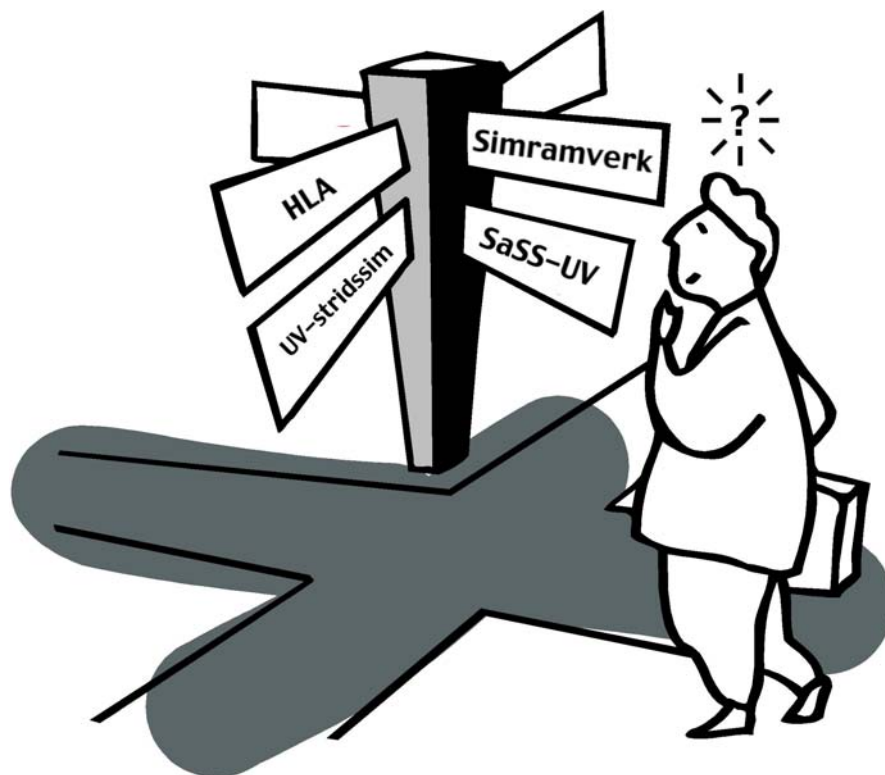


Vahid Mojtahed, Göran Bergström, Mattias Karlsson

En jämförelse av ansatserna SaSS-UV och UV-strid



TOTALFÖRSVARETS FORSKNINGSINSTITUT

Systemteknik
172 90 Stockholm

FOI-R--1568—SE

Januari 2005

ISSN 1650-1942

Underlagsrapport

Vahid Mojtahed, Göran Bergström, Mattias Karlsson

En jämförelse av ansatserna SaSS-UV och UV-strid

Utgivare Totalförsvarets Forskningsinstitut - FOI Systemteknik 172 90 Stockholm	Rapportnummer, ISRN FOI-R--1568--SE	Klassificering Underlagsrapport
	Forskningsområde	
	Månad, år Januari 2005	Projektnummer I60125
	Delområde	
	Delområde 2	
Författare/redaktör Vahid Mojtahed Göran Bergström Mattias Karlsson	Projektledare Vahid Mojtahed	
	Godkänd av Monica Dahlén	
	Uppdragsgivare/kundbeteckning FOI – Avd. för Systemteknik	
	Tekniskt och/eller vetenskapligt ansvarig Vahid Mojtahed	
Rapportens titel En jämförelse av ansatserna SaSS-UV och UV-strid		
Sammanfattning (högst 200 ord) <p>Det utvecklas idag en mängd olika simuleringssystem för att simulera undervattensverksamhet. Två projekt där FOI deltar i utvecklingen är SaSS-UV och demonstratorprojektet UV-stridssimulator. Trots att dessa projekt till viss del liknar varandra skiljer de sig åt på en rad olika punkter. UV-stridssimulatorens är ett projekt som genomförs i samarbete med andra organisationer och den bygger på arkitekturen HLA. SaSS-UV är ett internt FOI-projekt som baseras på simuleringsramverket FLAMES. I denna rapport undersöks för- och nackdelar med att utnyttja en HLA-baserad simulering respektive att utnyttja simuleringsramverket FLAMES för att simulera undervattensverksamhet.</p> <p>Jämförelsen mellan arkitekturerna har gjorts genom att ställa upp en rad olika kriterier för vilka deras egenskaper studeras. Målet med rapporten är således inte att jämföra projekten UV-stridssimulator och SaSS-UV, utan att jämföra deras underliggande arkitekturer. Trots att arkitekturernas kvalitativt skiljer sig åt på flera punkter går det inte att peka ut den ena som bättre än den andra. Syftet med rapporten är snarare att påvisa vid vilka tillfällen den ena arkitekturen passar bättre än den andra.</p>		
Nyckelord Modellering och simulering, Simuleringsarkitektur, Simuleringsramverk, HLA, FLAMES, Undervattenssimulering		
Övriga bibliografiska uppgifter	Språk Svenska	
ISSN 1650-1942	Antal sidor: 24 s.	
Distribution enligt missiv	Pris: Enligt prislista	

Issuing organization FOI – Swedish Defence Research Agency Systems Technology SE-172 90 Stockholm	Report number, ISRN FOI-R--1568--SE	Report type Base data report
	Programme Areas	
	Month year January 2005	Project no. I60125
	Subcategories	
	Subcategories 2	
Author/s (editor/s) Vahid Mojtahed Göran Bergström Mattias Karlsson	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) A comparison between the UV-stridsimulator and SaSS-UV approaches		
Abstract (not more than 200 words) <p>For the time being several simulation systems are developed to simulate underwater activities. Two projects, in which FOI is involved in the development, are SaSS-UV and UV-stridssimulator. Even though these projects partly are similar to each other they differ in several aspects. The UV-stridssimulator is a cooperation project with other organizations and it is built on HLA. SaSS-UV is an FOI internal project which is based on using the simulation framework FLAMES. In this report the benefits and disadvantages of using the architectures HLA and the simulation framework FLAMES for simulation of underwater activities are investigated.</p> <p>The comparison between the architectures is made by exploring their abilities for several different criteria. The aim of this report is not to compare the projects UV-stridsimulator and SaSS-UV, but rather to compare their underlying architectures. Even though the architectures qualitatively differ in several aspects, it is not possible to decide which one is better. Hence, the report is focused on indicating in which applications one architecture is better suited than the other.</p>		
Keywords Modeling and simulation, Simulation architecture, Simulation framework, HLA, FLAMES, Underwater simulation		
Further bibliographic information	Language Swedish	
ISSN 1650-1942	Pages 24 p.	
	Price acc. to pricelist	

Innehållsförteckning

1. Inledning	6
1.1. Bakgrund	6
1.2. Syfte med rapporten	7
1.3. Läsanvisning	7
2. Simuleringsarkitekturer.....	8
2.1. En kort beskrivning av HLA.....	8
2.2. En kort beskrivning av Simuleringsramverk	9
2.3. En kort beskrivning av Flames	9
3. Intressanta kriterier	11
3.1. Stabilitet hos tekniken/arkitekturen.....	11
3.2. Stabilitet hos bakomliggande organisationer	11
3.3. Möjlighet till modellutveckling	12
3.4. Modellering och simulering av mänsklig beteende	12
3.5. Förmågan att göra detaljerade loggningar och efteranalyser	13
3.6. Möjligheter att ansluta tredjepartsprodukter	13
3.7. Krav på utvecklingsgruppen.....	14
3.8. Förändringar av scenariomiljöer.....	14
3.9. Nya funktionaliteter och uppgifter	14
3.10. Interoperabilitet / samverkan	15
3.11. Återanvändbarhet.....	15
3.12. Resurser för utveckling	15
3.13. Resurser för användning	16
4. Diskussion	17
4.1. Allmänna förutsättningar.....	17
4.2. Sammanfattning och slutsatser	17
4.3. Resultat	18
Bilaga 1 – En sammanställning av för- och nackdelar för HLA respektive simuleringsramverk	20

1. Inledning

1.1. Bakgrund

Sedan 1996 har arbete med och utnyttjande av simuleringsramverk pågått på FOI, och Institutionen för Systemmodellering på avdelningen för Systemteknik har av naturliga skäl haft en central roll i detta. Under dessa år har ett antal FOI-projekt studerat taktiska frågeställningar och prövat stridstekniska idéer med hjälp av olika luft-, mark- och sjöscenarier. På senare år har man känt ett behov av att undersöka om, och i så fall hur, man kan använda konceptet simuleringsramverk även för undervattensområdet. Som resultat av detta, startades det interna strategiska projektet SaSS-UV på FOI under våren 2003. SaSS-UV står för "System-av-System Simulering med simuleringsramverk - inriktning UV" och ett av huvudmålen i det projektet var att undersöka möjligheten att kunna bygga en syntetisk miljö för olika undervattenssimuleringar m.h.a. konceptet simuleringsramverk.

Under 2002 initierade högkvarteret ett arbete för att skapa ett demonstratorprojekt med namnet undervattensstridssimulator (i detta dokument kallad UV-Strid). FMV fick i uppdrag att beställa simulatoren och satte ihop en arbetsgrupp bestående av personal från Kockums, Saab Bofors Underwater Systems (SBUS), FOI Systemteknik samt konsultföretagen FrontEnd och Pitch, för att designa systemet. Främsta målet för projektet var att utveckla en simuleringsmiljö där produkter och koncept i tidigt skede kan utprovas och förbättras, i syfte att försöka underlätta och förkorta utvecklingsarbetet. Under 2003 kom själva utvecklingsarbetet av demonstratorn igång med Kockums som huvudleverantör.

Båda dessa projekt har således som huvudmål att bygga en simulering där man kan simulera undervattensscenarier på stridsteknisk och taktisk nivå. Projekten skiljer sig dock på en rad olika punkter.

För det första har simuleringarna fundamentalt olika underliggande arkitekturer. UV-strid byggs enligt standarden HLA - "High Level Architecture" - vilket är en arkitektur som beskriver hur olika typer av modeller skall utformas för att kunna återanvändas och samköras i gemensamma simuleringar. (Se avsnitt 2.1 för en kort beskrivning av HLA.) SaSS-UV å andra sidan byggs i simuleringsramverket Flames. Ett simuleringsramverk är ett kraftfullt modellerings- och simuleringsverktyg som skall hjälpa användaren med bl.a. en utvecklingsmiljö, ett antal tjänster som är gemensamma för alla simuleringar, ett antal standardverktyg, en uppsättning av exempelmodeller, o.s.v. (Se avsnitt 2.2 för en kort beskrivning av simuleringsramverk generellt och avsnitt 2.3 för en kort beskrivning av Flames.)

För det andra har de olika ekonomiska och personella förutsättningar. UV-strid har en avsevärt större budget (ligger på flera miljoner kronor per år) än SaSS-UV (som ligger på några hundratusen kronor per år). UV-stridsprojektet är dessutom ett samarbete mellan ett stort antal personer med vitt skilda kompetenser från flera olika företag. SaSS-UV är däremot ett internt FOI-projekt i formen av ett samarbete mellan institutionerna för Systemmodellering, Signalbehandling och Numerisk

Vågutbredning och analys vid Avdelningen för Systemteknik på FOI och projektet bemannas enbart med FOI-personal. Slutligen bör det poängteras att för UV-stridsprojektet är den kravspecifierade funktionaliteten och tidsramarna hårdare definierad än i SaSS-UV.

1.2. Syfte med rapporten

Under 2003 koncentrerades arbetet inom SaSS-UV-projektet på ett typfall där möjligheten att kunna integrera en befintlig modell för akustisk sonar i simuleringsramverket Flames undersöktes. Det arbete som genomfördes visade bland annat på att konceptet med simuleringsramverk är lämpligt för användning inom undervattensområdet. Med tanke på beröringspunkterna med det stora UV-stridsprojektet, bedömdes det lämpligt att som en av uppgifterna i projektet under 2004 planera en undersökning som formulerades enligt nedan:

”Gör en analys som kan svara på frågan, vad det är som skiljer sig respektive är gemensam för ansatserna SaSS-UV och UV-stridsdemonstrator (vad är det som karaktäriserar användningen av ett Simuleringsramverk kontra en HLA-arkitektur).”

Även om arkitekturerna för de båda simuleringarna är helt olika finns det också en del likheter mellan dem. I denna rapport jämförs för- och nackdelar med att utnyttja en HLA-baserad simulering respektive att utnyttja simuleringsramverket Flames för att simulera undervattensverksamhet. Målet med rapporten är således inte att jämföra projekten UV-strid och SaSS-UV utan att jämföra deras underliggande arkitekturer. Det är inte heller meningen att visa att den ena arkitekturen är bättre än den andra utan snarare att, baserat på deras egenskaper, visa vid vilka tillfällen det ena systemet passar bättre än det andra. Slutsatserna i rapporten kan därför ses som en indikation på vad man kan göra med dessa simuleringar i framtiden.

Jämförelsen mellan arkitekturerna har gjorts genom att jämföra deras egenskaper för en rad olika kriterier. De kriterier som tas upp i rapporten är större frågor som exempelvis hur resurskrävande utvecklingen är, hur flexibel arkitekturen är för utvidgningar o.s.v. Mindre och tekniskt betingade frågor som t.ex. hur bra tids-hantering fungerar berörs inte.

1.3. Läsanvisning

Kapitel 2 inleds med en kortfattad beskrivning av simuleringsstandarden HLA. Detta följs en generell beskrivning av simuleringsramverk och kapitlet avslutas med en kort presentation av simuleringsramverket FLAMES.

För att kunna göra en jämförelse mellan de två arkitekturerna har ett antal kriterier tagits fram. Dessa återges i kapitel 3 tillsammans med en bedömning hur de två arkitekturerna uppfyller varje kriterium.

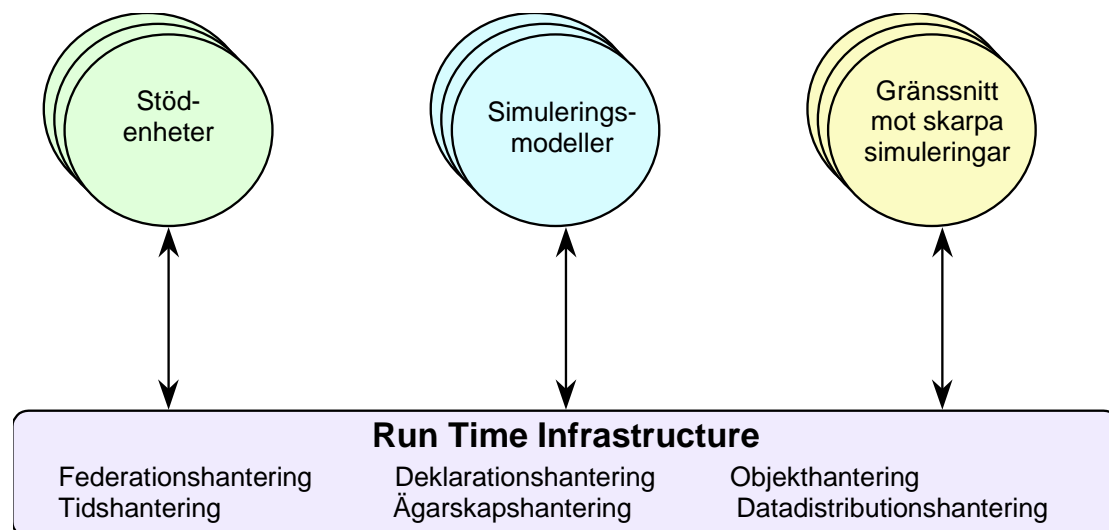
I kapitel 4 presenteras inledningsvis några allmänna förutsättningar för val av simuleringsarkitektur. Därefter följer en sammanfattning samt en resultatdiskussion.

2. Simuleringsarkitekturer

2.1. En kort beskrivning av HLA

HLA (High Level Architecture) beskriver den strukturella uppbyggnaden hos den infrastruktur som krävs för att olika typer av modeller skall kunna kopplas ihop i gemensamma simuleringar och samverka med varandra. Den beskrivna infrastrukturen syftar även till att modellerna skall kunna återanvändas som komponenter i framtida simuleringsstudier och övningar. En mängd modeller som samverkar med varandra i enlighet med HLA och som har sammanförts för en simulering i ett specifikt syfte kallas en federation och varje deltagande simuleringsmodell kallas för en federat, d.v.s. federationsmedlem. Dessa medlemsmodeller skall kunna vara av vitt skilda slag, ha utvecklats för olika syften, modellera helt olika aggregationsnivåer och till och med ha olika tidshantering. Denna arkitektur definieras av tre grundelement:

- Med *objektmodeller* beskrivs på ett standardiserat sätt de objekt som modelleras hos var och en av federationsmedlemmarna. De uttrycks i en standardiserad form, s.k. objektmodellsmallar, Object Model Templates (OMT), som tabeller vilka utgör formella "kontrakt" mellan federationsmedlemmarna om de objekttyper och de interaktioner som skall stödjas under en simulering. Varje federationsmedlem skall beskrivas med en SOM (Simulation Object Model), vari man redovisar alla de objekt som medlemmen modellerar och vill göra synliga för andra modeller samt relationer och interaktioner mellan dem. Till varje federation måste det finnas en Federation Object Model (FOM), som beskriver alla de objekt och interaktioner som är av intresse i kommunikationen mellan medlemmarna i den aktuella tillämpningen.



- En *gränssnittsspecifikation*, som beskriver de funktioner som federationsmedlemmarna skall utnyttja i RTI (Runtime Infrastructure). RTI är en standardiserad programvara, som betjänar medlemmarna på samma sätt som ett distribuerat operativsystem betjänar olika mjukvaror.

- För att åstadkomma en HLA-baserad simulering fordras att federationen och dess medlemmar lever upp till tio fundamentala regler. Dessa är dock mycket naturliga, som att varje medlem måste kunna uppdatera och återge alla de objekt och interaktioner som de lovat i sina SOM:ar, och att under en gemensam simulering får inget utbyte av data som finns definierat i FOM:en ske utanför RTI:n.

2.2. En kort beskrivning av Simuleringsramverk

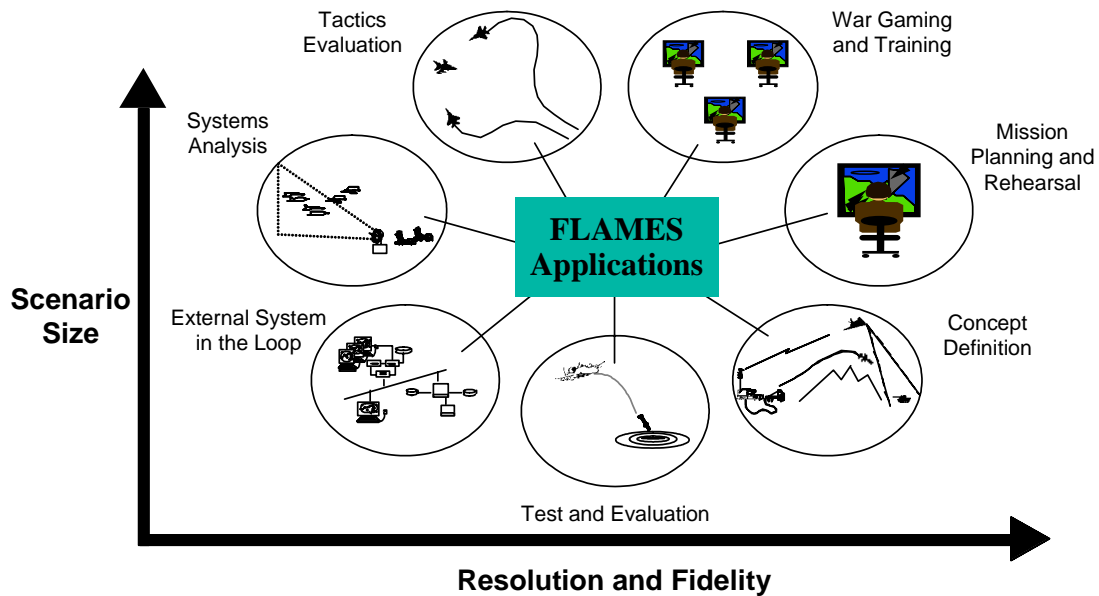
En betydande del av den funktionalitet som behövs för stora och generella simuleringsverktyg är ofta generell, och har ofta liten anknytning till själva problemområdet. Exempel på detta är tidshantering, databashantering, scenariogenerering, visualisering, etc. Ett av de största kraven man ställer på ett simuleringsramverk är att denna grundläggande funktionalitet tillhandahålls av dess kärna. Med anledning av detta, samt med tanke på det faktum att byggandet av komplexa och generella modeller och simuleringsystem från grunden i många fall är förknippat med stora risker, har en tydlig utveckling mot användande av olika simuleringsramverk skett.

Ett simuleringsramverk kan således sägas vara en infrastruktur, där tjänster för modellutveckling, exekvering, analys, scenarioupbyggnad och visualisering tillhandahålls. Om det finns en väl fungerande organisation som stödjer simuleringsramverket och ser till att det utvecklas kontinuerligt, så att det hela tiden möter nya behov, ökar nyttan betydligt. Finns det en uppsättning av färdiga modeller som ingår i verktyget minskar även tiden från anskaffning till produktion avsevärt. Under senare år har ett antal simuleringsramverk för militär modellering- och simulering (MoS) på stridsteknisk, taktisk och operativ nivå utvecklats, där FLAMES som används på FOI är ett exempel.

2.3. En kort beskrivning av Flames

FLAMES (FLexible Analysis and Mission Effectiveness System) är ett simuleringsramverk för simulering av komplexa militära scenarier på stridsteknisk/taktisk nivå. Ramverket är en kommersiell programvara, framtagen av företaget "Ternion Corporation", som också tillhandahåller support, utbildning och utveckling av modeller.

FLAMES är designat för att vara utbyggbart i bemärkelsen att det finns en väl dokumenterad process för att bygga egna modeller som utnyttjar de av ramverket tillhandahållna tjänsterna. Gemensamt för de olika tillämpningsområdena är att samtliga innefattar militära scenarier med ett antal aktörer, där aktörerna består av teknisk utrustning tillsammans med en kognitiv modell, vilken antingen kan styras av en beteendemodell eller av en operatör, s.k. man-in-the-loop-simulering. Ramverket tillhandahåller även applikationer för scenariogenerering, exekvering, visualisering och analys.



Ett enda simuleringsverktyg som kan användas för många olika syften, med varierande scenariostorlekar och omfattande antal deltagare med olika upplösningar och trovärdighetsgrader.

FLAMES är ett av de kraftfullaste simuleringsramverken för simuleringar av mångamot-många-dueller på marknaden idag. Verktöget är utvecklat för att användas inom den militära sektorn och tillhandahåller åtskilliga användbara modeller inom området. Figuren ovan skall visa på att FLAMES är utvecklat för att kunna användas för många olika syften, såsom exempelvis krigsspel, systemanalys, taktikutveckling. Vidare visar figuren att Flames klarar av olika upplösningar och olika nivåer av trovärdighetsgrader samt varierande och omfattande scenariostorlekar. Verktöget stödjer utöver modellering av fysiska egenskaper för olika materielsystem även beteendemodellering.

3. Intressanta kriterier

Även om det teoretiskt sett skulle gå att göra en kvantitativ jämförelse mellan en simulering utvecklad i enlighet med HLA-standard och en utvecklad med ett simuleringsramverk så är det inte praktiskt möjligt, främst beroende på de kostnader detta skulle medföra. Därför har en mer kvalitativ ansats valts vid jämförelsen i detta kapitel. För de valda kriterierna har en bedömning gjorts hur väl HLA-standardens respektive simuleringsramverket FLAMES uppfyller dessa. Målsättningen har varit att såväl val av kriterier som bedömningen av hur de jämförda produkterna uppfyller dessa ska ske så objektivt som möjligt. Syftet med kapitlet är att försöka visa på för- och nackdelar med HLA respektive simuleringsramverk.

3.1. *Stabilitet hos tekniken/arkitekturen*

Verktyg som kan klassas som simuleringsramverk för militärt bruk har funnits i ca 15-20 år, men trots det finns ingen gemensam definition för konceptet simuleringsramverk. Utvecklingen av ramverket Flames startades 1989 (enligt vissa utsagor skedde dock utveckling under hela 80-talet men under annat produktamn), och den första versionen var klar som en COTS¹-produkt 1992. Sedan 1995-96 uppfattas produkten som beprövad och stabil. Enligt företaget används FLAMES idag i ca 20-30 olika länder. I USA lär Flames användas av alla försvarsgrenar - flygvapnet, armén, flottan och marinkåren.

HLA initierades år 1995 av det amerikanska försvarsdepartementet, DoD, och det är en vidareutveckling av konceptet DIS, vilket står för Distributed Interactive Simulation, som initierades runt 1985. Därmed har även konceptet HLA ca 20 år på nacken, men de tekniska delarna som möjliggör att man praktiskt kan utveckla simuleringar i form av HLA-federationer har utvecklats först på senare år. Delar av konceptet är fortfarande under forskning och utveckling. Då utvecklingen startade relativt sent har tekniken inte hunnit mogna. Man brottas fortfarande med en del grundläggande frågeställningar vid utformandet av t.ex. nya RTI:er. Samtidigt, trots att tekniken är relativt ung och under utveckling, är den mycket beprövad, framförallt inom militära simuleringar. Arkitekturen har också vunnit internationellt genomslag och används i många länder idag.

3.2. *Stabilitet hos bakomliggande organisationer*

SaSS-UV är baserat på simuleringsramverket Flames, som ägs och utvecklas av företaget Ternion Corporation² med säte i Huntsville, Alabama i USA. För de företag som använder Flames finns det en möjlighet att träffas på användarkonferenser, anordnade av Ternion, för att kommunicera med varandra och dela med sig av sina problem och erfarenheter. I detta användarforum är det möjligt att hitta företag som har modeller man är i behov av. Dessa kan man byta mot modeller man själv har utvecklat. Användarna har också möjlighet att påverka Ternion i vilken riktning Flames skall utvecklas. Ternion är dock ett ganska litet företag med begränsade

¹ COTS - commercial off-the-shelf

² Ternion Corporation – För mer information se <http://www.ternion.com>

resurser, vilket kan leda till att föreslagna förändringar både kan kosta pengar och ta tid.

UV-strid bygger på HLA-arkitekturen som idag är en IEEE-standard³ och använder sig av pRTI⁴, vilket är en programvara utvecklad av företaget Pitch i Linköping. För att ta fram och exekvera en federation (d.v.s. en HLA-baserad simulering) använder man en särskild process kallad FEDEP⁵ - "FEderation Development and Execution Process", vilket också är en IEEE-standard. FEDEP föreskriver bl.a. att följande steg skall genomföras när man tar fram en federation: Kravdefinition - Konceptuell modell - Federations design - Integration och test - Exekvering och resultatanalys. Till stöd för utveckling av en federation finns en omfattande dokumentation med standarder och regler. Det börjar även dyka upp olika verktyg som stöder olika delar av utvecklingsarbetet.

3.3. Möjlighet till modellutveckling

I dagsläget är Flames bara anpassat till luft- och markstrid och är således inte ämnat för simuleringar av undervattensscenarier. Då en sådan utökning inte är planerad inom den närmaste framtiden, måste en dylik anpassning antingen göras av oss själva eller beställas av Ternion. För luft- och marksimuleringar finns dock mycket som redan är klart, t.ex. fördefinierade miljöer, terrängmodeller, generiska farkoster och beteendemodeller. Därför vore det eventuellt önskvärt att hitta ett annat simuleringsramverk som är avsett för undervattenssimuleringar, men som helst även bör klara "joint"-scenarier (Simulering av scenario som involverar gemensamma operationer med såväl luft-, mark- och undervattensstrid). Detta är för övrigt en av huvuduppgifterna för SaSS-UV-projektet under innevarande år.

Till skillnad från Flames finns varken miljö eller farkoster definierade på förhand i HLA-världen. Alla sådana modeller måste antingen byggas upp från grunden eller så bidrar de företag som deltar i federationsbygget med egna, redan befintliga modeller. För att ta fram en ny federation använder man FEDEP-processen. Idén med HLA är att när en modell (en federat) är utvecklad enligt HLA-standard för en federation skall det vara relativt lätt att senare utnyttja den i en annan federation. Det är tänkt att det skall så småningom finnas ett gemensamt bibliotek innehållande ett stort antal SOM:ar (deklarationer av federater). Från detta bibliotek skall man kunna plocka lämpliga federater till sin federation.

3.4. Modellering och simulering av mänsklig beteende

I Flames finns ett antal fördefinierade modeller av mänskligt beteende, så kallade kognitiva modeller. Dessa fördefinierade modeller kan vara till stor hjälp om deras beteende är relevant för frågeställningen, men dessa måste ändå ses som exempelmodeller vars mönster man följer när man skall skapa egna beteende-

³ HLA - IEEE Standard, [1516-2000](#), for Modeling and Simulation (M&S) High Level Architecture (HLA)---Framework and Rules

⁴ pRTI - The certified Run-Time Infrastructure for the open international HLA 1516 IEEE standard, from PITCH, <http://www.pitch.se/>

⁵ FEDEP - IEEE Recommended, [1516.3-2003](#), Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP)

modeller. Kognitiva modeller i Flames fungerar som avancerade tillståndsmaskiner, där utvecklaren har, i princip, fria händer att modellera det beteende som önskas.

Till skillnad från Flames finns inga fördefinierade modeller av mänskligt beteende i HLA. Dock finns det många företag som har utvecklat sådana modeller vilka går att få eller köpa och sedan försöka HLA:ifiera, d.v.s. anpassa det till HLA-regelverket. Vidare finns det även ett antal arkitekturer och COTS-verktyg, avsedda för modellering av mänskligt beteende, vilka börjar komma med HLA-portar.

3.5. *Förmågan att göra detaljerade loggningar och efteranalyser*

I UV-strid och många andra HLA-baserade simuleringar finns en resultatlogg som kontinuerligt registrerar alla objekt i simuleringen samt alla händelser som sker under exekveringen av ett scenario. Det finns därför stora möjligheter att göra detaljerade efteranalyser på dessa data off-line (efter en exekvering) med en extra programvara. Eftersom antalet händelser kan bli väldigt många är det bra att på förhand välja ut vilka händelser som är intressanta och därmed skall loggas. En sådan filtrering, vilken minskar storleken på loggfilen och underlättar efteranalysen, är möjlig att göra i en HLA-federation.

Man kan säga att i princip finns det motsvarande möjlighet i Flames också. I samband med scenariodefinitionen brukar man deklarerar vilka händelser man är intresserade av att logga, och hur ofta skall de loggas. På så sätt skapar man loggfiler som efter körning kan analyseras med olika analysverktyg. Ett SQL⁶-baserat analysverktyg följer även med ramverket.

3.6. *Möjligheter att ansluta tredjepartsprodukter*

För HLA-baserade federationer är det en rad olika kringprodukter som är aktuella. Loggningsverktyg och efteranalysprodukter brukar höra till de man i första hand tänker på, men allt från övervaknings- och visualiseringsverktyg till modeller för syntetisk naturlig omgivning (Synthetic Natural Environment, SNE) kan anskaffas och kopplas in som en tredjepartsprodukt. Produkter som efteranalysverktyg påverkar ofta inte federationen under gång utan använder sig bara av resultatloggen eller liknande. Olika visualiserings- och GUI⁷-verktyg är i regel lite svårare att föra in eftersom de ofta måste HLA-anpassas, vilket kan bli ett ganska omfattande jobb. Svårast är det troligen att ansluta komplexa tredjepartsprodukter, exempelvis SNE-modeller. Om produkterna är HLA-anpassade går det naturligtvis mycket lättare, men det kan ändå innebära en del besvär om produkterna är skrivna för ett annat RTI eller framtagna för ett helt annat scenario.

I och med att Flames idag är HLA-ifierat, så finns det i princip möjlighet att vid behov koppla in alla ovannämnda produkterna via HLA. Till skillnad från HLA-baserade federationer, som ofta börjar från noll, så finns i simuleringssammanhaningen en hel del redan på plats. I och med att visualiseringsverktyg, scenarioberedare,

⁶ SQL – Structured Query Language, ett s.k. frågespråk, vilket används för att ställa frågor till en databashanterare.

⁷ GUI – Graphical User Interface, grafiskt användargränssnitt

exekveringsövervakare etc. normalt sett hör till ramverket, så har behovet av att koppla in tredjepartsprodukter varit lite annorlunda. Därmed finns det lite olika möjligheter att koppla in sådana. Här har man skapat olika portar för att kunna koppla in andra simulatorer, C4I-system, MIL – Man In the Loop – d.v.s. mänskliga operatörer, röststyrningssystem och liknande.

3.7. *Krav på utvecklingsgruppen*

För båda arkitekturerna måste det finnas dels kompetens som kan beskriva hur olika entiteter agerar, dels kompetens för att implementera agerandet tekniskt. HLA är skapat bland annat för att den tekniska utvecklingen skall kunna ske distribuerat med många olika aktörer som sitter på olika orter. Utvecklingen testas sedan på gemensam ort och för att det hela skall fungera ställs det stora krav på ett ordentligt förarbete. För en effektiv utveckling av simulatören behövs det därför en stark kompetens i de inledande faserna av FEDEP-processen. Liknande krav finns inte för Flames, men där är det svårare att genomföra en distribuerad utveckling på flera orter.

3.8. *Förändringar av scenariomiljöer*

Det är viktigt att på sikt kunna simulera "joint"-scenarier. För Flames har det visat sig att inte vara helt lätt att på egen hand lägga till en ytterligare miljö (undervatten). Det går bra om den skall simuleras för sig, men det blir problematiskt om man skall köra ett "joint scenario". I UV-strid gick det bra att skapa en undervattensmiljö, men frågan är hur det senare kommer att gå att utvidga denna med mark- och luftsimuleringar. Genomförda simuleringar i USA visar dock att det går bra att genomföra en sådan utveckling i HLA, vilket verkar lovande för framtiden.

Sammanfattningsvis kan det sägas att tekniskt är det nog möjligt för båda arkitekturerna att klara av simulering av gemensamma operationer. Det är dock svårt att idag bedöma vilken av arkitekturerna som är billigast och smidigast, både i anskaffning och i användning på längre sikt. En mer intressant frågeställning i bägge fallen torde ändå vara den semantiska problematiken vid en sådan samkörning, d.v.s. hur det fungerar att gira med 6G för att lägga sig bakom ett fientligt flygplan, avfyra en jaktrobot och följa denna samtidigt som simuleringen räknar på en signals utbredning under vattnet, inklusive alla dess studsar mot havsbotten, havsytan och andra närliggande aktörer. Detta kan innebära att varken realtidskörningar eller MIL-funktion (Man in the Loop) kan stödjas.

3.9. *Nya funktionaliteter och uppgifter*

Varje gång en ny entitet skall läggas till en simulator i HLA skapas en ny federation, vilket egentligen kräver en ny FEDEP-process. Även om man ignorerar det och inte går igenom alla steg i processen är det ett omfattande arbetet med specifikationer av gränssytor o.s.v. för att lägga till den nya entiteten. Detta arbete kan däremot gå ganska snabbt i Flames där det finns färdiga generiska modeller av t.ex. en farkost. Att få in en enkel helikoptermodell i Flames kan därför gå väldigt snabbt jämfört med motsvarande arbete i HLA. Arbetet med att införa nya beteendemodeller, vilket endast indirekt påverkar datautbytet mellan modellerna, kan dock anses likvärdigt i de båda arkitekturerna. Likaså om man vill studera en ny uppgift med befintliga entiteter,

vilket inte heller leder till nämnvärt större arbete i någon av simulatorerna. Dock krävs det alltid en ny federation i HLA även om man vill studera nya uppgifter med befintliga federater och deras redan existerande funktionaliteter.

3.10. Interoperabilitet / samverkan

När det gäller samverkan mellan flera aktörer och samkörning av modeller fungerar HLA bra. Eftersom arkitekturen beskriver hur ingående modeller skall utformas, blir det endast ett begränsat arbete innan man kan få modeller utvecklade för olika federationer att agera med varandra. Man kan, åtminstone teoretiskt sett, till och med samköra flera federationer utan större komplikationer (Det finns dock inga egna erfarenheter av sådant arbete på FOI). Liknande möjligheter är inte lika stora i Flames. Även om man kan låna modeller av varandra är det svårt att koppla ihop simuleringar lika smidigt som i HLA. Det finns dock som tidigare nämndes både HLA-kopplingar och andra portar i Flames som möjliggör en sådan samverkan (se 3.6 Möjligheter att koppla in tredjepartsprodukter).

3.11. Återanvändbarhet

Modeller utvecklade i HLA är relativt lätta att återanvända i andra federationer. Trots detta har vi hittills i praktiken tyvärr inte sett att federater återanvänds i större utsträckning i nya federationer. Eftersom modellerna är utvecklade för ett specifikt RTI kan justeringar uppkomma i samband med det. Det är också troligt att vissa delar, som var specifika för den federation för vilken modellen utvecklades, måste justeras till den nya federationen (se FOM:ar och SOM:ar i 2.1 En kort beskrivning av HLA). För den nya simuleringen kan det t.ex. finnas krav på att modellen levererar mer information om sitt agerande än vad som krävdes tidigare. För skapande av nya entiteter som liknar redan befintliga är det värre. Även om man i förarbetet till en federation kan arbeta fram konceptuella, generiska modeller av exempelvis en farkost, är det ett omfattande arbete att sedan skapa de olika specifika farkosterna. Den delen fungerar bättre i Flames där de generiska entiteterna lättare överförs till specifika aktörer. I Flames går det förvånansvärd lätt och smidigt att överföra en specificerad aktör, uppdrag eller beteende från ett scenario till ett annat, eller från en simulering till en annan. Däremot kan det dock vara betydligt svårare att överföra en specificerad aktör från en annan simulator eller simulering till en simulering i Flames (om man inte använder de härför avsedda portarna, se 2.6 och 2.10).

3.12. Resurser för utveckling

Att utveckla en HLA-federation är dyrt. Det är särskilt viktigt att lägga mycket pengar i de inledande förstudie- och kravställande faserna. Även om tillgången på kompetens inte är begränsande kommer detta arbete att bli kostsamt. Fördelarna kommer senare i och med att det blir billigare att utöka simuleringssystemen då de bygger på en sådan standardiserade arkitektur som HLA. För Flames kostar det pengar att bygga upp en kompetens och en resurs som kan bygga simuleringar i ramverket. När denna kompetens har byggts upp, blir förstudie- och kravdefinitionsfaserna billigare och snabbare jämfört med motsvarande fas i genomförandet av en FEDEP. Det är dock nämnvärt att den stora komplexiteten i ett simuleringssystem som Flames gör att det är en ganska hög tröskel innan man kan klassas som en kompetent Flames-utvecklare.

3.13. Resurser för användning

När simulatorerna/simuleringarna väl är klara så beror det rätt mycket på utformningen hur enkla de är att använda. Om de har utvecklats med användarvänliga gränssnitt och är väldokumenterade, så är de ofta smidiga att använda. En fördel med det omfattande förstudiearbetet i FEDEP-processen (om man nu följer denna fullt ut) är att det genereras mycket dokumentation om de simulatorer och simuleringar som är byggda enligt HLA-standard.

Å andra sidan, att förse ett simuleringsramverk med användarvänliga gränssnitt samt de medföljande användar-, utvecklar- och referensmanualerna är en del av definitionen av konceptet.

En mer intressant fråga är för vilka syften dessa simulatorer ofta används. Tyvärr har de flesta federationer vi hittills har sett i Sverige (och även i USA) antingen varit testfederationer eller demonstratorer. I dessa fall är ofta användning begränsad till ett enda tillfälle, d v s man sätter upp ett stort system som skall fungera med en enda knapptryckning vid ett eller ett fåtal tillfällen. För Flames är det i princip tvärtom. Ett militärt simuleringsramverk har per definition 10-tals olika användningsområden (se 2.3 En kort beskrivning av Flames). Användning av Flames (t.ex. i analyssammanhang vilket har varit den flitigaste användningen på FOI) är som en evigt iterativ och interaktiv process. Där kommer man hela tiden med nya frågeställningar, vilket i sin tur kräver modifieringar och där detta inte räcker, får man lägga in nya funktionaliteter. Därefter blir exekvering och analys av resultat som i sin tur skapar nya frågeställningar o.s.v. I denna iterativa process är ofta alla från scenariogenererare, modellutvecklare till militärt områdesexpert och uppdragsgivare inblandade.

Kostnaden för att använda simulatorer och simuleringar byggda enligt olika arkitekturer beror utöver kostnaden för utveckling också på licensavgifter. Licensavgifterna för Flames eller olika RTI:er blir dock oftast endast en marginell kostnad av projektet.

4. Diskussion

4.1. Allmänna förutsättningar

Beskrivningen av olika kriterier i kapitel 3 kan ge sken av att det är en enkel process att välja antingen ett simuleringsramverk eller en HLA-federation. Det är dock viktigt att komma ihåg att det finns viktiga faktorer som påverkar valet, vilka samtidigt ligger utanför den presenterade listan över kriterier. En faktor att tänka igenom innan man beslutar sig för endera den ena eller den andra lösningen är *kontexten*, det sammanhang i vilket en simulering skall användas. Där ingår vilket syfte man vill nå med simuleringen och vilket användningsområde denna förväntas få. Det finns troligen även en beställare, extern eller intern, som har åsikter och önskemål som påverkar valet av simuleringsteknik. En annan viktig faktor är det *organisatoriska arvet* hos såväl utvecklare som användare. I de absolut flesta fallen kommer en simulering att utvecklas och användas av organisationer som har utvecklat och använt flera andra typer av simuleringsverktyg. Detta påverkar hur man inom organisationen ser på olika tekniska lösningar, vad som är möjligt och även vad som är önskvärt.

De ovan nämnda faktorerna kommer framför allt att påverka hur de olika urvalskriterierna "viktas", d.v.s. hur viktiga de anses vara, inför valet av en teknisk lösning på de simuleringsbehov som man har. Kontext och organisatoriskt arv kommer därför att vara avgörande för hur man använder urvalskriterierna och det är således viktigt att åtminstone fundera igenom dessa faktorer innan beslut tas.

4.2. Sammanfattning och slutsatser

Kraven och förväntningarna på en simulator är helt avgörande för valet av arkitektur. Inför anskaffandet av en simulator 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, tekniska resurser eller tillgängliga kompetenser. I vissa fall kan något av dessa randvillkor ensamt avgöra valet av tekniken. Exempelvis är HLA med sitt omfattande FEDEP-arbete känd för att vara en tidskrävande process som gör att det dröjer innan man har en fungerande första federation. Simuleringsramverk å andra sidan brukar vara väldigt bra på den så kallad "Rapid Prototyping", d.v.s. att snabbt kunna ta fram körbara scenarier som gör att man tidigt kan pröva sina idéer. Så, om man har kravet på sig att inom en kort tid ta fram en simulering som kan svara på en viss frågeställning, är en analys av de andra nämnda kriterierna inte till så stor nytta. Den tid som finns tillgänglig för att lösa uppgiften begränsar således valet av simuleringsteknik och fungerar som ett "randvillkor".

Interoperabilitet och återanvändning hör också till de viktiga aspekterna man måste ta hänsyn till i valet av arkitektur. Där tiden och ekonomin så tillåter, eller då helt nya miljöer eller typer av simuleringar ska skapas, förefaller det lämpligare att göra detta i HLA eftersom det kan ge en större flexibilitet, återanvändbara komponenter och en bättre förutsättning för framtida samverkan med andra modeller. För simuleringsramverk gäller det hela tiden att göra en avvägning mellan att förenkla för användaren och 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 därmed större möjlighet till återanvändning.

Det är vår uppfattning att utveckling i HLA inte bör väljas om det inte finns stora ekonomiska resurser, åtminstone inte förrän det existerar ett stort (internationellt) bibliotek med SOM:ar och FOM:ar att tillgå. Skälet för detta är att även mindre vidareutvecklingar av existerande federationer kan bli kostsamma. Å andra sidan tenderar omfattande justeringar av existerande simuleringsramverk 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 måste som nämnts Ternion påverkas att gå i den riktningen.

Införandet av nya objekt, nya beteende och variationer på simuleringar går dock betydligt smidigare i Flames än i HLA. På grund av den dyra utvecklingen i HLA skulle man därför till och med kunna tänka sig att göra förstudier av tilltänkta utvidgningar i till exempel Flames för att sedan, om resultaten verkar lovande, föra in funktionaliteten i den nya federationen.

4.3. Resultat

För att få en uppfattning om för och nackdelar med HLA kontra simuleringsramverk vid simulering av undervattensverksamhet har erfarenheterna från projekten SaSS-UV och UV-stridssimulatore för olika kriterier beskrivits i kapitel 3. Då de olika kriterierna är kvalitativa och inte helt lätt låter sig omsättas till siffervärden har ingen direkt jämförelse gjorts. Det skulle bli som att jämföra äpplen med päron. Dessutom är målsättningen, som tidigare nämnts, inte att framhäva någon arkitektur framför den andra, utan snarare att markera styrkor och svagheter för att underlätta för utvecklare att välja arkitektur baserat på de kriterier som är viktigast för dem.

Trots att vi medvetet har undvikit att kvantifiera våra kriterier, eftersom det då lätt kan uppstå missförstånd, förstår vi vikten av att ha en mer samlad bild av för- respektive nackdelarna mellan dessa två arkitekturer vid simuleringar av undervattensverksamhet. En liknande sammanställning av för- och nackdelar finns dock för alla typer av militära simuleringar i rapporten "Att hitta vägen i simuleringsrymden - En studie av parametrar och kriterier för utveckling av modeller och simuleringar i det svenska försvaret", [TRITA-NA-E02086]. Rapporten är resultatet av ett examensarbete som under hösten 2001 genomfördes på FOI i samarbete med Nada (Numerisk analys och datalogi), KTH. I sammanfattningen i rapporten nämns att vid modellering och simulering hos det svenska försvaret finns det främst två sätt som används. Det första är att göra distribuerade simuleringar, främst med hjälp av HLA, och det andra sättet är att använda sig av simuleringsramverk. Syftet med rapporten är att studera de standarder, tekniker, frågeställningar, konsekvenser och problemområden som finns hos de båda angreppssätten och hos modellering och simulering. Fördelar och nackdelar för varje angreppssätt diskuteras och några allmänna råd och riktlinjer ges.

Trots att slutsatserna i det ovannämnda examensarbetet gäller generellt för alla typer av simuleringar anser vi att de överensstämmer väldigt väl med våra slutsatser angående simuleringar av UV-verksamhet. Vi har därför valt att komplettera vår

rapport med ett utdrag ur den tidigare nämnda exjobbssrapporten, nämligen "*Chapter 8 - Conclusions and Discussion*" (se bilaga 1 - En sammanställning av för- och nackdelar för HLA respektive simuleringsramverk). Vi rekommenderar dock den intresserade att läsa hela exjobbssrapporten.

Bilaga 1 – En sammanställning av för- och nackdelar för HLA respektive simuleringsramverk

Utdrag ur rapporten "Finding One's way through Simulation Space -A Study of Methods and Parameters for Model and Simulation Development in the Swedish Defense", Marianela García Lozano, 2001.

Chapter 8 - Conclusions and Discussion

Summary of Observed Characteristics

When analyzing the advantages and disadvantages of using an HLA approach versus a simulation framework approach for every one of the application areas that were chosen to cover the ordinary spectrum of the military M&S areas and viewing the discussed parameters, a rather long list of observations was compiled. The following are the most prominent ones.

Simulation Framework Based Simulations

Of course, as there exist a great variety of simulation frameworks, it is difficult to settle generally valid characteristics. However, I think this set of observations are rather common. Let us first have a look at those qualities that are the strength of the framework concept.

- Simulation frameworks can significantly reduce the time required to build complex training and simulation systems if the framework's simulation space coincides with that of the project.
- A simulation framework optimizes the engineering resources since much of the repetitive and administrative tasks are already included, allowing the developers to concentrate on the modeling of the primary tasks.
- A simulation framework gives a rapid scenario and prototype creation and it is usually easy to enhance or replace the built-in models.
- The use and reuse of earlier and common models is easier since they are stored in a library. Examples could be sensors, terrain, weather, etc.
- Many models and other utilities come for free. They may not be needed for one application but can be used and added in a later stage if they are needed.
- The object oriented layout of most frameworks helps to maintain object orientation in the modeler's model code.
- Many simulation frameworks come with modules providing implementation support; thus contributing to consistent programs with fewer bugs.

- The simulation frameworks can be user friendly since there is the same environment for all simulations and the user always has the same well-known user interface.
- If the simulation framework is supported by a solid company that can guarantee future software maintenance and development it can help prevent simulations from becoming obsolete.

And then the disadvantages:

- Simulation frameworks may contain much more than is needed for a single application, scenario or simulation, which could be an efficiency problem.
- They can have a high learning threshold (especially for people with no computing science background).
- The simulation frameworks are often made with a specific purpose in mind and can thus lack in certain areas that may be just as important for the customer.
- Usually the models and programs are tightly coupled to the structure of the framework and are thus difficult to reuse and export to another system. For example, simulation frameworks contain a kernel that provides many useful but non-standardized services, this may put restrictions on what kind of models that can be incorporated or exported. This can make exchange with third party providers of models made in other systems impossible. Existing models might have to be rewritten entirely in order to be incorporated and used in a simulation framework.
- In some simulation frameworks the scenario creation support gives the users a high degree of freedom and as a consequence also a great responsibility. Some inconsistencies and errors may then pass undiscovered.
- Many times the GUIs are unintuitive and old fashioned.

There are some drawbacks that are related to the simulation frameworks as commercial products.

- Sometimes the documentation is not complete or up to date making it difficult to develop and use the simulation frameworks.
- Sometimes the simulation framework versions are not backwards compatible
- The simulation frameworks are usually costly, requiring expensive licenses and costly training of developers and users.
- The simulation framework's developers may disappear or decide to terminate the development process making the simulation framework gradually less usable.

HLA Based Simulations

The advantages of distributed simulations based on HLA are well known and appreciated.

- HLA is adopted as an international standard. It is promoted by the US DoD and is widely accepted by the defense M&S community which should also facilitate international cooperation. This should also improve the chances of finding suitable and reusable models but one has to keep in mind that it still needs extensive administration in order to be possible.
- The developers are encouraged to develop modularized programs that can make use of information hiding and have a clear separation between services and models.
- Due to the modularization of the applications it is rather easy to remove, add or change modules and also create custom applications (like GUIs) and update them when necessary. Besides, one doesn't always need to know the internal structure of the models in order to connect them to each other, it is sufficient with their interfaces.
- Reusing models is relatively easy. The models can be kept in a repository together with the SOM and additional documents describing what the models do (and perhaps even how they do it).
- It is easier to use models in a joint simulation that uses different time management methods and / or time step resolution.
- It is in most cases rather easy to adapt models to HLA, which makes it easy to integrate external simulations and third party software. It is also easy to modify the interface of an already HLA adapted model to a new application.

However, there are also some characteristics that could be a drawback in some situations.

- There is a high learning threshold to overcome.
- When models are loosely connected there is greater risk of having undetected fidelity discrepancies.
- Verification and validation can be difficult since models may participate in a simulation without revealing their inner structures.

There are some problems that are due to the fact that the RTI is a commercial product and that its implementation isn't standardized. These problems are somewhat related to some of the drawbacks that the simulation frameworks show in their capacity as commercial products. One of these difficulties is that it can be difficult to replace one RTI implementation from one vendor with another.

Other difficulties have to do with the distributed environment in itself and are therefore not exclusive of HLA based systems. They might in fact be true for simulation frameworks as well. I have chosen not to dwell on these problems since there already is extensive literature available on this subject but I will mention some issues that might be decisive.

- Security is more difficult when there is a network involved.

- Verification and validation can be more difficult due to the delay problems and other consequences that arise from using a network.
- It can be more difficult and time consuming to develop and run a distributed simulation. Especially if it also is geographically distributed.

Discussion

As we have seen there are advantages and disadvantages to both systems. There is no method that always gives us the answer to when one approach is better than the other. A primary result of the thesis work is that in most military application areas it would seldom be a very bad choice either you choose an HLA based solution or a solution based on a simulation framework. On the other hand the choice could be more or less convenient depending on the aspects that are the most decisive in the current modeling situation.

HLA does not specify anything about the development environment or the programming language, making it very flexible and versatile. Models and simulations are exportable.

The down side to HLA is the lack of support for development and running of models and simulations. Since everything usually must be done from scratch, unless there is a good repository of models available, there is always a certain amount of repetitive work and a high threshold to overcome before a simulation can be run. HLA in itself is complex and sometimes thought of as difficult to use for designing, running and testing programs. Some choose to use other systems due to difficulties they have had with HLA. Using HLA also requires a long time commitment since becoming familiar and efficient with it requires knowledge and experience.

The simulation frameworks are usually developed with a specifically targeted area in mind, for example air combat training or air combat analysis. The models and simulations are often deeply integrated into the framework making it difficult to export models and simulations or make them interact with simulations outside the simulation framework. Now many simulation frameworks come with interfaces to enable them to work with models compatible with HLA or even other simulation frameworks.

The good part about using simulation frameworks is that they are user friendly and it takes less time to develop and create a scenario and have a running simulation. They often come with a repository of working components and models that can be modified to suit the needs of the user. One does not need to be a skilled programmer to make a simulation.

Much of the future success of the two approaches lies in the availability of models and components. Having well organized and documented repositories is a key. The security and secrecy issues must be dealt with if it's ever going to be possible to have such repositories together with international cooperation.

The determining aspects that rule the choice of whether to use an HLA based or a simulation framework based simulation are as far as can be seen primarily the available time for the project, the need of scalability, verification and validation, the user friendliness, the effectiveness of the code, the security aspect, the reusability and the potential for cooperation with other organizations. Of all these factors especially the scalability and cooperation aspects tend to favor an HLA solution, while the effectiveness, security and to some extent the user friendliness aspects favor a simulation framework solution. The other aspects are too closely tied together with exterior factors or with the specific attributes of the simulation framework to be generally determining for one side or another.

The ideal would probably be to combine the two approaches, making it easier for models and simulations in simulation frameworks to interact with HLA federates than it is today.

My belief is that the simulation frameworks will adapt and become more compatible with the HLA standard. Already, we see these tendencies; there are some frameworks that try to draw on both approaches strength. These systems are still quite new and I haven't seen any evaluations from these frameworks. The future will reveal if that is the way to go. Either way, I still believe that in the future, simulation frameworks will be used to develop models and simulations and run them as federations and federates. Thus, combining the benefits of both systems, reuse, ease and speed of development with simulation frameworks and the distribution, standardization and interoperability possibilities of HLA.