer som ingående vill studera vad simuleringsramverken innehåller inför en eventuell anskaffning eller för personer som ska jobba med verktygen. Beskrivningarna innehåller därför flera simulerings-specifika fackuttryck som inte ges någon närmare förklaring.

5.7.1. MOSES

Inledning

Den utvärderingslicens som FOI har erhållit består av tre delar: 1) MOSES grundplattform som består av simuleringskärna, basobjekt, domänobjekt, observatörsobjekt samt applikationsspecifika objekt, 2) visualiseringsverktyget *ObserverMedia*, samt 3) redigeringsverktyget *ModEdit*.

MOSES grundplattform är kompilerad till en särskild demoapplikation som inkluderar simuleringskärnan och de olika objekten. Vid förändringar eller tillägg av källkoden måste MOSES kompileras om för att förändringarna ska ingå. *ModEdit* används främst till att konstruera scenarier, men även för att starta simuleringen. Visualiseringsverktyget används för att ge olika vyer av det scenario som exekveras. Verktöget är länkat till grundplattformen med en CORBA-anslutning. Enligt dokumentationen är det möjligt att köra de olika delarna, (grundplattform, scenarioverktyg och visualiseringsverktyg) på olika datorer eller processorer.

En fullständig version av MOSES innehåller, förutom de ovan nämnda komponenterna, även verktyget *Execution Control*, som används för att styra exekvering av simuleringen och för att göra batch-körningar, och en uppspelningsfunktion som gör det möjligt att titta på resultaten av en simulering utan att vara tvungen att exekvera scenariot igen.

Scenariohantering

För att skapa och redigera scenarier används verktyget *ModEdit*. Verktöget har ett Windows-liknande grafiskt användargränssnitt. Verktöget har tre olika vyer/flikar: en flik där man skapar och redigerar sitt scenario, en flik där det går att se strukturen på alla objekt som ingår i scenariot och eventuella kopplingar mellan olika objekt samt en flik där man får information om hur exekveringen av ett scenario går. Det kan också vara av intresse att notera att innan man börjar skapa ett scenario måste källkoden parsas, så att *ModEdit* får information om vilka komponenter som finns tillgängliga. Detta utförs enkelt via menyn *Options* i *ModEdit*.

MOSES har tre grundläggande objekttyper och dessa är plattformar, delsystem och omgivningen. Plattformobjekt har i sig själva ingen funktionalitet utan de tjänar endast som "lagringsplats" för olika delsystemobjekt. Delsystemobjekten som en plattform innehåller är det som definierar denna plattform uppträdande och funktionalitet. Delsystem beskriver så väl funktionalitet och beteende som egenskaper. Ett fartyg kan t ex innehålla delsystemen ubåtsjakttaktik, förflyttning och utstrålat buller. Omgivningsobjektet hanterar utbytet av information mellan olika plattformar.

Ett scenario skapas genom att lägga till önskat antal plattformar och sedan fylla på dessa med delsystem som behövs. Beroende på uppgift kan antalet delsystem på en given plattform variera kraftigt – i de mest enkla fall behövs i stort sett bara förflyttning och eventuellt ett enkelt beteende. Varje objekt innehåller ett antal parametrar som kan redigeras, t ex kurs och fart hos det delsystem som sköter förflyttning. Det är genom objekten i scenariot som behövda parametrar matas in i simuleringen. Att ändra parametrar är enkelt, men det är inte helt lätt att veta vad olika parametrar har för funktion utan att läsa källkoden (Förhoppningsvis skulle denna information bli tillgänglig om MOSES anskaffades till FOI).

Exekvering

Innan exekvering av ett scenario måste *ObserverMedia* startas. Det kan vara idé att även ställa in hur länge simuleringen ska exekvera samt om simuleringen ska ske i realtid, snabbare än realtid eller långsammare. Därefter startas simuleringen via meny eller snabbkommando. Vid start växlar *ModEdit* över från redigeringsfliken till exekveringsfliken, där start och stopp av simuleringen visas. Eventuella exekveringsfel syns även detta i denna vy. Resultat från simuleringen visas med *ObserverMedia*.

För att skapa utdata från en exekvering används något som kallas observatörsobjekt, vilket är

en del av grundplattformen, och av ett observatörsmedium, vilket är en del av visualiseringsverktyget *ObserverMedia*. Observatörsobjekt läggs in i scenariot och parametersätts via *ModEdit* på samma sätt som de vanliga simuleringsobjekten. Ett observatörsobjekt bevakar uppträdandet hos ett eller flera simuleringsobjekt och loggar parametrar från dem. Observatörsobjektet skickar insamlad data till observatörsmediumet, vilket sedan visualiserar informationen i *ObserverMedia* i form av till exempel en tabell eller en XY-plot. Ett observatörsobjekt kan inte på något sätt påverka ett simuleringsobjekt utan endast observera detta. Under en pågående exekvering går det att klicka sig runt i *ObserverMedia* för att t ex följa ett visst fartyg.

Visualisering

Som nämnts tidigare sker all visualisering i applikationen *ObserverMedia*. Hur information från simuleringen presenteras i *ObserverMedia* beror på vilken typ av observatörsobjekt som har knutits till scenariot när detta skapades eller vid någon redigering. Det går att ha flera observatörsobjekt i samma scenario, vilka presenteras i sin egen vy i *ObserverMedia*. I den demoversion som har testats går det att presentera information i en tvådimensionell XY-plot eller i tabellform. Det ska även gå att presentera data som grafer, men det övningsexempel som medföljer demoversionen går inte in på detta. I ett bidrag till konferensen UDT Europe 2004 hänvisas till verktyget SIMDIS, som har utvecklats av US Naval Research Laboratory i Washington, D C, för att göra 3D-visualiseringar i MOSES.

Vid användning av 2D-plotten går det att se fartyg på ytan samtidigt som farkoster under vattenytan visas. Det går dock inte att se vilket djup en undervattensfarkost befinner sig på. Vidare gick övningsexemplet inte in på om det går att visa sensortäckning grafiskt även om skärmdumpar från TNO-FEL verkar indikera att denna funktionalitet finns.

I jämförelse med en del kommersiell programvara är visualiseringsmöjligheterna i MOSES tämligen enkla. Beroende på syftet med en simulering kan detta vara en fördel eller en nackdel.

Analys

Av den dokumentation och det övningsexempel som medföljde demoversionen av MOSES går det inte att utläsa något definitivt om analysmöjligheter. Samtidigt vore det anmärkningsvärt om det inte finns möjlighet att logga data på något annat sätt än direkt i *ObserverMedia*. Ett av huvudsyftena med MOSES verkar vara just analys, så i en fullversion av programvaran borde det finns analysmöjligheter. I övningsexemplet görs vid ett par tillfällen hänvisningar till funktionalitet som inte finns med i demoversionen; det är fullt möjligt att någon form av analysverktyg ingår i detta.

Modifiering av befintliga modeller

Det är svårt att säga exakt vad en fullversion av MOSES skulle kunna innehålla i form av modellbibliotek och dylikt. Demoversionen innehåller endast en grundläggande funktionalitet och det påpekas i övningsexemplet att vissa saker inte ingår eller utnyttjas i demoversionen. Efter att ha gjort övningsexemplet känns det dock som att det inte vore så svårt att modifiera de befintliga modellerna.

Nyutveckling

Demoversionen innehåller den grundläggande funktionaliteten och infrastrukturen för att skapa egna modeller. Vidare har MOSES ett verktyg för att generera grundläggande kod för nya moduler som verkar tämligen bra. Det medföljande övningsexemplet går igenom hur man skapar en mycket enkel förflyttningsmodell för ytfartyg samt en inte fullt så enkel målföljningsmodell för torpeder. Av dessa att döma är det troligen inte alltför svårt att skapa nya modeller så länge dessa passar ihop med den infrastruktur och de designtankar som finns i MOSES. Med andra ord torde det gå ganska smärtfritt att utveckla en egen modell för en hydroakustisk sensor eller en

egen kognitiv modell för ubåtsjakt. Däremot är det osäkert om det går att införa t ex en elektromagnetisk sensor utan att skriva om eller modifiera stora delar av källkoden.

SNE (*Synthetic Natural Environment*) och miljö för undervatten

Utifrån demoversionen är det svårt att avgöra exakt hur MOSES hanterar undervattensmiljön. Med ledning av de moduler som medföljer verkar det dock som om programmet tar hänsyn till bl a ljudhastighetsprofil, bottenbeskaffenhet, regn och vind vid havsytan samt skiktningar i vattenvolymen. Det finns dock inget som ger någon fingervisning om bottenpografi finns med.

Stöd för undervattenplattformar

I det övningsexempel som medföljer demoversionen av MOSES hanteras endast torpeder och övervattensfartyg. Vidare sägs i presentationen av projektet TDS Testbed, som TNO-FEL driver, att man vill simulera ytfartyg och torpeder men även olika typer av motmedel och skenmål för undervattensdomänen samt olika typer av skrovmonterade och släpade hydroakustiska sensorer med hjälp av bl a MOSES.

Samtidigt ger övningsexemplet en bild av att det inte skulle vara så svårt att modifiera modeller och beteende för att simulera ubåtar och andra undervattensfarkoster utöver torpeder. Det går t ex alldeles utmärkt att låta ytfartyg använda målsökningsbeteendet från en torped för att jaga en bullrande torped.

Sensorer och signaturer

MOSES verkar ha ett bra stöd för att hantera hydroakustiska sensorer. Hydroakustisk målsignatur hanteras i det passiva fallet som en egenskap hos farkosten som finns i objektet *RadiatedNoise*. I demoversionen av MOSES läser *RadiatedNoise* in en textfil där bullernivån i decibel finns angiven för att antal hastigheter. Det verkar även gå att ange eventuellt kavitationsbuller. Vid användning av aktiv sonar används modulen *Reflector* som beräknar en plattformens aktuella målstyrka beroende på plattformens geometri och aspekt i förhållande till sonaren.

I demoversionen finns inget som tyder på att det skulle finnas stöd för elektromagnetiska sensorer och signaturer i MOSES. Däremot ska detta, enligt muntliga uppgifter från personal vid TNO-FEL, finnas i fullversionen av MOSES. Dessutom används MOSES för att simulera minjakt och minröjning vilket styrker att det finns stöd för att hantera elektromagnetiska signaturer och sensorer.

Tekniska aspekter

Enligt dokumentationen är systemkravet för MOSES PC med Intel-processor som använder operativsystemet Windows 2000 Professional SP4. För att kunna exekvera och göra enklare modifikationer i klass med de som finns i övningsexemplet krävs vidare att Microsoft Visual C++ .Net och Java Runtime Environment, version 1.4.2_04 är installerade på de datorer som ska användas.

För att kunna utveckla egna modeller till MOSES krävs en del annan mjukvara utöver det som nämns ovan. En lista över denna finns i dokumentationen till övningsexemplet och verkar till största delen bestå av öppen källkod. Det verkar även som att allt som behövs utom Microsoft Visual Studio .Net medföljer en fullversion av MOSES.

Aggregera/disaggregera

Det finns inget i demoversionen som tyder på att det skulle gå att aggregera och disaggregera förbandsenheter i MOSES.

Möjlighet till distribuerade körningar

Det är möjligt att genomföra distribuerade körningar på olika sätt. Först och främst kan MOSES

tre grundkomponenter (MOSES grundplattform, *ModEdit* och *ObserverMedia*) körs på olika datorer eller på olika processorer. Det är även möjligt att koppla ihop MOSES med externa modeller. Detta är görs antingen genom att använda CORBA eller via kommunikationsprotokoll som TCP/IP. TNO-FEL har kopplat ihop MOSES med modeller skrivna i FORTRAN, bl a signalpropageringsmodellen ALMOST, och Matlab.

Det framgår dock inte säkert om MOSES har någon form av HLA-koppling. TNO-FEL säger att det är möjligt att koppla ihop MOSES och andra modeller med hjälp av HLA, men det är osäkert om detta har gjorts eller är förberett.

Joint möjligheter

MOSES verkar inte ha några möjligheter att hantera annat än ytfartyg och undervattensfarkoster. Det är nog därför svårt att skapa scenarier för gemensamma operationer utan stora förändringar av mjukvaran.

Interaktion mellan entiteter

Interaktion mellan olika plattformar i ett scenario sker via omgivningsobjektet (*Environment*). I MOSES används händelser för att hantera kommunikation/interaktion mellan olika plattformar eller mellan en plattform och omgivningen. Händelser används även när kommunikationen är fördröjd. I huvudsak verkar händelser vara begränsade till att omfatta signalpropagering samt kollisioner och explosioner.

Inom en och samma plattform kommunicerar de olika delsystemen med meddelanden. Meddelanden är antingen kommandon (*commands*) eller upplysningar (*notifications*).

Användning för andra områden/syften

I huvudsak verkar MOSES användas för analytiska simuleringar. Om det används för andra användningsområden framgår inte av den tillgängliga dokumentationen.

Kurser och *User groups*

Det finns ingen information huruvida TNO-FEL arrangerar kurser eller har användargrupper.

Andra användare (vars erfarenheter vi kan ha nytta av)

TNO-FEL har enligt uppgift ett samarbete med QinetiQ och innan dess med DERA, där man bland annat har delat modeller med varandra, bl a lär QinetiQ ha prövat olika plattformsmodeller gjorda för MOSES i sin simuleringsprogramvara ODIN.

Vidare undersökningar/kvarvarande frågor

Det finns flera olika frågor att gå vidare med. De kanske två viktigaste frågorna är om det är möjligt att använda MOSES för att simulera gemensamma operationer samt att få mer kunskap om MOSES potential utöver det som syns i demoversionen. Därför bör den inledande frågan vara att reda ut huruvida MOSES går att koppla ihop med andra program eller ramverk i syfte att simulera gemensamma operationer. Därefter vore det av intresse att få en bättre bild av vad MOSES kan göra, t ex genom en ordentlig demonstration av personal från TNO-FEL, som gärna knyter an till något projekt där de har använt MOSES. Det vore vid ett sådant tillfälle bra att få möjlighet att diskutera designen hos MOSES med någon som har varit med och tagit fram programmet.

Några andra intressanta frågor som bör följas upp efter detta är i vilken mån TNO-FEL skulle kunna bistå med kurser, användargrupper, etc om FOI valde att anskaffa MOSES. Även frågan om licenskostnader vore av intresse att få utredd.

5.7.2. FLAMES

Inledning

FOI har sedan 1997 använt FLAMES i olika sammanhang, och kunskapen om verktyget i allmänhet är därför väl underbyggd. FLAMES är också det verktyg som använts för pilotstudien.

FLAMES har sitt ursprung i simulering av luftstrid, utan att för den skull vara begränsad till det området. På senare år har en förbättrad vektorbaserad terrängmodell tagits fram, som gör det möjligt att använda FLAMES också för markscenarier.

FLAMES är designat för att vara flexibelt, och kunna anpassas för olika ändamål. Samtal har förts med Ternion om möjligheterna att bygga in stöd för undervatten i FLAMES.

Scenariohantering

Scenarion i FLAMES skapas och redigeras med det grafiska verktyget Forge. Verktyget är byggt kring en kartvy, som visualiserar det aktuella geografiska området, och de olika aktörer som lagts till i scenariot. För att ge de olika aktörerna attribut och egenskaper finns ett antal formulär.

Varje aktör konfigureras med ett scriptspråk, som är uppbyggt kring en väl definierad syntax, men förhållandevis enkelt att använda även för en icke-programmerare. För att utnyttja språket fullt ut, krävs dock att användaren har en djupare förståelse för hur FLAMES och de ingående modellerna fungerar.

Data för såväl scenarios som modeller lagras i en databas. När scenariot sparas uppdateras både filen och innehållet i databasen. Flera personer kan därför samtidigt arbeta med ett och samma scenario. Redigeringsverktyget innehåller också en del verktyg för att kontrollera att konfigurationen av aktörer är konsistent, samt för att grafiskt redigera brytpunkter, mm.

Undervattenstillämpningar ställer inga nya omvälvande krav på redigeringsverktyget, även om det visserligen vore praktiskt att kunna använda ordet DEPTH i scriptspråket.

Exekvering

Exekvering av ett scenario görs med programmet Fire, genom att scenariofilen och tillhörande data läses in. Scenariot kan exekveras antingen i mjuk realtid, eller så fort som möjligt. Scenariot kan också exekveras interaktivt, då Fire interagerar med andra simuleringar och/eller mänskliga aktörer. Det finns också stöd för Monte-Carlo simuleringar, där samma scenario exekveras flera gånger med olika slumpvalsfrö, för att göra statistiska analyser. Simuleringarna är repeterbara, så att en körning av ett scenario med ett visst slumpvalsfrö ger samma resultat.

Normalt så exekveras scenariot på en instans av Fire. Möjlighet att lägga ut exekveringen av enskilda aktörer på annan instans för t ex lastbalansering är liten, och av oss ännu oprövad. Det finns en tilläggsoption som möjliggör effektivare utnyttjande av multiprocessormaskiner. HLA-kopplingen kan också nyttjas för distribuerade körningar, men detta kräver antagligen ett visst mått av förberedelser redan vid designfasen.

Beteendet för en aktör kan påverkas under exekveringen, dock inte genom att användaren redigerar scenariot. Med de kommunikationsmöjligheter som finns till FLAMES (DIS, HLA, klient/server) går det att påverka det mesta i ett scenario, svårighetsgraden varierar dock från fall till fall.

Det finns även en option för att återstarta ett scenario från i förväg definierade *checkpoints* vid större körningar som sträcker sig över längre tid.

Visualisering

Visualiseringsprogrammet som följer med FLAMES benämns Flash, och kan visualisera ett scenario i en 2D vy, samt i 3D. Under det senaste året har en ny option tagits fram som gör det möjligt att använda *OpenFlight* data för renderingen. I såväl 2D som 3D vyn används ikoner för att visualisera plattformar.

Flash kan fungera i två olika moder; antingen som ett uppspelningsprogram där de data som ska åskådliggöras läses från fil, eller som en monitor, där simuleringsdata sänds över nätverket direkt från exekveringen av scenariot.

För undervattenstillämpningar har det visat sig att 2D vyn fungerar väl för att visualisera plattformar som befinner sig under ytan. Det finns också möjlighet att bygga egna vyer, exempelvis där de olika aktörerna och omgivningen kan ses från sidan.

Analys

Data som genereras vid exekveringen kan lagras med sk *recorders*, en speciell klass av modell som skriver till resultatfiler. *Recorders* kan modifieras efter behov för att utöka de data eller händelser som ska lagras. Ur dessa filer kan data extraheras med verktyget Flare, som stödjer en delmängd av SQL.

De data som loggas kan sedan användas i andra verktyg för analys och visualisering.

Modifiering av befintliga modeller

Med FLAMES följer en uppsättning exempelmodeller, med tillhörande källkod och dokumentation. Syftet med dessa är dels att erbjuda en möjlighet att snabbt kunna komma igång med att använda FLAMES utan att behöva börja med att utveckla modeller, men kanske framför allt för att illustrera hur egna modeller kan utformas. Exempelmodellerna är på en förhållandevis enkel nivå, rörelseekvationer med tre frihetsgrader, radarmodeller baseras på radarekvationen, etc. Exempelmodeller finns för de modellkategorier FLAMES har färdigt stöd för. Saknas en modell eller en kategori går det att utöka samlingen. Det är också förhållandevis enkelt att modifiera de medföljande exempelmodellerna så att de passar ens egna syften.

Ternion tillhandahåller också väl tillfredställande dokumentation om hur man går tillväga för att utveckla egna modeller, även utvecklarkurser erbjuds. Vidare kan Ternion åta sig modellutvecklingsuppdrag.

En modell i FLAMES kan tillhöra olika kategorier: utrustning, kognitiv modell, omvärldsmo- dell, tjänst eller attribut.

Utrustningsmodeller kan med fördel göras generiska, då det finns mekanismer för att parametersätta dessa till att efterlikna verkliga system. Beteendemodeller utformas vanligen som en samling processmetoder, som representerar olika tillstånd av en aktör, tex. en pilot som kan vara under attack, eller flyga i route. En modell av en tjänst kan grovt beskrivas som en process som finns tillgänglig under simuleringen, t ex. väderservice eller dylikt.

Användandet av egenutvecklade modeller och parametersättning av dessa kräver ingen programmeringserfarenhet, dock erfordras naturligtvis en god dokumentation.

Nyutveckling

Utveckling av egna modeller i FLAMES görs i C eller C++. Lyckade försök att använda ADA har också gjorts av Ternion. Utöver god kännedom om programmeringsspråket krävs också kännedom om FLAMES och dess API.

Det är naturligtvis svårt att ange ett mått på hur lång inlärnings tiden är, men givet en väl formulerad och avgränsad uppgift har vi sett att examensarbetare efter en månad kan vara inne i implementeringsarbetet.

Nya modeller kan skapas antingen genom att utgå från en befintlig modell, eller genom att använda den kodgenerator som följer med.

Det försök som gjordes under 2003 att implementera en hydroakustisk sensor i FLAMES har visat att det är fullt möjligt att genomföra, men det kan vara värt att påpeka att resultatet till stor del är avhängigt en väl genomtänkt design som utnyttjar ramverket.

SNE och miljö för undervatten

Omgivningsmodellen i FLAMES kan sägas bestå av atmosfärsegenskaper, en topografisk modell och s k. *features*³. Den topografiska modellen kan vara antingen en ren höjddatabas (DTED), eller i vektorformat. I det senare fallet kan även olika attribut för landområden anges, och infrastruktur som vägnät kan då ingå i den topografiska modellen. Det finns inget färdigt stöd för undervattensmiljö i FLAMES, då de topografiska modellerna antingen beskriver endast vattenytan eller markförhållanden.

Den genomförda pilotstudien har visat att egenskaper så som ljudhastighetsprofil är tämligen enkla att lägga till på egen hand. I det försök som genomfördes under 2003 implementerades en tjänst som tillhandahöll omgivningsdata som salthalt och ljudhastighetsprofil.

Möjligheten att importera batymetridata med verktyget TerraVista är inte fullt utforskade, och kvarstår som ett intressant försök.

Stöd för undervattensplattformar

Även om det inte finns något färdigt stöd för undervattensmiljö, så finns det inga begränsningar i övrigt för hur en plattform i FLAMES positioneras, så länge dess position kan beskrivas med antingen ECR⁴ koordinater eller Latitud, Longitud och höjd⁵. Det går utmärkt att positionera en plattform under marknivå, eller under vattenytan.

Plattformars position definieras av en punkt, och orientering. De saknar utsträckning i rummet, så för visualisering används ikoner för utritning i två och tre dimensioner. Konsekvenser av detta för möjligheten att göra strålgångsberäkningar har inte utforskats närmare.

Plattformsmodeller kan ges olika attribut och egenskaper, som signatur och rörelsemodell. Mekanismer för att parametersätta generiska modeller till att efterlikna verkliga system finns.

Sensorer och signaturer

Bland exempelmodellerna till FLAMES finns radarmodeller av olika slag, samt en enkel siktlinjemodell. Det finns inget färdigt stöd för hydroakustiska och elektromagnetiska sensorer.

En signatur i FLAMES är bland exempelmodellerna implementerad som radarmålarea. Försöket under 2003 visade att det är fullt möjligt att implementera och använda en enklare akustisk signatur, i det fallet bestående av en tabell med värdepar för frekvens och signalstyrka. Även mer komplexa, och rikttningsberoende signaturer kan implementeras.

Tekniska aspekter

FLAMES finns för plattformarna Windows, Solaris och SGI. FLAMES använder ett eget *widgetsystem* (Jewel), vilket gör att grafiska applikationer ser likadana ut och fungerar under såväl Windows som Unix/X. Grafiken som används för visualisering av scenarios är baserad på OpenGL.

Aggregera/disaggregera

FLAMES har inget inbyggt stöd för aggregering/disaggregering. Däremot finns färdigt stöd för en *chain-of-command*, där varje aktör kan ha en överordnad befälhavare. I de grafiska programmen Forge och Flash kan användaren välja att visa/gömma underordnade enheter. Detta koncept påverkar endast visualiseringen, inte simuleringen.

³ Objekt som både kan vara en del av topografin, och påverka såväl som påverkas av händelseförloppet i scenariot. Broar och byggnader är exempel på detta.

⁴ Kartesiskt koordinatsystem med origo i jordens medelpunkt och z-axeln genom nordpolen.

⁵ En höjd som mycket väl kan vara negativ

Möjlighet till distribuerade körningar – HLA/RTI

FLAMES är i grunden ett monolitiskt simuleringsramverk, men det finns ett antal tilläggsalternativ som möjliggör distribuerade körningar. Såväl DIS som HLA stöds, och med de senaste versionerna börjar stödet för HLA bli riktigt användbart. Vidare finns en möjlighet att använda klient/serverlösningar, där fristående externa program genom att använda FLAMES klient-API kan interagera med en simulering. Denna lösning baseras inte på någon etablerad standard, men kan i många fall vara enklare att implementera än att genomföra samma sak genom att implementera en HLA federation.

Joint möjligheter

FLAMES är från början framtaget av personer med en bakgrund i US Airforce, något som naturligtvis påverkat utformningen, och FLAMES har sedan många år varit ett kraftfullt verktyg för MoS av luftstrid. Sjöstrid har mycket gemensamt med luftstrid, och i och med de nya alternativen för att importera omvärldsdata på vektorformat, finns det goda förutsättningar för joint-scenarios.

För undervattenstillämpningar finns dock inget färdigt stöd.

Interaktion mellan entiteter

Aktörer i ett scenario kan skicka meddelanden för att kommunicera med andra aktörer i ett scenario. Olika delmodeller (för en och samma aktör) kan utbyta data sinsemellan med en generisk datastruktur, s k. *ModelArgumentList*.

Användning för andra områden/syften

FLAMES har testats för analytiska studier, så som taktikutveckling. Det finns möjlighet att använda FLAMES även för träningsändamål och mer tekniska studier, men det har inte prövats av oss. Dock finns lyckade exempel på detta internationellt.

Den stora styrkan med FLAMES har i flera år varit den bakomliggande funktionaliteten och arkitekturen, snarare än möjligheterna till visualisering. På senare år har Ternion lagt ner mer energi på att förbättra användarvänligheten, och de grafiska aspekterna.

Kurser och User groups

Det finns ett antal användare av FLAMES, och sedan 2003 även en användargrupp. Vidare administrerar Ternion en webbplats⁶ för utbyte av FLAMES modeller. Nyttan med en användargrupp är svår att avgöra, men för en forskningsorganisation är detta ett komplement till konferenser och andra existerande forum.

Ternion tillhandahåller kurser för användare och utvecklare, och kan även själva åta sig modellutvecklingsuppdrag för kunds räkning.

Andra användare (vars erfarenheter vi kan ha nytta av)

Ternion har ett antal företag och organisationer i USA, Asien och Europa som kunder. Bland andra nämns US Army, US AirForce, US DoD, US Navy, US MarineCorps. Även NATO C3Agency, CAOC (NATO Combined Air Operations Centers), NPC (NATO Programming Centre). Ytterligare kunder är listade på <http://www.ternion.com/customers/customers.htm>

Bland de kundprojekt som särskilt omnämns på hemsidan finns BAE Systems, NC3A, AFRL (Air Force Research Lab) samt FOI.

⁶ FLAMES Resource Catalog, <http://flames.ternion.com/frc>

5.7.3. ODIN

Inledning

QinetiQ har liksom FOI en lång historia med utveckling av en rad specialiserade modeller för olika ändamål och syften. Motivet till utvecklingen av ODIN var att kunna knyta samman dessa, inte minst för att enklare kunna återanvända modeller.

ODIN är en händelsestyrd simuleringsmodell för undervattensdomänen. Grundstrukturen innehåller ett antal grundläggande komponenter som t ex simuleringskärna. Till detta knyts olika funktionella komponenter som omgivningsmodell, signalpropagering, etc. ODIN har en objekt-orienterad design och egenskaper ärvs av underklasser.

ODIN gav ett bra första intryck. En klar fördel är att det skapades som en fortsättning på simuleringsverktyget THOR, vilket säkerligen innebär att utvecklarna har dragit många slutsatser från THOR angående vad som fungerar och inte fungerar. ODIN var från början inte ämnad som en produkt, men det finns idag flera användare av ODIN som ger kontinuerlig feedback till utvecklarna. Förutom engelska organisationer finns även användare från USA, Norge och Canada. En ny förbättrad utgåva av ODIN släpps varje år från utvecklargruppen, men det känns som om det redan nu är ganska genomarbetat.

Den nuvarande policyn hos QinetiQ är att ODIN visserligen behöver ha en tillräckligt stor bredd för att kunna tillmötesgå behov från olika kunder, men man försöker uppnå detta genom att just fokusera på dessa specifika behov snarare än att implementera funktionalitet av det mer allomfattande slaget. Vid en anskaffning av ODIN ingår programvarorna HOMER och TRACKPLOT, samt ett antal exempelmodeller. Vidare ingår två kurstillfällen. Det första för att ge en allmän bild av ODIN och för att bekanta sig med verktyget. Det andra tillfället är ämnat att visa på hur kunden kan tillämpa ODIN på sina specifika uppgifter.

Scenariohantering

Ett scenario i ODIN består av ett antal entiteter, som var och en kan byggas upp av olika underklasser/attribut. Till en av dessa, *tactics*, hör ett scriptspråk för att kontrollera beteendet.

Scriptspråket, HLL (*High Level Language*) är unikt för ODIN, och påminner litet om någon sorts "pseudo-Fortran". Språket innehåller styrsatser som *if*, *then*, *while*, etc, och är betydligt mer kraftfullt än en tillståndsmaskin. Dock kan det bli något krångligt att implementera mer raffinerade beteenden. Från scriptet går det (enl. uppgift) att anropa klassmetoder från den underliggande C++ modellen. Valet av om funktionaliteten ska läggas i script eller i C++ koden var dock inte självklart. Funktioner av mer statisk karaktär kunde med fördel implementeras i C++, medan det på experimentstadiet kunde finnas en fördel att hålla det på scriptnivå. Det framkom att arbetet med beteendemodeller var en av de mer resurskrävande bitarna, jämfört med utrustningsmodeller.

De från QinetiQ som deltog i demonstrationen tyckte att HLL var ett smidigt och lättanvänt språk, men om detta stämmer återstår att se. Från tidigare kontakter med norska FFI vet vi att norrmännen inte tyckte att det var lättanvänt.

Exekvering

ODIN gick snabbt att köra, oftast mycket snabbare än realtid. Det finns stöd för att göra Monte Carlo-simuleringar och batch-körningar. Det fanns dock inget direkt stöd för variabler på scenarionivå, men diskussionen gav att det gick att lösa genom att köra från någon scriptfil där nödvändiga variabler kan definieras/manipuleras.

Visualisering

ODIN har ett enkelt tvådimensionellt visualiseringsverktyg som kallas TRACKPLOT. Det kan plotta rörelsebanor, pingar, låsning, mm. Olika aktörer i ett scenario representeras av en bokstavs- och sifferkombination av typen E1, E2, etc. Det verkade som att man i huvudsak är intresserad av

de olika dataloggar som skapas och inte så mycket av hur det ser ut.

Det går dock att spela upp en sparad simulering i det amerikanska 3D-verktyget SIMDIS. SIMDIS verkade vara ett ganska lättanvänt verktyg som gav snygga visualiseringar. Tyvärr kan SIMDIS i dagsläget endast visualisera amerikanska fartyg och vapensystem. Verktyget är utvecklat av NRL (*Navy Research Laboratory*), USA, och för att kunna använda det krävs kontakter med NRL.

Koordinatsystemet i ODIN är kartesiskt med XYZ-axlar och origo som referenspunkt i scenariot. Latitud/Longitud används inte, då det inte funnits behov av annat än en "flat-earth-approximation". Odin har dock kopplats till en OA-modell på QinetiQ, som använder sig av ett koordinatsystem med latitud/longitud.

Analys

ODIN genererar data som textfiler, men saknar mer raffinerade postprocessningsverktyg. För efteranalys är man hänvisad till externa programvaror, vilket faktiskt inte behöver vara en nackdel. Excel är ett kraftfullt verktyg för att generera grafer etc, och det är en process som enkelt kan automatiseras med t ex några rader i Perl. Det postprocessningsverktyg som FOI tagit fram till FLAMES skulle förmodligen också gå att använda här.

Modifiering av befintliga modeller

Diskussionen kring detta gav att det i de flesta fall inte brukar vara några problem, svårigheterna uppkommer när man försöker införliva helt nya koncept. (Vilket stämmer väl överens med våra egna erfarenheter från FLAMES och t ex robotar som samverkar med varandra.) Det avtal om *annual support* som bör/brukar tecknas täcker detta så länge som det rör sig om att använda befintlig funktionalitet, t ex svara på frågor. Om det krävs utvecklingsarbete blir det förstås en fråga om kostnader.

Nyutveckling

Möjligheten att integrera egna, befintliga modeller var nog den aspekt som avhandlades minst under mötet. Klart är dock att det går att göra, bl a Norge har erfarenhet av detta. QinetiQ har även använt ODIN för tillämpningar utanför UV-domänen. Lyckade försök att koppla in hårdvara i fartyg till simuleringen har också gjorts, även om det betonades att detta ligger långt utanför det tänkta användningsområdet.

SNE och miljö för undervatten

Metodiken för att hantera flera ljudkällor och sensorer fick också en kort förklaring. Hanteringen av signaler sköttes av ramverket, ingen direkt kommunikation mellan de olika entiteterna behövs. Om en aktör var utrustad med en sensor så skickades information, t ex den för tillfället aktuella akustiska signaturen till sensorn för bearbetning. Reflektion mot bottenlager och yta beräknas med algoritmer för strålgångsberäkning. Lösningen föreföll skalbar, i bemärkelsen att det går att använda olika grader av upplösning.

Normalt sett arbetar man med en planskiktad, 3-lagers modell för vattenvolymen där man använder tabellerade data. Det går att öka detta till n antal lager, men på bekostnad av prestanda. För tillfället finns inget stöd för bottenpografi, men det står på QinetiQs önskelista och kan komma att implementeras under nästa år.

Stöd för undervattenplattformar

ODIN är framtaget främst för att simulera undervattenskrigföring och har ett visst arv från torpedsimuleringsmodellen THOR. Vid demonstrationen av ODIN förevisades scenarier med ubåtar, ytfartyg, torpeder och olika typer av motmedel. Därför kan det nog sägas att stödet för undervattensplattformar är bra.

Sensorer och signaturer

Verktyget är utvecklat huvudsakligen för scenarion med högfrekvent akustik på djupa vatten. Det fanns inget stöd för lågfrekventa akustiska signaler, och endast ett litet stöd för elektromagnetiska signaler. Det senare utvecklades för ca fem år sedan och de som arbetade med området har sedan dess lämnat QinetiQ. Intresset för elektromagnetiska signaturer är enligt utvecklarna på QinetiQ lågt. Implementeringen av den högfrekventa akustiken verkade dock mycket genomtänkt. T ex delades en utsänd signal upp i ett antal olika signaler, var och en representerande en stråle från emitterande objekt till mottagande objekt, vilket leder till goda möjligheter att göra noggranna beräkningar.

Alla entiteter kunde tilldelas ett godtyckligt antal reflektionspunkter och signaturerna för dessa punkter var beroende på riktning på infallande signal. Reflektioner av signalerna beror endast på dessa punkter och inte på formen på entiteten (som bara användes för kollisionsdetektering). Reflekterade signaler behandlades sedan som vanliga utsända signaler vilket underlättar simuleringar av multistatiska fall. Vidare sköttes mycket av signalpropageringen i en omvärldsmodul i simulatören som filtrerade bort onödiga signaler. Designen är mycket lik det upplägg som finns i UV-strid. Dessutom användes inte bara sonarekvationen, det fanns även stöd för tidsserier.

Simulering av vakor som påverkade de akustiska signalerna var implementerat. Det borde även kunna utökas till att innefatta hydrodynamiska effekter.

Sammanfattningsvis fanns det i stort sett bara stöd för högfrekventa akustiska signaler, men i och med att den implementeringen verkade så genomtänkt, med användande av miljö, avancerade signaturer, etc finns det säkerligen goda möjligheter att utöka simulatören med fler typer av signaler. Det enda stora orosmomentet var att det inte fanns stöd för varierande bottenpografi, endast planskiktade modeller användes. Detta verkade inte helt lätt att införa, men det är ett måste inför framtida användning.

Tekniska aspekter

ODIN fungerar på olika plattformar, Windows, Solaris och Linux var de som nämndes under dagen. ODIN är implementerat i C++, och enligt har enligt uppgift en objektorienterad design. Det är svårt att ge en bedömning på hur pass stort ODIN är, men programvaran (troligtvis inklusive tillhörande modeller) omfattar mer än 300 klasser.

De programvaror som demonstrerades under dagen var:

- ODIN - Själva ramverket, men som det också föreföll verkar även vara det program som används för att exekvera ett scenario.
- HOMER - Ett Java-baserat GUI (*graphical user-interface*) för att konfigurera scenarios. De i ett scenario ingående entiteterna presenteras som en trädstruktur på liknande sätt som i MOSES. HOMER kan användas för att redigera parametrar, etc. Till saken hör dock att ingen av deltagarna använde sig av HOMER speciellt ofta eftersom de använde sig av textfilerna då de behövde ändra något.
- TRACKPLOT - Ett förhållandevis enkelt 2D visualiseringsverktyg, som kan plotta rörelsebanor, pingar, låsning mm. Inte helt olikt den 2D-plot som finns i MOSES. Går även att ändra färgkoder mm, och förmodligen mycket användbart.
- SIMDIS - NRLs 3D visualiseringsprogramvara. ODIN kan generera data som kan läsas in i SIMDIS. Dock fanns i den version vi fick se endast ikoner som motsvarar amerikanska plattformar.

Det API som ODIN tillhandahåller är också C++. Språk som C och Fortran är ofta direkt kompatibla med varandra, men för C++ gäller att de olika API-funktionerna förses med tillägg i namngivningen, sk *name-decoration*. Därför måste de modeller man vill integrera i ODIN förses med en *wrapper* i C++, vilket brukar vara mycket enkelt att åstadkomma.

Aggregera/disaggregera

Det finns inget som demonstrerades under vårt besök som tyder på att det skulle gå att aggregera och disaggregera förbandsenheter i ODIN. I den mån man vill simulera stora scenarier kopplade man ihop ODIN med en annan modell, *BattleLab*, som används för att simulera större flottstyrkor.

Möjlighet till distribuerade körningar

QinetiQ har kopplat ihop ODIN med ett antal olika modeller och simuleringsverktyg. I ett projekt har man kopplat ihop ODIN med ett program som heter *BattleLab*, vilket simulerar sjöstrid på (högre) taktisk nivå. Denna koppling gjordes som en HLA-federation med hjälp av en FOM (*Federation Object Model*) som kallas VMSA. Vid några tillfällen har man även kopplat in ytterligare en modell, SURVIVE, som används för att beräkna skador hos fartyg som träffats av en torped eller som har gått på en mina. I de fallen används *BattleLab* för att simulera det övergripande scenariot till dessa att torpedskott har skett. Därefter har ODIN använts för att simulera torpedskottets förlopp. Slutligen har resultatet av torpedträff simulerats i SURVIVE.

Man har även testat att koppla ihop ODIN med en målsökare till den brittiska marinens *Sting Ray*-torpeder. Torpedens målsökare har monterats på ett fast stativ vid en testbana och sedan har man låtit ett fartyg gå fram och tillbaka längs testbanan. Under försöken har styrdata från målsökaren kontinuerligt matats in i ODIN.

Då man önskar mer detaljerade data från torpedernas målsökare eller vill kunna testa nya signalbehandlingsalgoritmer brukar man koppla ihop ODIN med en signalbehandlingsmodell som kallas *Nereus*.

Om man så önskar går det att ha flera användargränssnitt igång samtidigt på olika datorer som är kopplade mot en och samma instans av ODIN. Teoretiskt går det även att ha flera instanser av ODIN igång som kommunicerar med varandra, men utvecklarna från QinetiQ verkade inte tycka att det fanns några skäl för att göra så.

Joint möjligheter

QinetiQ har även integrerat en del entiteter ovanför ytan. Det fanns simuleringar på grundläggande nivå som hade med satelliter och helikoptrar. De såg inte heller några begränsningar om man ville simulera joint-scenarion. Det bör dock observeras att ODIN främst är avsett för att simulera undervattensscenarier.

Användning för andra områden/syften

Inom Storbritannien används ODIN för operationsanalys, studieverksamhet, taktik- och doktrinutveckling samt för materielupphandling (en variant av *Simulation Based Acquisition*). Många av de utländska kunderna använder ODIN för träning och utbildning.

Kurser och User groups

Det ingår två 1-veckas användarkurser vid köp av ODIN. Den första kursen ger en allmän bild av ODIN och går ut på att bekanta sig med verktyget. Den andra är ämnad för att QinetiQ ska kunna visa hur kunden kan tillämpa ODIN på sina specifika uppgifter.

Det verkade inte finnas några regelrätta användargrupper eller forum för användare. Tydligt är det så att om en kund betalar för att få någon specifik funktionalitet så brukar andra användare normalt sett inte få ta del av detta. Det har dock funnits några undantag. Däremot kan man, mot en årlig avgift, få kontinuerliga uppdateringar från QinetiQ, mest av typen bugggrätning. Enda kunden som verkade ägna sig åt egen utveckling för ODIN är Norge.

Andra användare (vars erfarenheter vi kan ha nytta av)

QinetiQ har ett antal utländska kunder. USA har erhållit ODIN genom ett MoU (*Memorandum of Understanding*) och de personer från QinetiQ som deltog i mötet verkade inte veta eller vilja säga

vad USA har använt ODIN till. Övriga kunder är Kanada, Norge och Singapore som alla har köpt ODIN, främst för olika typer av utbildnings- och träningsverksamhet. Singapore är tydligen intresserat av grunda vatten (*shallow waters*) och har erhållit utbildning i hur ODIN kan konfigureras och användas för detta av QinetiQ. Norrmännen är tydligen ganska självständiga och ägnar åt egen utveckling av modeller och funktionalitet för ODIN.

Vidare undersökningar/kvarvarande frågor

Det som för tillfället är viktigast är att få tillgång till en demolicens av ODIN så att vi kan testa verktyget själva. Alternativet är att någon från QinetiQ kommer på besök och hjälper till att testa de olika saker vi vill kunna göra.

Det vore även mycket intressant att fråga norrmännen om deras erfarenheter av att använda ODIN och få en bättre bild av vad de har gjort och tänker göra.

5.7.4. VR-Forces

Inledning

MÅK var ett av de företag som svarade snabbt på vår förfrågan om simuleringsramverk för undervattensmiljö. Vi har haft diskussioner kring verktyget, och också fått ett flertal demonstrationer av VR-Forces, dels då MÅK och deras återförsäljare från Antycip besökte FOI, och på I/ITSEC, Orlando 2004.

Vi har dock inte själva genomfört några tester eller försök med verktyget, utan endast tagit del av befintligt material och ställt frågor som vi fått besvarade.

Verktyget är indelat i en *frontend* (*vrfGui*) för grafiskt användargränssnitt etc, samt en *backend* (*vrfSim*) för simuleringen.

Scenariohantering

Av demonstrationerna framgick att det är enkelt för en van användare att skapa nya scenarier. Det finns stöd för att klicka med musen i en karta och placera ut nya entiteter. Efter utplaceringen kan entiteternas beteende anges. Man kan exempelvis anvisa brytpunktsbanor de ska följa och ge dem order om att vara aggressiva eller passiva vid möte med fienden. Beteendet kan sedan om behov uppstår sättas om under pågående simulering. En ovan användare kanske finner scenariohanteringen något svårare, även om det såg ut att vara ett ganska lättanvänt grafiskt gränssnitt. Det är dock lovande att det redan verkar finnas så mycket stöd för att skapa nya scenarier.

När man sparar ett scenario i användargränssnittet *vrfGui* sparas även information om antalet startade instanser av *vrfSim*. Vid start från ett sparad scenario ges operatören möjlighet att antingen fortsätta exekvera scenariot med samma antal instanser av *vrfSim* som tidigare eller att samla alla objekt till endast en instans.

VR-Forces har i grundutförandet ett tämligen enkelt orderverk implementerat i det grafiska användargränssnittet. Orderverket består av ett antal uttryck av typen *while*, *if*, etc som kan kombineras med att man klickar i kartan. Regelverket är i viss mån utbyggbart, det går att lägga till egna tillståndsvariabler, däremot förefaller det som att "språket" som man bygger villkoren med är fixt, dvs *if-then-while*.

Det är också fullt möjligt att byta ut hela *vrfGui* mot ett egenutvecklat GUI (tex. SOAR) tack vare att VR-Forces har ett GUI API.

Exekvering

Verktyget är indelat i en frontend för grafiskt användargränssnitt etc samt en backend för simuleringen. VR-Forces backend går att bädda in i befintliga applikationer. Det går även att integrera egna modeller i en backend och att bygga en helt egen frontend. Möjlighet till lastbalansering finns genom att schemalägga exekveringen av olika entiteter på separata *backends*. Flera

frontends kan också användas samtidigt, t ex. vid träningsstationer. Det är också möjligt att "fjärrstyra" en frontend från en annan process, däremot inte att styra en backend på samma sätt.

Objekt som skapas i en instans av *vrfSim* kopplas till denna. Om en instans skulle krascha är det möjligt att länka om objekt skapade i denna instans till en annan instans.

Det är också möjligt att starta flera instanser av simuleringsmotorn, för att dela upp belastningen på simuleringen. En eller flera instanser av *vrfGui* går att koppla mot en eller flera instanser av *vrfSim* (många-till-många-förhållande).

Visualisering

Det finns ett fristående, till synes påkostat, grafikverktyg för 3D-visualisering som heter VR-Stealth/Stealth viewer. Vid en första anblick verkade det väldigt tjustigt med välgjord grafik. Funktionaliteten kan dock diskuteras. Ett grundläggande problem för vår del är att vattenytan inte är transparent, vilket kan leda till problem om man vill ha en överblick över en undervattenssimulering.

Ett annat potentiellt problem är illustrationen av sensortäckning. Det kanske allvarligaste problemet är att sensortäckningen inte baseras på data från simuleringen utan på externa data. Detta beror säkerligen på att det i grundutförandet inte finns några sensormodeller implementerade i VR-Forces annat än en enkel siktlinjemodell. Även om visualiseringen var väldigt snyggt gjord för en radar på ett flygplan är frågan om det är användbart för en undervattenssimulering med en mängd sensorer på ett antal olika plattformar. Det skulle antagligen se väldigt rörigt ut. Dessutom ändras sensorstäckningen för flertalet undervattenssensorer över tiden och med ändrad position.

Förutom sensortäckning kan VR-Stealth/Stealth viewer även användas för att visualisera vapentäckning. För närvarande kan VR-Stealth/Stealth viewer endast använda OpenFlight-/DTED-data för att generera terräng.

Med VR-Stealth/Stealth viewer går det att öppna flera olika vyer samtidigt (varje vy får ett eget fönster). Det går inte att begränsa vilken typ av objekt som en viss vy kan visa, så alla som har möjlighet att starta verktyget kan se alla tänkbara vyer.

Analys

Möjligheter till efterbearbetning av data från simuleringar finns, men vilka dessa är och hur de används har inte beaktats, i och med avsaknaden av undervattensmiljö.

Modifiering av befintliga modeller

Vid skapandet av nya objekt till simuleringarna förespråkade de att utgå ifrån deras generella modeller. De nya objekten skapas genom att modifiera värdena på en uppsättning parametrar som beskriver den generella modellen man utgår ifrån. Det framgick inte hur generella deras modeller var, dvs om det är troligt att de kommer att vara tillräckliga för våra behov. MÄK rekommenderade oss därför att gå en av deras kurser för att förstå detta bättre, och för vi bättre ska kunna uppskatta hur lång tid modifieringar av befintliga modeller kan ta.

Skapandet av nya objekt/entiteter görs genom redigering av en textfil. Detta kräver alltså ingen programmeringsfärdighet.

Modeller kan utvecklas som generiska plattformar, och parametersättas för att efterlikna ett större antal verkliga system.

Somliga av exempelmodellerna kan behöva synas ordentligt, exempelvis så modelleras inte rörelsebanan för ballistiska projektiler, utan endast p_{kill} används. Det är dock fullt möjligt att använda mer raffinerade skademodeller, t ex uppslagstabell för att få olika verkan mot olika typer av mål. Även polynomfunktioner kan användas för skademodeller.

Nyutveckling

Som tidigare nämnts är det rekommenderat att utgå ifrån MÄKs generella grundmodeller vid nyutveckling av objekt. Det förefaller svårare att skapa helt nya objekt ifrån grunden. Något som måste undersökas innan användandet av de generella modellerna, är hur mycket funktionalitet som redan är inkluderat. Enligt presentationen verkade det vara mycket (dödräkning, koordinat-transformationer etc) och det framgick inte om detta är lätt att göra om eller om det är smidigt att lägga till ytterliggare funktionalitet.

Ett konkret exempel som diskuterades var problematiken att lägga till en ny sensor. De verkade inte tycka att det skulle vara några problem att integrera t ex en EM-sensor under vattnet. För att avgöra om sensorn kommer att detektera ett mål eller ej kan man koppla bort deras detektionsalgoritmer och integrera sin egen signalpropageringskod. Observera att enligt nuvarande arkitektur avgör varje sensor hur den beräknar sin detektion. Detta är inget problem men ställer krav på att olika aktörer använder samma typ av signalpropageringsberäkning för att relevanta jämförelser ska möjliggöras.

SNE och miljö för undervatten

Ingen av MÄKs befintliga modeller påverkas idag av miljöparametrar. Trots att de försäkrade att detta är lätt att inkludera i en ny modell är det oroväckande. Det faktum att man inte har använt någon modell som påverkas av miljö när simuleringsramverket utvecklades medför att man riskerar att stöta på problem när en sådan modell ska införas. De problem som då kan uppkomma kan dessutom visa sig svåra att lösa med nuvarande design av ramverket. Detta är därför en stor risk som bör studeras vidare.

I VR-Forces finns möjlighet att representera bottenpografi. Programmet i sig självt har stöd för att hantera terrängdata av formaten OpenFlight, DTED och Shapefiler (ESRI). Dock så stödjer visualiseringsverktyget VR-Stealth/StealthViewer endast de två förstnämnda formaten.

För att ta in egen omgivningsmodell, vilket vi antagligen behöver göra, kan denna härledas från en befintlig klass, den som kontrollerar och styr simuleringen. Det är oklart hur mycket arbete detta medför, men det är inte olikt det sätt vi använt för att införliva en undervattensmiljö i FLAMES. Även för att ta in enstaka parametrar, som t ex salthalt eller vattentemperatur krävs detta ingrepp.

Stöd för undervattenplattformar

Det fanns i stort sett inget stöd alls för undervattensplattformar. Det ingick en mycket enkel ubåtsmodell i VR-Forces. Den är dock inte mer än en förklädd helikoptermodell som inte påverkas av terrängen (vilket bl a leder till att "ubåtar" kan köra rakt igenom undervattensterräng som t ex grynnor). Att det går att använda en exempelmodell för helikopter som ubåt tyder visserligen på att det är ett flexibelt verktyg, men det är också en svaghet att det inte finns en färdig ubåtsmodell.

Beräkningarna som gjordes med sensorerna lämnar mycket att önska vad avser graden av realism, och för undervattensstillämpningar måste troligtvis en modellsamling byggas upp från grunden.

Sensorer och Signaturer

CGF-delen, d v s modellbibliotek med datoriserade aktörer ska vara den starkaste sidan hos VR-FORCES.

De har utvecklat egna modeller för sensorer, men så att dessa var lätta att koppla bort och ersätta med egentillverkade. VR-Forces har i grundutförandet endast en siktlinjebaserad sensormodell för upptäckt och bekämpning av fientliga mål.

Icke-ballistiska vapen (i stort sett allt annat än kanoner, kulsprutor och eldhandvapen) skapas som egna objekt när de avfyras och kan därmed påverkas av andra objekt i scenariot. Således är

det t ex möjligt att skjuta ner en kryssningsmissil med jakt- eller luftvärnsrobotar.

Tekniska aspekter

MÄK's programvara (VRForces och VR-Stealth/Stealth viewer) finns för plattformarna Windows och Linux (och i viss mån SGIs operativsystem IRIX).

VR-Forces API är skrivet i C++. Enligt uppgift har en grundläggande filosofi varit att all funktionalitet ska vara tillgänglig via olika API:er (Simulation API, GUI API, Remote Control API och Terrain API).

Aggregera/disaggregera

VR-Forces har stöd för aggregering/disaggregering, i alla fall vad gäller förbandsobjekt. Det är dock "pseudo-aggregering", dvs för ett aggregerat förbandsobjekt hanteras de ingående förbandsobjekten som separata entiteter, inte som en helhet. Det finns dock vissa möjligheter till "sann aggregering" (men vi är osäkra på hur detta görs).

Det finns inga direkt uppenbara begränsningar för vad som kan aggregeras. Det är t ex fullt möjligt att aggregera ett antal grupper till en pluton, ett antal plutoner till ett kompani osv. Vidare kan dessa aktörer behandlas antingen individuellt (vilket är normalfallet) eller som grupp. En fråga är dock hur pass generella dessa mekanismer är.

Aktörer i ett scenario kan skicka meddelanden för att kommunicera med andra aktörer i ett scenario.

Möjlighet till distribuerade körningar

MÄK har vidareutvecklat sitt RTI och håller på att certifiera det mot den kommande standarden HLA 1516 DLC (*Dynamic Link Compatible*). Detta är en striktare specifikation än HLA 1516 och syftet är att få RTI:er som är *plug'n'play*-utbytbara.

MÄK erbjuder även verktyget RTI Spy som används för att logga den trafik som sker i RTI:et, bl a vem sänder och mottager vad, övervaka belastning, prestanda mm.

Joint möjligheter

De finns redan idag en joint-miljö inbyggd i systemet vilket är väldig lovande. Om det skulle visa sig att det är lätt att bygga ut stödet för undervattenstillämpningar är steget till att köra ett joint-scenario troligen ganska litet.

Användning för andra områden/syften

Det har inte framgått av demonstrationerna på vilka områden detta ramverk har sin styrka. Det är lite oklart om det bäst används som träningsanläggning eller för analys eller något annat.

Samtidigt har MÄK lagt ganska mycket kraft på visualisering och att det ska vara lätt att bygga ett scenario. Därför känns det som att fokus ligger på att VR-Forces ska kunna användas för att snabbt och enkelt implementera ett tänkt scenario och sedan visualisera hur det gick. Typiskt skulle detta kunna användas för att pröva planer och nya koncept utan alltför stora krav på detaljupplösning..

Vidare kan VR-Forces säkert fungera alldeles utmärkt som "stimulator" till olika typer av *Man-in-the-loop*-simulatorer, exempelvis stridsvagnsbesättningar, robotskyttar och liknande.

Att använda VR-Forces för analytiska simuleringar verkar kräva en hel del arbete med att implementera lämpliga modeller och algoritmer.

Kurser och User groups

MÄK saknar för närvarande en *UserGroup* för VR-Forces. Tanken på en sådan har funnits hos MÄK, men inga konkreta planer på att gå vidare verkar finnas. De har tydligen även diskuterat att ha ett webbforum, men hittills har arbetsinsatsen bedömts vara för stor.

MÄK erbjuder sig att hålla introduktionskursen där vi så önskar. Antingen blir den bara för oss eller så kan den bli tillsammans med andra som också är intresserade av att evaluera VR-Forces eller som har köpt programmet.

Andra användare (vars erfarenheter vi kan ha nytta av)

- Projekten OneSAF/ModSAF (USA). Det är gratis att använda CGF:er framtagna i projekten OneSAF och ModSAF är för användare i USA. Att de ändå köper VR-Forces CGF:er kan tolkas som en styrka hos produkten.
- Danska Flottan
- Norska försvarsmakten

5.7.5. STRIVE

Inledning

STRIVE är en produkt som har sina rötter dels i CAEs tidigare verktyg ITEMS, men som även har sin bakgrund i CAEs roll som leverantör av träningsanläggningar. Inom företaget såg man såväl behovet som nyttan av en infrastruktur som var generell och återanvändbar. STRIVE har en arkitektur som är baserad på HLA.

STRIVE har varit under utveckling i ett flertal år nu. Första gången vi såg det var på I/ITSEC 2000, då i en förhandsversion. Sedan dess har företaget investerat ansevärd summa i STRIVE⁷.

Under utvärderingsperioden har vi förfogat över en demoversion av STRIVE CGF. Denna version är en betarelease av STRIVE 2.0, och motsvarar inte till fullo den slutliga produkten. Den del av STRIVE som berör undervattensområdet (STRIVE SONAR) har vi vid utvärderingen inte haft tillgång till, då produkten fortfarande är under utveckling. Vi har dock haft tillgång till en omfattande PowerPoint presentation av STRIVE SONAR, samt korrespondens med CAEs utvecklingsteam.

STRIVE är en modulär produkt, dvs det går att införskaffa endast de delar som man är intresserad av. Ramverket för STRIVE benämns SFX, och kan sägas motsvara utvecklingspaketet för FLAMES. Där ingår API:er etc som behövs för att utveckla egna modeller.

CGF innehåller den syntetiska taktiska omgivningen, och tjänstgör även som ett användargränssnitt. Det är inte helt uppenbart var gränsen mellan de ovan nämnda byggblocken går, men då vi utgår från att de behövs för så gott som all användning av STRIVE lämnar vi den frågan därhän.

Scenariohantering

Scenarioeditorn, eller CGF Studio har en modern utformning, med så kallad *drag-and-drop* funktionalitet där man kan dra ut plattformar och släppa dem på kartan. CGF Studio är en rätt avancerad grafisk tillämpning, och de gränssnitt som är mest exponerade för en användare av verktyget ger intryck av att vara väl genomtänkta.

Ett scenario i STRIVE benämns *exercise*, eller övning på svenska. I CGF Studio kan man redigera nya övningar, spara, men också ansluta till dem. Aktörer i en övning benämns entiteter. Dessa kan redigeras och konfigureras i CGF Studio. I CGF Studio finns också en 2D kartvy, där de olika entiteterna placeras ut på kartan.

CGF Studio är också gränssnittet för att specificera parametrar för olika delmodeller, och redigera det taktiska beteendet. Beteendemodeller i STRIVE benämns doctrine, och byggs upp av ett antal olika variabler samt villkorssatser (*if-then*). Variablerna är knutna till en datatyp (logisk,

⁷ \$35,000,000 redan investerat i NeTT/STRIVE concept, och företaget räknar med att lägga ytterligare \$100,000,000 i en nära framtid. Källa: MS&T, 1/2005

numerisk, text), och kan representera tillstånd hos den egna eller någon annan aktör i scenariot. Konstruktionen förefaller mycket flexibel, men kräver förmodligen en viss förståelse för grunderna i programmering för att kunna utnyttjas fullt ut. De exempel som följer med demoversionen är tyvärr antingen triviala, eller för omfattande att gå igenom inom ramen för denna utvärdering.

Exekvering

Från CGF Studio kan exekveringen av ett scenario startas, alternativt går det att ansluta till en simulering som körs på annat håll. Det går också att stoppa, pausa och återuppta exekveringen (även om flera andra federater i en HLA-federation är involverade). En händelselogg rullas upp och visar meddelanden om taktiska händelser i scenariot.

Det finns två nivåer för att ange hur exekveringen av ett scenario ska schemaläggas: hårdvarunivå, där man kan välja att synkronisera med systemklockan, eller att inte göra det. På mjukvarunivån går det att synkronisera med systemklockan, med en valfri faktor, t ex 2x verklig tid. Det går också att välja att simuleringen ska stegas fram med tidssteg av en fix storlek. Om simuleringen inte skulle uppnå önskad exekveringshastighet, efter t ex införande av noggrannare beräkningar för någon del, exekveras scenariot i snabbast möjlig takt.

Vi har inte sett något färdigt stöd för Monte-Carlo körningar, men variabler som är globala för alla deltagare, men specifika för en viss övning kan definieras. Proceduren att starta och stoppa en övning går också att automatisera, och därmed finns åtminstone de mer grundläggande förutsättningarna.

Visualisering

Scenariot kan visualiseras i 2D eller i 3D med CGF Studio under exekvering. Det finns också möjlighet att implementera egna, specialiserade *overlays*, där taktisk information om scenariot kan visualiseras.

Möjligheterna att visualisera output från olika akustiska sensorer är mycket goda. En stor mängd av de typiska grafer som används för att visualisera sonardata i olika praktiska undervattentillämpningar finns redan inkluderade i programmet. Detta ger utmärkta möjligheter att använda verktyget för att skapa verklighetstroga träningsscenarion. För sonartillämpningen finns också tillhörande analysverktyg där akustisk vågutbredning i olika miljöer kan studeras i detalj.

Analys

Det finns möjligheter att analysera ett scenario i STRIVE. Verktyget CUI Studio gör det möjligt att studera och manipulera data för utrustningsmodeller, scenario och federater under simuleringens gång. Data kan plottas över tiden, och redovisas i tabeller. Valet av vilka data som ska presenteras, och hur, kan göras utan att avbryta exekveringen. Under simuleringens gång kan meddelanden om vad som händer i simuleringen loggas till fil.

I den demoversion vi haft tillgång till finns dock inget stöd för att spara en playbackfil, eller annan kvantitativ efterbearbetning av data. Däremot kan beteendemodeller analyseras under simuleringens gång, genom att visa vilken villkorssats som för närvarande är aktiv, och även värdet på de variabler som ingår.

Modifiering av befintliga modeller

Hela uppbyggnaden av STRIVE är modulär och ger intrycket av att vara skapad för att ge användaren goda möjligheter för att genomföra modifieringar. Användarna kan antingen byta ut hela, delar eller ändra parametrar i befintliga modeller. För att underlätta det senare medföljer en mängd generiska exempelmodeller som kan parametreras för att efterlikna andra system. Modellerna av fysiska system (i stort sett allt utom modeller för beteende och omgivning) är hierarkiskt uppbyggda och olika delmodeller knyts samman till en enhet med textfiler. Omgivningsmodellerna för t ex akustisk vågutbredning är designade för att hela delar ska vara lätta att

byta ut. Det finns API:er som beskriver hur befintliga externa modeller måste modifieras för att kunna inkluderas i simuleringen. I ramverket är dessutom redan förberett för att nya vågutbredningsmodeller ska kunna använda alla möjliga typer av tänkbara inputparametrar, vilket är ett ytterligare exempel på att designen är gjord för att det ska vara lätt att göra egna modifieringar. CAE uppskattar tidsåtgången för att inkludera en ny modell, som t ex en ny vågutbredningsmodell, till 1-3 dagar för en van användare.

Nyutveckling

Egna modeller av exempelvis rörelsedynamik kan göras som plugins till STRIVE. Utvecklaren följer då ett antal riktlinjer för hur modellen ska implementeras, och STRIVE ser till att den blir åtkomlig för användaren. Det går också att kopiera befintliga parameteruppsättningar, som använder samma underliggande implementering.

Att utöka ramverket med nya typer av sensorer, som tex elektromagnetiska för undervattensdomänen, verkar görbart så länge man håller sig till deras ursprungliga struktur (se "SNE och miljö för undervatten" för närmare beskrivning). Mer omfattande ändringar verkar däremot inte lika enkelt.

Utveckling av egna modeller i STRIVE kan göras på olika nivåer. Dels kan den så kallade CSF (*core simulation framework*) användas, som hanterar instansiering, objekthantering, mm, men mer användbart är förmodligen CSE (*core synthetic environment*), som enligt manualen tillhandahåller algoritmer och specifika implementeringar. Enligt uppgift ska det också vara möjligt att implementera egna omvärldsmodeller⁸, det är dock oklart vilken arbetsinsats som erfordras. STRIVE kan också läsa in terrängdata från verktyget TerraVista.

SNE och miljö för undervatten

I STRIVE finns en *Ocean Server* som hanterar miljön för simuleringen. Där finns miljödata, brusdata och modeller som hanterar signalpropageringen. All hantering av akustiska signaler är i frekvensdomän och styrs av sonarekvationen. Ramverket hanterar såväl lågfrekventa som högfrekventa fall. Beräkningar av vad som kommer till de akustiska sensorerna baseras på de signaturer som finns i vattnet, det omgivande bruset och försvagningen av signalerna då de propagerar genom vattnet. Denna hantering försäkrar användaren om att alla sensorer kommer att använda sig av samma miljö och vågutbredning till skillnad från de ramverk då detta styrs av sensorerna.

Stöd för undervattensplattformar

Redan idag finns stöd för ett stort antal entiteter som ubåtar, ytfartyg, torpeder och en mängd motmedel. Alla dessa är även modulärt uppbyggda för att beskriva rörelser, beteende, signatur mm, vilket underlättar att föra in nya typer av plattformar.

Sensorer och signaturer

Det finns ett väl utbyggt stöd för både sensorer och signaturer. Alla signaturer är lagrade i en del av programmet som kallas *Target model*. De är uppdelade i en passiv signatur med motorljud mm som förändras med entitetens hastighet, och en aktiv signatur samt hur denna förändras i förhållande till aspektvinkeln mellan sonar och målentitet.

De akustiska sensorerna är uppbyggda för att både kunna hantera passiva och aktiva signaler. Detta medför att hänsyn tas till intercept under passiv spaning. Förutom den tidigare beskrivna väl utvecklade visualiseringen finns även en ljudgenereringsmodul vilken medför att en användare kan lyssna på resultatet i en akustisk sensor i alla dess lobber. Även detta ger utmärkta möjligheter att skapa träningsscenarier med hjälp av ramverket. Detaljerna kring hur ljudet skapas med tanke på att akustiska beräkningar sker i frekvensdomän har inte framgått i vare sig dokumentation från

⁸ Den omvärldsmodell som följer med den utgåva av produkten vi fått är inte särskilt väldokumenterad.

eller korrespondens med CAE. Övriga typer av sensorer i undervattensdomänen är inte utvecklade. Av tillverkaren framgår dock att det är genomförbart att lägga till ytterligare sensorer så länge man håller sig till deras ursprungliga arkitektur.

Tekniska aspekter

STRIVE finns för plattformarna Windows och Linux. De grafiska användargränssnitten använder Qt, och den rörliga grafiken använder OpenGL för visualisering. STRIVEs RTI följer med installationen.

Gränssnittet för att utveckla egna modeller är C++, och då behöver även Microsofts utvecklingsmiljö vara installerad.

Aggregera/disaggregera

Inga kända uppgifter.

Möjlighet till distribuerade körningar

Att sätta upp en distribuerad körning förefaller ganska enkelt, men då vi inte haft tillgång till mer än en licens har vi inte prövat detta.

Som det nämndes i inledningen så har STRIVEs arkitektur som är baserad på HLA, och stödjer ett flertal RTI:er, bl a MÅK och DMSO. CAE har dessutom ett eget RTI som följer med produkten.

Joint möjligheter

STRIVE har stöd för terränghantering, undervattensområdet och flygande farkoster. Allt övrigt tillgängligt informationsmaterial tyder också på stora möjligheter till simuleringar av joint-scenarier.

Interaktion mellan entiteter

Aktörer i ett scenario kan kommunicera och utbyta data med varandra. Vilka mekanismer som finns för kommunikation mellan delmodeller har inte beaktats här.

Användning för andra områden/syften

CAE har en stark ställning på marknaden som leverantör av träningsanläggningar, och STRIVE är mycket mer än ett verktyg för analytiska simuleringar. Själva ramverket (SFX), kan naturligtvis tillämpas i många olika sammanhang. Men STRIVE är inte bara ett verktyg för undervattensområdet, eller för CGF:er. Utan att knyta an till mark-, luft-, eller sjöstrid finns det ett antal tilläggsmoduler för olika områden. Exempelvis *Electronic warfare*, som stöder framtagande av displaysystem, MMI:er och analysverktyg för telekrigområdet. Terrain Server, är precis som namnet antyder en terrängserver som kan förse andra modeller med siktlinjeberäkningar och information om terrängens beskaffenhet. Men också verktygen STRIVE RADAR och STRIVE FLIR förefaller ha god funktionalitet för såväl simulering som visualisering av sensorapplikationer.

Kurser och user groups

CAE har ett antal kurser som riktar sig mot användare av STRIVE.⁹

⁹http://www.cae.com/www2004/Products_and_Services/Military_Simulation_and_Training/Modeling_and_Simulation/et.shtml

Andra användare (vars erfarenheter vi kan ha nytta av)

Vi känner inte till någon som använder STRIVE för analytiska tillämpningar i undervattensområdet.

Kvarvarande frågor

Den version av STRIVE vi haft tillgång till är en betaversion, och alltså ännu ej släppt. Det är inte bara möjligt, utan också troligt att de många av de buggar och saknade features vi noterat är kompletterade när denna rapport skrivs (vilket förmodligen gäller även de andra verktygen). För att få en rättvis bild av STRIVE behöver utvecklingen följas upp. Vidare har STRIVE SONAR bara studerats genom informationsmaterial. Denna del behöver testas för att undersöka om produkten fungerar enligt beskrivning.

Bristen på verktyg för kvantitativ efteranalys behöver inte betyda att STRIVE inte går att använda för mer analytiska applikationer, men kostnaden för att komplettera analysverktygen bör undersökas närmare. Det är också möjligt att CAE har planer på att ta fram egna verktyg för det ändamålet.

Möjligheten att påverka CAEs utveckling av verktyget är oklar. Om man känner behov av modifieringar som är svåra att genomföra själv, så är möjligheterna att få CAE att göra dessa och vilka kostnadsnivåer det skulle röra sig om, lika oklara.

STRIVE är en omfattande och komplex produkt, och i den här studien har vi bara skrapat på ytan. För att få en bättre bild behöver mer omfattande tester göras, och kompetens om STRIVE byggas upp genom bl a deltagande i anpassade kurser.

6. Sammanfattning

Inom ramen för SaSS-UV har vi bedömt fem olika verktyg: FLAMES från Ternion, STRIVE från CAE, VR-Forces från MÅK, ODIN från Qinetiq samt MOSES från TNO. Utvärderingen har gjorts dels mot kriterier som baseras på tidigare års arbete inom detta projekt, och dels på praktiska erfarenheter från ett stort antal forsknings- och utvecklingsprojekt som har utnyttjat simuleringsramverk på FOI under många år. Vidare har vi försökt ta till vara på alla de krav och behov som har visat sig vara gemensamma för flertalet av de stora simuleringsprojekt som har bedrivits på FOI sedan början av 80-talet. Kriterierna är dock inte viktade mot varandra, eftersom en sådan viktning förutsätter en betydligt mer detaljerad kännedom om för vilka uppgifter verktyget ska användas till, och därför är en punktvis jämförelse av de olika verktygen inte lämplig i detta skede.

Kriterierna berör såväl tekniska som affärsmässiga aspekter för verktyg och leverantörer. Av de kriterier som vi funnit relevanta (se kapitel 4) så är det endast en del av dessa som behandlar de mer tekniska aspekterna för vad ett simuleringsverktyg bör klara av. De affärsmässiga aspekterna på en leverantör är minst lika viktiga. Vid genomgången av kriterieuppfyllelse i föregående kapitel skiljer sig de olika leverantörerna och deras respektive produkter avsevärt för dessa aspekter.

6.1. Kriterieuppfyllelse

TNOs och QinetiQs produkter uppfyller många av de undervattensspecifika tekniska önskemålen som finns på ramverket. De viktigaste plattformarna, signaturerna och sensortyperna finns redan implementerade. Båda dessa leverantörer har dock brister i de affärsmässiga aspekterna. Deras respektive produkter är inte kommersiella, vilket bland annat leder till att det är svårt att få demoversioner av verktygen. Det är även oklart vilken support en köpare kommer att få. I mötena med företagen verkade det dock som om de strävade efter en kommersialisering av sina produkter, och enligt vår bedömning har QinetiQ så här långt kommit längst. I den mer tekniska jämförelsen av produkterna anser vi också ODIN vara den mest avancerade. MOSES förefaller fortfarande vara en omogen produkt som är under utveckling.

I jämförelse med de militärt näraliggande organisationerna uppfyller MÅK, CAE och Ternion de affärsmässiga önskemålen mycket bättre. De tillhandahåller gärna demoversioner av sina respektive verktyg och har väl utvecklade hjälporganisationer för sina användare. Ingen av deras produkter har dock fokus på undervattensdomänen. MÅK har exempelvis en visuell ubåtsplattform, men kinematiskt är det en helikoptermodell som kan användas i en undervattensmiljö. Hanteringen av akustiska signaler under vattnet är väldigt förenklad. Vid en första analys verkar det dessutom inte vara helt enkelt att bygga ut ramverket så att det klarar av våra grundläggande krav. I Flames har vi redan med framgång testat att lägga till en undervattensmiljö med tillhörande akustiska sensorer (se kapitel 2). Problemet med Ternions produkt är att stöd för undervattensmiljön behöver byggas in i ramverket, något som måste göras av Ternion. Den undervattensmiljö som implementerades i pilotstudien gjordes i syfte att visa att konceptet är realiserbart, inte för att vara robust nog för att köra jointsimuleringar och storskalig användning. Företaget har dock inte uttryckt intresse av att själva finansiera denna utökning av ramverket, eftersom deras kundbas huvudsakligen är fokuserade på flygtillämpningar. I CAEs presentationer däremot framgår det att STRIVEs omfattande modellering av undervattensdomänen går att köra tillsammans med övriga tillämpningar. Detta är mycket lovande med tanke på att deras undervattensmiljö redan idag verkar vara väl utvecklad och lätt att utvidga med nya applikationer. En tydlig brist är dock att STRIVE SONAR än så länge bara har presenterats med ett PowerPointdokument. Någon demoversion finns ännu inte tillgänglig, och det går således inte att bedöma om produkten håller vad den lovar.

En normalt sett viktig aspekt vid val och utvärdering av verktyg är dess kostnad. Vi har här

valt att utelämna detta, delvis för att uppgifter om licenspriser etc vanligen är konfidentiella, och inte får publiceras. Men en ytterligare orsak är att det i detta skedet är svårt att bedöma kostnaderna. Priset för anskaffning av licenser är beroende på hur många personer som kommer att använda verktyget, och vilka moduler och tillägg som behövs. Men ett hårt faktum är att den största kostnaden inte utgörs av utgifter för licenser, utan kostnader för att bygga upp kompetens inom organisationen, samt ta fram modeller av de egna materielsystem och doktriner som bör finnas tillgängliga. De sistnämnda är dessutom inte engångskostnader, då kunskap och kompetens behöver vidmakthållas under ett antal år.

6.2. Verksamhetsnytta

Ett simuleringsramverk är ett komplext verktyg och kan uppfattas som en infrastruktur för M&S, som kan bedömas utifrån ett flertal perspektiv. Erfarenhet från dessa perspektiv finns dock redan, dels från flerårig användning av simuleringsramverk på FOI, dels från studier av möjligheten att finna/anpassa infrastruktur för M&S till undervattensområdet. Denna typ av verktyg vänder sig mot en mycket bred grupp av användare inom hela försvarsmakten och har stor potential att göra arbetet effektivare och livet enklare inte bara för oss utan även för våra kunder.

Att etablera användandet av ett simuleringsramverk som bör klara av såväl olika problemområden (mark, luft, sjö, undervatten) som olika tillämpningar är en svår uppgift. För att lyckas krävs god kännedom om behov, såväl tekniska som organisatoriska. En alltför hög grad av specialisering gör visserligen det komplexa verktyget lättanvänt, men medför ofta oönskade begränsningar.

Interoperabilitet och återanvändning hör också till de viktiga aspekterna man måste ta hänsyn till i valet av verktyg. För simuleringsramverk gäller det hela tiden att göra en avvägning mellan att förenkla för användaren och för utvecklaren att enkelt och snabbt producera nya körningar, och att erbjuda största möjliga flexibilitet till både samverkan och återanvändning. En av fördelarna med att använda ett simuleringsramverk är att det tvingar utvecklaren att skapa modeller som fungerar modulärt, och ger därmed större möjlighet till återanvändning. Problemet är att de ekonomiska effekterna av dessa egenskaper är svåra att mäta, användningen av simuleringsramverk ger istället mer strategiska fördelar, som kortare ledtider, ökad förmåga att ta fram större och/eller bättre modeller, bättre konkurrenssituation, mm. Dessa verktyg kan till en början uppfattas som dyra, komplexa och svårbemästrade, men efterhand brukar det visa sig att åtskilliga manår kan sparas in, jämfört med att på egen hand utveckla motsvarande funktionalitet.

En annan viktig faktor att tänka på är hur man ska dra nytta av den stora investering som det rör sig om. En trolig hypotes är att verktyget inte kan vara begränsat till att endast omfatta undervattensområdet. Med tanke på den investering man gör är det i högsta grad önskvärt att verktyget även ska kunna omfatta joint-scenarier i framtiden.

Det huvudsakliga syftet med SaSS-UV har varit att utreda om det är tekniskt möjligt att använda simuleringsramverk för UV området, och ge en helhetsbild av området simuleringsramverk och en uppdatering på det kommersiella utbudet. Målet har varit att enhetligt sammanställa hur väl de mest lovande produkterna uppfyller våra krav för att möjliggöra rekommendationer för lämpliga framtida simuleringsramverk. En bedömning av hur en organisation bäst kan dra nytta av de verktyg som utvärderats ligger därför utöver målsättningen, men diskuteras kortfattat i följande avsnitt.

6.3. Slutsatser

Krav och förväntningar på simuleringsverktyg är helt avgörande för valet av arkitektur och ramverk. Inför anskaffandet av ett simuleringsramverk så finns det ofta ett antal "randvillkor" som i praktiken kommer att slå in som styrande krav. Dessa kan t ex vara tidsramar, ekonomiska förutsättningar, användningsområde, lämplig detaljeringsgrad, tekniska resurser eller tillgängliga kompetenser. I vissa fall kan något av dessa randvillkor ensamt avgöra valet av produkten.

Simuleringsramverk kan vara kraftfulla verktyg för så kallad *Rapid Prototyping*, d v s att snabbt kunna ta fram körbara scenarier som gör att man tidigt kan pröva sina idéer, eller på kort tid ta fram en simulering som kan svara på en viss frågeställning. Införandet av nya objekt, nya beteenden och variationer på simuleringar går smidigt i ett sådant verktyg. Å andra sidan är det viktigt att man väljer rätt simuleringsramverk som är avsedd för användningsområdet, ty omfattande justeringar av existerande simuleringsramverk tenderar att bli svåra och ibland dyra att genomföra. Införandet av en enkel UV-miljö i Flames gick visserligen ganska bra, men för att få en fullt utvecklad UV-miljö behöver anpassningar av FLAMES göras av Ternion.

Såväl MOSES som ODIN är framtagna för att användas inom den egna organisationen, och för att möta behovet av modellering och simulering specifikt för UV området. En organisation med liknande behov skulle förmodligen, om de affärsmässiga aspekterna kan hanteras, kunna dra stor nytta av såväl MOSES som ODIN. Men om behoven går utanför UV området, finns det goda skäl att beakta de tekniska begränsningarna.

VR-Forces och FLAMES är framtagna för att vara kommersiella M&S verktyg, där kunderna själva ska kunna utveckla egna modeller, och respektive företag stå för infrastrukturen. Både Ternion och MÅK är förhållandevis små företag, där de som jobbar på företaget ofta är delägare. Ett intryck från utvärderingen av STRIVE är att ett av designkriterierna har varit att verktyget ska passa en så stor kundkrets som möjligt, utan att behöva modifieras eller anpassas. CAE är dessutom en börsnoterad internationell koncern, och förutsättningarna att påverka verktygets fortsatta utveckling är därför annorlunda än för Ternion och MÅK.

6.4. Rekommendationer

Vi är övertygade om att användande av simuleringsramverk kan åstadkomma en förnyelse och dynamik i försvarsforskningen, samtidigt som det kan stärka FOIs externa nätverk och kundkrets, inte minst med väsentligt nya möjligheter att studera olika hotbilder.

Baserad på de argument som gavs och de diskussioner som fördes under avsnitt 6.1 så kan varken MOSES eller VR-Forces rekommenderas. Vi bedömer däremot ODIN som en lovande produkt som har ett omfattande stöd för undervattensområdet. Affärsmässigt förefaller dock Flames och STRIVE som klart starkare kandidater. Möjligheterna att genomföra vidareutvecklingar och framförallt utvidgningar mot joint-scenarier framstår också som större i dessa verktyg. I en vidare jämförelse mellan Flames och STRIVE framstår STRIVE som den mest lockande kandidaten med tanke på det mycket väl genomarbetade stöd som redan finns för undervattensdomänen. Visserligen finns det ännu ingen demoversion av STRIVE SONAR, men enligt den information vi har verkar detta vara ett verktyg med mycket god potential.

Även om CAEs simuleringsramverk STRIVE framstår som det bästa alternativet kvarstår en del frågor. I och med att vi inte har kunnat testa det fullt ut går det inte att i dagsläget rekommendera en anskaffning av produkten. Verktyget uppvisar dock en mycket stor potential som motiverar ytterligare tester för att utvärdera om det kan vara värt att anskaffa. En viktig punkt vid testerna är möjligheterna för nyutveckling och införande av egna applikationer. Med tanke på att CAE är en stor organisation går det inte att räkna med att önskemål om hur produkten ska uppdateras alltid kommer att tillgodoseas. Å andra sidan kan sådana särskilda anpassningar nästan alltid beställas, av leverantören eller annan part, eller lösas med egen utveckling. En förutsättning är dock att det finns ett tillräckligt starkt motiv för anpassningen, och därmed möjligheter till finansiering.

6.5. Fortsatt arbete

SaSS-UV var en intern strategisk satsning på FOI i form av ett samarbetsprojekt mellan de tre institutionerna Systemmodellering, Signalbehandling och Numerisk vågutbredning och analys vid avdelningen Systemteknik. Syftet med projektet har bland annat varit att undersöka om konceptet simuleringsramverk kan användas för UV-området, att belysa de viktigaste aspekterna inför ett

anskaffande av ett simuleringsramverk, att göra en grundläggande undersökning av existerande ramverk och baserat på detta göra rekommendationer för framtiden. Strategin med satsningen var att resultaten och rekommendationerna från projektet sedan skulle kunna användas av andra projekt som har behov av att genomföra simuleringar i UV-området. Framförallt projekt som behöver anskaffa ny simuleringsprogramvara. Om projektet SaSS-UV skulle ha fortsatt hade den naturliga fortsättningen varit att införskaffa en demoversion av STRIVE och fortsätta undersökningen på en djupare nivå. Innan en eventuell anskaffning av ett simuleringsramverk hade det också analyserats för vilka ändamål och i vilken omfattning det kan användas på FOI. Projektet kommer dock inte att fortsätta under 2006, utan avslutas med denna rapport. Det ligger nu därför på de aktuella projekten som har behov av en ny simuleringsmiljö att gå vidare med uppgiften.

7. Referenser

- [1] Vahid Mojtahed, Göran Bergström och Mattias Karlsson, En jämförelse av ansatserna SaSS-UV och UV-strid, FOI rapport FOI-R--1568--SE, 2005.
- [2] Per-Axel Karlsson och Lisbeth Pers, MUMS - En modell för simulering av undervattens motmedel, FOA rapport FOA-R--96-00374-313--SE, 1996.
- [3] Robert Sigg och Jenny Schiöld, COMBIS version 3.0, FOI rapport FOI-R--1349--SE, 2004.
- [4] MS&T magazine, issue 3, 2002.
- [5] Proceedings of the 2004 Interservice/Industry Training, Simulation and Education Conference (I/ITSEC-2004), Orlando, Florida, U.S.A., December 6-9, 2004.

Appendix A – Request For Information (RFI)



At correspondence refer to
FOI designation

Date
27 Sep 2004

Nr

Your reference

Our reference
Vahid Mojtaheed

Looking for a Maritime Simulation Framework

FOI, the Swedish Defence Research Agency, has developed and used computerized simulation models for more than 30 years. Recently a project has been started with the purpose to study if and how commercial simulation tools could be used in maritime studies. One of the activities in this project is to look for a tool that supports modelling and simulation of naval (military) scenarios with subsurface, surface, ground, and airborne units, mainly at the tactical level.

We have two main criteria for potential candidates. First of all, the tool should be a simulation framework. By using the term framework, we mean that the tool should have some sort of simulation kernel, which takes care of the more low-level aspects of simulation, e.g. time handling, database management, etc. In our view, a useful simulation framework for maritime applications has to be a fully functional, end-to-end simulation system complete with off-the-shelf applications that support definition, execution, post-processing and visualization of scenarios. The most important characteristic of a simulation framework is its underlying architecture; an architecture that allows not only the applications supplied with the framework to be customized, but also allows new framework-based applications to be developed rapidly and inexpensively. Also, this architecture should be based on a simulation kernel that schedules and orchestrates the different parts in executing a scenario: participating actors (such as submarines, torpedoes, or C2 sites) and their sensors, inter-actor communication, actor motion, and cognitive behaviour.

Secondly, it should allow us to use our own in-house developed models. When it comes to using in-house developed models with the simulation tool, we would like to be able to implement a range of different models. These include numerical modelling of physical platforms (submarines, ships, etc.) and wave propagation as well as environmental models. Furthermore, we would like to be able to simulate tactical behaviour and decision processes using cognitive models. For this "customer model" functionality we would like to have control of the implementation at source code level. So, to summarise we are looking for a

Systems Modelling

Postal address	Visiting address	Telephone	Fax	Web page
SE-172 90 Stockholm Sweden	Enköpingsvägen 126	+46 8 555 030 00	+46 8 555 033 97	www.foi.se

simulation framework that supports incorporation of subsurface wave propagation models and associated environmental parameters and variables, rather than a set of models.

This request for information is primarily initiated by one project to fulfil one type of requirement. However, it would be an advantage if the proposed framework can be of use for other types of usages within the field of modelling and simulation as well. Any simulation tool that fits the profile for the maritime simulation framework outlined above will be evaluated and may become part of our toolkit for modelling and simulation for quite some time. We are currently using a simulation framework that has up until now fulfilled our needs, both for air warfare and ground combat simulations, but it has no support for underwater applications. It is therefore desirable that the proposed maritime simulation framework in some way is interoperable with other simulations, thus enabling us to study scenarios regarding joint operations, i.e. scenarios with units from different branches of the armed forces. Needless to say, if the new framework could be shown to supersede our current simulation tool, it may gradually replace that in many other applications.

If your company has already an existing simulation framework that fits the description above, please contact us. We will evaluate all such products that are made available to us to see to what extent they fulfil our needs.

Please address questions about the interpretation of these request to one of the FOI researchers listed below.

POC on Simulation Framework issues: Johan Pelo, johan.pelo@foi.se

POC on Maritime issues: Göran Bergström, goran.bergstrom@foi.se

Sincerely yours,

Project manager
Vahid Mojtahed

Appendix B – Företag till vilka RFI:n skickats

Aerosim-Mechtronix
Aerotech Telub
Aerotech World Trade Ltd.
Alenia Marconi Systems
Benntec Systemtechnik GmbH
BGT Bodenseewerk
Boeing Autometric
BVR Systems.
CAE
Datamat SpA
DT Media Ltd.
EDM Ltd.
Evans & Sutherland
ISD Data AB
Kockums AB
Lockheed Martin Training
MÄK Technologies
Micro Analysis & Design
MOOG Inc.
MultiGen-Paradigm, Inc.
Northrop Grumman Information Technology
Orchard Communications Design Group Ltd.
QinetiQ (*)
Rafael – Systems Division
Saab Bofors Underwater Systems
SAIC (Science Applications International Corp.)
SimAuthor Inc.
Simulis LLC
SSPA
ST Training & Simulation Pte. Ltd.
STN Atlas Elektronik GmbH
Systems & Management Solutions International
Systems Technology Inc.
Ternion (*)
Thales Training & Simulation
The AEGIS Technologies Group
TNO/FEL (*)
VEGA Group Plc.
ZDeep

*) RFI ej skickad, då kontakt med företaget redan etablerats.