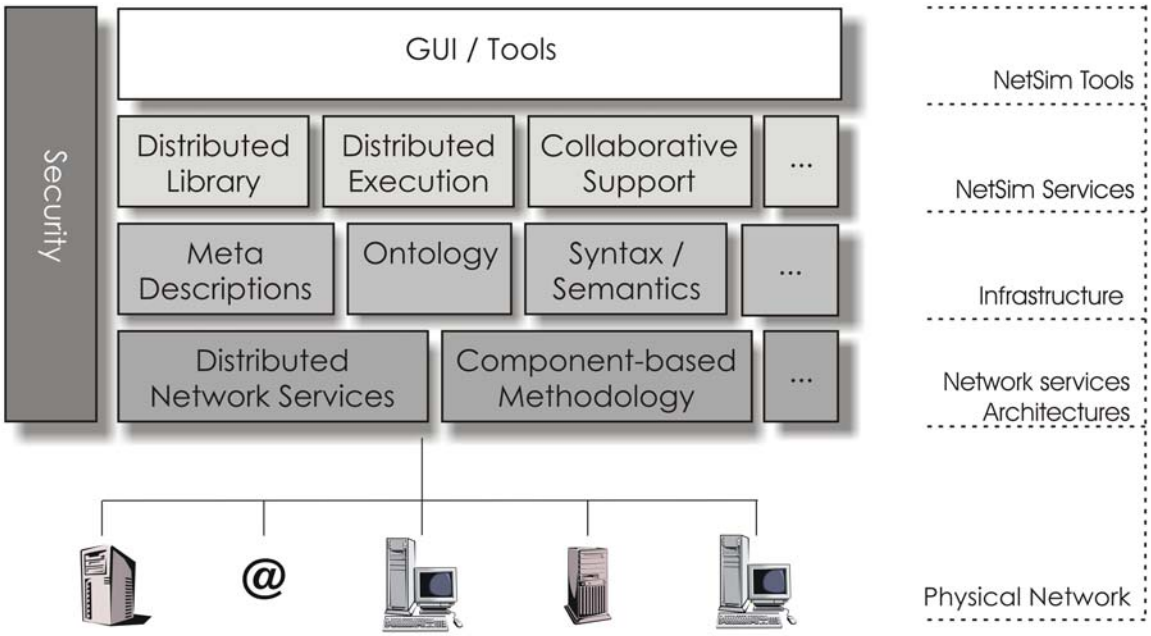
FOI-R--1439--SE
December 2004
ISSN 1650-1942

Metodrapport

Martin Eklöf, Marianela García Lozano, Farshad Moradi,
Dan Nordqvist, Magnus Sparf, Jenny Ulriksson

Nätverksbaserad Modellering och Simulering – Arkitekturen



Systemteknik
172 90 Stockholm

TOTALFÖRSVARETS FORSKNINGSINSTITUT

Systemteknik
172 90 Stockholm

FOI-R--1439--SE

December 2003

ISSN 1650-1942

Metodrapport

Martin Eklöf, Marianela García Lozano, Farshad Moradi,

Dan Nordqvist, Magnus Sparf, Jenny Ulriksson

Nätverksbaserad Modellering och Simulering – Arkitekturen

Utgivare Totalförsvarets Forskningsinstitut - FOI Systemteknik 172 90 Stockholm	**Rapportnummer, ISRN** FOI-R--1439--SE	**Klassificering** Metodrapport
	Forskningsområde 2. Operationsanalys, modellering och simulering	
	Månad, år December 2004	**Projektnummer** E6023
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 21 Modellering och simulering	

Författare/redaktör Martin Eklöf Marianela García Lozano Farshad Moradi Dan Nordqvist Magnus Sparf Jenny Ulriksson	**Projektledare** Farshad Moradi **Godkänd av** **Uppdragsgivare/kundbeteckning** FM **Tekniskt och/eller vetenskapligt ansvarig**

Rapportens titel

Nätverksbaserad Modellering och Simulering – Arkitekturen

Sammanfattning (högst 200 ord)

Utveckling och användning av webb- och nätverksbaserade system har under det senaste decenniet ökat betydligt. Exempel på detta är det senaste paradigmskiftet från plattformsbaserat till nätverksbaserat försvar och utvecklingen av Internet-teknologier. Även MoS-området har följt denna utveckling. På Försvarsmaktens uppdrag initierades 2004 ett projekt för att undersöka och studera området Nätverksbaserad Modellering och Simulering (NBMS), samt identifiera vilka möjligheter som detta kan erbjuda försvaret. För att experimentera med framtagna produkter och tekniker, samt visa på fördelarna med dessa, utvecklas en prototyp till en NBMS-miljö som stödjer aktiviteter inom MoS-processen. Projektet, som förkortas NätSim, berör många delområden såsom: Datorbaserad samverkan, Distribuerad exekvering, Distribuerade filsystem, Säkerhetsfrågor inom NBMS mm.

Projektets första år har inriktats mot att utforma en gedigen arkitektur för NBMS, vilket resulterade i en flerlagerarkitektur med modulär uppbyggnad. Förutom detta har delområdena behandlats, t.ex. har en infrastruktur för säkerhet utformats, en grund för distribuerad exekvering och hantering av simuleringskomponenter har byggts, implementering av en infrastruktur för datorbaserad samverkan har initierats, samt har en bibliotheksstruktur med sökprocesser och resurshantering designats. Projektarbetet utförs dessutom med koppling till interna FOI-projekt och externa universitet m.fl. för att bl.a. få återkoppling.

Denna rapport presenterar projektet, projektets verksamhet hittills, samt framtida inriktning och viktiga frågeställningar som måste beaktas.

Nyckelord

Modellering och simulering, HLA, komponentbaserad M&S, Distribuerad M&S, Datorbaserad samverkan, CSCW, Distribuerad exekvering, Distribuerad resurshantering, Säkerhet

Övriga bibliografiska uppgifter	**Språk** Svenska
ISSN 1650-1942	**Antal sidor:** 75 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Systems Technology SE-172 90 Stockholm	FOI-R--1439--SE	Method report
	Research area code 2. Operational Research, Modelling and Simulation	
	Month year December 2004	**Project no.** E6039
	Customers code 5. Commissioned Research	
	Sub area code 21 Modelling and Simulation	

Author/s (editor/s)	Project manager
Martin Eklöf Marianela García Lozano Farshad Moradi Dan Nordqvist Magnus Sparf Jenny Ulriksson	Farshad Moradi **Approved by** **Sponsoring agency** Swedish Armed Forces **Scientifically and technically responsible**

Report title (In translation)

Network Based Modelling and Simulation – The Architecture

Abstract (not more than 200 words)

The development and use of web and network based systems have increased significantly during the last decade. An example of this is the change from a platform based to a network based defence, and the development of Internet technologies. The M&S area is no exception for these changes. In the spring of 2004, commissioned by the Swedish Armed Forces, a project was initiated with the aim to study the area of Network Based Modeling and Simulation (NBMS), and also identify what possibilities this area could offer the defence. To experiment with developed products and techniques, and to demonstrate the advantages, a prototype environment for NBMS is being developed that supports the activities of the M&S process. The project, called NetSim, concerns many subareas such as: Computer-based cooperation, Distributed execution, Distributed file systems, and Security issues within NBMS.
The project's first year has been focused on designing a stable architecture for NBMS, and resulted in a layered architecture with a modular structure. Apart from this, the subareas have been handled, for example an infrastructure for security has been designed, a foundation for distributed execution and management of simulation components has been constructed, implementation of an infrastructure for computer-based cooperation has been initiated, and a library structure including search processes and resource management has been formed. The project also has connections to FOI projects and universities etc. to receive feedback and input.
This report presents the project, the project activities so far, and future directions and important research questions at issue that will be considered.

Keywords

Modelling och simulation, HLA, Component-based M&S, Distributed M&S, Computer-based collaboration, CSCW, Distributed execution, Distributed resource management, Security

Further bibliographic information	Language Swedish
ISSN 1650-1942	**Pages** 75 p.
	Price acc. to pricelist

Innehållsförteckning

Sammanfattning

Utveckling och användning av webb- och nätverksbaserade system har under det senaste decenniet ökat betydligt. Flera trender visar på denna utveckling och betydelsen av nätverk och nätverksbaserad kommunikation inom framtida försvarssystem. Bl.a. kan nämnas det senaste paradigmskiftet från plattformsbaserat till nätverksbaserat försvar och utvecklingen av Internet-relaterade teknologier. Området MoS har också följt samma utveckling, såväl inom den civila som den militära sektorn. Genom introduktionen av SimNet, DIS och HLA har en ny generation av MoS-arkitekturer trätt fram som kännetecknas av en komponentbaserad utvecklings- och exekveringsmetodik med fristående komponenter som kommunicerar via nätverk. Dessa MoS-arkitekturer föranleder en utveckling av en gemensam plattform för utveckling och utnyttjande av simuleringsmodeller.

Området nätverksbaserad modellering och simulering (NBMS) omfattas av olika aspekter relaterade till kombinationen av nätverk och MoS. På Försvarsmaktens uppdrag, och med motivering i behovet som presenteras ovan, initierades våren 2004 projektet ”Nätverksbaserad Modellering och Simulering”, förkortat NätSim. Syftet med projektet är att undersöka och studera området NBMS, samt identifiera vilka möjligheter som detta kan erbjuda försvaret. Projektet är framförallt inriktat på att ta fram en generell arkitektur för NBMS med målet att studera och forska inom den nya generationen av MoS-arkitekturer. Förutom detta utvecklas metoder och infrastruktur för att bygga en tjänstebaserad miljö för utveckling och exekvering av nätverksbaserade simuleringsmodeller. För att experimentera med framtagna produkter, tekniker och verktyg, samt visa på fördelarna med att använda dessa, utvecklas en prototyp till en nätverksbaserad utvecklings- och exekveringsmiljö för NBMS. Miljön är tänkt att vara en infrastruktur, som kan stödja olika aktiviteter inom hela MoS-processen. En sådan miljö kan stödja simuleringsmodeller under hela deras livslängd, från kravspecifikation, konceptuell modellering och design, till utveckling, exekvering och analys. Ett effektmål har också varit att identifiera generella metoder för hur nätverksbaserad MoS kan stödja olika verksamheter på olika nivåer inom FM.

Arbetet har främst bedrivits i två spår, ett för forskning och ett för utveckling. I forskningsspåret identifieras, anpassas och utvecklas lämpliga metoder, tekniker och verktyg, i syfte att undersöka fördelarna med att kombinera Modellering och Simulering (MoS) med nätverks- och webbteknologier och förstå hur dessa påverkar (förbättrar/förenklar) våra sätt att modellera och simulera. Mer specifikt undersöks delar såsom: ramverk för beskrivning av resurser, samverkansformer för utveckling och exekvering av simuleringsmodeller, distribuerad exekvering av modeller, distribuerade databaser och filsystem, komponentbaserad modellutveckling, sammankoppling av olika arkitekturer för distribuerad simulering, samt säkerhetsfrågor inom NBMS. Under utvecklingsspåret utvecklas en prototyp till en nätverksbaserad utvecklings- och exekveringsmiljö med målet att testa olika metoder, tekniker och verktyg. Ett annat syfte med prototypmiljön är att kunna redovisa framtagna forskningsresultat och demonstrera idéer och förslag.

Arbetsspåren har olika tidsperspektiv. Forskningen har givetvis ett längre tidsperspektiv och berör inte endast dagens teknik, utan även framtida möjligheter. Utvecklingsarbetet är dock framförallt baserat på dagens teknik, men miljön skall även kunna utnyttjas under en längre tid för att testa och pröva olika tekniker, även sådana som inte existerar idag. För att på ett effektivt sätt kunna bedriva dessa aktiviteter parallellt, behövs en robust och skalbar struktur/arkitektur för miljön tas fram och det är det arbetet som projektet har fokuserats kring första året.

Årets arbete har främst inriktats mot att utforma en gedigen, komponentbaserad arkitektur som ska kunna användas för att realisera NätSim-plattformen. Resultatet blev en flerlagerarkitektur med modulär uppbyggnad. Dessutom har arbete ägnats åt kunskapsspridning och presentationer inom NBMS och projektet, dels för att sprida kunskap om det nya området och skapa nya kontakter, och dels även få återkoppling på det arbete som utförs. Vidare har aktiviteter i de separata delområdena i projektet fortsatt. Då säkerhet har setts som en viktig fråga har en infrastruktur för detta och specifikt för plattformen utvecklats konceptuellt. En grund har lagts för automatisk distribuerad exekvering och hantering av simuleringskomponenter och beräkningsresurser (DRMS), som är tjänstebaserad (*Web services*) och där metadatamodellerna baseras på OWL. Vidare har en första implementering av en infrastruktur för datorbaserad samverkan implementerats som en modul (CC) baserad på HLA och XML. För båda aktiviteterna har noggrann analys av vilka tjänster och krav som är gällande utförts.

Detta har även gjorts för biblioteksfunktionen, och en *Grid*-baserad (*Grid* är en distribuerad nätverksteknologi) prototyp av ett mellanlager för hantering av sökfunktioner mm. för NätSim har börjat implementeras.

Årets arbete har ytterligare bekräftat behovet av en försvarsgemensam miljö för MoS-verksamhet. Den arkitektur som har utformats under året för detta syfte är tjänstebaserad och modulär, något som har visat sig positivt på många sätt. Därför fortsätter implementeringen av de separata projektaktiviteterna i denna riktning, varefter vi sätter ihop delarna och utvärderar dem. Vi fortsätter även samarbetet och kopplingen mot andra relevanta projekt och verksamheter. Plattformen är tänkt att utnyttjas av andra projekt för att dels tydliggöra kundnyttan samt även få återkoppling på vårt arbete.

Dessutom har årets arbete resulterat i beslut om framtida arbete samt identifierade frågeställningar för fortsatt forskning och utveckling. Vad gäller biblioteksfunktionen kommer prototypen för söknings-processer samt ett lager inkluderande ontologi att färdigställas. I arbetet insågs att det intressanta att undersöka är just sökfunktion, matchning och representation av resurser, snarare än att utveckla en teknisk lösning för biblioteksfunktionen. För DRMS fortsätter implementeringen av tjänsterna, men arbetet kommer att inriktas mot att bl.a. effektivisera hanteringen av feltolerans i systemet, samt undersökning av metoder för lastbalansering. Arbetet med CC kommer att fortsätta med kompletterande implementering av modulen och mot att göra den mer generaliserad. Forskningsfrågor som har identifierats under årets arbete kommer att hanteras närmare, såsom dynamiska synkroniseringsmekanismer för datorbaserad samverkan. Säkerhet kommer att vara en viktig del även i nästkommande års arbete och kommer att tillägnas mer tid framöver. Generellt är säkerhet i dessa typer av system, såsom distribuerade försvarssystem, en viktig och komplex fråga. Implementering av de framtagna principerna för säkerhet i miljön kommer att initieras och undersökning av identifierade frågor såsom säkerhetsaspekter för metadata och kryptering av trafik kommer att fortsätta. För själva NätSim-miljön är förhoppningen att NetScene – ett FOI-internt utvecklat verktyg vid avdelningen för Ledningssystem – skall vidareutvecklas i syfte att användas dels som basapplikation för NätSim och dels som scenarioverktyg för MoS i plattformen.

Akronymlista

ACL	*Access Control List*
AES	*Asymmetric Encryption Standard*
AFS	*Andrew File System*
API	*Application Programmning Interface*
CA	*Certificate Authorization*
CC	*Collaborative Core*
CHI	*Computer-Human Interaction*
CORBA	*Common Object Request Broker Architecture*
CSCW	*Computer Supported Collaborative (alt. Cooperative) Work*
DCOM	*Distributed Component Object Model*
DHT	Distribuerade Hashtabeller
DIS	*Distributed Interactive Simulation*
DRMS	*Distributed Resource Management System*
DSA	*Digital Signature Algorithm*
FEDEP	*Federation Development and Execution Process*
FM	*Försvarsmakten*
FW	*Firewall*
GAS	*(Terraplay) Game Access Server*
HLA	*High Level Architecture*
HTTP	*HyperText Transfer Protocol*
HTTPS	*HTTP Secure*
IASTED	*the International Association of Science & Technology for Development*
IDS	*Intrusion Detection System*
IEEE	*the Institute of Electrical and Electronics Engineers*
IMIT	Mikroelektronik och Informationsteknologi
IP	*Internet Protocol*
IPS	*Intrusion Prevention System*
IPSec	*IP Securiy, VPN implementation*
JVM	*Java Virtual Machine*
KTH	Kungliga Tekniska Högskolan
MIT	*Massachusetts Institute of Technology*
MoS	Modellering och Simulering
MOSART	*Modelling and Simulation for Analysis and Research Test-bed*

NBF	Nätverksbaserat Försvar
NBMS	Nätverksbaserad Modellering och Simulering
NUS	*National University of Singapore*
OCL	*Object Constraint Language*
OGSA-DAI	*Open Grid Services Architecture data Access and Integration*
OWL	*Web Ontology Language*
PDA	*Personal Digital Assistant*
PDCS	*Parallel & Distributed Computing & Systems*)
PKI	*Public Key Infrastructure*
P2P	*Peer to Peer*
QoS	*Quality of Service*
RDF	*Resource Description Framework*
RLS	*Replica Location Service*
RSA	*Rivest, Shamir and Adelman, public-key encryption*
RPC	*Remote Procedure Call*
RTI	*Runtime Infrastructure*
SAML	*Security Assertion Markup Language*
SBM	Simuleringsbaserad Materielförsörjning
SCS	*The Society for Modelling & Simulation International*
SHA-1	*Secure Hash Algorithm 1*
SimNet	*Simulator Networking*
SOA	*Service-Oriented Architecture*
SSL	*Secure Socket Layer*
SSO	*Single Sign-On*
TCP	*Transmission Control Protocol*
XACML	*Extensible/XML Access Control Markup Language*
XKMS	*XML Key Management Specification*
XMI	*XML Model Interchange Format*
XML	*Extensible Markup Language*
XVCL	*XML-based Variant Configuration Language*
SLA	*Service Level Agreement*
TS	*Terraplay System*
UDP	*User Diagram Protocol*
UML	*Unified Modelling Language*

UoH	Universitet och Högskola
URI	*Uniform Resource Identifier*
VPN	*Virtual Private Network*
VV&A	Verifiering, Validering och Ackreditering
VO	Virtuell Organisation
WS	*Web Services*
WSRF	*Web Services Resource Framework*
W3C	*World Wide Web Consortium*

1 Introduktion

1.1 Bakgrund

Nätverksbaserad modellering och simulering (NBMS) är en gemensam benämning på ett antal områden inom domänen för Modellering och Simulering (MoS), som visar på olika aspekter av kombinationen av nätverk och MoS. På Försvarsmaktens uppdrag initierades våren 2004 projektet ”Nätverksbaserad Modellering och Simulering”, förkortat NätSim, med målet att undersöka och studera området, samt identifiera vilka möjligheter som detta kan erbjuda försvaret. Projektet förväntas fortlöpa under totalt tre år, och ska framförallt vara inriktat på att ta fram en generell arkitektur för NBMS. Arbetet genomförs med en fokusering på komponentbaserad modellutveckling, datorbaserad samverkan för modellutveckling och exekvering, samt ett gemensamt ramverk för beskrivning och exekvering av modeller, applikationer, verktyg, etc. Samtidigt arbetar projektet med att sprida kunskap om området och skapa en gedigen grund för fortsatt arbete inom NBMS genom seminarier, föredrag (egna och inbjudna), deltagande i konferenser, samt samarbeten med KTH och SU och internationella organisationer. Denna rapport presenterar resultatet från det första årets arbete, samt diskussioner och visioner för framtida verksamhet.

1.2 Mål

Utveckling och användning av webb- och nätverksbaserade system har under det senaste decenniet ökat betydligt och dessa system används allt oftare inom olika samhällssektorer, som till exempel Försvarsmakten (FM). Flera parametrar visar på denna utveckling och betydelsen av nätverk och nätverksbaserad kommunikation inom framtida försvarssystem. Bl.a. kan nämnas det senaste paradigmskiftet från ett plattformsbaserat till ett nätverksbaserat försvar och utvecklingen av Internet-relaterade teknologier. Området MoS har också följt samma utveckling inom såväl den civila som den militära sektorn. Genom introduktionen av SimNet *(Simulator Networkin*g), DIS *(Distributed Interactive Simulatio*n) och HLA *(High Level Architecture)* presenteras en ny generation av MoS-arkitekturer. Dessa kännetecknas av en komponentbaserad utvecklings- och exekveringsmetodik med komponenter som kommunicerar via nätverk.

Målet med projektet är således att studera och forska inom den nya generationen av MoS-arkitekturer och utveckla metoder och infrastruktur för att bygga en tjänstebaserad miljö för utveckling och exekvering av nätverksbaserade simuleringsmodeller. Detta innebär att projektet siktar på att identifiera, anpassa och utveckla lämpliga metoder, tekniker och verktyg för att utveckla en tjänstebaserad arkitektur, som används till design, utveckling, dokumentation, exekvering och analys av nätverksbaserade simuleringsmodeller. För att experimentera med framtagna produkter, tekniker och verktyg, samt visa på fördelarna med att använda dessa, utvecklas en prototyp till en webbaserad utvecklings- och exekveringsmiljö för NBMS. Miljön, fortsättningsvis omnämnd som NätSim-miljön, är tänkt att vara en infrastruktur som kan stödja processer som FEDEP[1] *(Federation Development Proces*s), SBM *(*Simuleringsbaserad Materielförsörjning) och VV&A *(*Verifiering, Validering och Ackreditering). En sådan miljö kan stödja simuleringsmodeller under hela livslängden, från kravspecifikation, konceptuell modellering och design, till utveckling, exekvering och analys.

Med tanke på att detta projekt behandlar ett brett och stort område har även ett mål varit att utnyttja det arbete som har gjorts inom området både nationellt och internationellt, genom att etablera kontakter, samarbeta med andra aktörer för att undvika dubbelarbete, samt sprida kunskap om vår verksamhet. Detta har gjorts genom att skriva artiklar, hålla föreläsningar, delta i konferenser och engagera olika parter för att bygga upp ett nätverk av kompetenta personer som arbetar inom området. Samtidigt har en teknisk infrastruktur upprättats i form av ett nätverkslabb för att experimentellt forska kring och studera nätverksbaserad modellering och simulering. Ett effektmål har också varit att identifiera generella metoder för hur nätverksbaserad MoS kan stödja olika verksamheter på olika nivåer inom FM.

---

[1] FEDEP står för *Federation Development and Execution Process* och är är en process med ett antal grundläggande steg för utveckling av HLA-federater och -federationer.

1.3 Nyttan för FM

Konceptet NBF (det Nätverksbaserade Försvaret), med egenskaper som behovssammansatta system och hög tillgång till information, erbjuder bl.a. större flexibilitet, effektivitet, lägre kostnader och bättre informationsöverlägsenhet, men leder samtidigt till en ökad komplexitet. Idag är vi tyvärr långt ifrån att ha en klar bild av vad NBF innebär och hur det skall förverkligas. MoS och i synnerhet NBMS är ett kraftfullt verktyg (om inte det enda sättet) för att kunna tydliggöra denna bild och bana vägen för ett förverkligande av konceptet. Många av de metoder och erfarenheter som byggs inom NBMS kan direkt, i vissa fall med viss anpassning, användas för framtagning av olika delar av det nätverksbaserade försvaret.

Idag ser vi dessutom en utveckling inom försvaret som tyder på att FM kommer att använda simuleringsmodeller som informationsbärare genom hela Försvarsmaktsutvecklingen, från behovsdefiniering till skarp insats på förbandsnivå. Denna utveckling ställer krav på livscykeln för simuleringsmodeller, dvs. för såväl systemering, realisering och implementering, som på förvaltning, drift och avveckling. NätSim-plattformen erbjuder en tjänstebaserad arkitektur som kan stödja nätverksbaserade simuleringsmodeller under olika faser i livscykeln.

Genom att utveckla ett system (en miljö) för utveckling av nätverksbaserade simuleringsmodeller, skapar vi också en plattform för att undersöka möjligheterna att utnyttja och utveckla den senaste metodiken och tekniken inom NBMS. Metoder som har identifierats, anpassats eller framtagits torde dock vara generella nog för att kunna utnyttjas vid utveckling av försvarssystem på alla nivåer, och inom olika tillämpningsområden.

Projektet beaktar dessutom säkerhetsfrågor inom NBMS, genom att utföra en säkerhetsanalys och utveckla en säkerhetsarkitektur för plattformen. Säkerhetsarkitekturen kan med fördel anpassas och användas även vid utveckling av andra nätverksbaserade system, såsom t.ex. NBF.

Fördelarna med nätverksbaserad MoS kan t.ex. vara:

- Återanvändbarhet av simuleringsmodeller
- Ökad tillgänglighet av modeller
- Effektivare och snabbare utveckling av simuleringsmodeller (genom att utnyttja komponentbaserad utvecklingsteknik)
- Ökat samarbete (mellan utvecklare och användare) genom samordnad utveckling, dokumentation och exekvering av modeller via nätet
- Bättre stöd för träning och utbildning
- Skalbarhet och ökad datorkraft
- Feltolerans
- Effektivt utnyttjande av befintliga datorresurser
- Säkrare utveckling och exekvering av modeller

1.4 Genomförande

Arbetet inom projektet bedrivs i två parallella spår, forskning resp. prototyputveckling.

Forskningsspåret

I forskningsspåret identifieras, anpassas och utvecklas lämpliga metoder, tekniker och verktyg, i syfte att undersöka fördelarna med att kombinera Modellering och Simulering (MoS) med nätverks- och webbteknologier och förstå hur dessa påverkar (förbättrar/förenklar) våra sätt att modellera och simulera. Följande områden har varit och är av störst intresse under detta spår:

- Metamodellering: ett ramverk för beskrivning av resurser (t.ex. simuleringsmodeller, verktyg, datorer, applikationer, etc.).
- Samverkansformer för utveckling och exekvering av simuleringsmodeller (t.ex. metoder och tekniker för distribuerad CSCW för heterogena klienter).
- Distribuerad exekvering av modeller (exempel på intressanta frågor är: effektiv migrering av simuleringsmodeller, feltoleranta exekveringar och lastbalansering).
- Distribuerade databaser och filsystem (t.ex. metoder för lagring och sökning av simuleringsmodeller).
- Komponentbaserad modellutveckling (återanvändbarhet och interoperabilitet). Hur man kan åstadkomma "*composability*" på olika nivåer (syntaktisk till semantisk).
- Samverkan mellan och koppling av olika arkitekturer för distribuerad simulering.
- Säkerhetsfrågor inom NBMS.

Arbetet inom detta spår har bl.a. resulterat i en informationsmodell för NBMS och NätSim-plattformen. Vi har också identifierat och anpassat ett antal metoder och tekniker för NätSim-miljön som redovisas i senare avsnitt.

Utvecklingsspåret

Under utvecklingsspåret utvecklas en prototyp till en nätverksbaserad utvecklings- och exekveringsmiljö med målet att testa olika metoder, tekniker och verktyg, både existerade och sådana som har tagits fram eller anpassats inom projektet. Ett annat mål med miljön är att kunna redovisa framtagna forskningsresultat och demonstrera idéer och förslag.

Dessa spår, dvs. forskningen inom nätverksbaserad MoS och prototyputvecklingen har olika tidsperspektiv. Forskningen har givetvis ett längre tidsperspektiv och berör inte endast dagens teknik, utan även framtida möjligheter. Utvecklingsarbetet är dock framförallt baserad på dagens teknik, men miljön skall även kunna utnyttjas under en längre tid för att testa och pröva olika tekniker, även sådana som inte existerar idag. För att på ett effektivt sätt kunna bedriva dessa aktiviteter parallellt, behöver en robust och skalbar struktur/arkitektur för miljön att tas fram.

Under året har vi utfört analyser och undersökning av miljön baserad på ett antal tänkbara användningsområden. Resultatet har använts för att designa en skalbar och robust arkitektur, vilken redovisas i kapitel 2.

Arbetsgrupper

Projektarbetet har under året delats i sju arbetspaket som har haft nära samarbete:

- Arkitektur och design
- Distribuerad resurshantering
  - Distribuerad exekvering av modeller
  - Distribuerat resursbibiliotek
- Samverkansformer för MoS
- Komponentbaserad modellutveckling
- Samverkan mellan olika arkitekturer för distribuerad simulering
- Säkerhetsaspekter av NBMS

För att få stöd med arbetet inom de olika grupperna, samt öka vårt samarbete med UoH har under året ett antal examensarbetare engagerats. Resultatet från olika arbetsgrupper redovisas i senare avsnitt.

För att kvalitetssäkra resultaten inom projektet läggs dessutom arbete på att lämna bidrag till konferenser och tidskrifter, föreläsa om projektet, samt samarbeta med andra aktörer inom och utanför FOI. Därigenom erhålles (från andra forskare) en värdefull granskning av våra metoder och resultat.

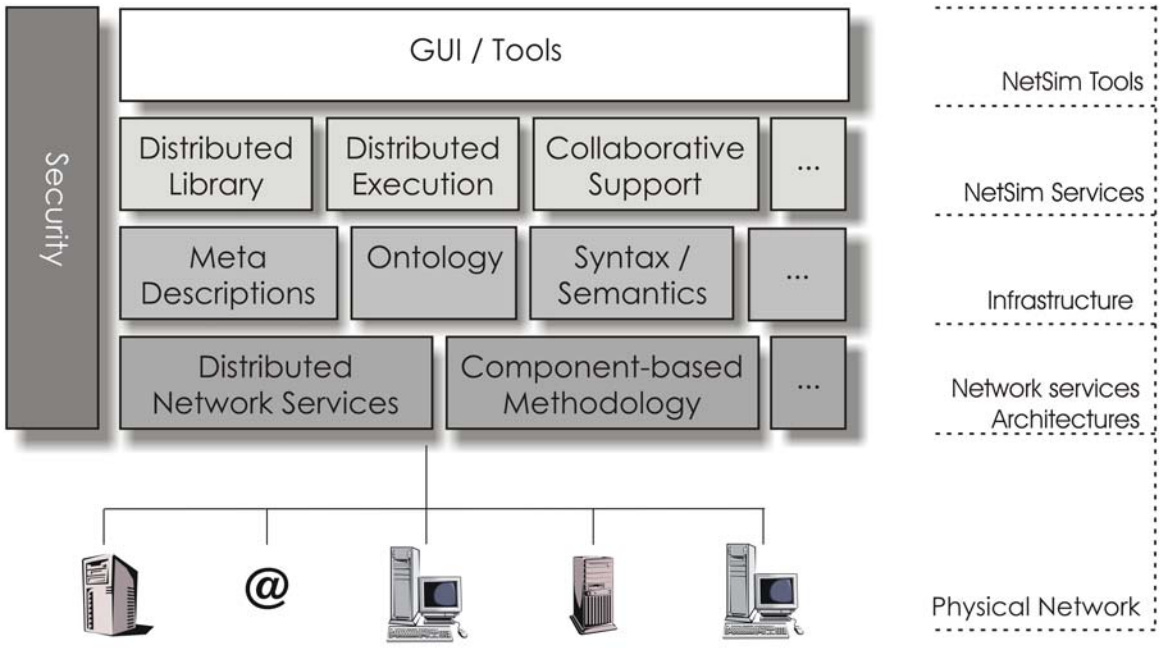
1.5 Rapportens struktur

I kapitel 2 beskrivs arkitekturen för NätSim-miljön. Här ges en beskrivning av hur arkitekturen ser ut med de ingående lagren, utan någon fördjupning i tekniska detaljer. Kapitel 3 redovisar den tekniska plattformen genom att förklara de tekniker, metoder och verktyg som har valts eller utnyttjats för att realisera de olika lagren inom arkitekturen. Vi beskriver ansatser för att kvalitetssäkra arbetet och våra resultat i kapitel 4. I kapitel 5 redovisas våra slutsatser efter årets arbete och beskrivs vår framtidsvision och fortsatt arbete.

2 Arkitektur för nätverksbaserad modellering och simulering

Utveckling och användning av webb- och nätverksbaserade system har under det senaste decenniet ökat betydligt. Flera trender visar på denna utveckling och betydelsen av nätverk och nätverksbaserad kommunikation inom framtida försvarssystem. Bl.a. kan nämnas det senaste paradigmskiftet från plattformsbaserat till nätverksbaserat försvar och utvecklingen av Internet-relaterade teknologier. Området MoS har också följt samma utveckling, såväl inom den civila som det militära sektorn. Genom introduktionen av SimNet, DIS och HLA har en ny generation av MoS-arkitekturer presenterats som kännetecknas av en komponentbaserad utvecklings- och exekveringsmetodik med fristående komponenter som kommunicerar via nätverk.

Dessa MoS-arkitekturer föranleder utveckling av ett gemensamt beskrivningsramverk för utveckling och utnyttjande av simuleringsmodeller. Ramverkets syfte är att bl.a. möjliggöra återanvändbarhet, interoperabilitet och komponentbaserad utveckling av simuleringsmodeller. För att ramverket skall kunna stödja simuleringsmodeller under deras livscykel på ett optimalt sätt bör det ha en tjänstebaserad struktur. Som nämndes i Avsnitt 1.4 arbetade projektet under 2004 med att designa och utveckla en plattform som baserades på ett sådant ramverk och en sådan struktur. Plattformen skulle dessutom vara tillräckligt generell för att kunna utnyttjas under en längre tid för att testa och pröva olika tekniker, även sådana som inte existerar idag. Resultatet blev en flerlagerarkitektur med modulär uppbyggnad, enligt Figur 2.1. Fördelen med denna arkitektur är att den är skalbar och att moduler som är baserade på olika teknologier kan ersätta varandra utan att strukturen i sin helhet behöver ändras eller bytas ut. En annan fördel med detta tillvägagångssätt är att olika tekniker kan utnyttjas inom de olika lagren. Därmed fås tillgång till en mer flexibel miljö och är inte begränsade till en enda teknik.



Figur 2.1: Schematisk bild av NätSim-arkitekturen och dess olika lager.

2.1 Nätverkslager

Det understa lagret i arkitekturen består av olika nätverksbaserade teknologier som möjliggör utbyte av information och tjänster mellan distribuerade simuleringar, spel, applikationer och andra nätverksresurser. Dessa teknologier kan delas in i arkitekturer för komponentbaserad utveckling/exekvering av simuleringsmodeller (t.ex. HLA, DIS, Terraplay, etc.) och distribuerade nätverkstjänster som webbtjänster, *Grids* och *Peer-to-Peer* (P2P).

2.1.1 Komponentbaserade arkitekturer

2.1.1.1 Composability

Med komponent menas här en självständig bit mjukvara, t.ex. simulering, som kan distribueras och användas i andra applikationer än den ursprungligen utvecklades för. Genom att sätta samman mindre och mer flexibla/generella simuleringskomponenter i olika konstellationer skapas mer komplexa simuleringsmodeller. De främsta fördelarna med komponentbaserad utvecklingsmetodik är återanvändbarhet av simuleringskomponenter, kortare utvecklingstider och distribution av exekveringsresurser. Detta kräver dock att komponenterna är baserade på samma standard så att de kan kopplas samman.

Under de senaste åren har det utvecklats flera standarder för komponentbaserade arkitekturer inom både den civila och militära sfären, som skall stödja och underlätta att modeller av skilda slag och lokaliserade på vitt skilda datorer skall kunna samverka i gemensamma simuleringar. Dessa arkitekturer har utvecklats och anpassats för olika syften och olika domäner. Idag existerar ingen arkitektur som kan tillgodose alla de krav och behov som de olika användningsområdena har. Under det gångna året har projektet tittat närmare på en teknik som skulle kunna användas som grund för en infrastruktur som kan hantera dessa krav (se Avsnitt 2.1.1.2 nedan).

HLA är ett exempel på den senaste i raden av arkitekturer som har framtagits inom den militära domänen. HLA möjliggör sammankoppling av modeller, system och andra komponenter till ett större simuleringssystem. Det är utvecklat i syfte att tillåta återanvändning av komponenter oavsett vilket programspråk de är utvecklade i eller vilken plattform de körs på. Alla ingående komponenter i det större simuleringssystemet kallas för federater och samlingen av federater kallas för en federation i HLA. Finessen med HLA är möjligheten till återanvändning av komponenter i andra samanhang än de ursprungligen var utvecklade för. En civil motsvarighet till en sådan arkitektur *Terraplay system* (TS) som har utvecklats av företaget Terraplay och är en generisk infrastruktur för spel med många deltagare. TS erbjuder en infrastruktur där olika typer av noder, såväl stationära och mobila, kan koppla sig och spela mot varandra.

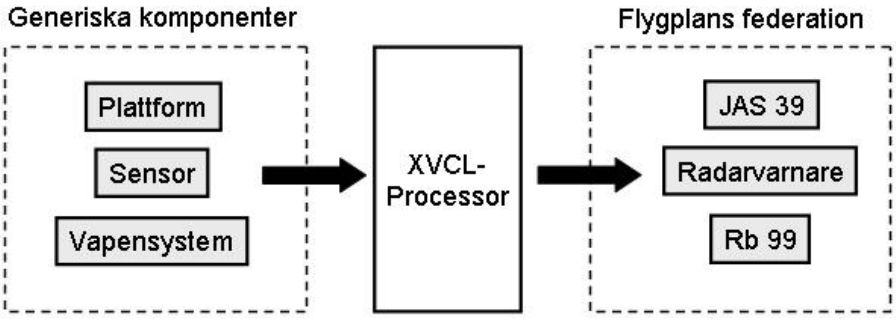
Ett nyckelord inom komponentbaserad modellutveckling är *composability*. *Composability* är förmågan att kunna sätta samman komponenter och utveckla sammansatta simuleringar. För att kunna utföra detta krävs det dock att komponenterna är interoperabla. Med interoperabilitet mellan komponenter (simuleringsmodeller) menas att komponenterna skall kunna kommunicera och utbyta information på ett meningsfullt sätt för att uppnå ett bestämt syfte. Vad som erbjuds idag är framförallt interoperabilitet på syntaktisk nivå, vilket betyder att komponenter via givna gränssnitt kan kommunicera och skicka information till varandra. Det finns dock inget enhetligt sätt att åstadkomma interoperabilitet och förståelse på semantisk nivå, egenskaper som innebär att komponenterna inte endast tar emot informationen utan även kan tolka och uppfatta den på rätt sätt.

Ett mål med NätSim är att erbjuda en infrastruktur för komponentbaserad utveckling/exekvering av simuleringsmodeller som stödjer olika komponentbaserade arkitekturer. Detta stöd utformas i två faser. I den första fasen skall komponenter som tillhör samma arkitektur kunna kopplas samman och exekveras i den underliggande körtidsmiljön. I den andra fasen skall modeller som tillhör olika arkitekturer sättas samman. För båda faserna krävs det detaljerade beskrivningar av simuleringskomponenter, inte enbart på syntaktisk nivå utan även på semantisk nivå för sammansättning av simuleringskomponenter.

2.1.1.2 XVCL

XVCL (*XML-based Variant Configuration Language*) är en metaprogrammeringsteknik som utnyttjar regelsystem för att konfigurera generiska komponenter [webb: XVCL]. Konfigureringen resulterar i specifika komponenter som är anpassade för en specifik applikation. På detta sätt kan krav från skilda miljöer, eller applikationer, påverka de generiska komponenterna så att dessa kan återanvändas. I XVCL representeras en generisk komponent av ett XML-dokument, innehållande källkod och XVCL-specifika instruktioner. En mängd sådana dokument bearbetas av en XVCL-processor som konfigurerar och sammankopplar komponenterna till ett nytt system (applikation).

Inom NätSim undersöks för närvarande nyttan med integrering av HLA och XVCL. Mer specifikt utreds möjligheterna till utveckling av generiska federater (kodskelett) som anpassas till en specifik federation. Detta kommer att underlätta återanvändning av HLA-specifik kod, samt dölja HLA-specifik logik för federationsutvecklaren. Tanken med detta arbete är att möjliggöra utveckling av federater/federationer i ett ramverk där användaren konceptuellt definierar informationsutbytet mellan entiteter. Utifrån den konceptuella modellen genereras federater, utifrån generiska komponenter, genom bearbetning av en XVCL-processor, se Figur 2.2 nedan.



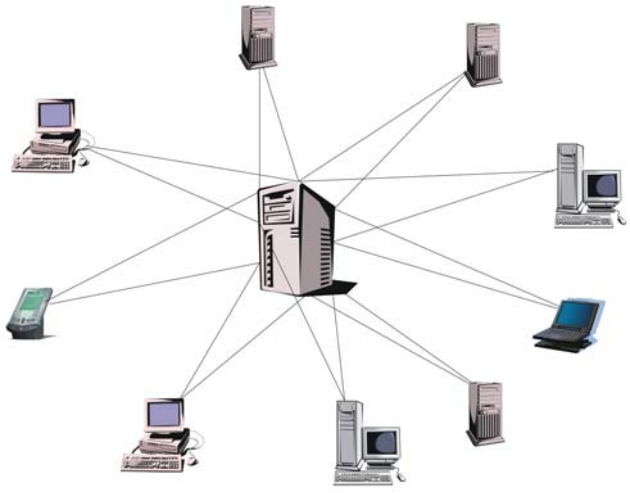
Figur 2.2: Konfigurering av generiska komponenter i syfte att skapa en flygplansfederation.

2.1.2 Nätverksarkitekturer

En ofrånkomlig fråga vid utveckling av distribuerade system är vilken underliggande nätverksarkitektur man skall använda sig av. Exempel på arkitekturer kan vara *client-server* eller *peer-to-peer* och i flera fall används en kombination eller modifikation av olika arkitekturer. Varje arkitektur har sina förrespektive nackdelar som utvecklaren måste ta sig an vid design och utveckling av systemet.

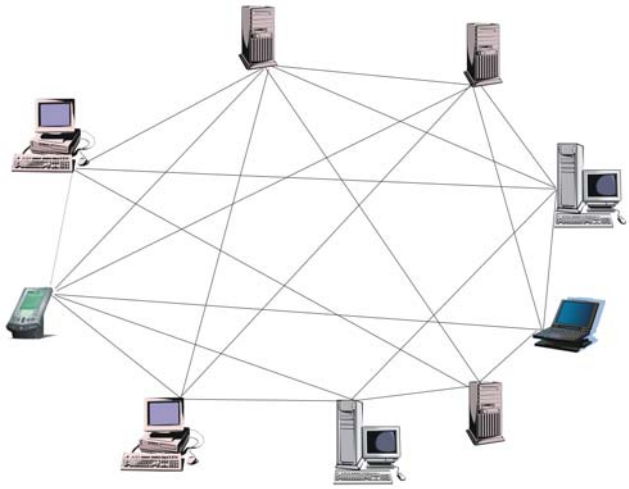
Genom att använda sig av nätverk med tillhörande distribuerade noder kan man utföra arbeten som inte annars skulle vara möjliga, till exempel att dela upp beräkningstunga applikationer på ett flertal noder (processorer) för exekvering och således spara tid. En nackdel med att göra på detta sätt är att det blir mer komplext genom att systemet blir svårare att förstå och administrera, det vill säga det ställer högre krav på utvecklarna att utveckla ett distribuerat system kontra ett monolitiskt system. En fördel med ett distribuerat system är att man sprider ut kraften i systemet på ett antal olika resurser. Exempel på detta är lagring av data och exekvering av beräkningstunga applikationer.

Ett exempel på nätverksarkitektur som nämnts i den inledande texten är *client-server*, som kan ses som ett centraliserat system där man centralt i systemet har en dator med hög kapacitet (*server*) som i sin tur betjänar ett antal klienter (*client*) som begär tjänster från servern, se Figur 2.3. Signifikativt för denna lösning är att ”all” bearbetning sker på servern och att klienten skickar en begäran om att få ett uppdrag utfört av servern och slutligen ta emot resultatet och presentera det.



Figur 2.3: Centraliserad nätverksarkitektur, client-server.

Ett annat exempel på arkitektur är *peer-to-peer*, detta angreppssätt (som arkitektur) bygger på direktkommunikation mellan klienterna, Figur 2.4. Med detta sägs indirekt att noderna är både *client* och *server* samtidigt. Det vanliga är således att alla noder exekverar en och samma applikation till exempel en fildelningsapplikation, där en nod X tillhandahåller filer som kan laddas ner av andra noder, t.ex. nod Y, samtidigt som nod X själv laddar ner en fil från en annan nod Z. Andra exempel utöver fildelnings-applikationer kan vara applikationer som fördelar ut beräkningsuppdrag i *peer-to-peer*-nätverket för beräkning.

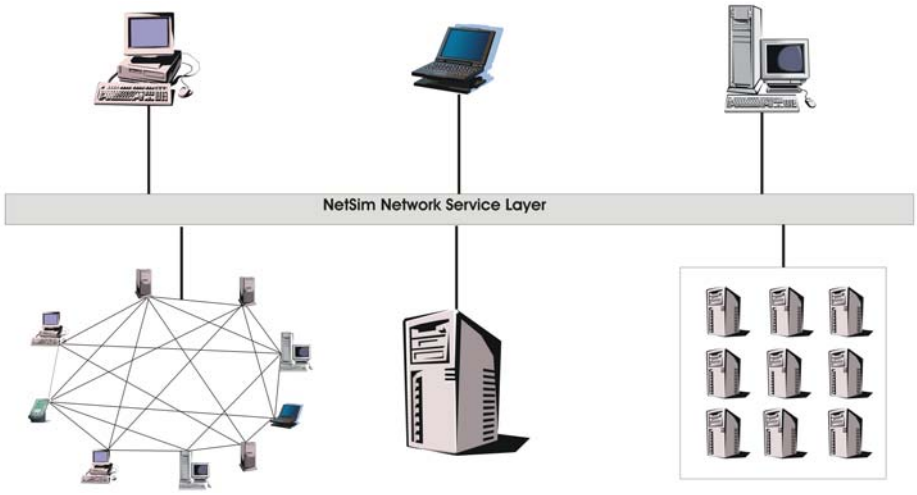


Figur 2.4: Helt distribuerad nätverksarkitektur, peer-to-peer.

Ytterliggare en arkitektur som ofta kommer upp i samband med beräkningstunga applikationer är *Grid* eller *Grid Computing*, Figur 2.5. *Grids* kan se som ett sätt att samla ihop stora mängder datorer/beräkningsresurser/processorer för att skapa en enda stor virtuell superdator. *Grids* lägger rent praktiskt ett lager mellan användaren och den mängd datorer som skall utföra beräkningen. Detta lager kan till exempel tillhandahålla tjänster för bland annat resursfördelning utifrån någon prioritets-bedömning, lastbalansering, uppdelning av beräkningsuppdraget, lagring med mera.

Figur 2.5: Virtuell superdator, Grid Computing.

I NätSim-konceptet kan alla ovan beskrivna arkitekturer förekomma som underliggande strukturer. Men precis som med alla former av distribuerade system döljs detta för användaren i form av ett mellanliggande lager som tillhandahåller och fusionerar tjänsterna för användaren, Figur 2.6. Detta med syftet att avlasta användaren från problematiken med hur man till exempel nyttjar tjänsterna i de olika arkitekturerna.

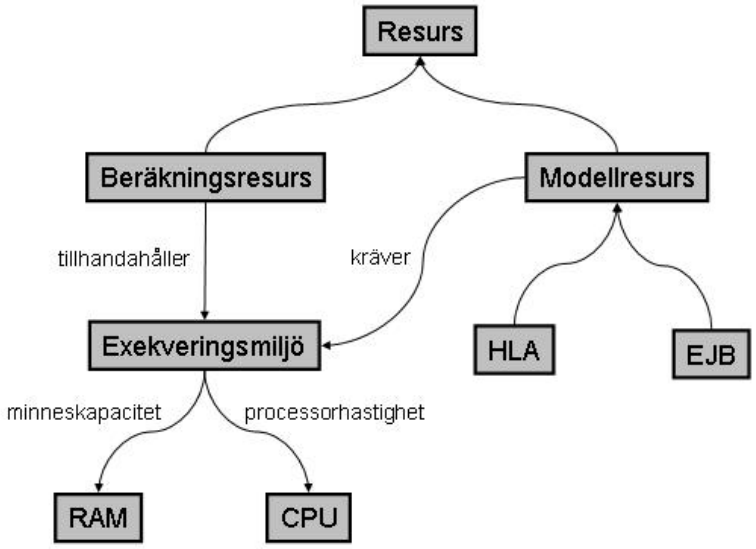


Figur 2.6: NätSim-konceptet för sammankoppling av nätverksarkitekturer.

2.2 Ramverk för beskrivning av resurser

2.2.1.1 Resursbeskrivningar

För att möjliggöra effektivt utnyttjande av resurser, i form av modeller, beräkningskraft etc., är det viktigt att beskriva dessa på ett standardiserat sätt. Genom detta möjliggörs mer precisa sökningar efter resurser, matchning av resurser som kan tillgodose krav på interoperabilitet (jämför med 2.1.1.1 ovan), samt effektivt resursutnyttjande vid exekvering. I detta sammanhang krävs en ontologi (informationsmodell) som representerar informationen på både syntaktisk och semantisk nivå[2], för att skapa en enhetlig syn på informationen inom systemet. Ontologin beskriver bland annat klasshierarkier, attribut för klasser, samt relationer mellan klasser. Figur 2.7 visar hur en delmängd av en NätSim-relaterad ontologi kan utformas.



Figur 2.7: Delmängd av NätSim-relaterad ontologi.

För att införa semantiskt rika resursbeskrivningar i NätSim-miljön krävs strukturering och formalisering av den information som är relevant för systemets funktionalitet. Det krävs ett språk som kan uttrycka den nödvändiga informationen på ett entydigt och maskintolkbart sätt. Språket används för att bygga

[2] Syntax – den formella struktur som tillämpas av ett språk för att uttrycka information

Semantik – med semantik avses meningen med information

upp en gemensam informationsmodell, eller ontologi, som uttrycker de regler som resursbeskrivningar måste följa för att kunna nyttjas av miljön. I Avsnitt 2.2.1.2 beskrivs ett antal språk som kan utnyttjas för detta syfte. Vidare krävs funktionalitet som kan extrahera relevanta resursbeskrivningar baserat på en specifik fråga, t.ex. vilka beräkningsresurser som har mindre än 50% belastning. Detta hanteras av en inferensmotor som vid sökning tar hänsyn till den gällande ontologin och de resursbeskrivningar som baserats på denna.

Ontologibaserad sökning är klart fördelaktig jämfört med sökning baserat på nyckelord. Denna typ av sökning tillåter framförallt mer precis lokalisering av de resurser som eftersöks. Metoden möjliggör sökning baserat på de klasser, samt relationer mellan klasser, som ontologin uttrycker.

Matchning av resurser kan förenklas genom att applicera ontologibaserade resursbeskrivningar. Inom NätSim-miljön finns flera fall då detta är relevant. Ett av målen med miljön är att tillgodose att simuleringskomponenter som tillsammans utgör en simulering är interoperabla. Resursbeskrivningarna representerar i detta fall metadata om simuleringskomponenter som kan nyttjas för att avgöra om en mängd simuleringskomponenter kan interagera på ett korrekt sätt.

2.2.1.2 Språk för ontologier

Idag satsas mycket resurser på utveckling av metoder för hantering av formaliserad information och kunskap i webbsammanhang, inom det som benämns den semantiska webben. Den semantiska webben förväntas bidra med förbättrad lokalisering, integrering och återanvändning av information på webben, genom formaliserade och maskintolkbara representationssätt för information. Genom W3C (*World Wide Web Consortsium*) [webb: W3C] hanteras många av de frågor som är relevanta för realisering av den semantiska webben. Inom detta forum pågår bland annat arbetet med att standardisera de språk som kommer att ligga till grund för den semantiska webben, nämligen RDF (*Resource Description Framework*) [webb: RDF] och OWL (*Web Ontology Languge*) [webb: OWL].

RDF utgör ett av de första stegen mot att realisera visionen om den semantiska webben. RDF är ett generiskt ramverk som kan användas för att representera metadata om resurser på webben på ett enhetligt vis, för utbyte och återanvändning mellan olika applikationer, organisationer eller andra grupperingar. För vissa syften visade sig RDF inte bidra med tillräcklig uttrycksfullhet, varför OWL har skapats som komplement. OWL finns i sin tur i tre olika versioner (*lite*, DL och *full*) som har varierande grad av uttrycksfullhet.

Det vanligast förekommande representationsformatet för RDF och OWL är XML, vilket gör språken lämpade för heterogena nätverksmiljöer. RDF och OWL är inte enbart skapade för webbsammanhang, utan kan även utnyttjas för informationsrepresentation i lokala nätverk. Inom NätSim-miljön skulle det vara fördelaktigt att använda RDF/OWL som ramverk för representation av de metadatamodeller som krävs. Detta skulle skapa förutsättningar för en enhetlig uppfattning av information inom systemet, underlätta vid sökning efter tillgängliga resurser, samt även effektivisera matchning av resurser då detta krävs.

2.3 Tjänstelager

Ovanpå det grundläggande lagret (Nätverkslagret) baserar sig NätSim på en tjänstebaserad arkitektur eller *Service-Oriented Architecture* (SOA) som det ofta benämns i systemutvecklingsdomänen. SOA är en arkitekturstil vars mål är att uppnå en lös koppling mellan kommunicerande mjukvaruagenter. En tjänst är i detta fall ett arbetspaket som tillhandahålls av en tjänsteleverantör (*provider*) för att uppnå ett önskat resultat för en tjänstekonsument (*consumer*). Både leverantör och konsument realiseras i form av mjukvaruagenter som agerar enligt sina användares instruktioner. SOA är således en arkitektur som erbjuder självständiga tjänster (funktionalitet) med väldefinierade anropbara gränssnitt i form av mjukvara som är tillgänglig och upptäckbar i ett nätverk. Dessa tjänster kan i sin tur anropas från klientapplikationer med definierade sekvenser. Konceptet SOA och även *Web Services* relaterar starkt till nätverksarkitekturen *client-server* som beskrevs i 2.1.2. [He 04], [Chatarji 04]

SOA är inget nytt revolutionerande koncept, tidiga och fortlevande exempel på detta är till exempel *Common Object Request Broker Architecture* (CORBA) och *Distributed Component Object Model* (DCOM) som

tillhandahåller SOA-liknande funktionalitet. Det sätt på vilket CORBA och DCOM erbjuder tjänster har dock ett par nackdelar som till exempel tätt sammankopplade scenarier. [Chatarji 04]

Kombinationen av SOA och *Web Services* (WS) löser till viss del det angreppssätt som CORBA och DCOM har gentemot SOA. WS har framförallt tagit sig igenom ytterliggare en barriär genom att möjliggöra för distribuerade applikationer utvecklade på olika plattformar och med olika programmeringsspråk att kopplas samman på ett neutralt sätt. Detta till exempel genom att använda sig av ett enkelt XML-baserat (*eXtensible Markup Language*) sätt att utbyta information, vilket möjliggör bl.a. att Java-applikationer kan nyttja Microsoft .NET-applikationer över ett nätverk. WS är en samling av möjliggörande tekniker för SOA, och SOA är på stark frammarsch som den arkitektur som kommer att gälla för utveckling av tillgängliga och anpassningsbara applikationer. [Chatarji 04]

LedsystT, som är ett av de centrala projekten inom utvecklingen av det Nätverksbaserade försvaret[3] (NBF), tar sig an det tjänsteorienterade konceptet. LedsystT producerar ingen konkret produkt utan snarare ett antal tekniska tjänster där information ska kunna lämnas och hämtas på olika sätt och vara tillgänglig för många användare på olika nivåer. [webb: LedsystT]

I NätSims tjänstelager återfinns ett antal SOA-liknande tjänster såsom till exempel tjänster för distribuerad exekvering, samt olika söktjänster för sökning i underliggande fil- och lagringssystem.

2.3.1 Distribuerad resurshantering

Det räcker inte bara att erbjuda tjänster fritt över ett nätverk utan man måste även ha mekanismer för att fördela ut uppdragen, det vill säga någon form av resurshantering för de distribuerade tjänsterna som erbjuds. Exempel på resurshanteringsfunktioner som kan erbjudas är jobbfördelning för exekveringstjänsten vilket gör att användarna av exekveringstjänsten bara lägger ut ett eller flera exekveringsuppdrag och sedan inte behöver tänka på var deras uppdrag exekveras i systemet. Ett annat exempel relaterat till resurshantering och filsystemet är lagring av till exempel simuleringsmodeller. Det måste finnas en mekanism som tillser att dessa alltid finns tillgängliga om delar av nätverket skulle gå ner, det vill säga en form av automatisk resurshanteringsmekanism som sprider ut simuleringsmodellerna på ett fördelaktigt sätt ur tillgänglighetssynpunkt.

2.3.1.1 Distribuerad resurshantering – Exekvering

DRMS (*Distributed Resource Management System*) är namnet på den exekveringsmiljö som utvecklas inom projektet och som ger användare av NätSim-miljön tillgång till den beräkningskraft som krävs vid exekvering av en simulering. Konceptet bygger på att skapa grupper, eller så kallade virtuella organisationer (VOs), inom beräkningskraft delas mellan medlemmarna. På detta sätt får den enskilda användaren tillgång till en pol med beräkningskraft som är långt större än den som finns tillgänglig på användarens egen dator. Exekvering inom DRMS är en transparent process, i bemärkelsen att medlemmarna i gruppen inte är involverade i att identifiera lediga beräkningsresurser, allokera (reservera) identifierade beräkningsresurser, eller att distribuera och konfigurera involverade simuleringskomponenter.

I [Eklöf et al. 04] presenteras föregående års arbete med DRMS som framförallt koncentrerades på delning av beräkningskraft i *peer-to-peer*-nätverk. Under innevarande år har konceptet för DRMS utökats något till att inkludera mer omfattande stöd för feltolerans, samt utveckling av en tjänstebaserad arkitektur för DRMS.

I ett löst samankopplat system för exekvering av simuleringar är det viktigt att beakta hur ett fel i en programvara, alternativt hårdvara, skall hanteras. Det finns idag inget stöd för feltolerans i arkitekturer som HLA, vilket kan medföra delvis felaktiga resultat om vitala simuleringskomponenter frånfaller, producerar felaktig data etc. under simuleringsexekveringen. I ett system med löst sammankopplade enheter är det inte möjligt att anta att allokeringen av resurser är statisk. Med andra ord tillkommer och frånfaller resurser spontant enligt ett oförutsägbart mönster. DRMS fokuserar på problemet med

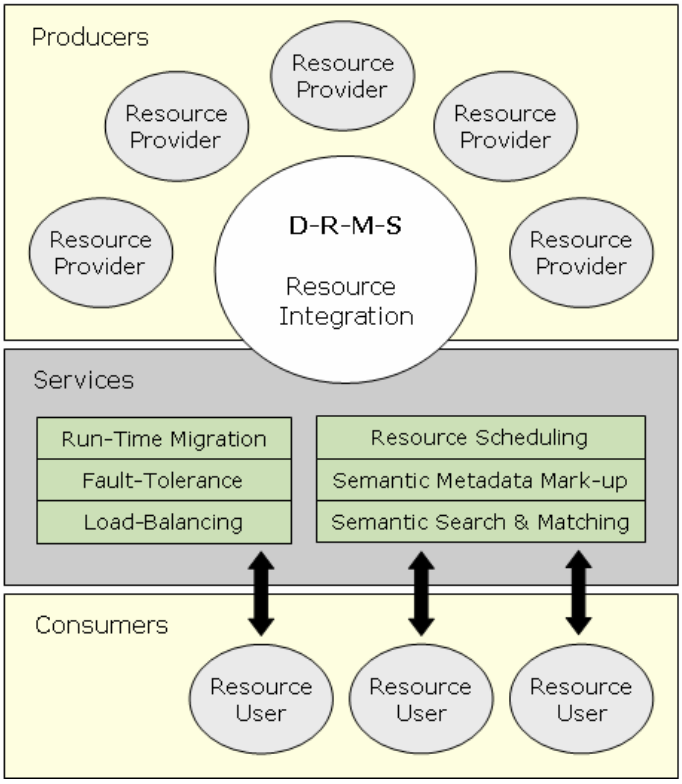
[3] Nätverksbaserat försvar (NBF), som kan ses som ett samlingsnamn för Försvarsmaktens utveckling mot ett flexibelt insatsförsvar [webb: FHS-NBF].

frånfallande exekveringsresurser och mindre på interna fel i applikationen. Förutsättningen för att implementera denna typ av feltolerans är att enskilda simuleringskomponenter, som tillsammans utgör en simulering, kan förflyttas (migreras) mellan skilda värdmiljöer under exekveringsgången.

Från ett övergripande perspektiv omfattas DRMS-konceptet av följande funktioner i syfte att stödja ovan nämnda aspekter (se Figur 2.8 nedan):

1. Lokalisering av lediga beräkningsresurser inom ett nätverk
2. Allokering av simuleringskomponenter till beräkningsresurser
3. Simuleringsexekvering med eller utan feltolerans
4. Simuleringsexekvering med eller utan lastbalansering

En grundläggande förutsättning vid migrering (även allokering) av simuleringskomponenter är att tillgodose den enskilda komponentens krav på en värdmiljö. Dvs. komponentens krav, tillika värdmiljöns egenskaper, måste beskrivas på ett standardiserat sätt för att möjliggöra en effektiv matchning. Vid matchning kontrolleras de krav en simuleringsmodell kan förväntas ha, t.ex. minsta möjliga RAM-minne, processorhastighet etc., mot exekveringsresursernas egenskaper, för att finna möjliga allokeringar. I detta arbete eftersträvas en generell metod för modellering av nödvändig information. Metoden bör stödja utveckling av ett NätSim/DRMS-specifikt schema som definierar hur kraven från komponent, tillika egenskaperna för en exekveringsresurs, skall uttryckas.



Figur 2.8: Övergripande koncept för DRMS-miljön.

2.3.1.2 Distribuerad resurshantering – Bibliotek

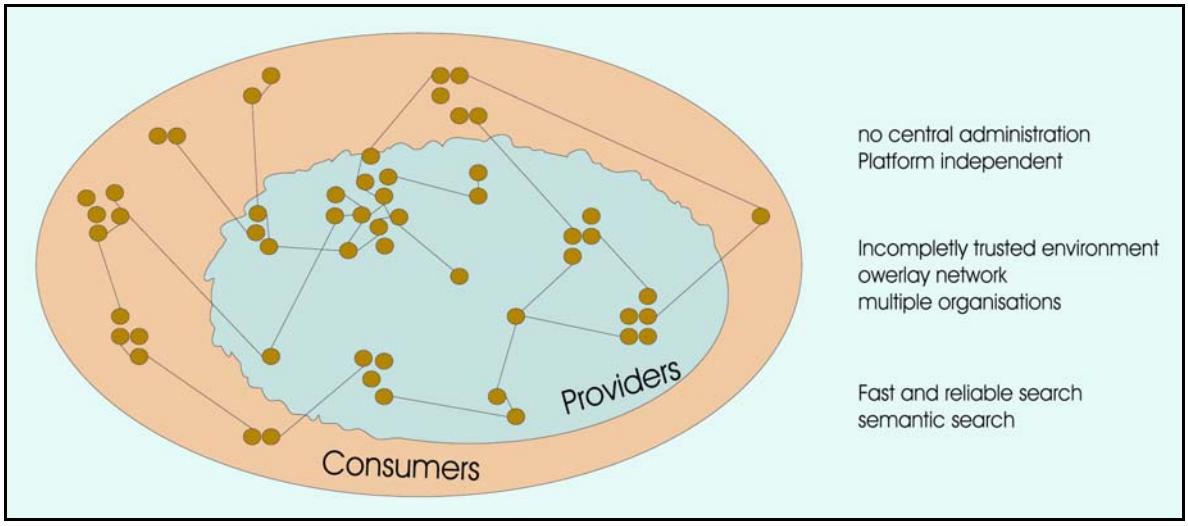
En virtuell organisation (VO) består av en grupp människor eller organisationer som har ett gemensamt mål eller intresse. Denna grupp kan sträcka sig över organisationsgränser, nätverk, plattformar etc. Syftet med det distribuerade biblioteket i NätSim är att skapa ett gemensamt forum med tjänster och resurser som kan delas mellan användarna, på samma sätt som i en virtuell organisation. Målet är att låta olika personer eller organisationer med tillgängliga, eller med behov av, resurser, modeller, filer, komponenter, osv. av olika slag, kunna utnyttja och dela resurser. Biblioteket befinner sig för närvarande på ett tidigt utvecklingsstadium.

Biblioteket i NätSim kan ses som ett informationsnätverk som är överlagrat på det befintliga samarbetsnätverket. Resurser, modeller och annat som flera kan ha intresse av beskrivs i en metadatafil som är uppbyggd enligt en för NätSim standardiserad informationsmodell och ontologi (se Avsnitt 2.2). Varje tillgänglig fil, modell, resurs etc. i biblioteket beskrivs i metadatafiler där det t.ex. finns en beskrivning av vad det är för något, vilka som har rättigheter till den, ägare, när metadatafilen som beskriver den skapades, av vem, när den gjordes tillgänglig osv. Metadatafilerna i biblioteket är uppdelade i två abstraktionsnivåer där den lägre nivån beskriver filen, modellen eller resursen i sig och den andra, högre nivån, beskriver metadata om metadatat, dvs. hur, när, var filen skapades, rättigheter osv.

I biblioteket kommer det att finnas metadata som framförallt beskriver tre olika domäner, det kan vara filer, resurser eller modeller. Filerna kommer framförallt från NätSim-miljön och kan t.ex. bestå av användardata, scenariobeskrivningar, loggfiler, samarbetsbeskrivningar osv. De är ofta genererade i miljön och används för det mesta endast av eller i miljön. Med resurser menas framförallt beräkningsresurser med beskrivningar av processorkraft, tillgänglighet, hårdvara etc. Dessa beskrivningar används främst av DRMS-modulen för att t.ex. kunna hitta lämpliga och lediga beräkningsresurser. Modellerna är simuleringsmodeller eller simuleringskomponenter som har utvecklats i olika sammanhang och som man inom en organisation vill göra tillgänglig för andra. Här ingår även all den dokumentation som kan finnas runtomkring med t.ex. API:er, dokumentation, erfarenheter osv. Användningen av dessa simuleringsmodeller, komponenter och övrig dokumentation sker vid modellering och simulering.

I de gemensamma delarna av biblioteket är det endast metadatabeskrivningarna som finns tillgängliga. Detta för med sig att även om filer och modeller finns hos olika organisationer som kanske inte är internt kompatibla, på grund av t.ex. olika plattformsval, olika säkerhetsföreskrifter eller annat, finns det ett överliggande lager som utgörs av biblioteket. Organisationerna kan i sin tur ha databaser eller FTP-servrar, åtkomliga utifrån, där modeller och andra filer kan hämtas.

Användarna av biblioteket är av framförallt två typer; de som producerar modeller och filer som de vill dela med sig av, samt de som endast är intresserade att använda biblioteket för att söka efter filer, resurser och modeller, se Figur 2.9. Naturligtvis kan användare tillhöra båda typerna.



Figur 2.9: En konceptuell bild av biblioteket. Användare i olika organisationer delar med sig av resurser, filer och modeller och andra användare använder dessa i olika projekt.

Den grundläggande funktionaliteten som är tänkt att finnas i biblioteket är att skapa, läsa, skriva och ta bort metadata, samt sökning. Detta betyder också att filer, modeller och resurser skall kunna läggas till och tas bort från biblioteket. Sökningen skall inte bara ske på nyckelord utan den skall även kunna ske baserat på ontologi, dvs. det skall vara möjligt att vid en sökning av JAS39-flygplansmodeller även kunna få t.ex. andra typer av flygplansmodeller, pilotmodeller och vapenmodeller som är relaterade med JAS39. Detta kräver både att det finns en ontologi för den domän modellerna är utvecklade för, och en fungerande informationsmodell för beskrivning av filer, modeller och resurser. Baserat på detta byggs inferensregler för att kunna lokalisera filer, resurser, modeller, etc. som stämmer överens med sökkriterierna.

2.3.2 Datorbaserad samverkan för M&S

2.3.2.1 Önskvärd funktionalitet

I en plattform för gemensam M&S och resurstillgång inom försvaret såsom avses med NätSim, är en tjänst för datorbaserad samverkan en eftersökt funktion. *Datorbaserad samverkan* är ett samlingsbegrepp för den typ av samverkan där verktyg och tjänster tillåter människor att på olika platser gemensamt operera på (s.k. ”dela”) ett verktyg eller en arbetsarea på sina datorer. Deltagare i samarbetet kan interagera med och samtidigt se vad som för användarna uppfattas som samma instans av dator-programmet. På detta sätt kan användare på distans samarbeta och lösa uppgifter i datormiljön. NätSim-plattformen utgör en sådan gemensam miljö, där en tjänst för datorbaserad samverkan är tänkt att stödja de som använder NätSim med att använda de tillhandahållna verktygen och funktionerna gemensamt. Plattformen tillåter på så vis personer inom försvaret att samarbeta mer lättillgängligt via sina datorer och mobila enheter vilket leder till att en mängd fördelar och möjligheter öppnas varav några presenteras nedan (se även Exempelscenario Appendix A för en praktisk exemplifiering av nyttan):

- Möjliggör och underlättar åtkomst av expertkunskap. Omedelbar tillgång till rätt expertis vid rätt tidpunkt.
- Möjliggör och effektiviserar datorrelaterade aktiviteter på distans, såsom projektarbete, distansutbildning, logistikplanering och distansträning.
- Bidrar till att komplettera *den gemensamma lägesbilden* (gemensam samtidig lägesuppfattning, gemensam informationsbas, gemensamma beslut etc.).

Förutom de generella fördelarna kan även nytta med datorbaserad samverkan vinnas särskilt mycket inom vissa områden och M&S-domänen är ett sådant område. Till exempel kan uppgifter som relaterar till VV&A-processen underlättas och effektiviseras, då VV&A-personal mer lättillgängligt kan inspektera och bistå arbetsprocesser under arbetets gång. Dessutom kan arbetets kvalitet säkras då tillgången till expertis förbättras. Resultatet kan även enklare säkerställas, då en kund eller överordnad kan delta i mer av arbetet på avstånd. NätSim kan alltså inte bara medföra att simuleringar och simulatorer blir tillgängliga på avstånd, utan kan dessutom bidra med direkt support från expertis till simuleringen, något som ofta saknas.

2.3.2.2 Tjänsten i NätSim – Ett stöd för användare

NätSim-tjänsten för datorbaserad samverkan tillhandahålls som en modul, *Collaborative Core* (CC), som stödjer miljön i bakgrunden transparent för användaren. CC levererar stöd för och de tjänster som är nödvändiga för att en användare ska kunna starta, administrera och delta i datorbaserade samarbeten, samt dela verktyg och funktioner, i NätSim. Detta inkluderar även kommunikationsverktyg som erfordras, såsom webbkamera, ljud- och videoöverföring och chatt. Dessutom erbjuds utvecklingsstöd för att kunna integrera nya verktyg för datorbaserad samverkan och implementera CC-funktionalitet för att kunna utnyttja tjänsten. CC beskrivs mer i detalj i kapitel 3.1.1.

2.3.2.3 CSCW och NätSim

CSCW är en benämning som ofta används för datorbaserad samverkan. Termen står för *Computer Supported Collaborative Work*, och avser de miljöer, produkter, tekniker och metoder som finns för detta och delade virtuella världar. CSCW hanterar även de sociala aspekter som uppkommer av att möta personer i datorbaserade världar. Det vill säga förutom tjänster och tekniskt stöd för denna funktion, behandlar CSCW frågor kring virtuell social närvaro av karaktären *Hur mycket kommunikation och av vilken typ behövs för att lösa en specifik uppgift? Hur demonstreras sinnesstämningar i datorrepresenterade personligheter (så kallade* avatars*)*? Denna typ av frågor behandlas inte i NätSim-arbetet, då syftet i projektet är att se på tekniska förutsättningar för datorbaserad samverkan inom en gemensam försvarsmiljö. I dessa sammanhang avses ofta krissituationer, snabba beslut och planering, situationer som inte kräver lika hög social närvaro som till exempel då personer i en virtuell värld möts och ska lära känna varandra.

2.4 Applikationslager

Som har tidigare beskrivits är målet med NätSim att erbjuda en tjänstebaserad infrastruktur som stödjer simuleringsmodeller under hela livscykeln. I de tidigare avsnitten beskrevs också ett antal tjänster som ingår i infrastrukturen. Det översta lagret i NätSim -arkitekturen består av applikationer och verktyg som utnyttjar dessa tjänster och används för kommunikation med användarna. Exempel på applikationer är verktyg för utveckling av simuleringsmodeller, UML-verktyg och olika verktyg som stödjer FEDEP från beskrivning och kravanalys till design och utveckling av federationer samt test och verifiering av dessa.

För att applikationer skall kunna utnyttja tjänsterna i de lägre lagren bör det finnas ett gränssnitt (API) som möjliggör det för applikationerna att anropa de metoder som behövs för att komma åt tjänsterna.

Under designarbetet har gränssnitt tagits fram för koppling mellan tjänsterna och applikationer genom att bl.a. identifiera informationsflödet mellan de olika lagren. Dessa gränssnitt är på en mer generell nivå och är frikopplade från tekniska implementeringar. Det finns också tekniska realiseringar av dessa gränssnitt som redovisas i kapitel 3.

2.5 Infrastruktur för säkerhet

Två viktiga säkerhetsbegrepp när det gäller kommunikation mellan användare är kryptering och autentisering. Kryptering används för att obehöriga inte ska kunna ta del av, eller sabotera kommunikation mellan legitima användare. Autentisering ser till att man vet vem man pratar med, och att det är en legitim användare. Ett specialfall av autentisering är att en användare signerar en datafil för att andra vet att de ska kunna lita på detta data.

2.5.1 Mål

NätSim är en kombination av ett militärt ledningssystem och forskningsplattform. För att kunna använda NätSim för militärt bruk så krävs separat hårdvara för kryptering av transmissionslänk och hantering av hemliga symmetriska nycklar. För forskningsbruk är det inte lika självklart att man vill tynga ned systemet med stark kryptering och krångliga inloggningsprocedurer, det ska vara enkelt och snabbt att använda säkerheten. Om systemet exekverar lokalt inom en organisation (exempelvis inom FOI) utan att använda känsligt data, så behövs egentligen ingen speciell säkerhet, men man vill undvika att missa säkerheten den dagen det behövs. Fokuseringen på säkerhet har gjorts för signalklassnings-nivån *restricted* (se Appendix B.4.3.1) vilken får användas i samband med mjukvaruimplementationer. Den säkerhetsnivån kan hantera känsligt och internt material, men inte hemligt. Tanken är att militär hårdvara kan läggas till senare för att skydda transmissionslänken om så behövs.

2.5.2 Autentisering och säkerhetspolicy

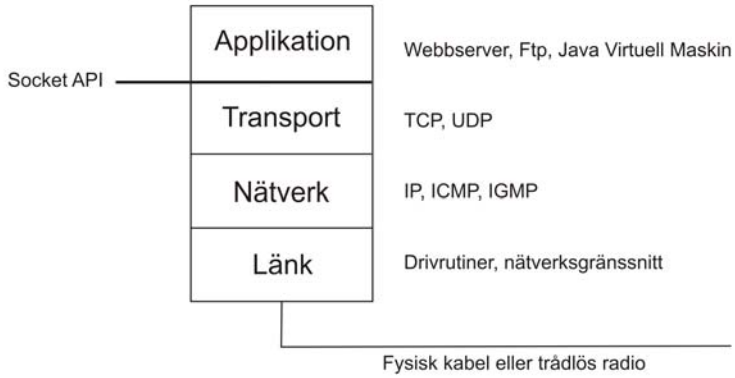
En användare måste verifiera sin identitet för att få tillgång till simuleringar och information som inte är publik. Tanken är att det ska räcka men en inloggning i NätSim per session (så länge klienten körs kontinuerligt) för att komma åt nödvändiga tjänster och resurser på olika noder. Detta kallas även *Single Sign-On* [Bengtsson et al. 01], dvs. en enda inloggning.

Det räcker inte att användaren är autentiserad för att avgöra dennes rättighet på en resurs (läsa, skriva, skapa eller exekvera), utan man behöver dessutom tillämpa en säkerhets-*policy* för att skydda resurserna. Varje resurs tilldelas en åtkomstlista *(access control list,* ACL) på användare eller grupp, samt tillåtna operationer för dessa. Innan åtkomst sker så matchas användaren och dennes grupptillhörighet mot åtkomsträttigheter på resursen. ACL beskrivs ytterligare i B.4.2.1.

Det finns tjänster i systemet som hjälper användare och andra tjänster att utföra uppgifter. Dessa delegeringstjänster måste vara av mer statisk karaktär samt pålitliga för att slavnoder ska kunna lita på dem. Hjälptjänster bör autentisera sig som tillhörande organisationen.

2.5.3 Kryptering

Det finns kryptering på flera nivåer av kommunikationen. I turordning underifrån och upp (se Figur 2.10) så kan eventuell militär hårdvara användas för att skydda **länken** (även kallad data- eller transmissionslänk).



Figur 2.10: De fyra lagren för TCP/IP kommunikation inom en nod [Stevens 94].

Ovanpå detta kan man använda sig av virtuella privata nätverk (VPN) för att skydda all trafik på **nätverksnivå** (kryptering av IP-paket). VPN kan skydda meddelandetrafik mellan noder som deltar i en simulering. Den kan också användas för att skapa en insynsskyddad tunnel mellan mobila noder, olika organisationer och domäner som är sammankopplade över en icke-skyddad kanal som Internet.

Längst upp har vi **applikationer och tjänster** som kan använda sig av *secure socket layer* (SSL, används av webbläsare i den krypterade motsvarigheten till HTTP: HTTPS) för att prata mellan två noder [Stallings 03]. Denna går dock inte att använda när fler än två noder samtalar i en konferens.

Kryptering kan också ske av en fil i filsystemet för att förhindra otillåten åtkomst.

2.5.4 Säkerhetsbehov på olika nivåer

NätSim är att betraktas som en applikation, vilken som helst, installerad i ett operativsystem. Även om man lyckas säkra applikationen helt, så kan den attackeras från underliggande nivåer som operativsystem, minne, drivrutiner, nätverk osv. Appendix B belyser, i ett brett perspektiv, hot och kravställningar mot hela systemet där NätSim exekveras. Även problem med administrativa rutiner beaktas. Men för att begränsa arbetet har vi valt att fokusera på säkerhetsfrågor för systemets tjänster och applikationer. Säkerhetsbehoven för ett antal prioriterade tjänster listas nedan.

Ett generellt krav för NätSims tjänster är att de kräver autentisering av användare, och denna ska göras med S*ingle Sign-On* (SSO).

Datorbaserad samverkan (*Collaboration*):

- Autentisering / *policy*-hantering:
  - Personprofil för användare bör matchas mot säkerhetsnivå på simuleringen.
  - Administratör kan tillåta nya användare att delta i simulering.
  - Kunna hämta och verifiera användare och klienters rättigheter.
  - Möjligt att ställa in säkerhetsnivå för användare, grupper och klienter.
  - Stöd för roller och grupper: administratör, läsare och skrivare av data.
  - Det måste finnas ett ramverk med användare, samt autentisering av dessa.

- Kryptering:
  - Överföring av känslig sensorinformation, positions uppdateringar, hanteringsinformation kring ett uppdrag etc.

Exekvering (DRMS):

- Autentisering / *policy*-hantering:
  - Kontrollera åtkomst på användare för beräkningsresurserna krångligt eftersom exekveringstjänsten tar över simuleringsarbete och fördelar det till beräkningsresurserna. Eventuellt reservera eller prioritera resurser för t.ex. den lokala organisationens användare.
  - Delegering av skapare mot resursregister?
  - Metadata bör begränsa var en viss modell får exekveras.
  - Scenariobeskrivning bör avgränsa var hemlig exekvering får exekveras.
  - Administrator får skapa och ta ned simulering indirekt via exekveringstjänst.

Bibliotek:

- Autentisering / *policy*-hantering:
  - Kontrollera användarrättigheter innan sökning, endast visa behörigt resultat.
  - Metadata med åtkomstregler (flera ägare, grupp, säkerhetsnivå) skapas vid initial lagring (alternativ ägare kan vara simuleringens administratör).
- Kryptering:
  - Kommandon och parameteröverföring (av t.ex. metadata).
  - SSL/HTTPS/XML dokumentkryptering för att läsa, skriva och skapa distribuerade filer. Åtkomst-*policy* för operationer på en fil bestäms ytterst av den nod som lagrar filen.
- Behov av säkerhetsnivåer: publik, *restricted.*
- Kompatibilitet med *Grid*-säkerhet [webb: gsi].

För att summera tjänsternas behov så ligger tyngdpunkten på autentisering och *policy*-hantering.

2.5.5 Tekniker

För att tillgodose säkerhetsbehoven för tjänsterna har ett antal säkerhetstekniker identifierats (som behöver implementeras):

- Certifikathantering. För att styrka identiteten hos användare behövs en infrastruktur för skapande och verifiering av certifikat, *Public Key Infrastructure* (PKI), se B.4.1.1. Certifikaten används för att styrka användarens identitet genom att han kan signera sina meddelanden med sin hemliga nyckel. Dessutom kan nyckelparen användas för att säkert överföra en gruppnyckel för fortsatt krypterad kommunikation.
- Autentisering/*policy*-hantering:
  - Autentisering inom en domän, och emellan domäner i form av *Single Sign-On.*
  - Användar- och grupphantering.
  - Säkerhets-*policy* med *access control list* (ACL)

- Kryptering mellan noder:
  - VPN på nätverksnivå.
  - SSL, HTTPS eller möjligtvis XML kryptering på applikationsnivå.

3 Teknisk plattform

NätSim-infrastrukturen har stöd för olika arkitekturer för komponentbaserad utveckling/exekvering av simuleringsmodeller, vilket gör det möjligt att använda fördelarna med olika arkitekturer och undvika att begränsa sig till en standard. Idén är att tillhandahålla ett bibliotek, med tillhörande delar såsom beskrivningar, kategorier och dylikt av modeller som tillhör olika arkitekturer. Användarna skall kunna välja bland dessa modeller och t.ex. komponera sammansatta simuleringar av de heterogena komponenterna. Den tekniska realiseringen av de olika NätSim-tjänsterna och resultatet från arbetet detta år presenteras i detta Kapitel.

3.1 Teknisk realisering av de olika lagren

3.1.1 Distribuerad resurshantering – Exekvering (DRMS)

Under 2004 påbörjades arbetet med att implementera konceptet såsom beskrevs i 2.3.1.1. Målet har inte varit att göra en fullständig implementering i år, utan snarare att lägga grunden till DRMS. NätSim är baserad på en tjänsteorienterad arkitektur, där enskilda delar görs tillgängliga för potentiella konsumenter genom väldefinierade gränssnitt. Därför var det naturligt att även lösa den interna, DRMS-specifika, kommunikationen enligt detta angreppssätt. Som grund för implementeringen valdes *Web Services*. Initialt övervägdes även *Grid*-infrastrukturer som grund för implementeringen, närmare bestämt *Globus Grid Services*. Detta alternativ valdes dock bort beroende på en alltför stor *overhead* associerad till utveckling och driftsättning av *Grid Services*, åtminstone för denna fas av utvecklingsarbetet. I praktiken är det dock ingen större skillnad på *Grid Services* och *Web Services*, då båda plattformarna bygger på en tjänsteorienterad arkitektur och baseras på samma teknologier i stor utsträckning. Det finns även möjlighet till konvertering av *Web Services* till *Grid Services* i framtiden, om så krävs. Vidare finns även arbete som pekar på en tydlig konvergens av *Web Services* och *Grid Services* i framtiden, se [webb: WSRF] för mer information.

Idag finns ett flertal, fritt tillgängliga, verktyg/plattformar för utveckling av *Web Services* att tillgå. För DRMS valdes Axis [webb: Axis], som är en del av *Apache Software Foundation*, framförallt för att detta är en väldokumenterad och välanvänd plattform som även *Globus Grid Services* baseras på i stor utsträckning (vilket underlättar eventuell transformering i framtiden). DRMS implementeringen baseras på tre huvudsakliga typer av tjänster, dessa är:

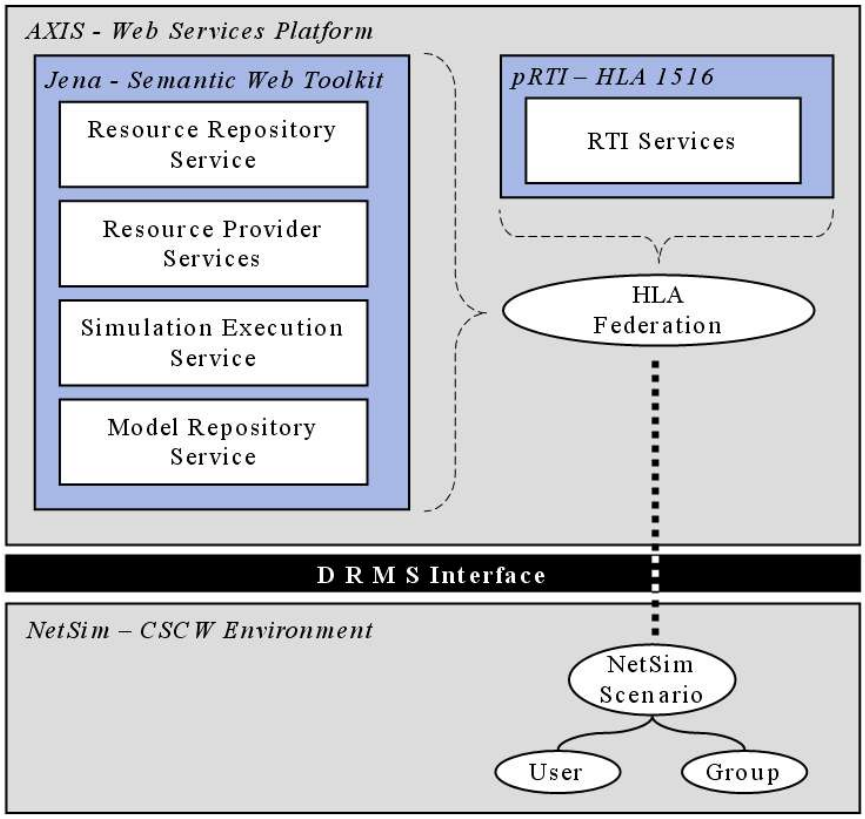
1. Beräkningstjänst
2. Resursregister
3. Exekveringstjänst

En beräkningstjänst representerar en nod i nätverket där beräkningsjobb kan allokeras för exekvering. Gränssnittet för en beräkningstjänst presenteras i Tabell 3.1. En beräkningstjänst registrerar en beskrivning av sina resurser hos ett resursregister. Gränssnittet för ett resursregister presenteras i Tabell 3.1. Resursregistret håller reda på tillgängliga beräkningsresurser inom ett nätverk och utnyttjas av en exekveringstjänst för att finna lediga beräkningsresurser. En exekveringstjänst är ansvarig för exekvering av en simulering som utvecklats i NätSim-miljön. Den har ansvar för identifiering av beräkningstjänster, samt allokering av simuleringskomponenter till dessa. Vidare har den även funktionalitet för detektering av fel och migrering av simuleringskomponenter under en simuleringsexekvering. Med detektering av fel avses förmågan att uppfatta när en simuleringskomponent (HLA-federat) inte fungerar korrekt, vilket resulterar i en omstart av federaten på en ny nod (migrering). Exekveringstjänsten har för närvarande inget externt gränssnitt, då kopplingen till övriga NätSim-komponenter inte är etablerad ännu.

Tabell 3.1. Gränssnitt för DRMS-tjänster.

Beräkningstjänst		**Resursregister**	
Metod	**Beskrivning**	**Metod**	**Beskrivning**
allocationRequest	Gör förfrågan till beräkningstjänst om den accepterar ett jobb.	**addResourceInfo**	Registrera info om beräkningstjänst hos resursregister.
downloadResources	Begär att beräkningstjänst laddar ner berörda resurser för exekvering av jobb.	**updateResourceInfo**	Uppdatera info om beräkningstjänst hos resursregister.
startJobExecution	Startar exekvering av ett jobb hos beräkningstjänst.	**removeResourceInfo**	Ta bort info om beräkningsresurs hos resursregister.
stopJobExecution	Stoppar exekvering av jobb hos beräkningstjänst.	**getAllResources**	Begär info om alla beräkningstjänster från resursregister.
checkJobStatus	Kontrollerar status för jobb hos exekveringstjänst.	**getResource**	Begär info om specifika beräkningstjänster från resursregister.

Allokeringen av simuleringskomponenter till beräkningstjänster baseras på rika metadatabeskrivningar. För att tillgodose de krav som en simuleringskomponent har på en värdmiljö sker en matchning mellan en simuleringskomponents krav och resurser som en beräkningstjänst tillhandahåller. Denna matchning bygger i grunden på en metadatamodell som definieras i OWL (*Web Ontology Language*) [webb: OWL]. OWL används för att etablera ett NätSim/DRMS-specifikt schema som resurser av skilda slag beskrivs utifrån. Schemat definierar ett antal relevanta metadata kring en beräkningstjänst och en simuleringskomponent. Detta inkluderar metadata gällande hårdvara och mjukvara, men kan även inkludera säkerhetsrelaterade aspekter. De delar av implementationen som hanterar metadata bygger på Jena, ett API som ger tillgång till funktioner som kan operera på OWL-strukturer [webb: Jena]. Figur 3.1 visar en konceptuell bild av systemet.



Figur 3.1: Schematisk bild av DRMS-implementeringen.

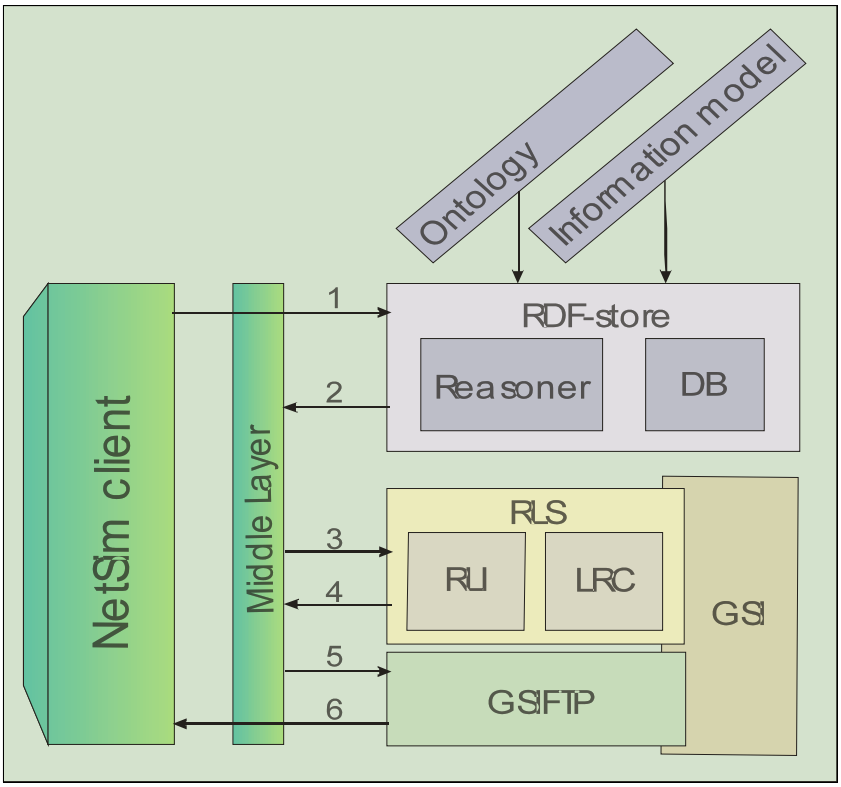
3.1.2 Bibliotek

Under andra halvan av 2004 har en förstudie av bibliotekets funktionalitet utförts. Olika krav har sammanställts i en enkel specifikation och använts som underlag för att utvärdera olika arkitektur- och teknikförslag. Utvärderingskraven har varit av typen, *open source*, plattformsoberoende, fungerar med olika organisationer, ingen central administration, distribuerat, säkerhet, fungerar med resten av NätSim-arkitekturen, osv. En viktig fråga var hur biblioteket skulle förhålla sig till filer, resurser och modeller. *Skulle biblioteket fungera som en databas eller ett filsystem där allt lagras? Skulle biblioteket vara frånskilt från de deltagande organisationernas system? Skulle det vara centraliserat eller distribuerat, kunde det finnas en övergripande organisation med en administrativ och styrande roll, kunde plattformsberoende accepteras, osv.?* Med utgångspunkt från dessa frågor och fler diskuterades olika alternativ och vilka avvägningar eller konsekvenser olika arkitektur- och teknikförslag förde med sig.

De tre huvudförslagen som undersöktes var; distribuerade filsystem såsom AFS, *Andrew File System,* se [webb: AFS], att bygga ett eget filsystem och indexera filer med hjälp av distribuerade hashtabeller, DHT, såsom Past och Pastry, se [webb: PAST, PASTRY], samt att använda *Grid* och Globus, se [webb: Globus], som bygger på tjänste- och VO-begreppet. Tillvägagångssättet som har valts för biblioteket är att använda *Grid*-tjänster och utnyttja den arkitekturen. Det som framförallt talade för *Grids* och Globus-alternativet var att det var det som hade flest färdiga tjänster och grundläggande funktionalitet som kunde användas för bibliotekets syften. Det skulle inte behövas något centralt administrationssystem och det skulle bli ett tämligen distribuerat och säkert system som inte skulle vara beroende av de deltagande organisationernas interna system eller plattformsval. Ett annat viktigt argument var att trots att det är ett omoget koncept så finns det ett stort intresse för och vilja att vidareutveckla *Grids* och Globus i omvärlden. Det som talade emot AFS var framförallt att det skulle krävas en central administrationsorganisation och att biblioteket skulle bli hårdare knutet till organisationernas interna system. Past och Pastry har många lovande fördelar såsom total distribuering samt bra och fungerande mekanismer för att hantera nodbortfall. Det som var den stora nackdelen med detta alternativ var att för många komponenter saknades för att kunna realisera bibliotekskonceptet inom ramen för NätSim-

projektet. En framtida vision är att använda de bra egenskaperna med DHT tillsammans med *Grids* och Globus.

Två examensarbeten har formulerats och genomförs för närvarande inom biblioteksaktiviteterna. De beräknas bli klara under nästa år. I år har de gjort förstudie och utvärdering av arkitekturer och kommer nu att inrikta sig på att implementera en avgränsad del av sökprocessen. De kommer framförallt att implementera och utforska processen från det att en sökningsinitiering görs till dess att svar returneras.



Figur 3.2: Sökningsprocessen. En NätSim-klient skickar en sökförfrågan till bibliotekstjänsten.

Sökningsprocessen kommer att gå till så att en *NetSim client* skickar en sökningsförfrågan till bibliotekstjänsten. Sökningskriterierna kommer att skickas till en RDF-Store där de tolkas och matchas mot metadatafilerna med hjälp av inferensregler som har implementerats utifrån en enkel ontologi (se steg 1 i Figur 3.2 ovan), som *back-end* finns databaser som innehåller metadatafilerna. I framtiden kan det tänkas att *Grid*-tjänsten OGSA-DAI, *Open Grid Services Architecture data Access and Integration*, se [webb: OGSA-DAI], som hjälper till att bättre samordna sökningen i de distribuerade databaserna används. Svaren med logiska filnamn från sökningen skickas tillbaka till biblioteksmellanlagret (se steg 2 i Figur 3.2). De logiska filnamnen skickas sedan till RLS, *Replica Location Service*, [webb: RLS] som svarar med en URI, *Uniform Resource Identifier*, se steg 3 och 4 i Figur 3.2. Sedan, beroende på vad det är som har eftersökts, kan t.ex. *Grid*-FTP [webb: GFTP] användas för att hämta hem modeller eller filer som resultat av sökningen.

3.1.3 Datorbaserad samverkan för M&S (CC)

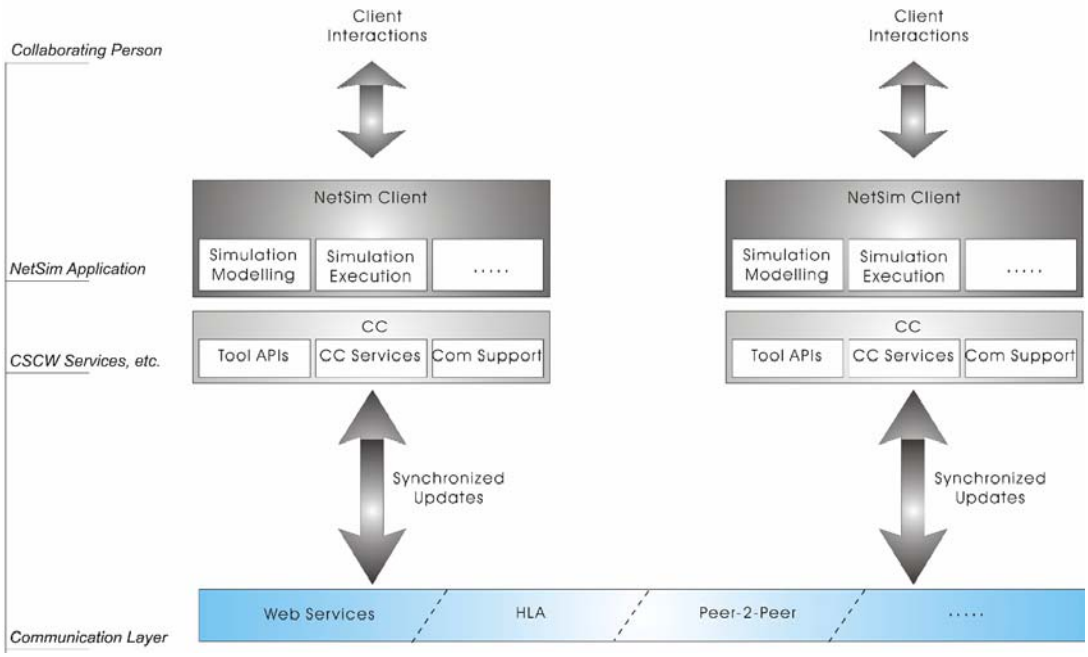
Eftersom fokus för NätSim-arbetet detta år rör arkitekturfrågor, riktas aktiviteterna för CC (*Collaborative Core*) först mot att grundligt undersöka vilka specifika tjänster som behövs för datorbaserad samverkan i miljön. För detta bryts först den avsedda aktiviteten ner, dvs. datorbaserad samverkan för främst M&S, så att nödvändiga funktioner och tjänster kan identifieras, varefter en kravspecifikation för CC kan sammanställas. Utifrån kravspecifikationen har en infrastruktur för datorbaserad samverkan utformats, vilken kommer att implementeras som en modul i NätSim-arkitekturen och erbjuda tjänster transparent för användaren (se Figur 3.3).

3.1.3.1 Modulens uppbyggnad

CC-klienten består huvudsakligen av tre komponenter (se även Figur 3.3):

1. Kommunikationsmedel (integrerade verktyg för chatt, webbkamera och audio).
2. Applikationsgränssnitt (API:er och mjukvara samt *guide-lines* för att koppla in nya applikationer (verktyg) i NätSim som ska kunna användas under datorbaserad samverkan).
3. Grupphanteringstjänster (skapa och förstöra grupper, gruppspecifika inställningar etc.).

Den första komponenten kommer att utgöras av existerande gratisverktyg som integreras i modulen. Arbetet koncentreras därför inte på detta, utan på frågeställningar för och implementering av de båda andra komponenterna. Två examensarbeten har formulerats inom respektive del som behandlas parallellt och integreras i slutfasen av arbetet i den gemensamma CC-modulen (dvs. våren -05).



Figur 3.3: Schematisk bild av hur CC erbjuder en NätSim-klient sina tjänster.

3.1.3.2 Arkitektur – HLA och XML för datorbaserad samverkan

Metod och teknik

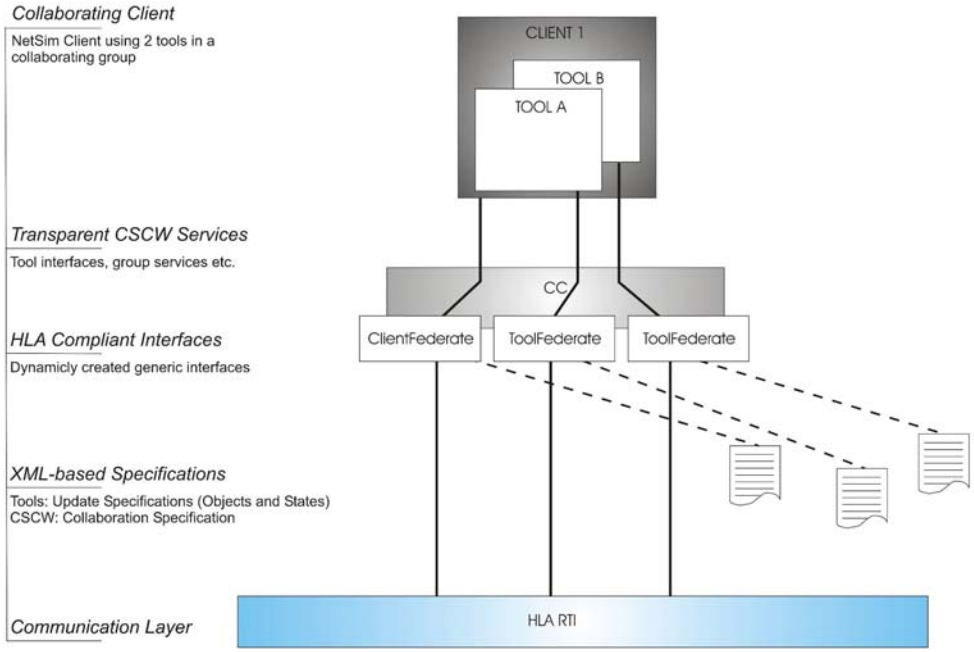
Den föregående prototypen för datorbaserad samverkan i NätSim var enkel men gav värdefulla erfarenheter [Ulriksson et al. 03]. Implementeringen baserades på en P2P-arkitektur som utvärderades för syftet. Resultatet visade på en omogen teknik, att ett centraliserat koncept har begränsningar och att arkitekturen bör vara mer komponentbaserad samt erbjuda flexibla mekanismer för synkronisering. Därför fortsätter arbetet istället med att implementera en mer beständig, genomarbetad lösning som svarar mot dessa krav, samt mer avancerade funktioner för grupphantering, synkronisering och generiska gränssnitt för verktyg.

Det som eftersöks är en distribuerad arkitektur, som innehåller stöd för grupphantering, tidshantering, synkronisering mm. En tänkbar teknik för detta är till exempel WS, men en arkitektur som pga. M&S-tillämpningen redan låg nära till hands, och som har de eftersökta egenskaperna, är HLA. HLA är ursprungligen avsett för distribuerade simuleringsprocesser, men har på senare tid även börjat utnyttjas till andra typer av applikationer, såsom distribuerade *multiplayer*-spel [webb: Magnetar] och samordning av distribuerade applikationer, exempelvis [Shen et al. 99]. I detta fall utnyttjas redan existerande funktionalitet i HLA för att implementera gruppsamordning och verktygsgränssnitt för datorbaserad samverkan. För att infrastrukturen ska implementeras på ett plattformsoberoende och återanvändbart sätt (oavsett om HLA som teknik byts ut), så valdes XML för att beskriva och formulera all information som utbyts i ett samarbete och mellan verktyg. Detta medför även att information som

skickas mellan olika heterogena klienter, kan *parsas* (tolkas och presenteras) utifrån klientens specifika behov.

Implementering: Grupptjänster

För grupptjänster utnyttjas bl.a. *HLA Federation Management* för att hantera grupper och *HLA Time Management* för att säkerställa *consistency*[4] mellan deltagarna. En annan HLA-specifik funktion är *Data Distribution Management*, som undersöks för syftet att kunna skapa subgrupper, samt för att filtrera information, utifrån specifika egenskaper och önskemål. För att klienten ska kunna använda HLA-funktioner måste den ha ett gränssnitt mot HLA. Detta utgörs i praktiken av en federat som CC-modulen skapar dynamiskt och som är konfigurerad för klienten, *Client Federate* (se Figur 3.4 nedan). *Client Federate*, och den funktionalitet CC stödjer den med, är baskomponenten för att en klient ska kunna samarbeta i NätSim. Den ansvarar för klientens roll i samarbetsgruppen, koppling till verktygen, grupphantering mm., samt att en klient uppdateras korrekt om den inträder i en existerande grupp, eller om den har befunnit sig utanför en grupp en tid och återkommer.



Figur 3.4: Arkitekturkoncept för datorbaserad samverkan baserad på XML och HLA. För klienten finns en klientspecifik ClientFederate *och för varje verktyg, här två st., finns en* ToolFederate. *XML-specifikationerna utgör den gemensamma informationen för CC.*

För att hantera samarbeten behöver information för och konfigurering av grupper lagras på ett gemensamt format. För detta utarbetas en informationsmodell i XML och CC ansvarar för att skapa en sådan för varje samarbete. Informationsmodellen omfattar information såsom:

- Gruppnamn och beskrivning av samarbetet
- Gruppmedlemmar och deras aktiva status
- Verktyg som samarbetet utnyttjar
- Inställningar för administration, säkerhetsnivåer, arkitektur (roller) etc.

En mer konkret uppställning av de huvudsakliga tjänster som CC tillhandahåller för grupphantering, se Tabell A.2 Appendix A.

[4] *Consistency* är ett engelskt begrepp som inte är helt översättningsbart till svenska. Det betyder ungefär "utan inre motsägelser", och kan översättas med "konsekvens" eller "logisk koherens". Begreppet avser alltså för sammanhanget att deltagares vyer inte skiljer sig från varandra utan uppträder identiskt.

Implementering: Applikationsgränssnitt

Generiska gränssnitt är en avsaknad som ofta behandlas i existerande CSCW-system idag, exempelvis [Sun et al. 04]. Det finns få CSCW-miljöer som kan stödja att flera olika typer av applikationer ”delas”, (ett exempel är NetMeeting [webb: NetMeeting]). Dessa är dessutom oftast *screendump*-baserade[5], eller kommersiella och inte öppna för vidare utveckling. I de flesta fall erbjuds datorbaserad samverkan endast som en extra tjänst i ett visst verktyg, vilket medför att ingen samordning finns om två olika typer av verktyg ska användas i ett samarbete. För att svara mot detta problem, tillhandahåller NätSim generiska gränssnitt för verktyg, dvs. stödjande mjukvara och API:er, för att utvecklare ska kunna integrera verktyg i plattformen och för datorbaserad samverkan.

För att synkronisera verktygen behövs tidshantering och mekanismer, vilket här implementeras mha. HLA, samt ett gemensamt format för att formulera den information som uppdateras på. Det senare utgörs av en generisk informationsmodell i XML och hanterar uppdateringar av två olika typer, *Status* och *Events*. *Status* beskriver ett verktygs momentana tillstånd, inklusive alla aktuella objekt, förhållanden och status, medan en *Event* beskriver en enskild händelse för exempelvis ett enda objekt. Båda typerna följer samma *XML Schema*[6]. Alla uppdateringar transporteras via HLA och all synkronisering av verktygen utförs mha. HLA:s *Time Management* (och med konservativa och optimistiska mekanismer). För detta måste även verktyget ha en federat att kommunicera genom. En lösning är att varje klient har en enda federat som alla verktyg kommunicerar genom och som även gruppspecifik information kommuniceras genom. Detta innebär dock att den federat som utnyttjas endast stödjer en typ av tidshantering och även att andra funktioner inte kan göras applikationsspecifika. Därför låter vi istället varje verktyg ha ett gränssnitt mot HLA. Verktyget tar således självt ansvar för uppdateringar mm. Gränssnittet utgörs av en generisk federat, *Tool Federate*, som skapas av CC för varje applikation och dess specifika behov (se Figur 3.4). Förutom denna funktion måste CC även tillhandahålla stöd till applikationen för att kunna använda CC-tjänsterna och presentera information enligt den gemensamma informationsmodellen. Information och uppdateringar måste mha. CC kunna skickas, tas emot, tolkas och synkroniseras på applikationsspecifika sätt En mer konkret uppställning av de huvudsakliga tjänster som CC tillhandahåller för applikationer, se Tabell A.3 Appendix A.

3.1.3.3 Synkroniseringsmekanismer för konsistenta samarbeten

En av de viktigaste frågeställningarna för CSCW är synkronisering, dvs. hur *consistency* mellan deltagarvyer kan säkerställas och med vilken metod. Som exempel kan en strikt konservativ synkroniseringsalgoritm vara nödvändig då samarbeten måste garanteras helt konsistenta. Men i vissa fall, då många interaktioner utbyts, är det omöjligt att med så strikta ramar erbjuda realtids-uppdateringar utan att uppdateringarna drabbas av långa tidsförskjutningar. I dessa fall kan en relaxering av kraven på synkronisering och *consistency* erfordras. För syftet studeras därför möjligheter med dynamiska synkroniseringsalgoritmer, som är en intressant forskningsfråga och som ses som en avsaknad och behöver studeras mer inom CSCW-området [Chung et al. 04]. Frågan kommer delvis att behandlas och implementeras inom ramen för ett av examensarbetena (generiska verktygsgränssnitt).

Parallellt med detta utförs även en utvärdering av olika synkroniseringsmetoders lämplighet för varierande typer av samarbeten, utifrån främst *antal interaktioner*. Utvärderingen sker liksom implementeringen i HLA. Metoder som studeras är: konservativ, optimistisk och tidssynkron synkronisering. För utvärderingen har ett spel implementerats, som har anpassats till CSCW-arbete för att spelas tillsammans och distribuerat, där synkronisering av deltagarna och kommunikation hanteras via HLA. Applikationen fungerar som en testplattform där synkroniseringsmekanismer utvärderas mot realtidssamverkan, vilket påbörjas efter årsskiftet.

---

[5] Med ”*screendump*-baserade” avses här den typ av verktygsdelning där uppdateringar och interaktioner i verktyget skickas som hela *screendumps* (bilder) av datormiljön. Att skicka så stora bilder i realtid orsakar mycket *overhead*.

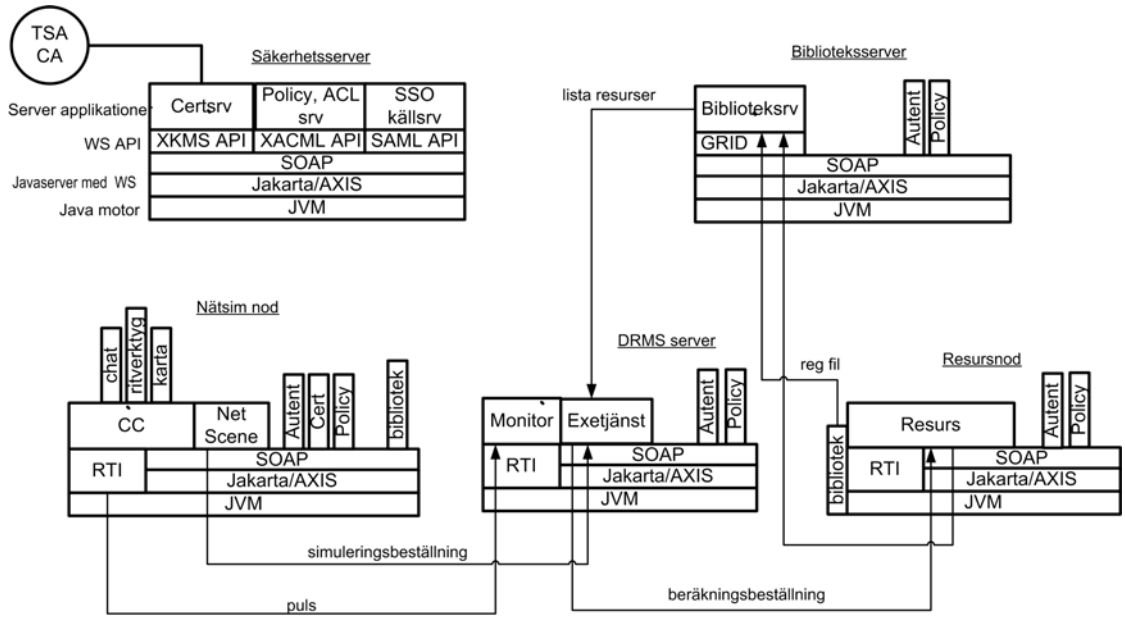
[6] *XML Schema* är mallar i vilka definieras ett antal regler som ett XML-dokument ska följa, t.ex. struktur och semantik för innehållet. Mallen kan sedan användas för att validera innehållet i dokumentet och att det beskrivs på en form som är enhetlig och tolkas på ett korrekt och gemensamt sätt.

3.1.4 Säkerhetslösning

För att implementera tjänsternas säkerhetsbehov krävs en del serverfunktionalitet samt logik i noderna för att stödja tjänsterna och applikationerna. För att kommunicera säkerhetsparametrar, meddelanden, nycklar och rättigheter mellan noderna så behövs ett standardiserat API.

Inom *Web services* finns en del intressanta säkerhetsgränssnitt som kan användas [Bengtsson et al. 03]. *SOAP* är en standard för att paketera XML-meddelanden och binda dem till ett kommunikationsprotokoll som t.ex. HTTP. Ett annat alternativ till *SOAP* är XML-RPC som inte visas här. XKMS utgör ett gränssnitt för bl.a. nyckelhantering (se t.ex. Appendix B och även PKI samt CA). SAML är ett ramverk som klarar att transportera autentiseringsinformation inom och utanför organisationsgränsen. XACML specificerar *policy* regler samt API för åtkomst av dem [Nagappan et al. 03].

Figur 3.5 åskådliggör säkerhetskomponenter och kommunikationsgränssnitt som behövs. De vertikala boxarna anger att det handlar om nödvändig logik på klientsidan. De horisontella innehåller server- och grundläggande funktionalitet i noderna. Figuren visar även lite av interaktionen som pågår mellan tjänsterna, vilken är intressant ur ett säkerhetsperspektiv. Det kan finnas både fler NätSim- och resursnoder i systemet, på bilden är för enkelhets skull bara en av varje sort inritad. Interaktionen mellan de olika NätSim-noderna är intressant, men syns tyvärr inte i bilden: den som startar upp simuleringen ska typiskt bevilja rättigheter för andra användare att delta.



Figur 3.5: Överblick av säkerhetskomponenter samt interaktioner mellan tjänster i NätSim.

Säkerhetsservrar:

- TSA *Certificat Authorization* (CA) server. Genererar privat/publikt nyckelpar för den svenska militära organisationen. Denna server måste emuleras eftersom funktionaliteten ännu inte finns.

- Certifikat och nyckelhanteringsserver. Pratar XKMS och döljer bakomliggande PKI med TSA CA. Kunna ta fram och individuellt överföra (mha individuellt asymmetriskt krypto) ny symmetrisk nyckel för exempelvis IPsec, säker chatt osv.

- *Policy*-server. Användargränssnitt för att skapa ACL-lista, vilken eventuellt även stödjer tidsbegränsningar och signalskyddsklasser. Kommunicerar med XACML API med andra *Web services*. Verifierar trippeln vem, objekt och operation mot *policy* regler. Kan spara inställningar per användare eller gruppnivå, samt skapande av grupper.

- *Single Sign-On* källserver. Autentiseringsserver för kommunikation inom och mellan domäner eller organisationer. Pratar SAML mot andra *Web services*.

För att en användare ska få logga in i NätSim så måste den först skapas med lösenord, samt få rättigheter tilldelade i *Policy*-ACL-servern. Efter inloggning måste ett certifikat med tillhörande publik och privat nyckel genereras, för att andra användare och tjänster ska kunna verifiera personen. Om genereringen sker i TSA-CA-servern så uppstår ett distributionsproblem av den privata (hemliga) nyckeln. Den får inte överföras via nätverket utan måste antingen skickas med diskett eller via smarta kort. Om den förs över med diskett så måste nyckeln lagras lokalt på hårddisken. Denna måste naturligtvis krypteras och får bara vara synlig för rätt användare. Den måste vidare lagras på alla maskiner som användaren kan tänkas logga in från.

Egentligen bör SSO inom och över organisations- eller domängränsen implementeras med Kerberos-servrar eller liknande [Bengtsson et al. 03]. Det måste finnas minst en Kerberos-server per domän. Applikationerna måste göras Kerberos-beroende eftersom servern ska tillfrågas för varje ny destination. Om servern fallerar så fungerar inte systemet, vilket är mycket olämpligt för ett *peer-to-peer*-nätverk. Det går att installera redundans med flera servrar, men centralstyrningen kommer man inte ifrån.

En trevligare lösning för SSO vore att använda sig av certifikat och publika nycklar som kan buffras i noderna. Första gången som två noder ska kommunicera med varandra så kan de verifiera varandra med certifikatservern. Därefter behövs egentligen bara certifieringservern för att distribuera revokeringar (invalidering eller återtagande) av befintliga certifikat. En autentiseringstjänst på noderna gör att användaren bara behöver logga in en gång, och tjänsten döljer alla autentiseringar som måste göras mot de andra tjänsterna. Detta är inte äkta SSO, men upplevs som det för användaren. Eventuellt behövs inte SSO källservern, certifikatservern kan vara tillräcklig om man väljer denna distribuerade implementation. Det får undersökas vidare om certifikatservern kan prata SAML.

Säkerhetstjänster på klienter:

- Certifikattjänst:
  - Ny användare ska kunna begära generering av certifikat med publik och privat nyckel.
  - Verifiera användare (autentisera) antingen med buffrad publik nyckel, eller via certifikatserver.
  - Kommunicerar med XKMS API.
- Autentiseringstjänst:
  - Inloggning (buffrad autentisering så att det ser ut som SSO lokalt på domänen).
  - Kommunicerar med XKMS och eventuellt SAML API.
- *Policy*-tjänst:
  - Administratör ska kunna skapa ny användare.
  - Verifiera mha. En ACL-server vad användare och tjänster har för rättigheter.
  - Stödjer XACML API.
- Dessutom behövs kryptering med SSL, HTTPS eller på XML dokumentnivå mellan två enskilda noder för att utbyta information.

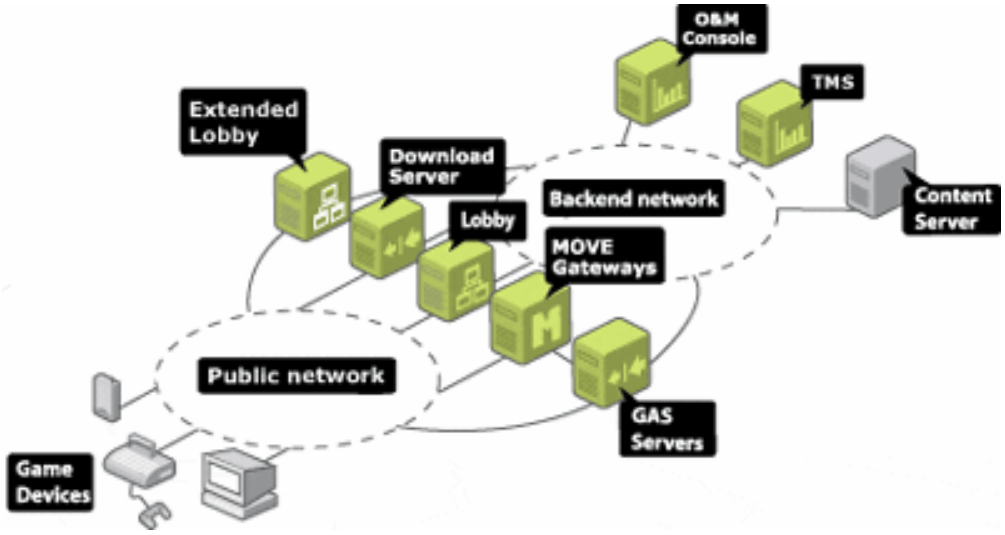
Gränsdragningen mellan certifiering och autentisering vad det gäller verifiering av användare är här en aning oklar. Implementationsbeslut i Java kommer att klargöra designen.

3.2 Koppling mellan HLA och Terraplay

En egenskap som eftersträvas i NätSim-arkitekturen är att komponenter på ett lättillgängligt sätt ska kunna kopplas samman, även om de är utvecklade i olika arkitekturer. Under året satsade projektet att koppla samman HLA-arkitekturen och TS. Genom att koppla samma HLA och TS finns det en potential att kunna ha tillgång till ett stort antal kommersiella nätverksbaserade spel.

3.2.1 Terraplay System

Som även nämnts tidigare är *Terrapay System* (TS) en generisk infrastruktur för spel med många deltagare, som kan hantera olika typer av spel och nätverk. Som exempel har TS använts som nätverkslösning för utveckling av PlayStation®2 spelet ”This is Football” från Sony Computer Environment. TS erbjuder två system, ett för fasta nät; *Terraplay System*, och ett för mobila nät, *Terraplay MOVE*. *Terraplay MOVE* är baserad på *Terraplay Systems* kärna och systemen kan kombineras för att erbjuda spel på både mobila och fasta noder som en enhetlig tjänst. TS är tänkt att erbjuda en komplett lösning med nödvändiga verktyg, komponenter och tjänster för att stödja nätverksbaserade spel och att göra dem tillgängliga för konsumenter via en kommersiell speltjänst (se Figur 3.6).



Figur 3.6: Schematisk bild av den komponentbaserade arkitekturen för Terraplay [webb: Terraplay].

Systemet har avancerade nätverkstekniker och egenskaper, och är optimerat för att erbjuda nätverksbaserade *online* speltjänster med relativt hög prestanda. Systemet är skalbart och har bl.a. stöd för TCP, XML, http, UDP, CDMA, GSM, 2.5G och 3G.

TS har två synkroniseringsmodeller, klient-server och *peer-to-peer*. Klienter kommer åt spelen via GAS (*Game Access Server*). GAS hanterar datakommunikation mellan spelare, lagrar sessionsdata och styr och anpassar datamängden till den tillgängliga kapaciteten. *Terraplay Management System* hanterar och allokerar systemresurser för att optimera prestanda och tillgänglighet. Det övervakar alla komponenter och verifierar (*Authenticates*) externa komponenter. TS tillhandahåller ett API, som möjliggör det för spelutvecklaren att komma åt de tjänster som TS erbjuder och utveckla spel med många deltagare.

3.2.2 HLA och TS

Nätverksbaserade spel är ett av de mest växande områden inom den kommersiella sektorn. Idag finns det ett stort antal högkvalitativa och relativt billiga spel på marknaden. Genom att länka samman HLA och TS finns det en möjlighet att komma åt ett stort antal spel till en relativt låg kostnad. Arbetet under 2004 har handlat om att studera TS, jämföra TS och HLA, samt utveckla en brygga för att länka HLA-federater med TS-spel. En fördel med denna koppling är att eftersom TS är mer lämpad för realtids-simuleringar, kan man tillåta simuleringar med högre realtidskrav koppla sig mot varandra genom TS och utföra givna scenarion. Senare kan resultatet från dessa simuleringar föras över till HLA-sidan, där simuleringar som har lägre realtidskrav körs. Ett bra exempel är aggregerings/disaggregerings-scenarion, där krigsspel förs på högre aggregeringsnivå (lägre resolution). Vid givna tillfällen när vissa villkor uppfylls eller då spelledaren önskar så zoomar man in i spelen genom att disaggregera (hög resolution) modellerna. De aggregerade modellerna skulle kunna köras i HLA-miljön, medan de disaggregerade modellerna körs i TS.

Arbetet med kopplingen mellan HLA och TS formulerades som ett examensarbete i år och genomfördes av två examensarbetare i samarbete med KTH och Blekinge Tekniska Högskola (BTH). Arbetet var lyckat och resulterade i en brygga som klarar av att koppla HLA-federater och TS-spel för

ett bestämt scenario bestående av ett antal pansarvagnar som slåss mot varandra. Nästa steg är att generalisera denna brygga för att kunna hantera kopplingen mellan olika typer av federater och spel.

3.3 Applikationer

NätSim är en plattform inte bara för gemensam M&S inom försvaret, utan även en gemensam miljö för projektarbete och resurstillgång. I miljön kommer olika verktyg att tillhandahållas, såsom M&S-verktyg och andra. Nedan beskrivs tre verktyg som studeras för integrering i NätSim.

3.3.1 NetScene

NetScene är en scenarioredigerare och en enkel scenariomotor för distribuerade simuleringar framtagen som en delprodukt i FOI-projektet *Modelling and Simulation for Analysis and Research Test-bed* (MOSART) [webb: MOSART]. NetScene befinner sig till dags dato fortfarande i utvecklingsstadiet, men har på relativt kort tid utvecklats till ett användbart verktyg (se Figur 3.7 nedan).

Figur 3.7: Skärmdumpar av NetScene [webb: MOSART].

Projektet NätSim har ett intresse i NetScene i och med att den funktionalitet som skall erbjudas till del behöver samlas i en gemensam applikation/användargränssnitt för de tänkta användarna. Exempel på relevant funktionalitet som måste lyftas upp i ett användargränssnitt som direkt berör slutanvändare är det distribuerade filsystemet som tillhandahåller lagring av komponenter, sökfunktioner gentemot den tjänstebaserade exekveringsmiljön, verktyg för distribuerat samarbete med mera. NetScene har redan en del av den funktionalitet som efterfrågas av NätSim-användare, framförallt som ett verktyg för att sätta ihop en simulering av komponenter. Med NetScene som grund har därför NätSim börjat vidareutveckla NetScene för att implementera de önskemål som NätSim har avseende funktionalitet.

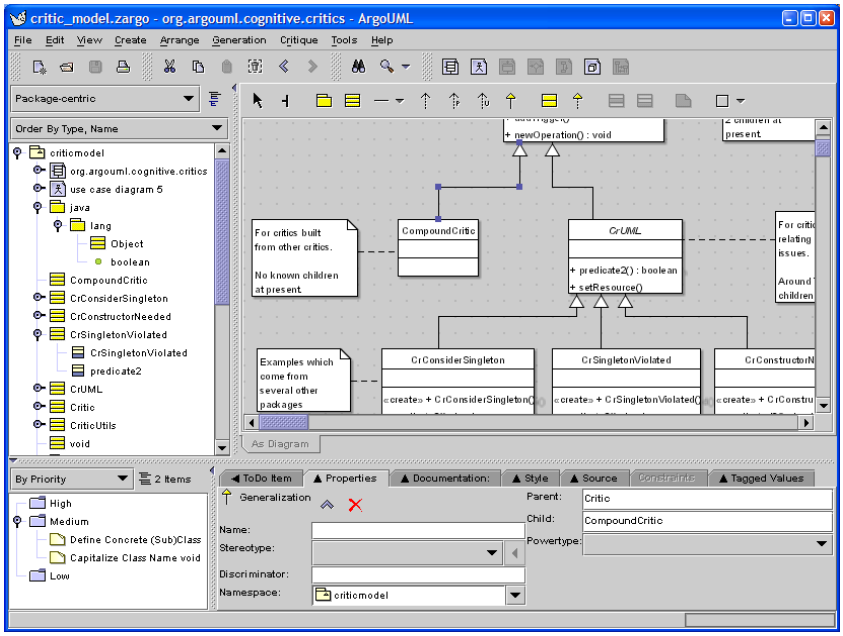
NetScene baseras i sin tur på NetBeans IDE som är en fri och öppen källkodsbaserad utvecklingsmiljö för mjukvaruutveckling [webb:NetBeans]. NetBeans IDE är framförallt ett utvecklingsverktyg för att utveckla mjukvara baserat på programmeringsspråket Java. NetBeans-projektet tillhandahåller, utöver utvecklingsmiljön, funktionalitet som gör att det relativt enkelt går att lägga till egenutvecklade moduler/funktioner som till exempel moduler för andra programmeringsspråk, modellering med *Unified Modelling Language* (UML) med mera. Modulbyggande i NetBeans baserar sig på *NetBeans Open APIs* [webb: Open IDE] som erbjuder ett generellt ramverk för att utveckla skrivbordsapplikationer. *NetBeans Open APIs* gör att man kan lägga till funktionalitet utan att behöva ändra i själva koden i grundapplikationen NetBeans IDE. Det är detta modultänk som NetScene nyttjar sig av, det vill säga man har med NetBeans som grund utvecklad NetScene med ett modulbaserat angreppssätt för att skapa sig en egen skrivbordsapplikation för sina egna behov.

3.3.2 ArgoUML

ArgoUML är ett verktyg för analys och design av objektorienterad mjukvara [webb: ArgoUML]. Verktyget baseras direkt på UML 1.3-specifikationen och till skillnad från många andra UML-verktyg följer ArgoUML specifikationen exakt. ArgoUML stödjer bl.a. OCL[7] (*Object Constraint Language*) samt XMI (*XML Model Interchange Format*). Produkten är inte ett kommersiellt verktyg med mängder av

[7] OCL är ett språk för att beskriva krav och ramar för objekt i UML.

funktionalitet likt kända modelleringsverktyg såsom Rational Rose [webb: Rational], utan utvecklades ursprungligen som ett forskningsprojekt som numera är gratis och *open source*. ArgoUML vidareutvecklas öppet och alla är välkomna att delta i projektet.



Figur 3.8: Exempelbild av användarvy för modellutveckling ArgoUML [webb: ArgoUML].

För M&S och de aktiviteter som ska utföras i NätSim är ett modellutvecklingsverktyg likt ArgoUML önskvärt (se Figur 3.8 ovan för en exempelvy av verktyget). Denna typ av verktyg används i de tidiga faserna av M&S-processen och till exempel i de tidigaste stegen i FEDEP-processen. ArgoUML stödjer dessutom konvertering av UML-diagram till både Java-kod och XML (XMI). Att verktyget har öppen källkod och är helt utvecklat i Java gör verktyget lämpligt för integrering i NätSim. Verktyget utvärderas nu i CSCW-aktiviteterna i projektet, till en början för syftet ”generiska gränssnitt” och alltså även för integrering i NätSim-plattformen.

3.3.3 CEPA 11.13

Euclid RTP (Research and Technology Programme) 11.13 var ett stort samarbetsprojekt som bedrevs under november 2000 till november 2003. Målet med programmet var att överkomma hindren för ett utbrett europeiskt utnyttjande av syntetiska miljöer (simuleringar) genom att utveckla en process med ett antal integrerade prototypverktyg, för att minska kostnaden och tiden för utveckling och användandet av simuleringar för träning, uppdragsgenomgång och simuleringsbaserad materielförsörjning (SBM). Projektet resulterade i ett antal verktyg och november 2003 genomfördes en stor demonstration i Paris för att redogöra resultatet.

Under året har vi försökt titta på några av dessa verktyg för att undersöka möjligheten att integrera vissa i NätSim-miljön. Dessa verktyg är bra exempel på applikationer som kan utnyttja tjänsterna i miljön. Under nästa år skall vi integrera ett par av dessa.

4 Kvalitetssäkring

4.1 Kunskapsspridning

NBMS är ett område med stor potential för framtidens försvar och industri. Det är många forskare och organisationer, både nationella och internationella, som har verksamheter som tangerar detta område. Därför är det av stor vikt att kunskap inom området sprids och att skapa samarbete för att undvika dubbelarbete. Det är vidare intressant att samla kompetenta intressenter och skapa forum för diskussioner inom ämnet. Därför har en viktig del av projektverksamheten varit fokuserad på att sprida kunskap, etablera nätverk och skapa sådana fora. Förutom att presentera artiklar om vår forskning, vid nationella och internationella konferenser, satsade projektet på att hålla föredrag om verksamheten vid olika tillfällen för FM, FMV, FOI, industrin och internationella organisationer. Dessa interaktioner har resulterat i värdefulla återkopplingar som har bidragit till att öka forskningens relevans samt hålla den på rätt nivå.

4.1.1 Referensgrupp

För att kvalitetssäkra och förankra arbetet och resultaten har en referensgrupp på tre personer bildats, med representanter från FM och FMV. Avsikten är att samla referensgruppen 2-3 gånger per år och presentera pågående arbete, olika vägval och resultat. Projektet hade ett möte med gruppen under 2004, vilket har resulterat i värdefulla synpunkter och *feedback*.

4.1.2 Presentationer och seminarier

NätSim-projektet presenterades och demonstrerades vid flera tillfällen under året. Bl.a. redovisades projektet på FOIs informationsdag om MoS februari 2004. Projektet presenterades också vid olika besök, bl.a. för besökare från Singapore (*Defence Science & Technology Agency* - DSTA och DSO) och Norge (Forsvarets forskningsinstitutt - FFI).

4.1.3 Artiklar

Under 2004 har två artiklar producerats och presenterats. En var en tidskriftsartikel till *SCS Simulation* (*Special editions: "Modelling and simulation applications in cluster and grid computing")*. Artikeln handlade om NätSims distribuerade resurshanteringsmiljö. Den andra var en konferensartikel som presenterades vid SIMSafe´04 och beskrev NätSim och våra visioner, tankar och tillvägagångssätt.

4.2 Samarbeten

Under projektets verksamhet har samarbeten upprättats och utförts med både Universitet och Högskola (UoH), projekt inom FOI och med andra organisationer. Detta stycke redogör för dessa samarbeten.

4.2.1 Inom FOI

Eftersom NätSims mål och ambitioner är på en allmän försvarsnivå och NätSim-miljön inte är intressant utan innehåll (modeller) och användare, har projektet haft som avsikt att samarbeta med projekt inom och externt FOI. Som exempel har ett samarbete initierats med projektet MOSART [webb: MOSART] på FOI, Ledningssystem i Linköping. Målet med samarbetet är bl.a. att ta fram ett scenariogenereringsverktyg som utnyttjar de tjänster som erbjuds av NätSim-miljön, och arbeta fram ett modellbibliotek som kan utnyttjas i projektet. En första version av scenariogenereringsverktyget finns och vi arbetar med att integrera det i NätSim-plattformen. Detta arbete planeras att bli klart till slutet av 2004 eller början på 2005 (diskuterades även i Avsnitt 3.3.1).

Ett annat intressant samarbete som har planerats under året är tillsammans med Avalon-projektet på institutionen för Flyg och Autonoma System. Avalon är ett referensbibliotek med egen arkitektur för sammansättning av modeller. Målet med samarbetet är att integrera Avalon i NätSim-plattformen. Det arbetet planeras att utföras under våren 2005.

Det görs också en samordning med FOI-projektet, ”Datorgenererade styrkor” för att dels utveckla modellbiblioteket och dels förse det med modeller av datorgenererade aktörer/stridskrafter. Dessutom sker ett kontinuerligt utbyte av information med andra projekt inom FOI genom att medarbetare som deltar i projektet också är involverade i dessa projekt och agerar som bryggor för utväxling av idéer och lösningar.

4.2.2 Med UoH

Under året har projektet haft ett regelbundet samarbete med Kungliga Tekniska Högskolan (KTH). Detta skedde framförallt genom Rassul Ayani som är forskningschef vid institutionen för Systemmodellering och professor på Institutionen för Mikroelektronik och Informationsteknologi (IMIT) på KTH.

Under 2004 engagerade projektet 8 examensarbetare som behandlade olika aspekter av projekt-verksamheten. 7 av dessa examensarbetare handleddes av KTH och 1 av Blekinge Tekniska Högskola. Av dessa var 2 stationerade på KTH, 2 på *National University of Singapore* (NUS) och fyra på FOI. Dessa arbeten beskrevs tidigare i rapporten.

4.2.3 Med industrin

Under året etablerades ett samarbete med företaget Terraplay [webb:Terraplay]. Som framgick av beskrivningen i Avsnitt 3.2 är målet med samarbetet att integrera TS i NätSim-plattformen och länka ihop HLA och TS. Arbetet bedrevs framförallt inom ramen för två examensarbeten i samarbete med KTH.

Projektet har också haft samarbete med företaget Pitch AB, genom att utnyttja det RTI som har utvecklats av företaget.

4.2.4 Internationellt

institutionen för Systemmodellering har sedan 1998 haft kontakter och samarbete med NUS inom området distribuerad MoS. Under 2000 fördes diskussioner med NUS om NätSim, då den fortfarande befann sig på idéstadiet. Ungefär samtidigt som NätSim startades som en strategisk forskningskärna, initierades ett systerprojekt vid NUS. Målet var att projekten skulle undersöka alternativa lösningar och olika aspekter av Nätverksbaserad MoS och samarbeta framförallt genom utbyte av erfarenheter och framtagna resultat. Sedan dess har utbyte med NUS skett på olika sätt, bl.a. genom examensarbeten.

2004 genomfördes också ett gemensamt examensarbete tillsammans med NUS och KTH. På NUS har en arkitektur för komponentbaserad programvaruutveckling utvecklats, XVCL som presenterades i 2.1.1.2. Främsta målet med XVCL är återanvändning av programmoduler. Examensarbetet handlade om att studera XVCL och alternativa metoder för komponentbaserad systemutveckling, samt att ta fram återanvändbara HLA-federater med hjälp av XVCL-arkitekturen. Arbetet genomfördes under sommaren och hösten av två examensarbetare. Delar av arbetet utfördes i Singapore.

Under hösten 2004 hade projektet två möten med representanter från FFI (avdelningen för Ledningssystem), presenterade arbetet och diskuterade eventuella samarbeten. Diskussionerna har resulterat i att vi tillsammans skriver ett gemensamt projektförslag som skall fungera som ett pilotfall och utföras under 2005. Förslaget skall beröra ett för oss relevant och intressant problem som ligger i linje med planerat NätSim-arbete under 2005.

4.3 Konferenser

Under året besöktes ett antal konferenser, dels för planering av framtida bidrag från projektet, dels för att säkerställa att riktningen i vårt arbete sammanfaller eller kanske till och med är före utvecklingen i resten av området.

4.3.1 CSCW – Kärnforum för datorbaserad samverkan

CSCW-konferens, Hotell Hilton, Chicago, USA, 6-10 november 2004.

Den första CSCW-konferensen hölls 1986 och hålls sedan dess vartannat jämnt år och bistår med för CSCW-domänen värdefulla artiklar samt diskussioner mm. 1989 hölls den första europeiska motsvarigheten, *ECSCW*, och den hålls sedan dess vartannat ojämnt år. CSCW är kanske den enda konferensen som riktar sig enbart till CSCW-området. Den är ett forum för design, implementering, utveckling samt utvärdering av tekniker och metoder som utnyttjas för att öka och effektivisera samarbete inom och mellan grupper och organisationer. Konferensen inleddes på lördagen med *workshops*[8], samt diskussionsgrupper som främst riktade sig till universitetsgrupper och doktorander för att diskutera sin forskning, och fortsatte i veckan med artikelsessioner och paneldebatter.

På konferensen fanns förutom artiklar som genomgått riktiga *review*s, även så kallade *short papers*, som till skillnad från vid många andra konferenser även de genomgått extensiva *reviews*. *Posters*, *workshops*, demonstrationer och *tutorials*[9] var ytterligare medel för presentation. Artiklarna riktade sig till två grupper av människor varför artikelsessionerna varierade mycket i ämne, teknikorienterade med teknik och implementation i fokus, samt beteendeorienterade med fokus på sociala aspekter och utvärdering. Som diskuterades där så behandlar CSCW många olika områden, såsom *distributed computing*, *networking* och *social psychology*. Detta gjorde konferensen intressant, men bidrog till att de olika ämnena inte berördes så djupt. Den konferens som istället ansågs bättre och mer kvalitativ, trots sitt betydligt bredare fokus, var *CHI*[10] som rekommenderas för fortsatta konferensbesök.

Bland de områden som behandlades fanns t.ex.:

- *Open source* mjukvara och applikationer.
- CSCW för mjukvaruutveckling.
- Gemensam taxonomi för CSCW.

De frågor som poängterades som brister och avsaknader inom CSCW-domänen, och som härigenom bekräftade att NätSim-projektets verksamheter inte bara är tidsenliga utan även före sin tid, var bl.a.:

- Dynamiska arkitekturer för *consistency* (centraliserade och distribuerade m.fl.), se t.ex. [Chung et al. 04].
- Generiska infrastrukturer för delade applikationer (gränssnitt mm.), se t.ex. [Sun et al. 04].
- Realistiska tillämpningar (verklig tillämpning och utvärdering utifrån detta).

4.3.2 PDCS – Konferens för parallella och distribuerade system

PDCS, MIT, Boston, USA, 9-11 november 2004.

PDCS (*Parallel & Distributed Computing & Systems*) är en årligt återkommande konferens som hålls vid MIT (*Massachusetts Institute of Technology*). Konferensen organiseras av IASTED (*the International Association of Science & Technology for Development*) och var den 16: e i ordningen [webb: PDCS].

PDCS har relativt brett fokus och behandlar många aspekter av området distribuerade och parallella system, allt från *Grid*-miljöer till trådlösa nätverk. Av speciellt intresse för NätSim-projektet var de spår som behandlade *Grid*-miljöer, *Web Services*, allokering och schemaläggning av resurser, feltolerans och lastbalansering.

Det bestående intrycket av konferensen är att *Grid*-miljöer av många anses ha stor potential vad gäller integrering av distribuerade resurser i heterogena miljöer. Inom detta område avhandlades främst frågeställningar relaterade till schemaläggning av resurser, samt metoder för effektiv lastbalansering. Av intresse för utvecklingen av DRMS var bland annat profilering av beräkningsresurser för estimering av

---

[8] Med *workshops* avses här öppna arbetsforum där diskussioner kring särskilt relevanta ämnen tas upp och personer får en möjlighet att dela erfarenheter och identifiera frågeställningar samt möjliga lösningar.

[9] *Tutorial* är en kort genomgång (snabbkurs) i något för konferensen relevant ämne.

[10] CHI står för *Computer Human Interaction* och är en ledande konferens inom ämnet människa-dator-interaktion.

dess förmåga att utföra ett beräkningsjobb. Profilering av en beräkningsresurs kan baseras på kunskap om dess tidigare agerande i systemet och kan ge en uppfattning om dess tillförlitlighet. Slutmålet är att ta fram ett SLA (*Service Level Agreement*) som specificerar den QoS (*Quality of Service*) som en konsument kan förvänta sig av det distribuerade systemet. Vidare presenterades ett antal, förhållandevis enkla, algoritmer för lastbalansering som skulle kunna utgöra ett alternativ för effektivisering av simuleringsexekveringar inom DRMS. Många av dessa hade utvärderats genom simulering, dvs. de hade inte implementeras i en ”verklig” miljö. Det skulle vara intressant att testa och utvärdera ett antal av dessa i DRMS-miljön och finna en effektiv form av lastbalansering för HLA-federationer.

I ett *keynote speech* presenterade Nancy G. Leveson (MIT) sin syn angående *”Making embedded software reuse practical and safe”*. Detta anförande avsåg främst återanvändning av mjukvara i inbyggda system, mer specifikt kontrollsystem inom rymdindustrin. Inom denna domän finns många fall där återanvändning av mjukvara har fått katastrofala följder.

Vid återanvändning av mjukvara är det viktigt att inte enbart se på återanvändning av källkod. Det är kanske ännu viktigare att se på återanvändning av de resurser som tas fram under tidiga faser av utvecklingsarbetet, i form av dokumentation, simuleringsmodeller etc. Vid utveckling av mjukvara står ungefär 20% av arbetsinsatsen av programmering utifrån systemspecifikationen. Det finns således stora vinster att göra om även andra projektresurser kan återanvändas på ett bra sätt. Detta kommer att bli alltmer betydande då mer och mer kod kan automatgenereras utifrån konceptuella modeller eller andra specifikationer. I det ultimata fallet görs enbart förändringar av de konceptuella modellerna vid anpassning av programvara till en ny miljö.

Vidare ansåg talaren att betydelsen av objektorienterad mjukvaruutveckling är överskattad, till och med felaktig, inom vissa domäner. Vid utveckling av programvara för inbyggda system är det bättre att utnyttja proceduriell programmering (*functional decomposition*) av programkoden. Användning av OO i dessa sammanhang omöjliggör praktiskt taget validering av ett system, vilket är en kritisk del vid utveckling av kontrollsystem av skilda slag. OO är bra att tillämpa i många situationer, men bör inte appliceras i de sammanhang som det inte är lämpat för, t.ex. vid utveckling av inbyggda system.

4.3.3 ESS – Europeiskt simuleringssymposium

ESS-Symposium, Budapest, 17-20 oktober, 2004.

ESS sponsras av *The Society for Computer Simulation, SCS*. Inriktningen på konferensen var simuleringar inom industrin, vilket innebar allt från simuleringar inom telekomindustrin, logistik, ekonomi, hårdvara, robotar osv. Exempel på sessioner som kunde ses var: ”*Simulation in manufacturing and production”, ”Micro-simulations, Simulation in logistics”, ”Traffic and transport”, ”Simulation methodologies”, ”Methods and techniques”, m.fl.* Konferensens huvudinriktning ”simuleringar inom industrin” kunde lustigt nog inte direkt ses på vilka som deltog. De flesta var från universitetsvärlden och då från framförallt datalogi- samt elektronikavdelningarna. I år var det en ganska liten konferens, men enligt vad som sades brukar den vara åtminstone dubbelt så stor och ha mer fokus på schemaläggning och algoritmer. En av förklaringarna som gavs till varför det i år hade varit så pass få artiklar var att huvudmännen hade gått i pension och inte riktigt hade haft entusiasmen att organisera konferensen längre.

Keynote speaker var Dietmar P.F. Möller från Hamburg-universitetet som öppnade konferensen med en entusiastisk presentation om *Virtual Reality* (”*VR: Computational Modelling and Simulation for industry”*). Han började med att berättade allmänt om vad VR var för något, krav, tekniker och hårdvara och gick sedan in på olika tillämpningar. Allmänt var nivån på artiklarna och presentationerna ok, det var mest tillämpningsartiklar, men tyvärr lades genomgående inte så mycket tid på att presentera metoderna.

Trender som kunde ses var neurala nätverk, genetiska algoritmer, agenter och spelteori. Ord som knappt nämndes under hela konferensen var interoperabilitet, fidelitet, aggregering, disaggregering, verifiering och validering.

Övrigt

HLA var inte helt okänt och några hade tom. jobbat med det för att koppla ihop olika modeller. Genomgående kunde dock ses att distribuerade modeller eller standarder inte riktigt var något som det

lades något krut på. Många utvecklade egna system och såg till att anpassa modeller till sina system. *In-house* verkade vara ledordet. Konferensens relevans för NätSim-projektet är inte så hög. För att läsa konferensartiklarna se [ESS 04].

4.3.4 DS-RT – IEEE-konferens för Distribuerade simuleringar mm.

DS-RT-konferens, Budapest, 21-23 oktober, 2004.

DS-RT Sponsras av IEEE. Konferensens inriktning var Distribuerade simuleringar samt Realtidsapplikationer. Det var en liten konferens med endast en *session* åt gången. Det var dock många välkända ansikten och bra med tillfällen att nätverka. Konferensen kan sägas bestå av två delar, där den ena handlade om distribuerade simuleringar, med tyngdpunkt på webb, metoder, *Grids* och den andra om realtidstillämpningar. Majoriteten av realtidstillämpningarna var inom VR, specifikt för virtuella samarbetsmiljöer.

På denna konferens fanns två stycken *keynote speakers*. Den första var Anthony Steed från University College London som talade om *Virtual Reality* (”*Being there together?*”). Detta föredrag var ganska likt det som hölls på ESS-konferensen men med skillnaden att Steed berättade mer om den subjektiva upplevelsen av att vara innesluten i den virtuella verkligheten. Han berättade om olika experiment de hade utfört samt testpersonernas reaktioner. Mycket tydde på att upplevelserna i VR-miljöer var i hög grad verklighetsnära för testpersonerna. Det andra *keynote*-föredraget hölls av Dr. Axel Lehmann från *University of the Federal Armed Forces* i Munchen. Hans föredrag handlade om komponentbaserad modellering och simulering (”*Component-Based Modelling and Simulation – Status and Perspectives*”). Med utgångspunkt från svårigheterna med att hantera komplexa system över tiden tog han upp hur området för komponentbaserad modellering och simulering hade gått framåt och vilket status det har nu. Han tog upp hur man på ett strategiskt sätt skulle kunna förhålla sig till komplexa system genom att gå enligt devisen ”*divide and conquer*”, dvs. att bryta ner storlek och komplexitet till mindre och mer hanterbara komponenter.

Trender som kunde ses var *Grids*, *Web Services*, Distribuerade simuleringar och VR. Konferensens relevans för NätSim-projektet bedöms som hög i nätverkssyfte och som medel ur forskningsnivå. För att läsa konferensartiklarna se [DSRT 04].

5 Diskussion och framtida arbete

Resultatet från detta års aktiviteter bekräftar våra tidigare antaganden att det finns behov av en gemensam tjänstearkitektur som stöd för MoS-verksamheter inom t.ex. FOI. Vi anser att NätSim är en bra första insats för att tillgodose detta behov.

Vi är nöjda med utvecklingen så långt. Arkitekturen och våra insatser inom de olika delarna ser lovande ut. Vi skall fortsätta under nästkommande åren med att implementera alla delar i plattformen, koppla samman och utvärdera dessa mha. ett eller flera typfall och scenarier. Vi fortsätter även samarbetet och kopplingen mot andra relevanta projekt och verksamheter. Plattformen skall utnyttjas av andra projekt för att dels tydliggöra kundnyttan samt få återkoppling på vårt arbete. Detta kan hjälpa oss att identifiera och arbeta med ytterligare frågor eller aspekter som är relevanta för kunden. Bl.a. planeras ett samarbete med Merlin-projektet på FOI, med syftet att länka Merlin-arkitekturen till NätSim, för att t.ex. möjliggöra att köra Merlin-modeller i NätSim-miljön.

NätSim har initierat samarbeten internationella organisationer och universitet, en verksamhet som har varit fördelaktig för projektet i form av att sprida idéer, samt för att få input och bekräftelse på riktlinjer och tillvägagångssätt. Under åren har även ett flertal examensarbeten inkluderats i projektet, vilket dels har stärkt kopplingen till UoH, och dels har bidragit till nya idéer och en objektiv syn på delar av NätSim. Denna typ av samarbeten har gett värdefulla bidrag till projektverksamheten och de kommer att fortskrida även under nästföljande år.

5.1 Delspåren

Nedan följer diskussioner kring de separata deluppgifterna i NätSim och beskrivningar av möjliga framtida aktiviteter inom dessa.

DRMS

Implementeringsarbetet är långt ifrån avslutat och många av de funktioner som redovisas i Tabell 3.1 kommer att vidareutvecklas under 2005. För närvarande implementeras en exempelfederation som skall ligga till grund för testing av DRMS, speciellt för utvärdering av algoritmer för feltolerans. Under nästa år kommer arbetet inriktas mot att effektivisera hanteringen av feltolerans i systemet, i syfte att minimera den kommunikations-*overhead* som kan relateras till denna funktion. Vidare kommer även metoder för lastbalansering att undersökas för att se på dess potential för effektivisering av en simuleringsexekvering. För detta bör ett antal algoritmer utvärderas vid exekvering av ”skarpa” simuleringsmodeller, tex. via koppling till Merlin och/eller MOSART.

Biblioteket

Under nästa år kommer prototypen för sökningsprocessen att bli klar. Mellanlagret för biblioteket som tar emot och hanterar sökningen med hjälp av *Grid*-tjänster kommer att vara implementerat. En enkel ontologi och sökning med hjälp av den kommer att kunna utföras. En fortsättning på årets arbete planeras och då kommer fokus att ligga på den övriga funktionaliteten, dvs. att kunna fullfölja processen från utdelandet av en fil, resurs, modell och skapandet av dess metadatafil till sökning och användning, till läsning, modifiering och borttagande av det metadatafilen beskriver. De andra delarna i funktionaliteten i biblioteket kommer också att behöva implementera såsom t.ex. gränssnitt för att göra simuleringsmodeller och -komponenter tillgängliga.

Exempel på frågor som behöver besvaras framöver:

- Metadatafiler:
  - Hur hantering av metadatafiler som antingen har döda länkar eller ägs av någon som inte längre finns i miljön skall hanteras.
  - Hur skall replikering av modeller hanteras i biblioteket?

- Hur ska *consistency* för filer eller modeller som läggs ut för ankomst med hjälp av biblioteket kunna garanteras?

– Säkerhet:

  - Hur skall åtkomst kunna säkerställas för de användare som har rättigheten och nekas för de som inte har rättigheter?

  - Skall sökning på filer som användare inte har rättigheter till fortfarande kunna utföras men åtkomst nekas?

– Arkitektur:

  - Utvärdering av arkitekturval.

CC

Arbetet under hösten samt resultatet från de båda examensuppgifterna förväntas lägga grunden för den infrastruktur för CSCW i NätSim som kommer att börja implementeras under nästkommande två år. Nedan presenteras några av de aktiviteter och forskningsfrågor som kommer att beröras.

Aktiviteter som ska utföras avseende CC-modulen:

– Implementering av generiska verktygsgränssnitt

– Implementering av grupptjänster

– Integrering av kommunikationsmedel

– Tillämpning och utvärdering (genom praktisk tillämpning, t.ex. med ArgoUML)

Frågeställningar för fortsatt forskning:

– Dynamiska synkroniseringsmekanismer ses som en intressant frågeställning som kommer att hanteras dels genom en teoretisk utvärdering, samt ev. även implementeras.

– Att utnyttja WS i sammanhanget vore intressant och fördelaktigt, se Appendix A.

– Det vore även intressant att parallellt med arbetet göra en studie i framtida möjligheter för denna typ av applikationer, där inga gränser ses. Vad skulle t.ex. virtuella världar innebära för krigföring och internationella insatser, eller för krisberedning och krishantering?

Säkerhet

Implementering under 2005-2006 kommer att ske i följande prioriteringsordning:

– Autentisering mellan klienter i NätSim nätverk inom domänen. Ska fungera som *Single Sign-On*.

– Autentisering kompatibel med *Grid*-autentisering i klienter.

– Certifikat server som stödjer certifikat inom domänen, SSO.

– *Policy*-server med användar- och grupphantering som stödjer XACML.

– Gränssnitt mot servrarna för NätSims tjänster.

– Kryptering mellan 2 noder mha SSL, HTTPS eller XML kryptering av dokumenten.

Undersöka vidare:

– Autentisering med SSO utanför domänen mha SAML API.

– Möjligheter med *Grid*-säkerhet. Är det ett bättre ramverk för säkerheten i NätSim?

- Säkerhetsaspekter för metadata, NetScene och startapplikation.
- Kryptering av all trafik mellan alla noder med VPN (t.ex. IPsec).
- *Open source* och MicroSoft varianter av ett globalt *Single Sign-On Passport*.
- Minimera behov av stationär säkerhetsserver alt. förenkla installation av denna. Anpassa säkerhetslösning mot *peer-to-peer*-teknik.

Identifiera 2 examensarbeten inom lämpliga delar ovan.

NetScene

Genom att framgent använda oss av NetScene och konceptet med moduler i NetBeans har vi skapat oss en grund för att dels skapa ett slutanvändargränssnitt samt även öppna dörrarna för samarbete med andra FOI-projekt som väljer MOSART som integrationsplattform. NetScene kommer i sin grund att leva inom ramen för MOSART-projektet och genom detta säkerställs en viss redundans inom FOI avseende framtida utvecklarkompetens mot den plattformen. Genom att projektet MOSART samt deras verktyg har rollen som integrationsplattform för simuleringsrelaterade projekt inom bland annat FOI fås även en koppling till ”verkliga” projekt som indirekt kan verka som kravställare på NätSim och dess framtida utveckling.

Referenser

[Bengtsson et al. 01] A. Bengtsson, A. Hunstad, L. Westerdahl, *Autentisering i nätverksbaserade system.* FOI-R—0331—SE, FOI, Ledningssystem, Linköping, december 2001.

[Bengtsson et al. 03] A. Bengtsson, A. Hunstad, L. Westerdahl, *Identitetsverifiering över systemgränser.* FOI-R—1024—SE, FOI, Ledningssystem, Linköping, november 2003.

[Chatarji 04] J. Chatarji, *Introduction to Service Oriented Architecture (SOA).* Internetreferens: Open Source Web Development – Dev Shed. Tillgänglig från http://www.devshed.com/c/a/Web-Services/Introduction-to-Service-Oriented-Architecture-SOA/. Senast besökt november 04.

[Chung et al. 04] G. Chung, P. Dewan, Towards *Dynamic Collaboration Architectures.* Proceedings of ACM conference on Computer Supported Cooperative Work, Chicago USA, november 2004.

[DSRT 04] *Distributed Simulation and Real-Time Applications*, Edited by Stephen J. Turner, David J. Roberts and Linda F. Wilson, oktober 2004, ISBN 0-7695-2232-7.

[Eklöf et al. 04] M. Eklöf, M. Sparf, F. Moradi, R. Ayani, *Peer-to-Peer-Based Resource Management in Support of HLA-Based Distributed Simulations*, SIMULATION, Vol. 80, p. 181-190, maj 2004.

[ESS 04] *European Simulation Symposium*, Edited by György Lipovszki and István Molnár, oktober 2004, ISBN 1-56555-286-5.

[Fåk] V. Fåk, *Kompendium i IT-säkerhet.* Linköpings Universitet, odaterad.

[Försvarsmakten 04] P. Gruvö, *Krav för signalskyddssystem avsedda för Totalförsvaret.* MUST ITSA TSA, Försvarsmakten, 2004-03-04.

[He 04] H. He, *What is Service Oriented Architecture?* O'Reilly webservices.xml.com. Tillgänglig från: http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html. Senast besökt november 2004.

[Nagappan et al. 03] R. Nagappan et al., *Developing Java Web Services*, Wiley, 2003.

[NätSim 03a] M.G. Lozano et al., *Use Case – "Simulering av stridsplan"*, NätSim-projektet, Ursvik, 2003.

[NätSim 03b] M.G. Lozano et al., *Use Case – "Simulering som beslutsstöd vid logisk planering"*, Nätsim-projektet, Ursvik, 2003.

[Peltier 01] Peltier, *Information Security Risc Analysis.* Auerbach Publications, 2001.

[Schneier 95] B. Schneier, *Applied Cryptography.* Wiley Publications, 1995.

[Shen et al. 99] X. Shen, R. Hage, N.D. Georganas, *Agent-aided Collaborative Virtual Environments over HLA/RTI.* Proceedings of the IEEE/ACM Third International Workshop on Distributed Interactive Simulation and Real Time Applications (DIS-RT '99), University of Maryland College Park MD, USA, oktober 1999.

[Stallings 03] W. Stallings, *Network Security Essentials, Applications and Standtards, Second Edition.* Prentice Hall 2003.

[Stevens 94] W. R. Stevens, *TCP/IP Illustrated Volume 1. The protocols.* Addison Wesley, 1994.

[Sun et al. 04] D. Sun, S. Xia, C. Sun, D. Chen, *Operational Transformation for Collaborative Word Processing.* Proceedings of ACM conference on Computer Supported Cooperative Work, Chicago USA, November 2004.

[Ulriksson et al. 03] J. Ulriksson, F. Moradi, R. Ayani, *Collaborative Modelling and Simulation in a Distributed Environment*. Proceedings of European Simulation Interoperability Workshop, Stockholm, Juni 2003.

[webb: AFS] Hemsida för AFS. Tillgänlig från: http://www.openafs.org/. Senast besökt november 2004.

[webb: ArgoUML] Hemsida för ArgoUML. Tillgänglig från http://argouml.tigris.org/ Senast besökt november 2004.

[webb: Axis] Hemsida för Axis – Web Service plattform från Apache Software Foundation. Tillgänglig från: http://ws.apache.org/axis/. Senast besökt november 2004.

[webb: FHS-NBF] Försvarshögskolan Samordningsgrupp NBF hemsida. Tillgänglig från http://nbf.fhs.mil.se/. Senast besökt november 2004.

[webb: Globus] Hemsida för Globus. Tillgänglig från: http://www.globus.org/. Senast besökt november 2004.

[webb: gsi] Hemsida för *Grid Security*: http://www-unix.globus.org/toolkit/docs/3.2/security.html Senast besökt novemer 2004.

[webb: intrusions] Hemsida för *Technical intrusions, Internet Security Systems*, http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical/. Senast besökt maj 2004.

[webb: Jena] Hemsida för Jena - Verktyg (API) för hantering av RDF och OWL strukturer. Tillgänglig från: http://www.hpl.hp.com/semweb/jena2.htm. Senast besökt november 2004.

[webb: LedsystT] LedsystT hemsida, Försvarets Materielverk (FMV). Tillgänglig från: http://www.fmv.se. Senast besökt november 2004.

[webb: Magnetar] Magnetar Games hemsida. Tillgänglig från http://www.magnetargames.com. Senast besökt november 2004.

[webb: MOSART] MOSART Projekthemsida. FOI Intranät. Tillgänglig från FOI Intranät: http://www-int.foi.se/MOSART/. Senast besökt november 2004.

[webb: NetBeans] NetBeans hemsida. Tillgänglig från http://www.netbeans.org/. Senast besökt november 2004.

[webb:NetMeeting] Microsoft NetMeeting hemsida. Tillgänglig från: http://www.microsoft.com/windows/netmeeting/. Senast besökt november 2004.

[webb: OGSA-DAI] Specifikation för OGSA. Tillgänglig från: http://www.ogsadai.org.uk/. Senast besökt november 2004.

[webb: Open IDE] *NetBeans Open APIs Home Page*. Tillgänglig från: http://openide.netbeans.org/. Senast besökt november 2004.

[webb: OWL] *OWL – Web Ontology Language* hemsida. Tillgänglig från http://www.w3.org/2004/OWL/. Senast besökt november 2004.

[webb: PAST] Hemsida för PAST. Tillgänglig från http://research.microsoft.com/~antr/PAST/. Senast besökt november 2004.

[webb: PASTRY] Hemsida för PASTRY. Tillgänglig från: http://research.microsoft.com/~antr/Pastry/default.htm. Senast besökt november 2004.

[webb: PDCS] Hemsida för PDCS-konferens, *PDCS – Parallell and Distributed Computing and Systems*. Tillgänglig från: http://www.iasted.org/conferences/2004/cambridge/pdcs.htm. Senast besökt november 2004.

[webb: Rational] IBMs hemsida för produktsviten Rational Rose. Tillgänglig från: http://www-306.ibm.com/software/awdtools/developer/rose/. Senast besökt november 2004.

[webb: RDF] Hemsida för RDF – *Resource Description Framework*. Tillgänglig från: http://www.w3.org/RDF/. Senast besökt november 2004.

[webb: RLS] Hemsida för RLS. Tillgänglig från: http://www.globus.org/rls/. Senast besökt november 2004.

[webb: Terraplay] Hemsida för Terraplay. Tillgänglig från: http://www.terraplay.com/. Senast besökt november 2004.

[webb: XVCL] Hemsida för XVCL – *XML-based Variant Configuration Language*. Tillgänglig från: http://fxvcl.sourceforge.net/. Senast besökt november 2004.

[webb: W3C] Hemsida för W3C – *World Wide Web Consortium*. Tillgänglig från: http://www.w3c.org. Senast besökt november 2004.

[webb: WSRF] Hemsida för: WSRF – *Web Services Resource Framework*. Tillgänlig från: http://www.globus.org/wsrf/. Senast besökt november 2004.

Appendix A – Datorbaserad samverkan i NätSim

Exempelscenario: Internationell insats och logistikplanering med NätSim

En plötslig förfrågan från regeringen initierar en beredning om en internationell insats. Ett förband ska snabbt förflyttas till landet, med nödvändiga förnödenheter och materiel. En förtrupp sänds omedelbart ner till området ifråga för att rekognoscera samt förbereda uppsättning av förband.

I beredningen planeras många detaljer, varav en är det behövda antalet och typen av transporter och bränsle på plats och i strid. Berörd ledningscentral i Sverige kopplar upp sig mot NätSim, där erforderliga applikationer och resurser kan tillgås. För att snabbt kunna planera insatsen startar befälhavaren ett NätSim-samarbete och inkluderar logistikplaneringsverktyg. Berörd personal med fordonskompetens blir kontaktad och går också med i NätSim och samarbetet. För att dessutom få direkt information från insatslandet går även en officer på plats med i samarbetet.

Samtliga samarbetsparter ser och hör varandra genom webbkamera, samt kan gemensamt interagera med logistikverktygen. Tillsammans kan de trots tidskritiska omständigheter planera förnödenheter och transporter, dessutom utifrån direkt information från insatsområdet. Ledningspersonal avgör de största besluten. Officeren på plats informerar om de rådande omständigheterna, eventuella förstörda transportvägar samt terrängen. Med hjälp av den aktuella informationen kan expertkompetens stödja planeringen på bästa sätt.

Antaganden

Vid implementation av datorbaserad samverkan i NätSim gjordes två huvudsakliga antaganden kring grupper:

- Den typ av CSCW-aktiviteter som kommer att utföras i NätSim antas oftast vara direkt uppgiftsrelaterade, dvs. livslängden på samarbetet antas kort. Detta ställer inte lika höga krav på t.ex. sociala aspekter.

- De grupper som samarbetar antas vara små, 2-8 personer. Med avseende på skalbarhet och arkitektur kan större grupper formas, men inget socialt stöd finns för att stödja att så många personer samarbetar och interagerar (virtuell konflikthantering).

CSCW-roller i NätSim

I NätSim finns för närvarande tre olika CSCW-roller. En roll bestäms utifrån personprofil (säkerhetsnivå etc.), situation eller datormiljö (prestanda eller nätverksuppkoppling). De tre användarna är: *Reader*, *Writer* och *Administrator*, som visas i Tabell A.1 nedan. En ytterligare roll som en klient kan inneha är *Owner*, som innebär att den klienten är ytterst ansvarig för eventuella producerade filer från samarbetet, som lagras i NätSim-nätverket när samarbetet avslutas. Denna roll har inte något med rättigheter avseende aktivitetstyper att göra och är därför inte med i tabellen nedan.

Tabell A.1: Tre olika typerna av klientroller. En Reader *kan läsa men inte operera på en delad applikation. En* Writer *kan både och. En* Administrator *är den enda som kan ändra inställningar för ett samarbete och godkänna nya deltagare i gruppen. Det kan finnas flera* Administrators *i ett samarbete. Tabellen kan även beskrivas enligt Figur A.1 nedan.*

Actiontype	**Reader**	**Writer**	**Administrator**
Visualisation	X	X	X
Interaction		X	X
Admin./Config.			X

$$R \Rightarrow R$$
$$W \Rightarrow R \cap W$$
$$A \Rightarrow R \cap W \cap A$$

Figur A.1: Schematisk beskrivning av hur de tre rollerna i CC förhåller sig till varandra.

Den typ av samarbeten som avses ska kunna vara snabba samarbeten och som inkluderar många olika klienttyper (såsom *pocketPC*, *handhelds* etc.) och som kan ha begränsade nätverksuppkopplingar. Därför antas mer komplexa möjligheter för CSCW, såsom virtuella världar, för krävande och komplexa för att studeras för NätSim och sammanhanget. Eftersom NätSim tillåter distribuerat samarbete, med många olika klienttyper, innebär det dessutom att synkronisering av deltagarna inte kan ske snabbare än den långsammaste klienten (vid strikt synkronisering). Att klienten är långsam kan bero antingen på dålig nätverksuppkoppling eller att klientens prestanda är sämre än de andras. I sådana fall kan det vara lämpligt med olika synkroniseringsmekanismer i gruppen, såsom diskuteras under 3.1.3.3. Den svagare klienten kan till exempel hanteras optimistiskt, medan de andra hanteras konservativt. Detta kan i sin tur innebära att den svagare klienten endast kan agera Reader, eftersom den klientens interaktioner kommer att komma ”*out-of-bounds*” för de andra deltagarna. Denna övergång av mekanismer skulle kunna hanteras dynamiskt och är en intressant frågeställning för fortsatt arbete.

Vidareutveckling – WS för att öka tillgänglighet och flexibilitet

Om hela infrastrukturen för CSCW baseras på HLA och XML inte används, så kan HLA-specifik funktionalitet, såsom HLA-objekt och –attribut, användas. Detta låser dock implementeringen mycket hårt till HLA, vilket vill undvikas. Föregående års prototyp [JXTA] var hårt knuten till tekniken och för integrerad med verktygen, vilket var ett problem och som nu vill undvikas. Konceptet som nu istället används tillåter dessutom enklare integration av flera tekniker i CC, vilket diskuteras här.

Det finns flera olika spår att följa för att vidareutveckla och utöka CC. Ett intressant spår är att utnyttja *Web Services* för att göra CC mer *lightweight* och lättillgänglig. Detta kan uppnås genom att låta CC vara rent komponentbaserad och finnas tillgänglig i olika versioner, som är tillgängliga via en webbtjänst. På detta sätt kan en klient när den startar samarbetet ladda ner endast de specifika tjänster den behöver för just det samarbetet. Detta gäller även verktyg som används i samarbetet, och som klienten kanske inte har installerade. Då klienten ska använda verktyget kan det antingen laddas ner på förfrågan av klienten (CC alarmerar om att verktyget inte finns och frågar om det ska laddas ner), eller alternativt utnyttjas verktyget på distans och som en webbtjänst. Detta gör dessutom att kommunikationen från klient till verktyg faktiskt inte baseras på HLA, vilket ger ytterligare möjligheter.

Som diskuteras i [Chung et al] är olika typer av arkitekturer bra för olika situationer. Centraliserade arkitekturer är t.ex. bra att använda då mycket stora grupper samarbetar, då det dessutom är troligt att inte alla interagerar lika mycket, dvs. detta är ett hierarkiskt samarbete såsom används i distansinlärning, träning, demonstrationer etc. En distribuerad arkitektur (ibland omnämnd som ”Replikerad arkitektur”) är bättre för mindre grupper där samtliga deltagare interagerar och det inte är önskvärt att vara beroende av en enda centraliserad nod. I NätSim antas till en början som beskrivet ovan att grupperna är små, men det skulle vara fördelaktigt om även stora grupper kunde hanteras. I detta fall skulle arkitekturen inte behöva vara så distribuerad som den blir av att baseras på HLA, utan skulle kunna kombineras även i detta fall med en centraliserad arkitektur och tekniker såsom *Web Services*.

Autentisering och säkerhetsnivåer

Då en grupp skapas sätts en säkerhetsnivå på gruppen. Denna avgör vilka resurser och aktiviteter som gruppen har tillgång till i NätSim-miljön. Då en klient går med i en grupp sker en autentisering av personen. Denna baseras på två faktorer, dels måste övriga medlemmar godkänna den nya medlemmen, dels måste klienten ha minst samma säkerhetsnivå som gruppen har eller högre. Säkerhetsnivån kontrolleras innan klienten kan förfråga gruppen. Godkänns detta skickas en förfrågan till gruppen. Denna presenteras för alla deltagare, men kan endast godkännas av en *Administrator*.

CC-tjänster

De tjänster CC huvudsakligen tillhandahåller för grupphantering, och som delvis är implementerade idag, beskrivs i Tabell A.2 nedan (mer detaljerad beskrivning bifogas slutliga implementeringen):

Tabell A.2: CCs grupptjänster.

Tjänst	Beskrivning	Implementering
GroupConfig	CC skapar en Gruppkonfiguration med alla inställningar och deltagare i gruppen etc. Ändras dynamiskt under arbetets gång.	XML-dokument Transporteras via HLA
ConfigUpdate	Uppdatering av gruppens GroupConfig. Sker vid två tillfällen: 1) Uppdatering helt ny användare, sker vid *Acknowledgement* från gruppen 2) Uppdatering återkommande deltagare	XML-dokument Transporteras via HLA
GroupManagement	1) Starta, avsluta grupper mm. 2) Hantera medlemskap i fler än en grupp och samtidigt.	*HLA Federation Management*
ClientFederate	CC skapar dynamiskt en ClientFederate som är konfigurerad klientspecifikt samt efter GroupConfig.	HLA

De tjänster CC huvudsakligen tillhandahåller för verktyg, och som delvis är implementerade idag, beskrivs i Tabell A.3 (mer detaljerad beskrivning bifogas slutliga implementeringen):

Tabell A.3: CCs tjänster för applikationer.

Tjänst	Beskrivning	Implementering
Interface	Dokumentation, funktionalitet samt Guide-lines för att integrera verktyg för CSCW och utnyttja funktionalitet i CC.	XML-dokument Javadoc mm
EventUpdate	Uppdatering av specifika händelser och interaktioner (s.k. *Events*) för ett verktyg, propageras till samtliga berörda verktyg.	XML-dokument Transporteras via HLA
StatusUpdate	Uppdatering (initiering) av verktyg avseende hela *Status*, dvs. samtliga objekt och tillstånd vid en viss tidpunkt.	XML-dokument Transporteras via HLA
DistributedLock	CC tillhandahåller för verktygen ett distribuerat lås för att undvika interaktionskonflikter mm.	HLA, Ring Algorithm
ToolFederate	CC skapar dynamiskt en ToolFederate som är verktygsspecifikt konfigurerad, särskilt är tidshantering och uppdateringsmekanismer.	HLA

Appendix B – NetSim Security Risk Analysis

B Contents

B.1 Introduction

This chapter was written early in the project as a means for the initial security discussions of NetSim. It analyzes the system from the bottom and up with all kinds of security threats in mind. The intention was to identify the security needs for this project. It is important to note that securing a distributed system is a complex task, and it is never possible to make it perfect. It is a matter of selecting the right level of security, or the system may become unusable. But when focusing on some parts, other things always remain.

The considerations and requirements were very general to fit architecture changes to NetSim. Some things might not be relevant to the project.

B.2 Scenarios

A couple of scenarios were initially discussed to understand the security needs of the system. Some scenarios already existed, while others were invented. They were extended to identify roles of the users, assets that NetSim wants to protect and threats against these assets.

B.2.1 Simulation of war plan

The scenario is described in [NätSim 03a]. NetSim is used as a communication tool by its members to generate a final war plan.

Input: a world model (map, units, persons) and intelligence about the enemy.

The simulation focuses on time aspects in movements, logistic issues and the map.

Output: a war plan and the briefing.

Analysis: simulation seems to be communicated locally within staff. Are people not locally present at the site allowed to connect with a remote mobile device? If so, what kind of military hardware is required to secure link layer of the mobile device?

Roles: battalion chief, staff members, sub chiefs and super chiefs.

Assets: a war plan, a world model (information about units), intelligence about the enemy and the simulation/communication in NetSim (sensible data).

Threats: affect/change intelligence about enemy, eavesdrop communication with mobile device, eavesdrop connection to/within staff and disturb traffic to/within staff.

B.2.2 Simulation as decision help in logistics planning

The scenario is described in [NätSim 03b]. A simulation is used to determine logistics and resource needs.

Input: pictures, task timetable, task size, model library, databases.

Assets: tool that searches resources & models, collaborative computer environment, computer capacity.

Output: equipment needs, material usage, number of airplanes, people, logistics model, briefing.

Analysis: distributed simulation runs in own separate military network.

Roles: remote analyst (logistics expert FMV), HKV staff (administrators), F6 flight staff (delivery).

Assets: resources, models.

Threats: affect/change logistic needs, eavesdrop information about resources, coordinates, people and models, disturb logistics channel.

B.2.3 FOI researchers cooperating with commercial company and foreign remote researchers

Roles: FOI researchers, foreign researchers, company.

Assets: models that may only be interacted with if you are authorized.

Threats: copy model and data that model possess.

B.2.4 FOI researchers in cooperation with military

Roles: FOI researchers, military users.

Assets: models with restricted data.

Threats: unauthorized user access models.

B.2.5 Military in cooperation with civil authorities (KBM, Landsting)

Roles: military users, civil authority users.

Assets: models with sensitive data.

Threats: eavesdrop of unencrypted traffic (problems with shared keys, traffic over Internet).

B.3 Threat analysis

The scenarios in the previous section were used to identify possible user roles, assets, and threats against the system. This information together with other common system threats [Peltier 01; webb: intrusions] was combined to find possible attacks. The Swedish military requirements to structure sensitive information are presented to illustrate possible security levels within military communication systems. Finally, found attacks were structured in three popular academic groups: confidentiality, integrity, and availability threats. The treats are accompanied by suitable countermeasures.

B.3.1 Attackers

This section identifies possible attackers against NetSim. A list of legitimate users for the system:

- Military in the field and in command control
- Researchers
- Simulation developers
- Remote user looking up information
- Remote expert
- Cooperating government
- Cooperating foreign military force
- Mobile code, web service (softweare using the system)

Attackers were derived from the scenarios; others came from the list of legitimate users, and some where added.

As can be seen from the table B.1, the hacker and a foregin military force are the most probable attackers towards NetSim. The hacker gains such a high ratio if the system is network based and connected towards Internet, because he has automated tools that with little effort can perform new attacks frequently. If the system is taken off-Internet and is only connected to a private network, the regular hacker can not target it. The military force, on the other hand, is the most powerful attacker since it has new advanced technology available and economical resources to hire any of the other attackers. Since the military force is the worst case, it is considered to be the attacker in the following sections.

Table B.1: Signal protection levels.

attacker	*motive/gain*	*resources*	*probability*
insider/intelligence	steal research, sell info	access to network, physical machine	4
user (accidental mistakes)	not enough education, ignorance, selfishness, bad routines	access to network, physical machine	2
SW (software malfunction)	(not enough testing)	application control	2
hacker	fun, terror	Internet, hack tools, passwords	5
foreign military force	spy, conquer, sabotage	advanced equipment, Internet, it-weapons, weapons	5

terrorists	terror	Internet, weapons	1
thief	sell equipment	burglar tools to gain physical access	1

B.3.2 Assets

An asset can be sensible information (data) or resources that the owners want to protect. These are interesting for an attacker that wants to access it or make it break down (in other words the target of the attack).

A list of assets within NetSim:

- Computer capacity
- Backups
- Communication: messages, speech, maps, pictures, data, coordinates
- Sensible database informaiton
- Distributed file system
- Information from search: services, information, nodes, models
- Availability of data and communication
- Meta description of federates etc.
- Physical machine (affect boot of OS, disk)

Threats are identified as the means of reducing or extinguishing the value of an asset.

B.3.3 System threats

This analysis starts from scratch, considering no present security. By combining system assets (described above) and attackers, various threats were identified. By analyzing the threats, counter measures were found. The threats were quantified and given scores (described below) to be able to compare them.

The threats have been divided into three classical security groups, CIA [Fåk]:

- Confidentiality
- Integrity
- Availability

They are further described below.

B.3.3.1 Score calculations (weights)

Physical threats and protection issues of administrative matter are listed in italic, and not considered in this risk analysis. The aim is focused on improving software security.

The risk factor was calculated like this:

```
risk factor = attacker probability * consequence score.
```

Attacker probability denotes the probability that the an attacker would perform this attack.

Consequence score dentoes the cost to repair a damage done by an attack.

The variables range from 1 to 5, where 5 is considered worst case and thus gives a greater score if multiplied with.

It is hard to evaluate protection cost for the protection mechanisms. E.g. VPN is mentioned as countermeassure against many different attacks, and once it is properly implemented against one attack, it will automatically protect against the others also. VPN is estimated to cost 4-5, but has for layout reasons been divided to multiple instances in the table of 1 instead. Access control list (ACL), intrusion detection system (IDS), firewall (FW) and antivirus are also mentioned multiple times.

B.3.3.2 Classifications on signal protection

Threats towards the system depend on:

- The kind of users:
  - Swedish military in training or sharp combat (herein referred to as ”Compat mode”)
  - Software developers of researchers (”Research mode”)
- The information sensitiveness of the data and software used: secret, sensitive, commercial or testing

When performing military training it is important to use the same security level as used in war times, otherwise the military users will be trained in the wrong way. The data used in both training and sharp situations are usually secret. Software developers are known to create automatic ways around the security not to be bothered by it. This could be a problem later on, if these ways exist in the final version of the software or if the security features are too user unfriendly. If the data used is not secret, a lighter form of security is enough.

TSA has also specified a range of classifications for signal protection (containing data, messages, documents, models etc.) which limits access. Table B.2 states the levels in an increasing order.

Table B.2: Signal protection levels.

Protection level	***What***	***Encryption***	***Sym Key***	***Transfer***	***Storage***
Unclassified	Existence of equipment	Not needed	-	any	any
Restricted	Velocity	Approved TSA Cots symmetric crypto (AES) or unencrypted if missing equipment	Keygen approved by TSA	Asymmetric keys by TSA (RSA, PKI ok) or double paper mail	locked room, metal cabinet
Confidential	Tested limits	Approved TSA Military secret hw sym crypto	Done by TSA	“Posten Rek”	safe
Secret	Detailed specifications	Approved TSA Military secret hw sym crypto	Done by TSA	Messenger	safe
Top Secret	Very sensitive	Approved TSA Military secret hw sym crypto	Done by TSA	Messenger	safe
Foreign Secret	Sensitive	By mail from foreign sender. Swedish military not allowed to correspond back with own sensitive info like this.	N/A	N/A	mail

Suitable signal protection for the Research mode of NetSim could be restricted. It is then possible to use PKI, RSA for asymmetric key transfer (see below) and AES for symmetric encryption [Schneier 95]. TSA must generate both parties' asymmetric keys; this could become a problem when working with foreign military or scientists, if they want to use their own certificates. TSA might not want to implement CA trust bridges (where two root CA's trust each other).

The symmetric keys have until today been transferred physically (on paper, in the future active cards) but things start to change with the recent specification on Restricted classification where asymmetric key transfer is mentioned [Försvarsmakten 04].

In a sharp combat situation the level better be Secret/Top Secret. To reach this level of security military crypto hardware boxes have to be added to all nodes. An asymmetric method is not trusted here, because it is mathematically very hard to verify how safe asymmetric encryption is. 9

B.3.3.3 Confidentiality threats

Confidentiality covers threats towards sensible data, communication, and information. Data should not be accessible to anyone. Access to classified data or passwords is considered severe, and gives a 4 in consequence score. If the users can not trust the system its credibility is lost and nobody wants to use it, and in the longer run the Swedish defense will loose their credibility to the public. Table B.3 outlines the confidentiality threats.

Italic text in the table denotes threats and countermeassures of a more administrative flavor. These are not considered further since the focus is on technical and software risk analysis. It has also been grayed out a bit, to make it easier for the eye to discard it.

Table B.3: Confidentiality threats

threat	*consequence*	*cons score*	*risk factor*	*protection*	*cost*
node physically compromised	*stolen HD, hw modification,*	*5*	*25*	*Restricted physical access*	-
node remotely compromised	sensor planted, remote it-weapon	5	25	IDS, firewall, antivirus	2
data classified too low	*classified data spread*	*5*	*25*	*Education, validation of classification*	-
lost backup	*classified data*	*5*	*25*	*Restricted access to backups, strict*	
theft	*classified data*	*5*	*25*	*Burglar alarm, guards, surveillance*	
cracked user domain PW	*user data and machine data out*	*4*	*20*	*regulations on key (21 bytes), admin, key hacker program test*	
crack symmetric key	access traffic communication	4	20	long keys of 128 bytes length + good algorithm like AES, VPN	1
crack asymmetric key	stolen certificate, access symmetric key	4	20	use long key if 2048 bytes lengths + good algorithm like RSA, VPN	1
crack asymmetric key pass phrase	stolen certificate, access symmetric key	4	20	good hash algorithm SHA-1 good entropy of password: at least 98 bytes	1
HW key logger	*passwords, messages*	*5*	*25*	*administrative routines*	-

SW key logger	passwords, messages	5	25	OS updates, antivirus, IDS	2
eavesdropping of traffic	passwords, classified data	5	25	encryption, VPN	1
intentional or accidental unencrypted traffic	passwords, class data out	5	25	configuration check tool, avoid fallback to clear text, VPN	4
accidental spreading of data	*classified data out*	*5*	*25*	*classified data policy, admin*	-
worms, virus	stealing class data and PW	5	25	antivirus, IDS, FW	2
spyware	spying on user behavior (or worse?)	2	10	spyware detector, application firewall	2
portscan, fingerprinting	find security hole, root shell	3	15	IDS, FW	2
attack not discovered, data log module	collect password, class data	4	20	IDS, antivirus, FW	2
buffer overflow	root shell	5	25	antivirus, IDS, OS configuration, SW patches, verify own code	5
man-in-the-middle attack	collect password, data	3	15	VPN	1
stolen model, service, tools	secret or proprietary code coming out	5	25	ACL	4
search sensible information, metadata	find node with sensible data (eases directed attack), reconnaissance of p2p network	2	10	ACL, limit view to authorized, perform own scans for sensible nodes	4
unauthorized access to p2p, simulation	access to classified data	5	25	ACL, authorization	4
interception electronic impulses (RÖS)	*passwords, screen dumps, messages*			*shield equipment*	-

To calculate length of a password corresponding to a 128 bytes long sentence: 26 small a-z + 26 big A-Z + 10 0-9 = 72 keys 72^x = 2^128 => x = 21. 21 bytes long. Do also see [Schneier, 2004] for more information about cryptography.

B.3.3.4 Integrity threats

Integrity attack results in that data becomes corrupted, modified or that the issuer of the data is not who he claims to be. The integrity attacks are given by table B.4.

Table B.4: Integrity threats

threat	consequence	cons score	risk factor	protection	cost
modified data, metadata	un-trusted data	4	20	dig sign, VPN	2
modified SW	unusable system	5	25	backup, CVS	2
man-in-the-middle	false identity, access traffic & passwords	5	25	VPN	1
inserted false messages, descriptions	un-trusted data	4	20	dig sign, VPN	
replay attack	un-trusted system	4	20	VPN	1
source spoofing attack	false node, hi-jacking	2	10	VPN (auth), OS patch	2
insider modifying data	un-trusted data	5	25	IDS	5
fake certificates	*can not trust other organization*	*2*	*10*	*up-to-date CA server patches, authenticated personal , requirements on remote CA*	-
Hacked certificate	ad hoc nodes not aware	3	15	Force periodic updates of revocation list on ad hoc nodes	3

B.3.3.5 Availability threats

Availability attacks on the system may crash a system completely or make it overloaded so it can not serve the users and fetch requested data when needed. The availability attacks are stated in table B.5.

Table B.5: Availability threats

threat	consequence	cons score	risk factor	protection	cost
spam	*floods mail system*	*1*	*5*	*spam filter*	-
external DoS (Denial of Service)	cripples Internet connection, disturbs simulation	3	15	configure routers to drop bad traffic from certain IP's, VPN	1
internal DoS	cripples internal communication, disturbs simulation	3	15	remove IP node from network	2
Lost backups	*classified info lost, not able to restore*	*5*	*25*	*admin routines*	-

Theft	*machine lost*	*5*	*25*	*locks, surveillance*	-
SW (rm) vandalism	*machine wont run*	*5*	*25*	*backup*	-
HW vandalism	*unusable system*	*5*	*25*	*locks, surveillance*	-
System resources misuse (games, spam, DoS, wares)	unnecessary system load	3	15	monitoring resources, IDS	3
bounce	attacks through own nodes, results in bad image	2	10	monitoring, IDS	1
Panic button (power down, offline)	*no system*	*5*	*25*	*backup resources*	-
Account problems	*users cannot login*	*3*	*15*	*system administrator can override*	-
p2p root not visible	p2p network looses connection	5	25	backup resource in p2p mechanism	3
accidentally remove data/ simulation	*Author has to restore data, system unusable for some time*	*2*	*10*	*better GUI, undo, backup*	-
spoofing attack, routing attacks	Override traffic routes, DoS	2	10	router configuration, OS patches	2
source routing	Override traffic routes	1	5	OS patch	1
EMP, HPM	*Destroy equipment*			*shielding?*	-

B.3.4 Considerations

A threat might be overlapped by more than one of the CIA threat groups. An attack could include several chained threats, starting up with small events. An attack could go on like this:

1. Find out information about the system (fingerprinting OS in accessible nodes).
2. Port scan nodes to find open ports, identify services running and their version numbers.
3. Attack a node with known vulnerability.
4. Gain a shell with root or user rights.
5. Install application that logs passwords and communication, scan internal computer network.
6. Continue attack against other nodes within the internal network.
7. Cripple entire network with internal denial of service (DoS): format all hard drives to cover tracks and waste administrator's valuable time.

B.4 Security requirements and recommendations

This section summarizes the protective means found in the threat analysis. It also recommends security precautions that should be taken.

B.4.1 Authentication

- A user must authenticate when starting up NetSim (before creating or joining a simulation). This is done only once, and it is the user's responsibility to exit NetSim. Screen savers, locked doors etc should protect the application when running.

- *Web services* should also be authenticated.

B.4.1.1 Public Key Infrastructure (PKI)

TSA should setup an own certificate authorization (CA) with PKI support, where one can verify users online. TSA does not trust other CA of security reasons, which becomes a big problem when wanting to cooperate with external organizations. Multiple CA servers are supposed to communicate through CA-bridges which endorse a non-hierarchical structure. A new organization could either setup its own top node CA or use a commercial one (like VeriSign).

The issuing process gives the user a certificate which among other things contains a public key. The user is also given a private secret key which should be stored encrypted. The user can then sign messages with his private key, and others can verify from whom the message came with the open public key. For more information about PKI and CA, see [Bengtsson et al. 03; Bengtsson et al. 01].

- While waiting for TSA to become available, we could setup an own CA server in the network where users authenticate to gain access to the symmetric key used in network communication. It is also easier to experiment with CA-bridges if we have control of the CA-server.

- The certificates should be of RSA X.509v3 (or possibly DSA type)

B.4.2 Access Control List (ACL)

Access control is a way to enforce a security policy and limit what users can do with the resources (typically files).

B.4.2.1 Resources and ACL rights

Resources within NetSim will allow certain actions applied from users or other resources. Here is a brief example of actions considered with the different resources. The triplet r,w,x means read, write (includes create and remove), and execute (run).

- Simulation r/w/x
  move = r+w+x
  create = [w+]x
  join = r+x

- Files, data, meta data descriptions, backups, model data: r/w/x

- Machine resource, remote node execution capacity: r/w/x

- Services, applications: r/w/x

- Certificates: r/w/x

B.4.2.2 Users

There are two levels of users within a desktop computer running NetSim. First the user has to logon to the operating system and administrative organization domain to access a user account. When starting NetSim, a new logon to the application is necessary since these users might not be the same, and are more application specific. To merge these two levels into one, full control of the administrator domain is required which is unecessary complex when only using the system for research purposes. It requires the researchers to set up a separate domain, with a separate domain controller server without support

from the IT unit. The user in NetSim can be a role, the same as the operating system username or even another user depending on the design. In an application dedicated device, like a PDA, the operating system might be hidden. Such a device only requires the NetSim login.

The table B.6 explores different user roles combined with allowed actions. The triplet u,g,o means user, group, and other.

Table B.6: User rights.

User	*Rights*
Superuser	Configures top node Add administrators Override simulation bridging
Sim Administrator **Librarian** **helpdesk** **(root)**	Override simulation (add/remove/move simulation) local/remote Bridge simulation to other networks Add users
Object owner **Programmer** **(u)**	Create simulation Add/remove/change own code Add/remove/change own simulation local
Group user **(g)**	Create simulation remotely, use computer resources of group. Join locked group simulation Additional group rights on code, files, simulation
All users **(o)**	Join open simulation Search services, tools, scenario descriptions Communicate with audio, video, text

- Access control needed for the resources (models, simulation, code, files etc)
  - user (owner) r/w/x (read/write/execute) rights
  - group r/w/x rights
  - other (guest, all) r/w/x rights
  - use sticky bits for inheritance (if applicable)
- Store ACL parameters in resources: meta data descriptions, code, files, simulation, objects.
- Simulation creator may specify a group to limit participation.

B.4.3 Encryption

Encryption is needed to protect non-public messages and files from revealing the information in cleartext. It will also protect against many attacks [Stallings 03]. Some encryption considerations are given regarding the signal protection level and protecting a peer-to-peer network.

B.4.3.1 Signal protection, keys and encryption algorithms

TSA requires that military command and control systems use **symmetric encryption**, which has the same key in all nodes that wants to communicate. It scales well with multiple nodes that share the same broadcast network, but additional personal key pairs are needed to make a conversation private.

For **research** purposes r**estricted signal protection** is recommended. This includes **X509v3 certificates** generated by TSA, the military Certificate Authority (CA). Own symmetric keygen approved by TSA, RSA asymmetric encryption to transfer symmetric keys to all authorized nodes, and at last AES symmetric encryption to scramble the conference/simulation. This mode should be sufficient for commercial companies as well.

In the event of a sharp **combat** situation, the more secure **Secret/Top Secret signal protections** must be used. This requires that separate military hardware crypto are added to all communicating nodes. Key transfer has to be done with messengers.

If one node leaves the NetSim group, the others should renegotiate a new symmetric key for communication. An upper data-limit for how much data to transfer with a symmetric key must be decided to avoid statistical key cracking.

B.4.3.2 Virtual Private Network (VPN)

A virtual private network on the IP-layer handles encryption of both TCP packets and UDP packets in a peer-to-peer network.

IPsec is a popular implementation of a VPN. It can be complex to setup in the nodes, which might lead to faulty configurations. VPN handles authentication, integrity check, automated key distribution with public key cryptography. Some kind of configuration verification of the setup is desired. Remove fallback/rollback functionality to cleartext in algorithm negotiations.

B.4.4 Network and host configuration

The needed security features should preferably be automatically configured when installing NetSim. Some considerations regarding network equipment and different kinds of nodes are given.

B.4.4.1 Routers, switches

Network equipment should be configured to avoid routing attacks and spoofing (pretending to have another MAC or IP adress adress).
An intrusion prevention system is recommended to be available on the network or within the nodes to discard attacks.

B.4.4.2 Stationary clients

The stationary clients typically contain the following software: NetSim, Java and Web services.

Needed security countermeasures are:

- Firewall.
- Antivirus with regular updates.
- Host intrusion detection system (HIDS) residing in operating system of the nodes, and possibly a network intrusion prevention system (NIPS) to avoid malicious network traffic.
- Automatically updated operating system and software.
- Revocation synchronization updates.

B.4.4.3 Mobile clients (PDA)

Mobile clients are probably not so powerful and should contain light versions of NetSim and Java.

Security means:

- Administrational access could be done from this node like a remote control; actions must be implemented by other stationary nodes.
- Automatically updated operating system synchronization.
- Revocation synchronization updates when connected properly.

If feasible:

- Firewall.
- Antivirus which updates definitions from other nodes when possible.
- HIDS + NIPS.