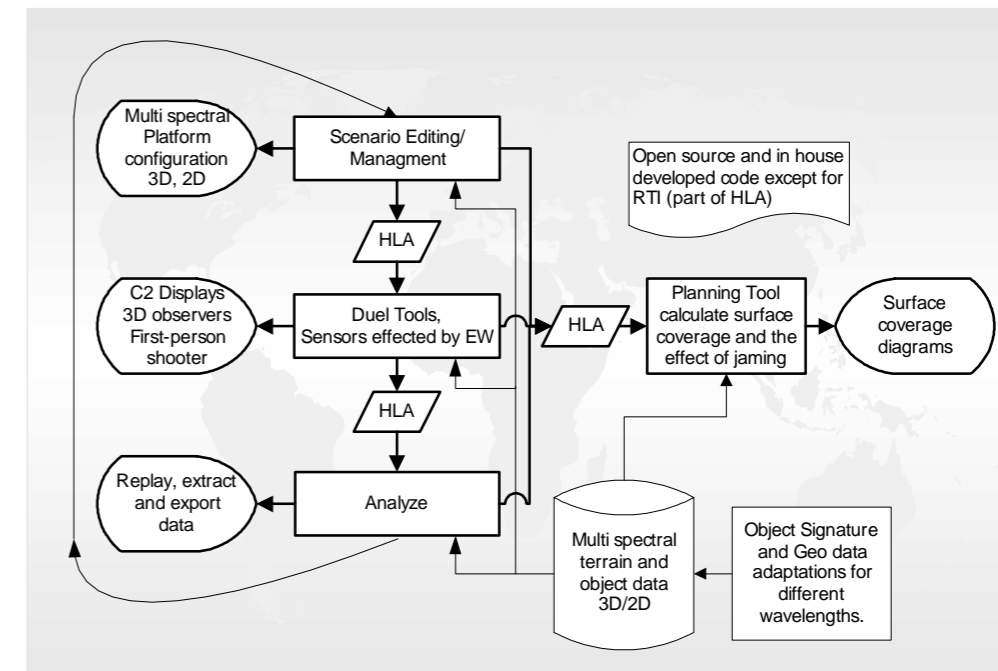


LARS TYDÉN, LEIF FESTIN, SEBASTIAN OLSSON, CHRISTER WIGREN



FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1250 anställda varav ungefär 900 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

LARS TYDÉN, LEIF FESTIN, SEBASTIAN OLSSON, CHRISTER WIGREN

Systembeskrivning LKS V2 Federationsdesign

Utgivare FOI - Totalförsvarets forskningsinstitut Ledningssystem Box 1165 581 11 Linköping	Rapportnummer, ISRN FOI-R--2274--SE	Klassificering Teknisk rapport
	Forskningsområde 6. Telekrig och vilseledning	
	Månad, år April 2007	Projektnummer E7546
	Delområde 69 Breda projekt inom telekrig och vilseledning	
	Delområde 2	
Författare/redaktör Lars Tydén Leif Festin Sebastian Olsson Christer Wigren	Projektledare Martin Castor	
	Godkänd av Mikael Sjöman	
	Uppdragsgivare/kundbeteckning FMV	
	Tekniskt och/eller vetenskapligt ansvarig Lars Tydén	
Rapportens titel Systembeskrivning LKS V2 Federationsdesign		
Sammanfattning <p>I denna rapport beskrivs designen av den tekniska simuleringsplattformen som är framtagen till LKS. Designen har utgått från att kunna återbruka så mycket som möjligt av det som redan finns och är relevant för LKS därav så har EWSim ramverket utnyttjats som en grund att stå på.</p> <p>Plattformen består i stora drag av tre delar scenariokonfiguration, dynamiska simuleringsmodeller samt loggning och analys. Designen är gjord så att olika plattformar (t ex stridsvagnar och helikoptrar) konfigureras med olika komponenter (t ex sensorer och vapen) och simuleras distribuerat över ett nätverk. Detta dokument beskriver framför allt det som är nytt, vilket är en nätverksmodell med routing som påverkas av telekrig och datornätverksoperationer. Dessutom har C2 funktionalitet utvecklats samt ett användargränssnitt som presenterar data för stabspersonal.</p> <p>Simuleringsplattformen är en HLA-federation (High Level Architecture) enligt standarden IEEE 1516.</p>		
Nyckelord HLA, CNO, Telekrig, Simulering		
Övriga bibliografiska uppgifter	Språk Svenska	
ISSN 1650-1942	Antal sidor: 72 s.	
Distribution enligt missiv	Pris: Enligt prislista	

Issuing organization FOI – Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 Linköping	Report number, ISRN FOI-R--2274--SE	Report type Technical report
	Programme Areas 6. Electronic Warfare	
	Month year April 2007	Project no. E7546
	Subcategories 69 Interdisciplinary Projects regarding Electronic Warfare	
	Subcategories 2	
Author/s (editor/s) Lars Tydén, Leif Festin, Sebastian Olsson, Christer Wigren	Project manager Martin Castor	
	Approved by Mikael Sjöman	
	Sponsoring agency FMV	
	Scientifically and technically responsible Lars Tydén	
Report title (In translation) Systemdescription LKS V2 Federationdesign		
Abstract <p>This document describes the design of the technical simulation framework that has been developed for the C2WS (Command and Control Warfare Simulator).</p> <p>A key concept in the design has been reusability. The EWSim (Electronic Warfare Simulation Interface Model) framework has a lot of relevant features already developed and for this reason it has been used as a starting point and has now been further extended to meet the requirements of the C2WS.</p> <p>The framework consists of three main parts, scenario configuration, dynamic simulation models, and a logging/analyzing tool. The design allows various platforms (such as tanks, helicopters etc) to be configured and equipped with a set of different components (i.e. sensors and weapon systems). Platforms and equipments can then be simulated distributed over a network using HLA (High Level Architecture) according to the IEEE 1516 standard.</p> <p>This document focuses on the new features which mainly include a networking model with routing that can be affected by EW (Electronic Warfare) and CNO (Computer Network Operations). Also additional Command and Control functionality and user interfaces presenting data to the staff commander has been developed.</p>		
Keywords HLA, CNO, EW, Simulation		
Further bibliographic information	Language Swedish	
ISSN 1650-1942	Pages 72 p.	
	Price acc. to pricelist	

Innehållsförteckning

1. Inledning	6
1.1. Bakgrund	7
1.2. Läsanvisning	8
2. Övergripande design	9
2.1. Simulerade objekt	10
2.2. Plattformer	11
2.3. Övergripande federationsdesign	13
2.4. Federater	14
3. Ledningsfunktion i LKS	16
4. Federater	19
4.1. Fartygsplattformsfederat	20
4.1.1. Fartygsplattformsfederatens simulerade objekt	20
4.2. Flygplansplattformsfederat	21
4.2.1. Flygplansplattformsfederatens simulerade objekt	21
4.3. Övriga markplattformarsfederat	22
4.3.1. Övriga markplattformars simulerade objekt	22
4.4. Stabsfederat	23
4.5. Nätverksfederat	24
4.5.1. Kommunikation	24
4.5.2. NätNod	26
4.5.3. Router	27
4.5.4. Gateway	28
4.5.5. NätNod – klassdiagram	29
4.5.6. Meddelandeöverföring	31
4.5.7. Länkuppdatering	31
4.5.8. Kabel och radio	32
4.5.9. Nät – CNO	33
4.6. IRSTplattformsfederat	40
4.6.1. IRSTplattformsfederatens simulerade objekt	40
4.6.2. IRST Interaktioner	40
4.7. Radarplattformsfederat	44
4.7.1. Radarplattformsfederatens simulerade objekt	44
4.7.2. Interaktioner	44
4.8. SiSplattformsfederat	47
4.8.1. SiSplattformsfederatens simulerade objekt	47
4.9. Federation Management federat	48
4.10. Spelledningsfederat	48
4.11. Loggerfederat	48
5. Infrastruktur och infrastrukturgränssnitt	49
6. Funktionsbeskrivningar	50
6.1. Signalspanings- och störsystemsfunction	51
6.1.1. Klassdiagram	51
6.1.2. Olika typer av SiS-/störsystem	52
6.1.3. Sekvensdiagram	54
6.2. Radarstörare	59
6.3. Visualisering av nät	60
6.3.1. Vägen från simulering till visualisering	60
6.3.2. Klassdiagram	60
6.3.3. CommNode	60
6.3.4. NodeType1 – N	60
6.3.5. CommLink	61
6.3.6. LinkCollection	61
6.3.7. TimeConstrainedLinkCollection	61
7. Akronymer och begrepp	62
Bilaga 1 FOM LKS	63
Bilaga 2 Publish Subscribe för federater I LKS V2	67
Bilaga 3 Publish and Subscribe för LKS objekt	68
Referenser	72

1. Inledning

Detta dokument syftar till att beskriva hur den tekniska plattformen ”Demonstrator Ledningskrigföringsimulator” version 2 är designad. Dokumentet är skrivet på en sådan nivå att det krävs god kunskap om distribuerad simulering via HLA samt telekrig (TK) och operationer i datornätverk (CNO) för att förstå innehållet.

1.1. Bakgrund

En förstudie avseende "Demonstrator Ledningskrigföringsimulator" bedrevs av FMV under år 2003 och 2004 [FS] . På Försvarmaktens (FM) uppdrag utreddes möjligheten att, helt eller delvis, åstadkomma en simulator avseende ledningskrigföring samt ge förslag på en demonstrator för en Ledningskrigföringsimulator (LKS). Resultatet visade att det är möjligt att åstadkomma en demonstrator. FMV förslag var att demonstratorn inriktas mot simulering av användning av TK samt CNO, där påverkan på såväl egen som annan aktörs ledningsförmåga utvärderas.

Under 2005 påbörjades etapp 1 i utvecklingen av en demonstrator Ledningskrigföringsimulator. Projektet leds av FMV och utvecklingen sker med deltagare från FMV, FM, FOI och FHS. Utveckling av en första version av demonstrator LKS har pågått sedan våren 2005 och under våren 2006 påbörjades utveckling av LKS v2. [FEDEP1v2] och [FEDEP2v2] beskriver steg 1 måldefinition och steg 2 konceptuell analys för version 2. Designen av LKS version 2 bygger på den tidigare framtagna designen för version 1 [DESIGNv1]

Utvecklingsprocessen som används är FEDEP (Federation Development and Execution Process IEEE 1516.3) vilket är en generell process för att ta fram en HLA-federation (High Level Architecture). En federation består av federater där en federat är en instans av en applikation som är ansluten till ett RTI (Run Time Infrastructure) med en federationsöverenskommelse gentemot övriga federater som kan vara med i federationen. Den federationsöverenskommelse som används i LKS är beskriven i "MOSART Federation Agreements" [AGR] och den "Federation Object Model" (FOM) som anger vilka objekt som kan finnas med i federationen finns redovisad i Bilaga 1 FOM LKS.

1.2. Läsanvisning

Läsaren rekommenderas att läsa och ta del av dokumenten [FEDEP1v2] och [FEDEP2v2] som berör steg 1 och 2 i utvecklingen av LKS v2 för att få ett sammanhang till det som beskrivs i detta dokument. Designen av LKS version 2 bygger på designen för version 1, med utökningar och mindre förändringar. De delar som är nya till version 2 omnämns speciellt i läsanvisningen nedan.

Kapitel 1: Ger läsaren en introduktion till LKS, presenterar syftet och målet med rapporten och beskriver övergripande den använda metoden FEDEP samt specifikt för FEDEP steg 2 som tas upp i denna rapport.

Kapitel 2: Ger en övergripande beskrivning av federater och deras innehåll. Federaterna är valda utifrån det scenario som finns beskrivet i [FEDEP2v2] samt så att de kan återbruka några befintliga applikationer som bl.a. EWSim [EW] för plattformsbaserade telekrigsimuleringar och NetScene [MOSART][EW] för spelledning och visualisering.

Kapitel 3: Ledning är en central funktion som alla simulerade plattformar har, denna återfinns i ett flertal federater och utvecklades till LKS v1.

Kapitel 4: Federater som ingår i federationen beskrivs här. Det nya för version 2 är det som berör CNO och nätverk i avsnitt 4.5.

Kapitel 5: Beskriver kortfattat infrastruktur och infrastrukturgränssnitt för federationen.

Kapitel 6: Här beskrivs funktioner som används i ett flertal federater så som radiosystem. Nytt för version 2 är avsnitt 6.3 visualisering av nät.

Kapitel 7: Syftar till att redovisa de referenser som är aktuella för denna rapport.

2. Övergripande design

Två grundläggande begrepp i federationen är objekt (simulerade objekt) och plattformar. Plattformar är en speciell typ av objekt som kan bära andra objekt. Exempelvis kan ett fartyg (en plattform) förses med t.ex. spaningsradar, IRST och SiS (andra objekt). I detta kapitel beskrivs först vilka objekt som kommer att simuleras, därefter beskrivs de plattformar som kommer att simuleras och vilka övriga objekt de kan bära. Slutligen beskrivs den övergripande federationsdesignen och en kort beskrivning av ingående federater.

2.1. Simulerade objekt

Med simulerade objekt avses objekt som kommer att simuleras i LKS-federationen version 2.

Kort beskrivning av simulerade objekt:

- Fordon, flyg, fartyg och människa – *Utrustningsbärare (dvs. plattformar)*
- Antenn – *Radarantenn som simulerar antenndiagram*
- Radar – *Parametersatt radarsystem*
- IRST – *Parametersatt IRST-system (Infrared Search and Track)*
- Radio – *Parametersatt radiosystem.*
- Radio SiS – *Parametersatt signalspanings- och pejlsystem mot radio, är ett specialfall av Radio.*
- Radio Stör – *Parametersatt radio störsändningssystem, är ett specialfall av Radio.*
- Radio TDOA – *Parametersatt radiosystem som kombinerar signalspaning-, pejl- och störförmåga mot radio, är ett specialfall av Radio.*
- Radio TVC – *Parametersatt Radiosystem med funktioner för att styra SiS- och störenheter samt att sammanställa SiS-data, är ett specialfall av Radio.*
- NätNod – *Kommunicerande nod i nätverket. NätNoden innehåller all funktionalitet för att beräkna förhållanden för radiomottagning och nätverk.*
- Radarstör – *Parametersatt radarstörare*
- Radarvarnare – *Parametersatt radarvarnare*
- Raketartilleri – *Utrustning för att kunna avfyra en raket*
- C2-objekt – *Hanterar lednings-/stabsfunktioner*

2.2. Plattformer

Plattformarna är uppbyggda hierarkiskt där grundobjektet bär övrig utrustning och ger möjlighet till förflyttning. Utrustning sitter placerad relativt föräldraobjektet. Alla plattformar är utrustade med radio och ledning. Nätverkskommunikation simuleras genom att ledningen äger en NätNod (accesspunkt) i nätverket. Trafiken som plattformen bidrar med till nätverket kan påverkas av CNO-applikationen (CNOApp). Det är via CNOApp som Computer Network Operation (CNO) attacker och försvar kan genomföras. Alla plattformar har en ledningsfunktion. Staber har även dessa en egen ledningsfunktion som är samma funktion som för plattformar men denna funktion har tillgång till mer information som kan visualiseras. All kommunikation med stab och övriga plattformar sker till/från denna ledningsfunktion.

De plattformar som ingår i simuleringen är konfigurerade med sensorer, ledning osv. är:

Radarplattform -Markgående fordon med radar.

- Fordon
 - Antenn
 - Radar
 - Radio
 - NätNod
 - Ledning
 - CNOApp (nätverksapplikation)

IRST-plattform -Markgående fordon med IRST.

- Fordon
 - IRST
 - Radio
 - NätNod
 - Ledning
 - CNOApp

SiS-plattform -Markgående fordon med SiS-kapacitet

- Fordon
 - Radio
 - NätNod
 - Radio SiS/Pejl
 - Ledning
 - CNOApp

Stabsfordonsplattform -Markgående fordon med stab

- Fordon
 - Stab
 - Radio
 - NätNod
 - Ledning
 - CNOApp

Fartygsplattform (Typ Kustkorvett) -Fartyg med radar, IRST, SiS/Pejl, radio och störutrustning mot radio

- Fartyg
 - Radar
 - Radio
 - NätNod

- Radio SiS/Pejl
- Radio Stör
- Radio TVC
- IRST
- Ledning
 - CNOApp
 - Ledning
 - CNOApp

Fartyg - RoRo-plattform -Fartyg

- Fartyg
 - Radio
 - NätNod
 - Ledning
 - CNOApp

Flygplattform -Flygplan med stab, radar, IRST, SiS/Pejl, radio och störutrustning mot radio

- Flygplan
 - Radar
 - Radio
 - NätNod
 - Radio SiS/Pejl
 - Radio Stör
 - IRST
 - Ledning
 - CNOApp
 - Stab
 - Radio
 - NätNod
 - Ledning
 - CNOApp

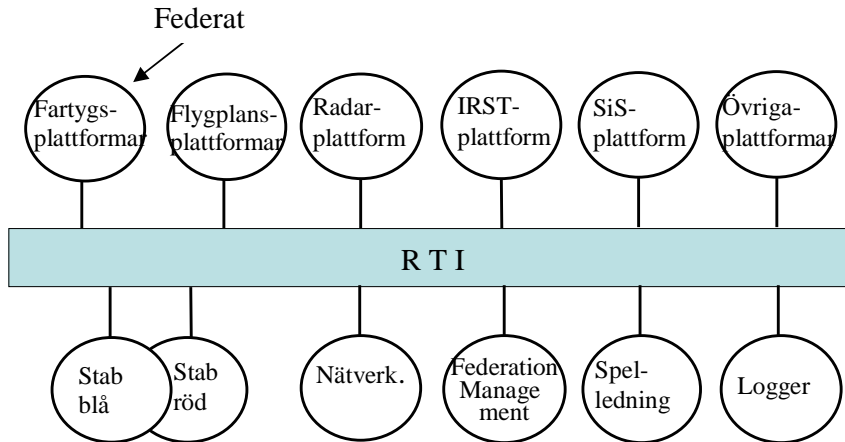
SF-plattform -Specialförband (SF) med radioutrustning med SiS/Pejl och störförmåga (Radio TDOA)

- Människa
 - Radio
 - NätNod
 - Radio TDOA
 - Ledning
 - CNOApp

Raketartilleriplattform -Markgående fordon med bekämpningsförmåga

- Fordon
 - Raketartilleri
 - Radio
 - NätNod
 - Ledning
 - CNOApp

2.3. Övergripande federationsdesign



Figur 1 Federationen och dess ingående federater vid exekvering.

Designen av LKS version 2 bygger på de tidigare framtagna dokumenten LKS FEDEP steg 1 måldefinition [FEDEP1v2] och LKS FEDEP steg 2 konceptuell analys [FEDEP2v2] för version 2. Dessutom baseras designen på designen för LKS version 1 [DESIGNv1]. Uppdelningen av federater utgår från två grundförutsättningar om hur ett möjligt spel kan gå till där staberna och ledningsfunktionen på plattformar skall kunna spelas samt återbruket av applikationer. För att röd och blå sida ska kunna spelas av Human in the loop (HIL) måste dessa kunna exekveras på olika maskiner och därmed vara skilda federater.

2.4. Federater

Vad som ingår i respektive federat är dynamiskt konfigurerbart men det som styr innehållet är hur scenariot ser ut som finns beskrivet i [FEDEP2v2] .

Samtliga federater som simulerar plattformar (Radarplattform, IRST-plattform, SiS-plattform, Fartygsplattform, Flygplansplattform, Övriga markplattformar) har CNOApp och C2-objekt med ledningsfunktion. På så vis kan plattformen kommunicera med andra plattformar samt genomföra CNA och CND.

Radarplattformsfederat

Simulerar objekt som hör till en fordonsmonterad spaningsradar med ledning. Följande objekt simuleras i denna federat:

- Fordon, Antenn, Radar, Radio

IRSTplattformsfederat

Simulerar objekt som hör till en fordonsmonterad IRST med ledning. Följande objekt simuleras i denna federat:

- Fordon, IRST, Radio

SiSplattformsfederat

Simulerar objekt som hör till en fordonsmonterad SiS med ledning. Följande objekt simuleras i denna federat:

- Fordon, Radio, Radio SiS/Pejl

Fartygsplattformsfederat

Simulerar objekt som hör till två fartyg med ledning, Kustkorvett samt RoRo (roll-on/roll-off transport fartyg). Staben är placerad på Kustkorvett men simuleras i stabsfederaten. RoRo fartyget transporterar materiel. Följande objekt simuleras i denna federat:

- Fartyg, Radar, IRST, Radio

Flygplansplattformsfederat

Simulerar objekt som hör till ett flygplan med ledning. Följande objekt simuleras i denna federat:

- Flyg, Radar, Radarstör, Radarvarnare, Radio

Övriga markplattformarfederat

Simulerar objekt som hör till de förband som säkrar infarten till kusten samt bekämpningsplattformen och det fordon som röd stab sitter i. Följande objekt simuleras i denna federat:

- Fordon, Människa, Radio, Radio SiS/Pejl, Radio TDOA, Radio Stör, Raketartilleri

Stab röd och blå federater

Simulerar objekt som hör till en stab. Följande objekt simuleras i denna federat:

- C2 med ledningsfunktion

Nätverksfederat

Simulerar NätNoder med routing i nätverket. Följande objekt simuleras i denna federat:

- NätNod

Federation Management federat

Federationskontrollfederat som styr scenariots exekvering genom att hantera tiden och händelser som start, stopp och paus, samt att specificera i vilket terrängavsnitt aktuell simulering skall köras, se [MOSART].

Spelledningsfederat

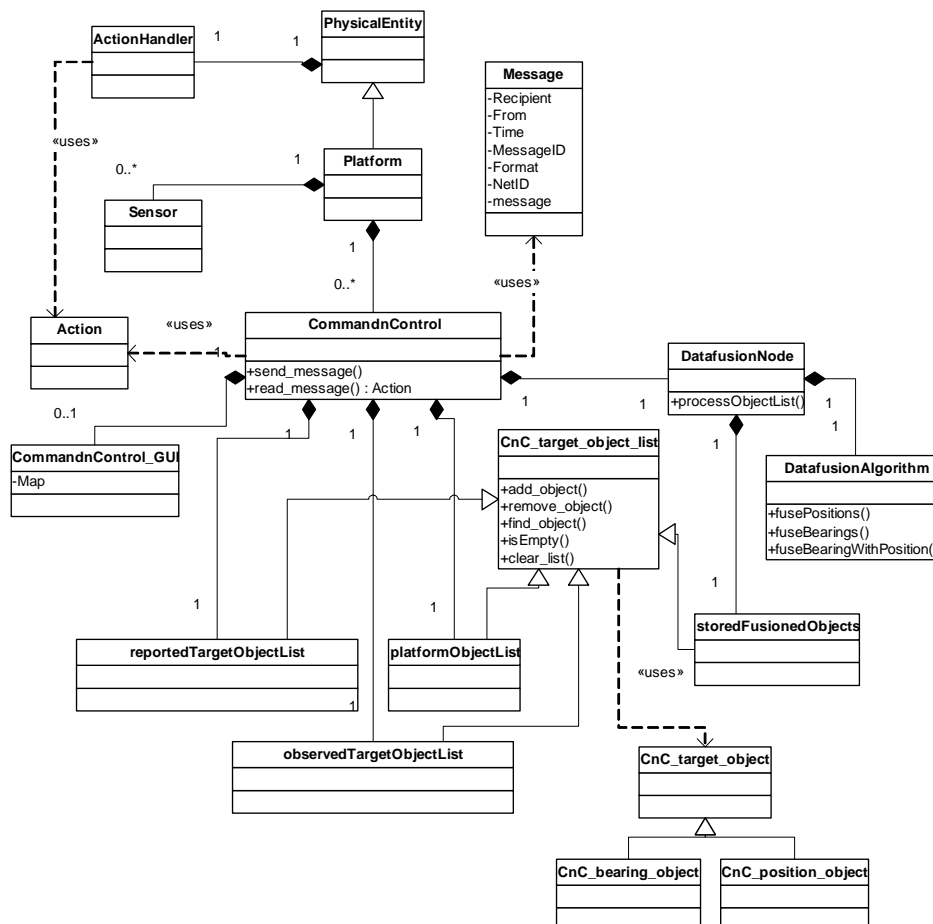
Distribuerar scenariokomponenter till respektive federat vid uppstart samt visar verkligt läge [MOSART][EW].

Loggerfederat

Logger kör hela tiden i bakgrunden och loggar kontinuerligt de data som utbyts över HLA. Den fungerar snarlikt en ”bandspelare” med funktioner som play, rec och pause. Record under en simulering sparar ner simulerad data (HLA objekt) till fil. Play spelar upp inspelade data från en loggfil i federater som prenumererar på de HLA objekt som finns i loggfilen [EW]. Denna federat är vidareutvecklad från LKS version 1 och sparar data till ett excel-dokument, samt loggar data kontinuerligt till en fil så att dessa finns kvar om simuleringen av någon anledning avbryts på ett icke-kontrollerat sätt. I federaterna kan loggpunkter sättas. Data från dessa loggpunkter går som HLA-medelanden i simuleringen och sparas på samma sätt som övrig HLA-trafik.

3. Ledningsfunktion i LKS

Ledningsfunktionen i LKS skickar och tar emot rapporter och order samt upprättar och underhåller en lägesbild. Ledningsfunktionen kan vara bemannad eller obemannad. Om den är obemannad sker ledningen automatiskt med hjälp av enkla regler som leder till en händelse (Action).



Figur 2 Klassdiagram för ledningsfunktion

Meddelanden (order och rapporter) skickas i det simulerade nätverket i form av FFM (Foi Fast Message). Då dessa ska förmedlas mellan olika federater skickas de som en HLA-interaktion med attributen:

- Recipient – mottagarens ID
- From – avsändarens ID
- Time – tid då meddelandet sändes
- message ID – unikt ID-nummer för meddelandet (Unikt heltal)
- format – nummer som anger vilket format meddelandet har (T.ex. format 501 för PC-DART, rapport enl. 7 S med given rad struktur se Figur 3)
- net ID – anger vilket nät som används för att skicka meddelandet, från det aktuella nätet hämtas meddelandets kvalitetsparametrar (t.ex. bitfelshalt, Signal to Noise Ratio (SNR) osv.)
- message – själva meddelandet

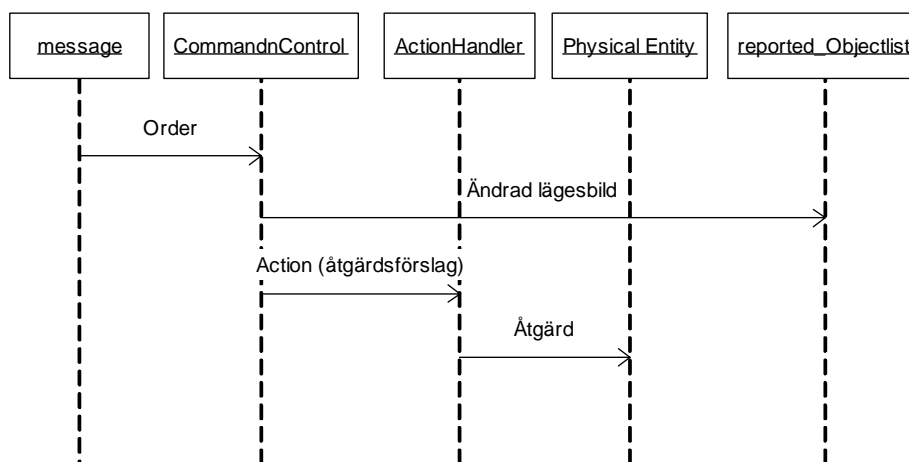
Rad	Innehåll rapport enl. 7 S
1	Stund , tidsnummer för observation
2	Ställe. Postition X, Y
3	Styrka, Antal
4	Slag, typ
5	Sysselsättning, fri text
6	Symbol, fri text
7	Sagesman, fri text

Figur 3 Exempel på format 501, med radanvisningar

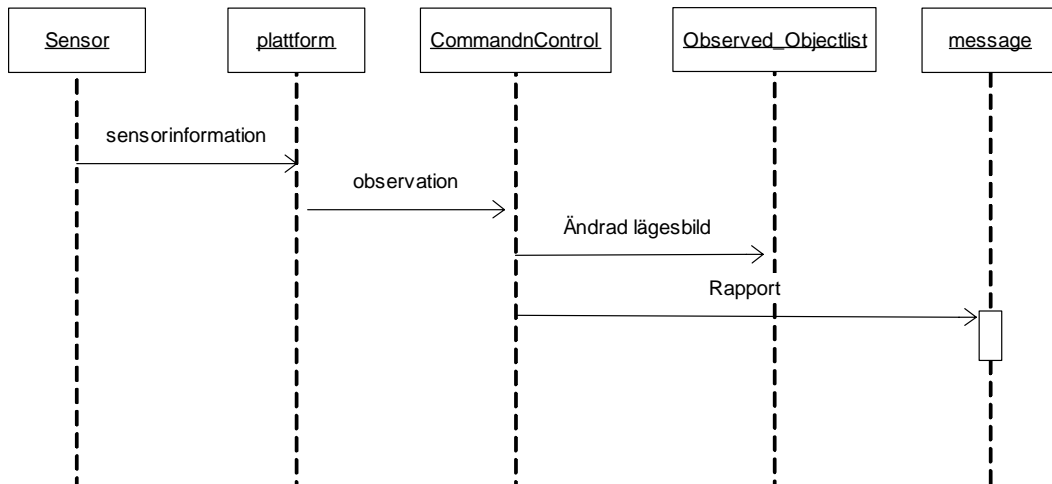
Ledningsfunktionen *CommandnControl* läser och skickar meddelanden. Lästa meddelanden kan resultera i en uppdaterad lägesbild och/eller en föreslagen åtgärd (action) t.ex. en omgruppering, se Figur 4.

Plattformen uppdaterar listan med av egna sensorer observerade objekt, se Figur 5, under tiden ledningsfunktionen uppdaterar listan med rapporterade objekt baserat på inkomna meddelanden. Den aktuella lägesbilden är en fusion av dessa listor, se Figur 6.

Lägesbilden kan visualiseras av *CommandnControls*'s grafiska gränssnitt i t.ex. en karta. För att styra objekt ger man order till dess ledning som sedan hanterar objektet genom olika händelser, *Actions*. Händelser hanteras i *ActionHandler*, vilken omvandlar händelserna till funktionsanrop i objektet och ordern utförs.



Figur 4 Sekvensdiagram från order till uppdaterad lägesbild samt åtgärd



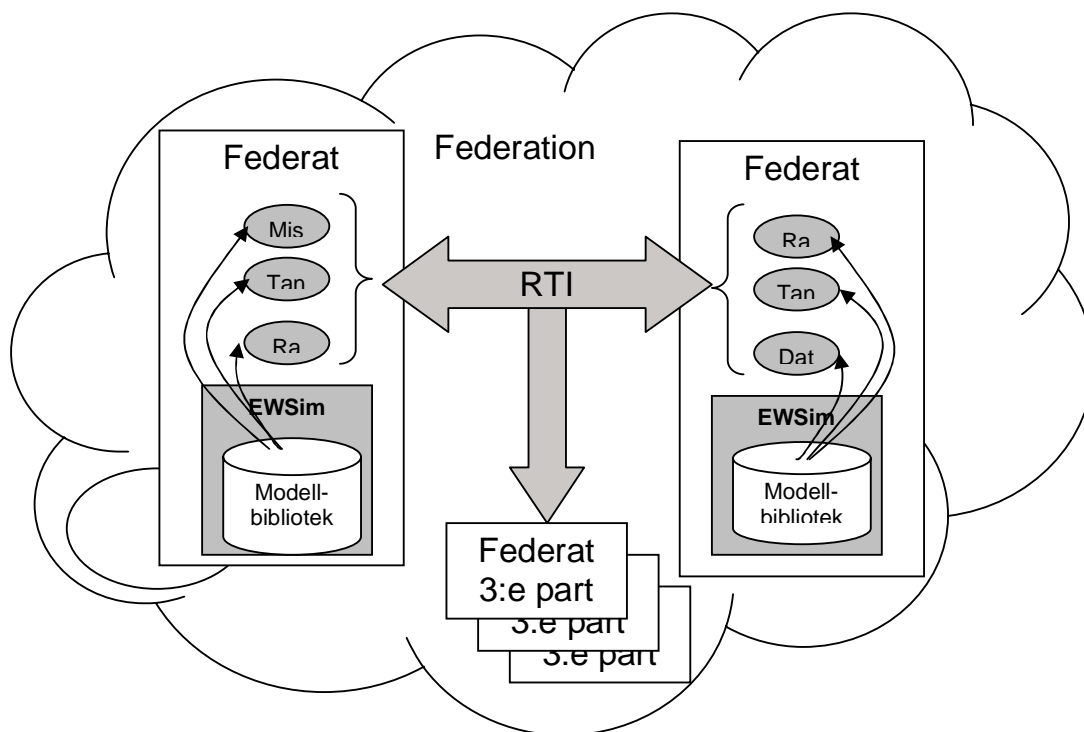
Figur 5 Sekvensdiagram från sensorutslag till uppdaterad lägesbild samt rapport



Figur 6 exempel på presenterad lägesbild

4. Federater

Många av federaterna inom LKS körs via ramverket EWSim. EWSim skapar en federat dynamiskt, vilket Figur 7 illustrerar. EWSim, som ramverk, innehåller modeller (plattformar och simulerade objekt) i ett bibliotek. Vid start av en federation startas en eller flera EWSim. Varje EWSim startar precis de modeller som behövs för varje enskild federat och inga andra. Det hela går att jämföra med en mataffär. Affären har alla upptänkliga varor på hyllan (modellbiblioteket), men en kund (federat) handlar och tar bara med sig hem det som behövs till matlagningen (federationen). Varje modell använder enbart RTI för kommunikation och betar sig därmed som en specialutvecklad federat.



Figur 7 EWSim som federat

Stabsfederaten och nätverksfederaten ingår inte i EWSims modellbibliotek utan körs helt separat. Enligt Figur 7 EWSim som federat, kan de då ses som tredjepartsfederater.

4.1. Fartygsplattformsfederat

I denna federat simuleras två fartyg, dels ett av typ Kustkorvett samt ett RoRo-fartyg. På Kustkorvetten finns en blå stab, vilken inte simuleras i denna federat utan i blå sidas stabsfederat. Fartyg bygger på plattformobjekt enligt RPR-FOM:en [RPR-FOM], där fartyg är en SurfaceVessel. Plattformen bär övrig utrustning och ger möjlighet till förflyttning av plattformen. Utrustning sitter placerad relativt föräldraobjektet. Fartygen konfigureras så att de är utrustade med radio och ledning (se avsnitt 3). Sensorer hanteras på samma sätt oavsett plattform, varför sensorfunktionen endast beskrivs en gång, se avsnitt 4.6, 4.7 och 6.1 för beskrivning av IRST, Radar och SiS/Pejl. Televapencentralen (TVC) simuleras som en del av stabens ledning, se 6.1.2.

4.1.1. Fartygsplattformsfederatens simulerade objekt.

De objekt som inte är gråmarkerade simuleras i denna federat.

Fartygsplattform (typ Kustkorvett)

- Fartyg
 - Spaningsradar
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Radio SiS/Pejl
 - Radio Stör
 - Radio TVC
 - IRST
 - Ledning
 - CNOApp Simuleras i nätverksfederaten
 - Stab
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Ledning Simuleras i stab blå federaten
 - CNOApp Simuleras i nätverksfederaten

Fartyg RoRo plattform

Fartyg

- Fartyg
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Ledning
 - CNOApp Simuleras i nätverksfederaten

4.2. Flygplansplattformsfederat

I denna federat simuleras ett flygplan som flyger över motståndarens sensorer. Flyg bygger på plattformobjekt enligt RPR-FOM:en där flygplan är Aircraft. Plattformen bär övrig utrustning och ger möjlighet till förflyttning av plattformen. Utrustning sitter placerad relativt föräldraobjektet. Flygplan konfigureras så att de är utrustade med radio och ledning (se avsnitt 3). Sensorer hanteras på samma sätt oavsett plattform, varför sensorfunktionen endast beskrivs en gång (se avsnitt 4.6, 4.7 och 6.1).

4.2.1. Flygplansplattformsfederatens simulerade objekt

De objekt som inte är gråmarkerade simuleras i denna federat.

Flygplansplattform

Flygplan med stab, spaningsradar, IRST, SiS/Pejl, radar och radiostörustrustning

- Flygplan
 - Spaningsradar
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Radio SiS/Pejl
 - Radio Stör
 - IRST
 - Ledning
 - CNOApp Simuleras i nätverksfederaten
 - Stab
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Ledning Simuleras i stab blå federaten
 - CNOApp Simuleras i nätverksfederaten

4.3. Övriga markplattformarsfederat

I denna federat simuleras fyra specialförband som är utrustade med det positionerande närpejlsystemet TDOA (Time Difference of Arrival). Stabsfordonsplattformen är endast till för att ha något att sätta staben på. Övriga markplattformar bygger på plattformsobjekt enligt RPR-FOM:en, där fordon är Ground Vehicle. Plattformen bär övrig utrustning och ger möjlighet till förflyttning. Utrustning sitter placerad relativt föräldraobjektet. Alla plattformar konfigureras så att de är utrustade med radio och ledning (se avsnitt 3). Sensorer hanteras på samma sätt oavsett plattform, varför sensorfunktionen endast beskrivs en gång (se avsnitt 4.6, 4.7 och 6.1).

4.3.1. Övriga markplattformars simulerade objekt.

De objekt som inte är gråmarkerade simuleras i denna federat.

4 st

SF-plattform

Specialförband med radio och SiS-/Pejl-/Störutrustning (Radio TDOA)

- Människa
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Radio TDOA
 - Ledning
 - CNOApp Simuleras i nätverksfederaten

Stabsfordonsplattform

Markgående fordon med stab

- Fordon
 - Stab
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Ledning Simuleras i stab röd federaten
 - CNOApp Simuleras i nätverksfederaten
 - Ledning
 - CNOApp Simuleras i nätverksfederaten

Raketartilleriplattform

Markgående fordon med Raketartilleri

- Fordon
 - Raketartilleri
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Ledning
 - CNOApp Simuleras i nätverksfederaten

4.4. Stabsfederat

Blå och röd sida har varsin stabsfederat. Blå sidas stab finns på kustkorvetten medan röd sida har sin på ett landbaserat ledningsfordon (Stabsfordonsplattform). Staben kan vara en bemannad federat och då består denna av verktyg för att visualisera samt sammanställa data och skicka rapporter samt order. Om staben inte är bemannad sköts detta med regelverk på samma sätt som i ledningsfunktionen (avsnitt 3). Tabell 1 nedan visar stabsledningsfunktionen och övrig ledningsfunktion.

Tabell 1 Beskrivning av olika ledningsfunktioner

Ledningsfunktion för plattform	Ledningsfunktion för stab
Lägesbild (Ikoner, bäringar, Pejlkryss mm)	Lägesbild (Ikoner, bäringar, Pejlkryss mm)
Möjlighet att skicka meddelanden (order och rapporter).	Möjlighet att skicka meddelanden (order och rapporter).
Möjlighet att ta emot meddelanden (order och rapporter).	Möjlighet att ta emot meddelanden (order och rapporter).
	Osäkerhetsellipser
	Nätverksvy (logisk och geografisk)
Dela med sig/ta del av andras lägesbild	Dela med sig/ta del av andras lägesbild

För visualisering av en uppspanad nätverksbild i stabsverktyget används en riktad graf, där grafens noder är noder i det uppspanade nätet och dess bågar är riktade kommunikationslänkar.

4.5. Nätverksfederat

Det simulerade nätverket ska fungera för överföring av information. Det skall kunna signalspanas och störas och dessutom påverkas av CNO åtgärder. Egenskaper för nätverket som är relevanta ur signalspaningssyften simuleras i nätverket och behandlas sedan av SiS sensorer (se kapitel 6.1). CNO i nätverket behandlas i kapitel 4.5.9 och är en del av det som simuleras i nätverksfederaten (även kallad ComNet).

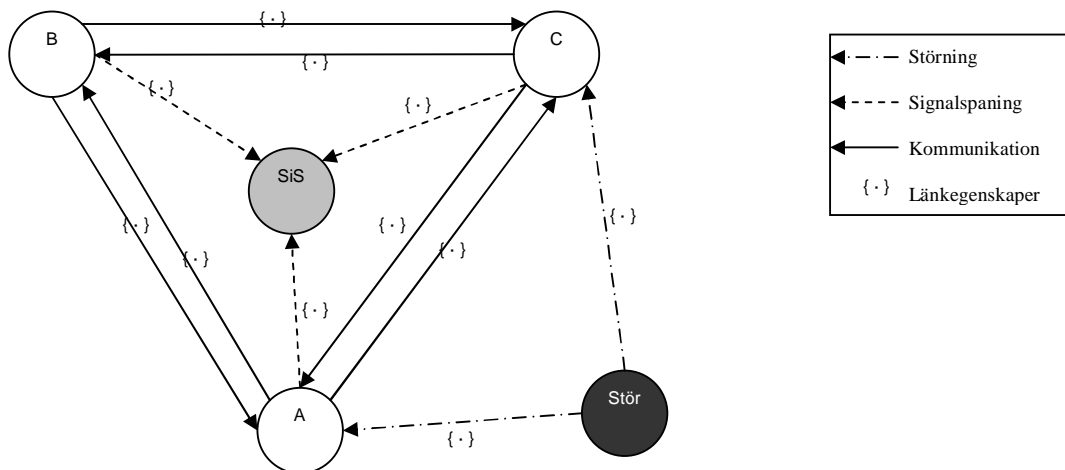
4.5.1. Kommunikation

Kommunikation hanteras med länkar mellan noder via radio eller kabel. En *nod* är ett kommunikationsobjekt som har en eller flera egenskaper avseende informationsöverföring, störning och signalspaning. All utrustning som på något sätt kan sända och/eller ta emot energi över ett medium (luften eller kabel) är en *nod*. Det betyder inte att all utrustning i realiteten kan utbyta information utan snarare att den kan påverka eller uppfatta överföring av information. Vi använder *nod* som en abstraktion för alla typer av kommunikationsutrustningar. Vi använder också benämningen NätNod som - till skillnad från nod - är en simulerad kommunikationsutrustning med datanätverkskapacitet.

Nätverket ska kunna överföra meddelanden med hänsyn tagen till bl.a. accessprotokoll och routing (se också LKS FEDEP steg 2 [FEDEP2v2]).

Då det finns ett krav på realtidsprestanda för nätverkssimuleringen kan man inte sända data traditionellt som ”på riktigt”, i form av bitar eller paket, utan det är nödvändigt att ta genvägar. Den modell som ansätts modellerar *länken* som den lägsta enheten.

En länk avser en överföringsväg från en *nod* till en annan *nod*. Mottagande nod äger länken och utför beräkningar för den. Med kommunikationsväg avses att mottagande nod förmår uppfatta den utsända energin. I figur 8 illustreras att den mottagna energin antingen kan vara ett meddelande från en annan nod (t.ex. länken A→C) eller störverkan (t.ex. länken Stör→A). Pilens riktning i figuren visar åt vilket håll energin strålar. Signalspanaren, som inte själv sänder utan bara lyssnar, har endast inkommande länkar.



Figur 8 Exempelnät med tre kommunicerande noder (A, B, C), en störare (Stör) samt en signalspanare (SiS)

En länk har ett antal egenskaper, t.ex. utsänd effekt, frekvens och bandbredd. Dessa egenskaper varierar med tiden och beror på de andra nodernas aktivitet både i kommunikationsavseende och i förflyttning av den fysiska enheten samt eventuella störares påverkan. Radiolänkar (till skillnad från kabellänkar) påverkar varandra. I Figur 8 påverkar t.ex. Störaren med länkarna Stör→A och Stör→C alla länkar i nätverket som är inkommande i nod A och nod C. Detta för att (enligt

figuren) nod A och nod C förmår uppfatta den energi störaren skickar och detta påverkar då andra inkommande länkar till dessa noder; dvs länkarna $C \rightarrow A$, $B \rightarrow A$, $B \rightarrow C$, $A \rightarrow C$. Motsvarande resonemang gäller även för vänligt sinnade noder i nätverket.

Dataöverföring

Det är den mottagande noden som gör en bedömning m a p samtliga sina inkommande länkars egenskaper, om ett meddelande kom fram eller ej.

En dataöverföring sker alltid över en (1) länk i taget. I fallet att en dataöverföring måste reläas över andra noder för att nå sin destination så innebär det att dataöverföringen sker i flera steg, ett över varje länk med tillhörande beräkningar. Detta för att ev. köande av meddelanden (dataöverföringar) ska kunna ske utan att påverka simuleringens kausalitet.

När en sändande nod har trafik till en annan nod meddelas alla andra noder som kan höra detta meddelande. När sändningen sker i ett kabelnät är det bara noder anslutna till det aktuella nätet som har möjlighet att uppfatta sändningen; sker sändningen över radio är det alla som förmår uppfatta den i luften utsända energin. I båda fallen är det med andra ord alla noder med inkommande länkar från den sändande noden som har möjlighet att uppfatta sändningen.

All kommunikation sker via ComNet som har global kontroll över signalvägen. Inga sändningskrockar antas ske inom nätet utan all kommunikation sker utan kollision via accessprotokollet TDMA (Time Division Multiple Access, används av t ex GSM). Dock tas hänsyn till störare och sändningar i andra nät. Störare måste därför tilldelas ett annat nät nummer än de noder de ska störa för att beaktas i störningsberäkningen.

Mottagande nod jämför inkommande meddelanden med annan, samtidigt inkommen, trafik/störning och beräknar vilka som nådde mottagaren och i vilken grad. Som underlag till dessa beräkningar har länkar och noder ett antal attribut. Några viktiga attribut följer.

Länkattribut

Frekvens	<i>Vilken bärvåg kommunikationen sker över</i>
Bandbredd	<i>Vilken bandbredd kring bärvågen som upptas</i>
Utsänd effekt	<i>Vilken effekt sändaren använder</i>
Kanal	<i>Vilken kanal kommunikationen sker över</i>
Kapacitet	<i>Vilken mängd datalänken förmår transportera per tidsenhet.</i>
Trafiklast	<i>Mängden data som länken för närvarande (i medel) transporterar.</i>

Dataöverföringsattribut

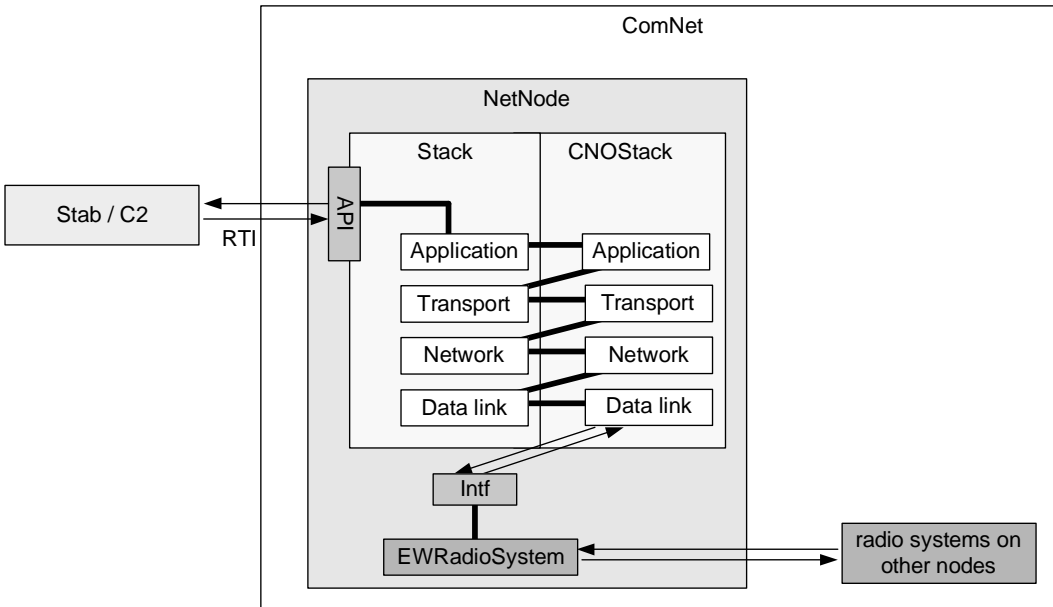
Sändningstid	<i>Tid för sändningen</i>
Mottagarid	<i>Destinationsnoden</i>
Storlek	<i>Antalet bitar sända (ger utbredning i tid tillsammans med datatakt)</i>
Innehåll	<i>Applikationsdata (Krypterat/autentiserat)</i>

Nodattribut

Fördröjning	<i>Den tid det tar för noden att vidareförmedla ett meddelande p.g.a. köer i noden.</i>
Mottagningsriktning	<i>Den geografiska yta som en trådlös mottagare förmår "höra i" (beror på antennen)</i>

4.5.2. NätNod

En NätNod är ett simulerat objekt som gör att en kommunikationsutrustning kan kommunicera i nätverk med en protokollarkitektur (jfr OSI referensmodell och Stack). NätNod benämns NetNode i figurer. I figur 9 visas schematiskt en NätNods ingående komponenter.



Figur 9 NätNod

När ett meddelande ska skickas från staben eller en plattform C2 (*command and control*, ledningssystem) lämnas det till ComNet. Utgående meddelanden (meddelanden som skickas från noden) går via ComNets API (*Application Interface*, applikationsgränssnitt) till applikationslagret i stacken. Om det utgående meddelandet tar sig igenom alla lager i stacken kommer det via "Intf" till EWRadiosystem som kan skicka det vidare till andra noder. Om ett meddelande inte tar sig igenom stacken, kan det t.ex. bero på att där finns en router (i "Network"-lagret) som talar om att det inte finns någon rutt (meddelandet kommer inte att kunna levereras) till mottagaren. Det kan även bero på attacker via CNO.

För inkommande meddelanden gäller att de går den omvända vägen. Först mottas signalen till EWRadioSystem som via "Intf" lämnar över till stackens accesslager. Om meddelandet tar sig hela vägen till applikationslagret kommer det att via API kunna levereras till stab/plattform C2. Om ett mottaget meddelande inte passerar genom stacken kan det bero på att det inte är den aktuella noden som är meddelandets slutliga adressat. Detta sker t.ex. då ett meddelande har reläats via den aktuella noden. Routern i stacken uppdaterar då mottagaradressen och skickar tillbaka meddelandet till EWRadioSystem som ett utgående meddelande via "Intf". Se även "4.5.6

Meddelandeöverföring” samt ”4.5.3 Router” där meddelandets väg beskrivs.

Stack

En nätverksnod innehåller en protokollstack. Stacken består av ett antal protokoll (lager) som man lagt på varandra. Protokollen implementeras som containrar med olika funktionalitet. Dessa containrar kan lätt bytas ut och rangeras om för att ge noden de egenskaper som önskas. Figur 9 speglar signalvägen som antingen är uppifrån och ned eller nedifrån och upp genom stacken. Olika protokoll är specialiserade på olika uppgifter (t ex routing, schemaläggning av kanaltillgång). Kopplingen mellan ComNet och stab/plattforms C2 (som finns utanför ComNet) sker via API medan kopplingen för sändning till andra noder sker via Intf (interface).

CNOStack

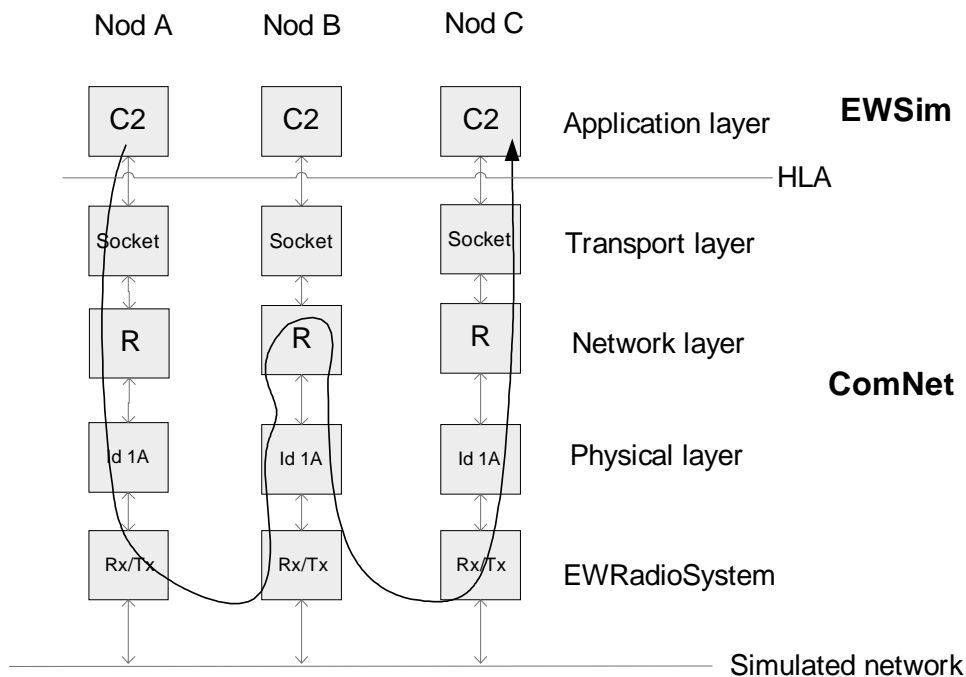
CNOStacken är en slags parallell stack för att för varje protokoll ha möjligheten att utföra CNO. Mer om hur detta går till finns under 4.5.9.

EWRadioSystem

Från en kommunikationsutrustning till en annan sker dataöverföringen endast via EWRadioSystem (se 6.1). En NätNod kan ha flera EWRadioSystem, där varje system är en anslutningspunkt mot en fysisk kommunikationskanal (t ex IEEE802.3 (Ethernet), IEEE802.11(WLAN),...), d v s en *länk*.

4.5.3. Router

Om det inte finns en direkt länk tillgänglig till den nod som ett meddelande ska skickas till används andra noder för att reläa meddelandet. Vilken väg som ett meddelande ska ta styrs av nodens router.



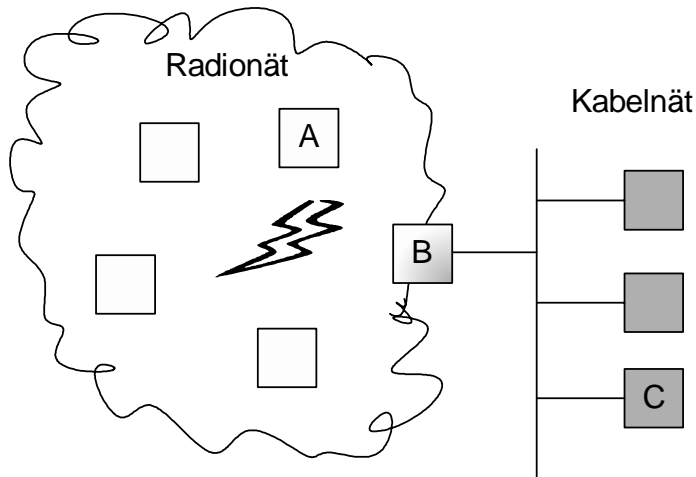
Figur 10 Signalflödet i stacken vid en router.

Varje nod har ett applikationslager (App), en router (R), fysiskt nod-id (Id) samt en mottagare/sändare (Rx/Tx). Meddelandet startar i applikationslagret hos nod A. Slutdestination är nod C, men det finns ingen direkt länk. Därför väljer routern i nod A att istället skicka meddelandet till nod B (vanligen kallat ”next-hop” i routingsammanhang). När meddelandet når routern i nod B ser routern att slutlig mottagare inte är den egna noden. Därför skickas meddelandet ut igen men nu med den nya mottagaren nod C. Routern i nod C ser att slutlig destination är den egna noden och levererar därför vidare meddelandet till sitt applikationslager.

Vilken väg ett meddelande ska ta väljs så att det blir minst antal hopp (antal noder som reläer) viktat med den kapacitet varje länk har.

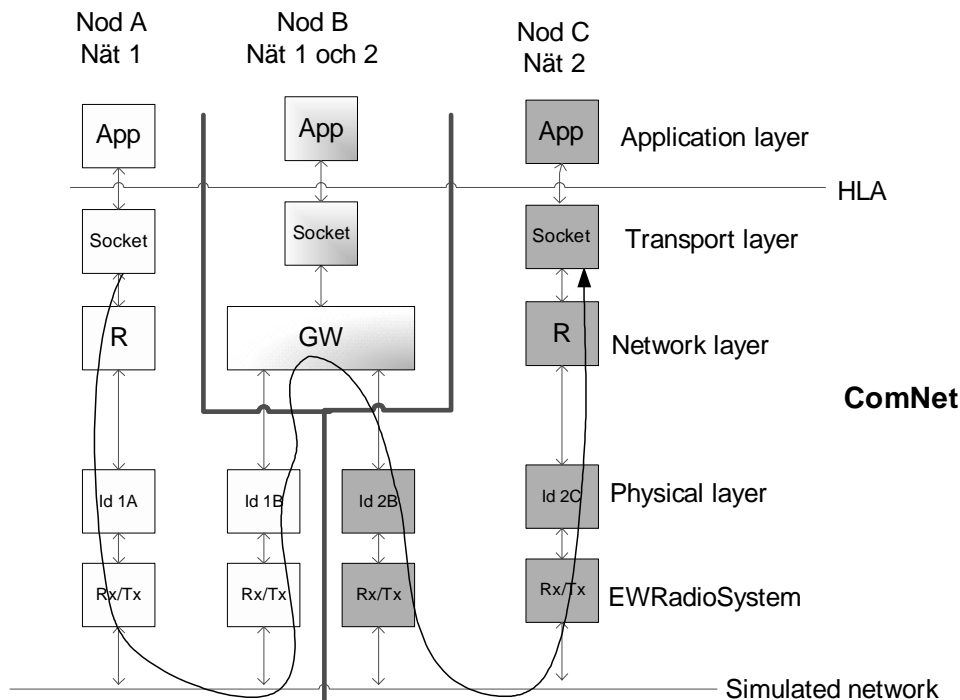
4.5.4. Gateway

För att kunna koppla ihop olika nät med varandra och skicka information från ett nät till ett annat måste en NätNod ha flera gränssnitt. Figur 11 nedan illustrerar kopplingen mellan två olika typer av nätverk.



Figur 11 Sammankoppling av två olika nätverk

För att hantera kopplingen har den tvåfärgade noden B två interface och två delvis parallella stackar. Sett schematiskt inuti noden får vi nod B i Figur 12. Linjen från nod till nod visar signalflödet då ett meddelande skickas från nod A till nod C.

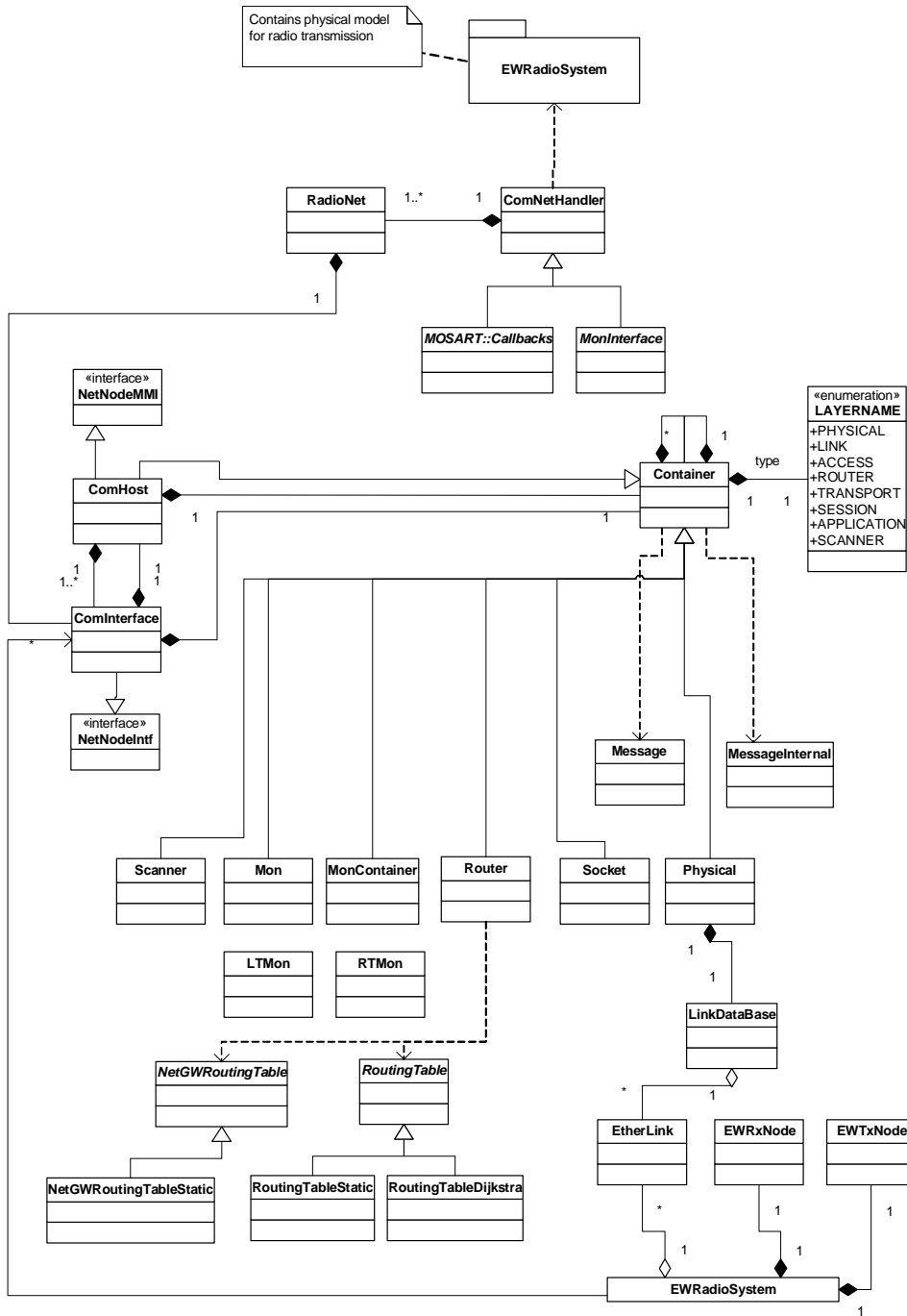


Figur 12 Signalflöde i stacken vid en gateway.

De båda noderna till höger och vänster är vanliga noder med varsin mottagare/sändare (Rx/Tx), fysiskt nod-id (Id) router (R) samt applikationslager (App). Noden i mitten innehåller istället för router (R) en gateway (GW) som förutom den funktionalitet som finns i en router även kan routa till andra nät. För att detta ska fungera krävs att routingtabellen utökas från att enbart innehålla information om noderna i nätet till att även ha information om hur ett meddelande kan hitta ett annat nät. En router har tillgång till det egna nätets routingtabell medan en gateway har tillgång till flera näts routingtabeller. I fallet ovan kommer alla meddelanden som skickas i nät 1 och är adresserade till noder i nät 2 att dirigeras till nod B. När meddelandet når nod B kommer gatewayen att kunna hänvisa vidare till noder i nät 2 eftersom den har tillgång till båda nätens routingtabeller.

4.5.5. NätNod – klassdiagram

En NätNod byggs i koden upp av klasserna ComHost och ComInterface. Detta för att kunna hantera situationen med delvis parallella stackar i en gateway. ComHost innehåller den gemensamma delen av stacken medan ComInterface innehåller de delar av stacken som dupliceras i en gateway. En nätnod består alltså av endast en ComHost samt en eller flera ComInterface. Det statiska klassdiagrammet är illustrerat i Figur 13.

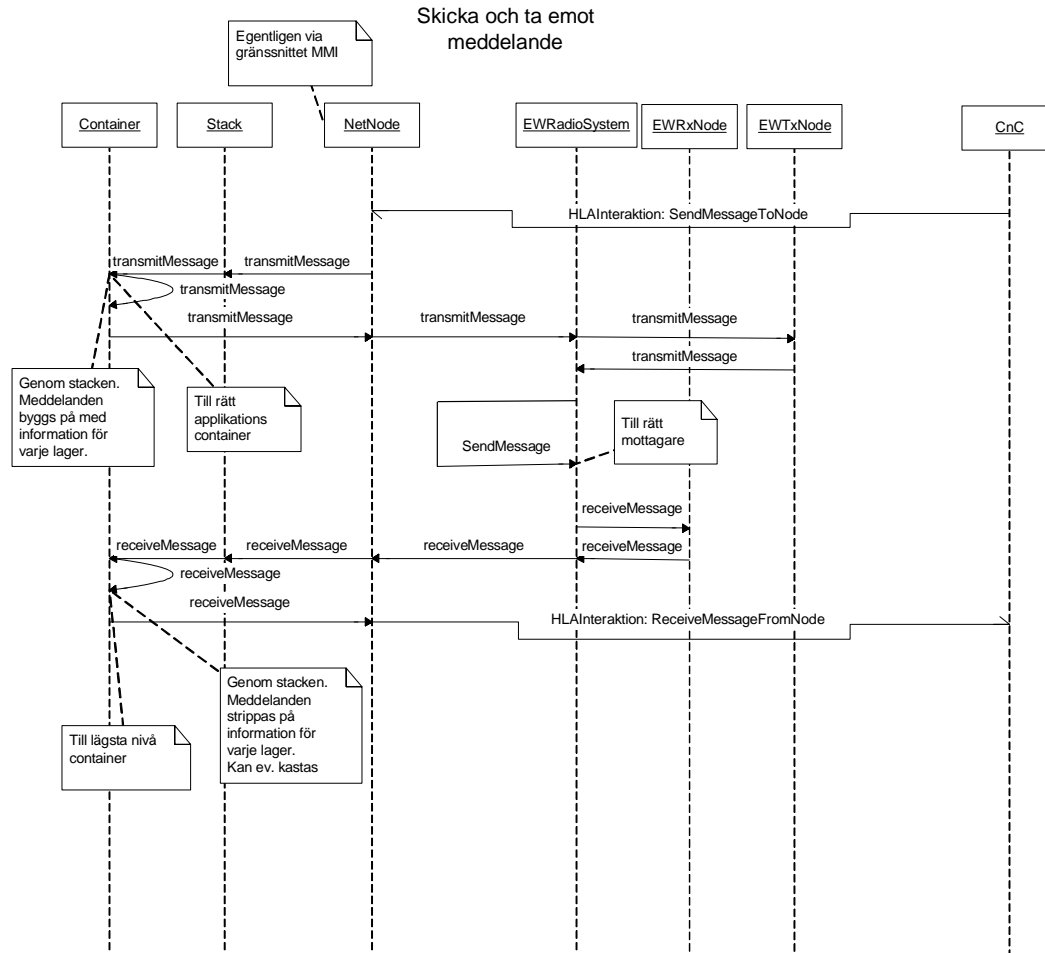


Figur 13 Klassdiagram för NätNod.

4.5.6. Meddelandeöverföring

HLA-Interaktionen ”SendMessageToNode” går från stab/plattforms C2 till ComNet då ett meddelande skickas iväg. Interaktionen innehåller data om sändare, slutlig mottagare samt själva meddelandet.

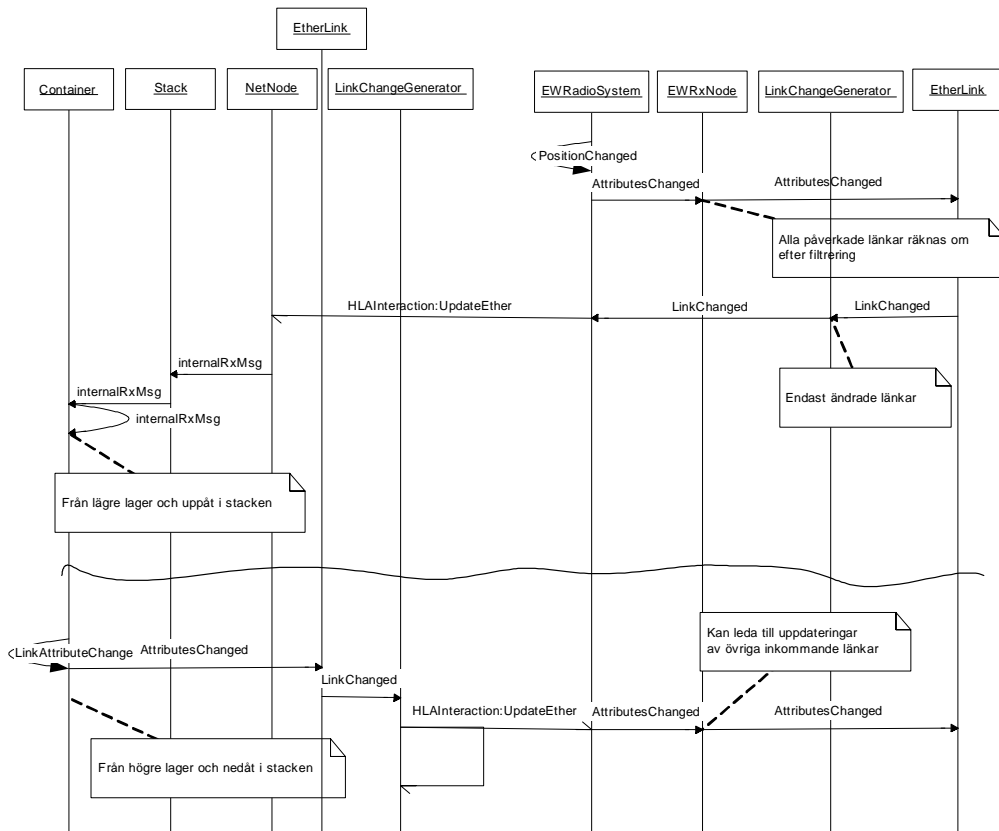
Den slutliga mottagarens stab/plattforms C2 får meddelandet från ComNet via HLA-Interaktionen ”ReceiveMessageFromNode”.



Figur 14 Sekvensdiagram för meddelandeöverföring.

4.5.7. Länkuppdatering

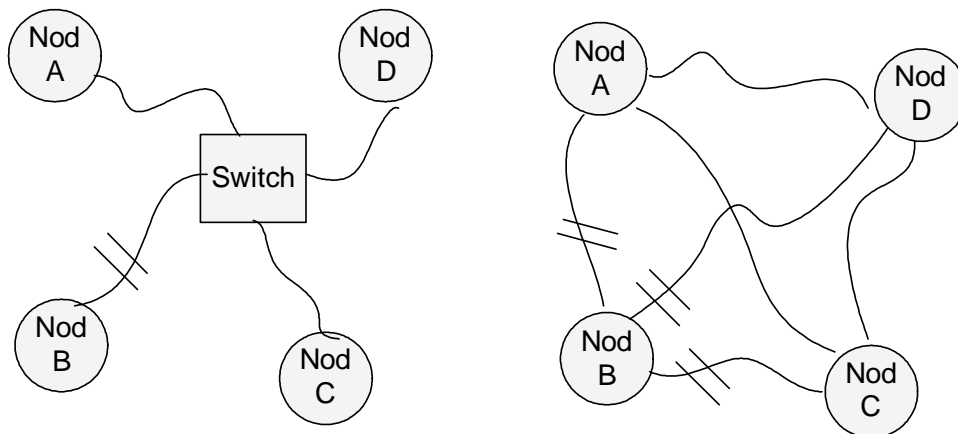
Alla beräkningar av resultatet av en meddelandeöverföring baseras på länkobjekt. Länken och dess attribut ägs visserligen av mottagaren, men den kan påverkas av alla andra noder i nätverket. Figur 15 visar ett sekvensdiagram för uppdateringsförloppet av en länk och hur förändringar sprider sig till alla berörda parter. Figuren är separerad i två delar av det vågiga strecket. I den övre delen börjar sekvensen med att en position har ändrats, men förloppet gäller för alla typer av attributförändringar vilka inkluderar t ex en aktiv störare, meddelandeöverföring, trafikförändringar. Den undre delen visar förloppet då attributförändringen kommer från Stacken. Sådana förändringar har att göra med att protokollet anpassar sig till omgivningen, t ex ändrad datatakt över en länk p g a störning.



Figur 15 Sekvensdiagram för länkändringar.

4.5.8. Kabel och radio

Beskrivningarna ovan om hur länkuppdateringar och dataöverföring sker är gjorda med hänsyn till radiofallet då det är det mest generella. Fallet då överföringen sker via kabel antas som ett specialfall av radiofallet. Kabelnätet modelleras som ett radionät där alla noder har direktkontakt med alla andra noder. Det vill säga kommunikationen behöver aldrig gå via en annan nods router för att nå mottagaren. Beräkningarna för signalens transport genom rymden byts ut till att gälla kabel istället för radio.



Figur 16 Kabelnät med switch modelleras som noder med multipla länkar

Att kabelnätet modelleras som ett radionät motsvarar att alla noder skulle ha en kabel direkt till alla andra noder vilket kanske inte alltid är fallet. Men resultatet blir i stort det samma som om noderna skulle ha kopplat in sig med varsin kabel till en switch, se Figur 16. För att motsvara ett switchat nät måste man ta hänsyn till alla länkar från aktuell nod om t.ex. ett kabelbrott mellan nod och switch ska simuleras.

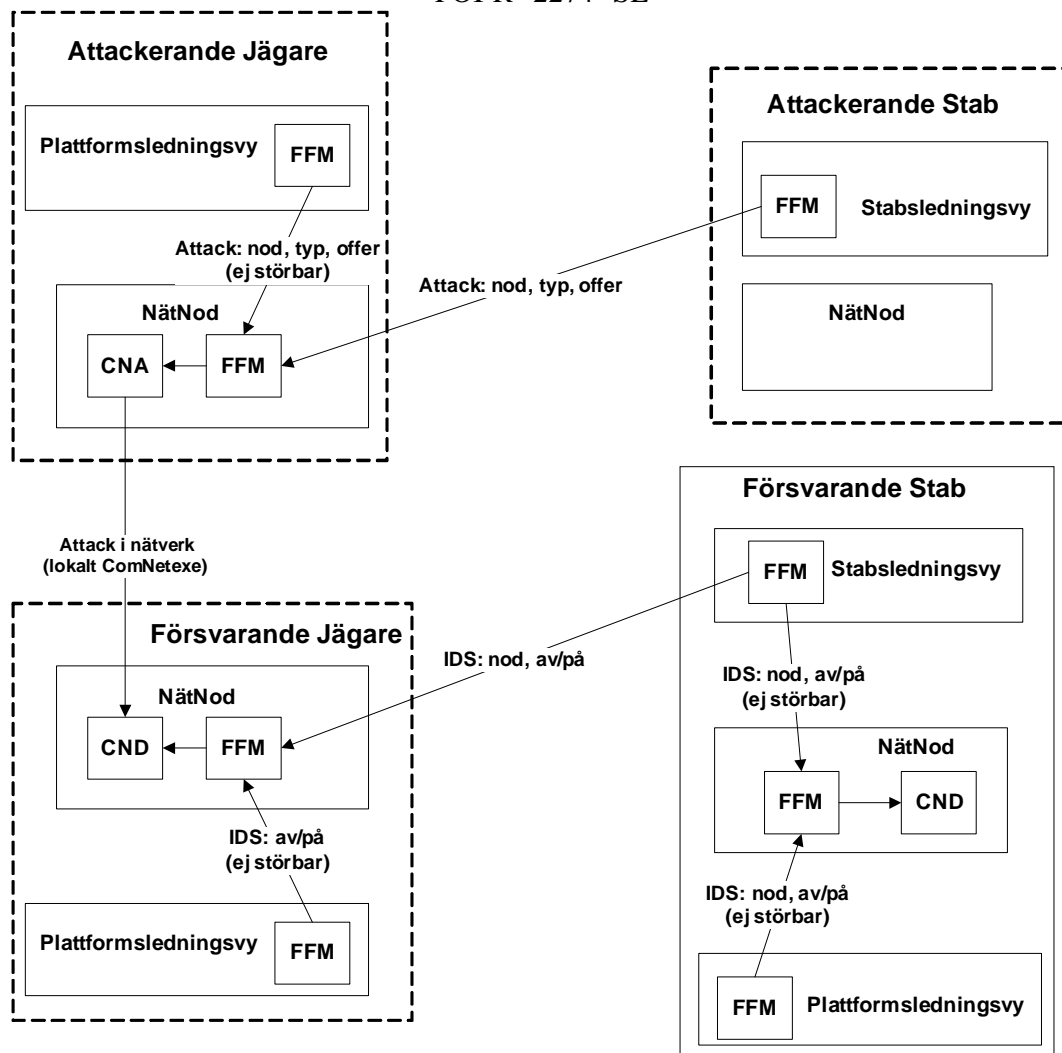
4.5.9. Nät – CNO

I detta kapitel beskrivs hur CNO-egenskaper är designade för att simuleras i nätverksfederaten, som en del av nätverket. En CNO-container finns i alla noders ComNet-stack. Gränssnitt för att utföra Computer Network Operation (CNO) attack och försvar tillhandahålls av alla vyer som implementerar CNO-förmåga.

Dataflöden

Figur 17 visar ordergivningen mellan noder i simuleringen. Staben befinner sig på en plattform (bara en nod) i figuren. Staben sköter vanligen ordergivning, och kan beordra noder (och deras plattform) att utföra order. Både stab och nod kan välja om IDS (Intrusion Detection System) ska vara påslaget eller ej.

Konfigurering av CNO och dess förutsättningar görs via staben och NetScene. Ordergivning sker via störbara FFM-meddelanden (FOI Fast Message) som går över HLA. Anledningen att vyer och noder inte hör ihop är att de finns i olika federater och måste alltså utbyta information med HLA (över RTI). Alla meddelanden (FFM) går över RTI. Det beräknas generellt om störningar påverkar mottagande av FFM-meddelanden. Ett viktigt undantag mot störning är när en vy på en plattform meddelar sin egen förmåga (jmf localhost). Själva attackerna sker internt i simulatoren på aktuell nätverksnivå, och genererar ingen HLA-trafik.

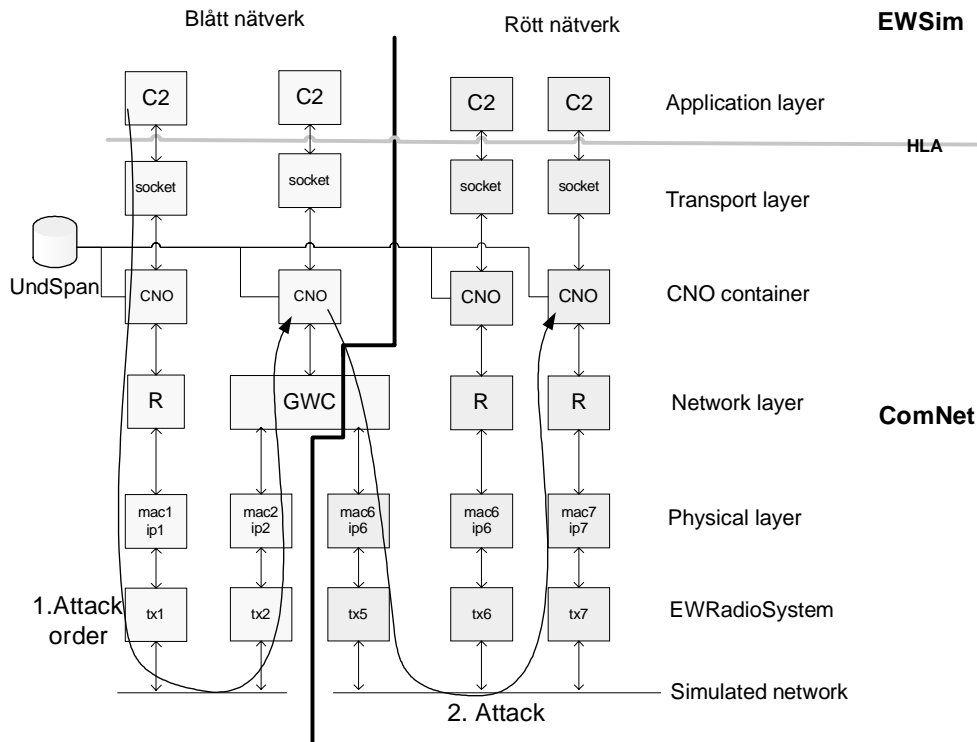


Figur 17. Ordergivning mellan nätverksnoder

Översikt CNO och nätverksstacken

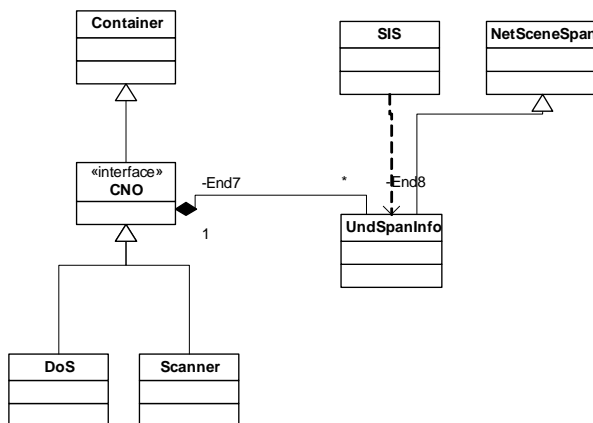
Figur 18. Översikt CNO och nätverksstacken, visar de lager som är implementerade i den generella nätverksstacken för några nät-noder. Stora delar av nätverksstacken måste abstraheras för att minska datamängden som skickas via det simulerade nätet och kunna utföra simuleringen i realtid. Attacker och försvar har förenklats genom att kommunicera effekter via FFM-meddelanden istället för att skicka verklig trafik.

En CNO container i en nod (id2) tar emot order om CNO attack, som skickar den vidare via en Gateway CNO (GWC) till fiendens nät. På röd sida kan man i figuren se hur meddelandet skickas via nod id6 (antar förfalskat id - även kallat spoofat IP; identitet är en abstraktion av IP-nummer och i övrig beskrivning av CNO funktionalitet används IP istället) vidare till mottagaren id7. Noderna använder en CNO container för att kommunicera attack (snd - send) och försvar (rec - receive). Notera att det finns två stycken id6 på röd sida, och båda hör trafik riktad till id6. GWC tillåter inte att trafik går från rött nät via router/gateway vidare till blått nät eller tvärt om.



Figur 18. Översikt CNO och nätverksstacken

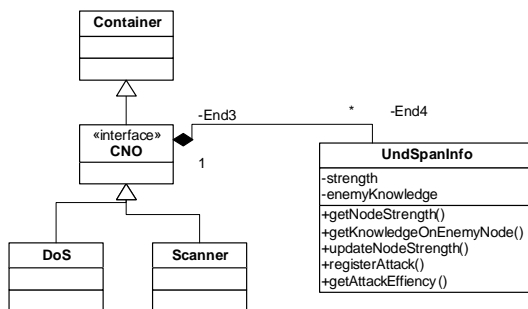
I Figur 19 finns en överblick av alla CNO containrar och vyer som kan samverka och visa resultat av CNO. Vyerna uppdateras genom att få meddelanden från de noder som ingår i en cno attack. De egna stabsvyerna kan kommunicera direkt med respektive CNO container för att sätta igång en attack.



Figur 19. Översikt CNO Klassdiagram

Sannolikheter

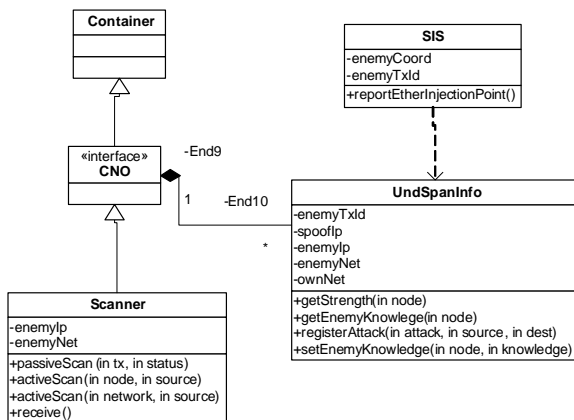
I denna version är ingen slump inblandad vid en CNO-attack utan den lyckas alltid och är det en "active scan" så returneras dessutom maximal enemyKnowledge. Storleken på enemyKnowledge är ett mått på hur mycket information som inhämtades under attacken. Denna information bestämmer vilka typer av attacker som kan utföras i fortsättningen.



Figur 20. Klassdiagram för CNO beräkningar (attack och försvar)

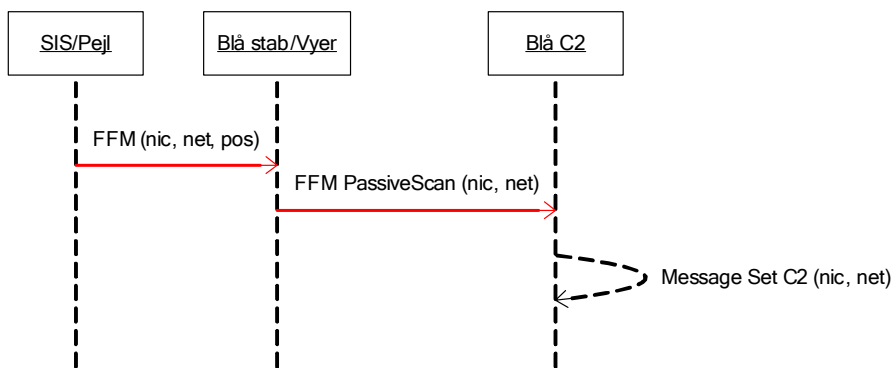
Attacker och försvar

För att kunna komma åt fiendens nät och plantera attacker behöver man först en injektionspunkt i nätet i form av en sändare. I radiofallet så kan man få tag i sändaren via SIS/Pejl. Kablar kan inte spanas upp men trafiken kan avlyssnas om det finns en övergång till radio.



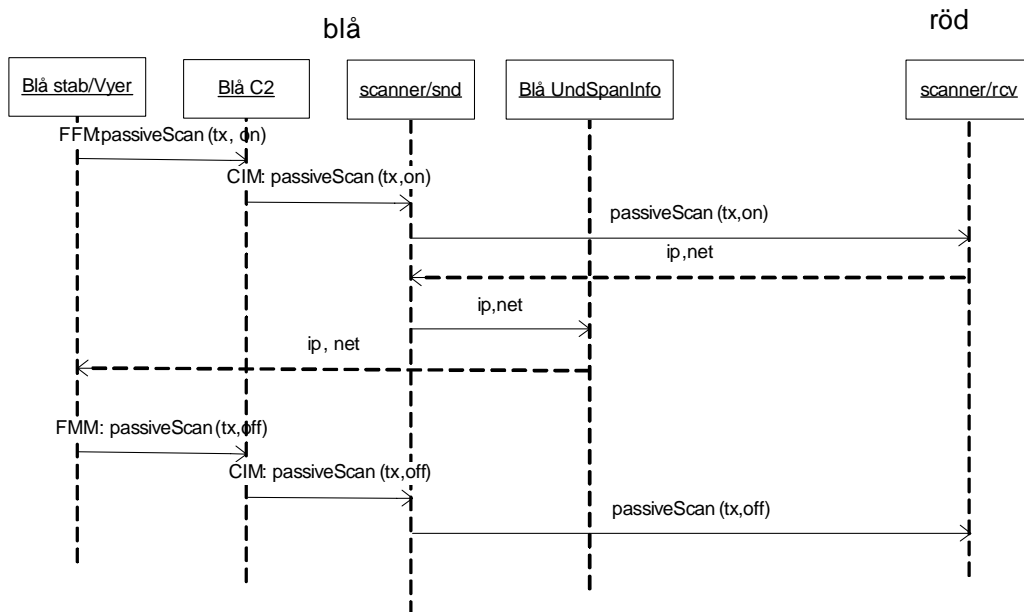
Figur 21. Injektion, scanner och SIS

Via SIS får man tillgång till en fiendenods NicId (motsvarar ip-nummer) och NätId (motsvarar nätverksmask). Genom att sätta dessa id på en egen nods interface (kan finnas flera men bara ett med cno förmåga) få man alltså en injektionspunkt som kan användas för att komma åt motståndarens nät.



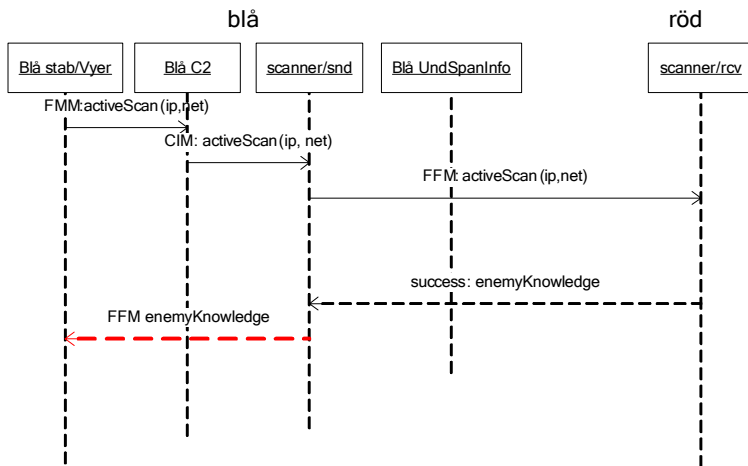
Figur 22. Sekvensdiagram Injektion

Samtidigt som en injektionspunkt sätts upp så aktiveras passiv scanning i samma nod. Den trafik som passerar noden tittar på meddelandena för att kunna uppfatta vilka ip- och nätnummer som är aktiva. På så sätt få man efter en tid veta vilka noder som finns i motståndarens nät under förutsättning att något skickas i det avlyssnade nätet.



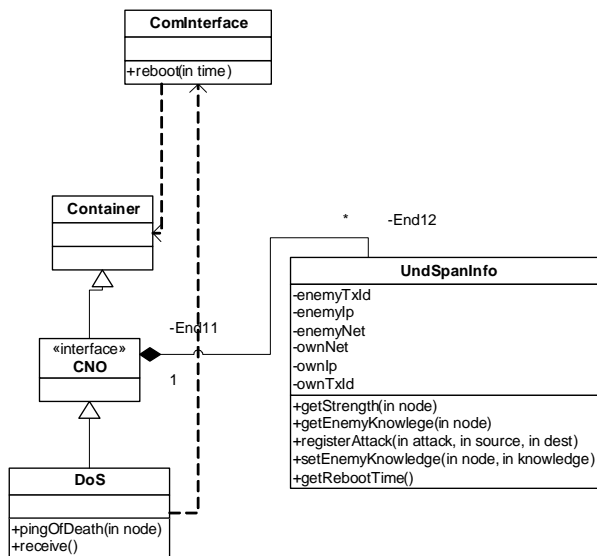
Figur 23. Sekvensdiagram för passiv scanning

När man känner till ip- och nätnummer så kan man fortsätta med en aktiv scanning (egentligen scanna noder på nätverket, deras öppna portar, samt göra fingerprint av tjänster och operativsystem) för att leta upp fiendens svagheter. Denna information samlas som en sannolikhet enemyKnowledge för varje upptäckt fiendenod. Just nu tilldelas alltid enemyKnowledge värdet 1, dvs man får reda på allt om motståndarens nod som behövs för att kunna göra andra typer av attacker. Just nu är den enda attacken som kan göras DenialOfService (DoS).



Figur 24. Sekvensdiagram för aktiv scanning

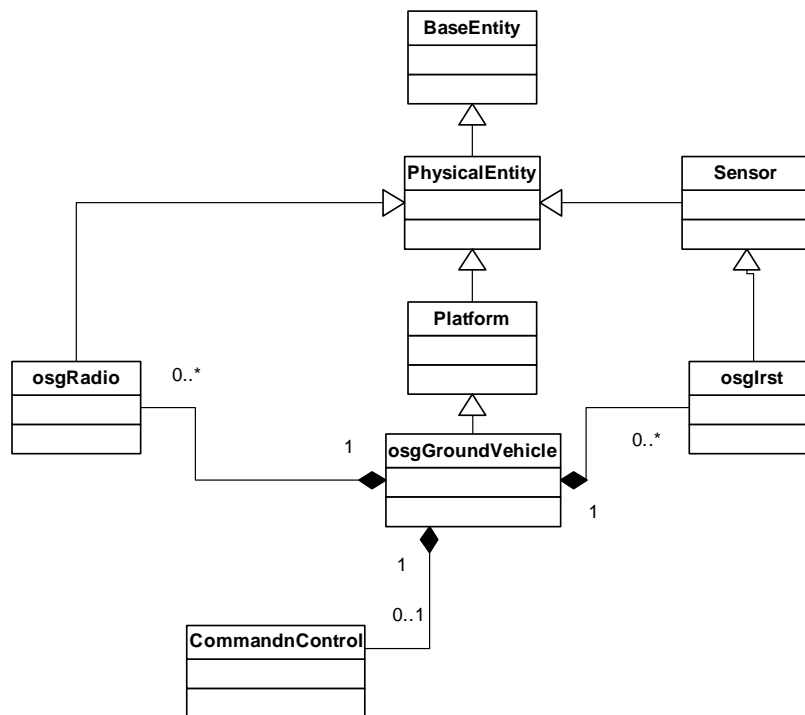
DenialOfService består i en implementation av pingOfDeath som gör att fiendens dator fryser (egentligen bootar om). Den är riktad mot en nod i taget. Den implementeras genom att alla meddelanden till och från noden kastas under en fördefinierad tid.



Figur 25. Klassdiagram DoS

4.6. IRSTplattformsfederat

I denna federat simuleras en IRST-enhet, vilken består av en plattform, t.ex. ett markgående fordon, med en IRST som spaningsensor och en radio (kan innehålla nätverksfunktion) för kommunikation med stab eller andra enheter. Plattformen (osgGroundVehicle eller någon annan farkost) med dess basklasser tillhandahåller spatial information om fordonet samt funktioner för att förflytta fordonet. Till fordonet är kopplat en IRST som spanar av omgivningen och kan rapportera upptäckta och eventuellt karakterisera objekt som närmar sig.



Figur 27 Klassdiagram för IRST-enhet.

4.6.1. IRSTplattformsfederatens simulerade objekt

De objekt som inte är gråmarkerade simuleras i denna federat.

IRST plattform

Markgående fordon med IRST.

- Fordon
 - IRST
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Ledning
 - CNOApp Simuleras i nätverksfederaten

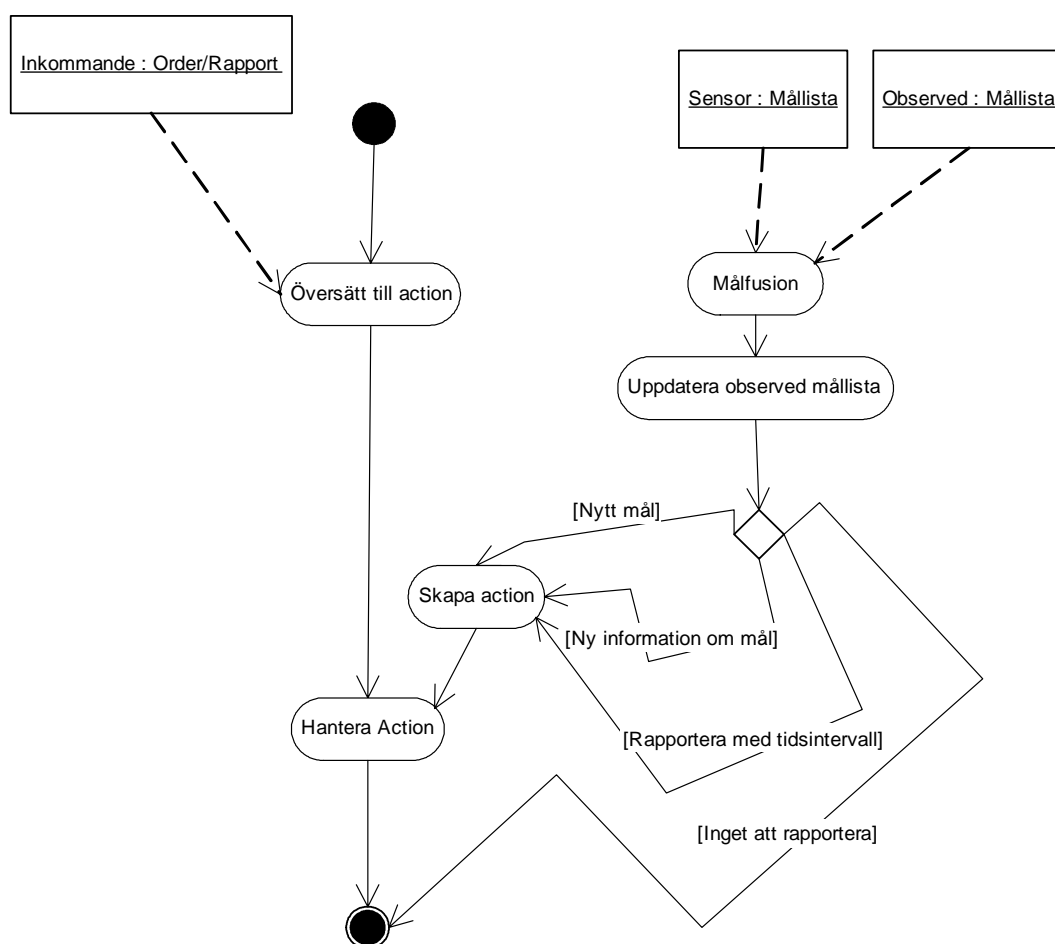
4.6.2. IRST Interaktioner

IRST-plattformen kan ta emot och skicka rapporter och order. Dessutom måste IRST:n känna till signaturen för de objekt den spanar efter. Dessa signaturer sätts för varje objekt via interaktioner som innehåller data om signaturen som funktion av aspektvinkel, våglängdsområde och motor- eller annan status. Index som beskriver status kan också ändras under simuleringens gång via interaktioner.

De rapporter och order som kan skickas är (vilken del som agerar eller skapar denna order eller rapport anges inom parentes):

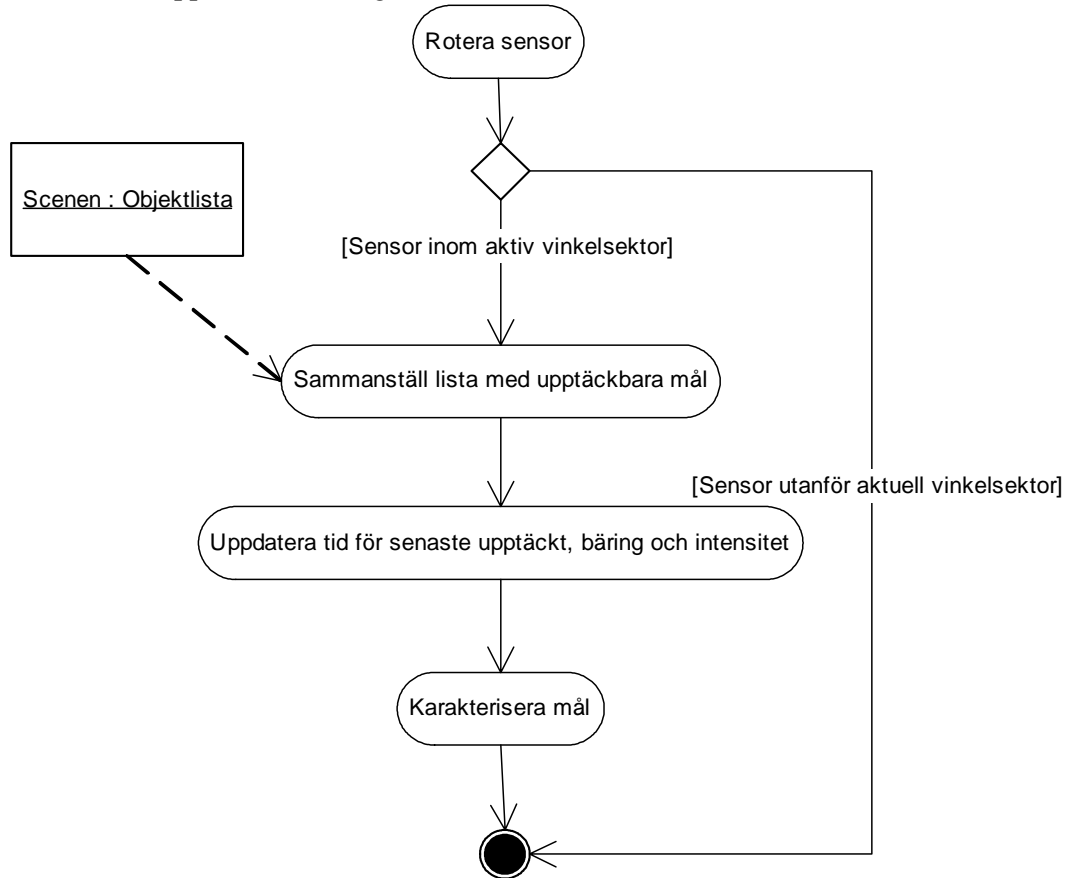
- Skicka rapport om egen position (CommandnControl)
- Skicka rapport om bäring och eventuell klassificering till ett eller flera målobjekt (CommandnControl)
- Ta emot order om att skicka lägesbild (alla eller enstaka mål) (CommandnControl)
- Ta emot order om att vara radiotyst (osgRadio)
- Ta emot order om att avbryta radiotystnad (osgRadio)
- Ta emot lägesbild (ett eller flera mål med bäring eller positionsangivelse) (CommandnControl)
- Ta emot order om förflyttning (osgGroundVehicle)
- Ta emot order om att bara rapportera om mål i viss vinkelsektor (osgIrst)
- Ta emot order om att rapportera med visst tidsintervall (CommandnControl)
- Ta emot order om att slå av eller på spaning (osgIrst)

Rapporter och order som kommer via nätverk eller radio översätts till en åtgärd (action) (om simuleringen är inställd så att IRST-plattformen skall fungera autonomt). Information från sensor kan också resultera i en åtgärd, dvs att en rapport skickas från IRST-plattformen till stab eller till andra plattformar. Flödesdiagrammet för detta visas i Figur 28. Detta sätt att hantera order, rapport och sensorinformation är gemensamt för ledningsfunktionen på alla sensorplattformar.



Figur 28 Hur order, rapport och sensordata tas om hand av ledning på sensorplattform (CommandnControl).

Objekt som är upptäckta och eventuellt karakteriserade av IRST:n jämförs med tidigare upptäckta mål (lokal lista i ledningsfunktionen) och ifall nya objekt upptäckts, ny information om målet framkommit eller om plattformen fått order om att rapportera regelbundet skapas en action som resulterar i att en rapport skickas. Figur 29 visar flödet för hur IRST-sensorn arbetar.

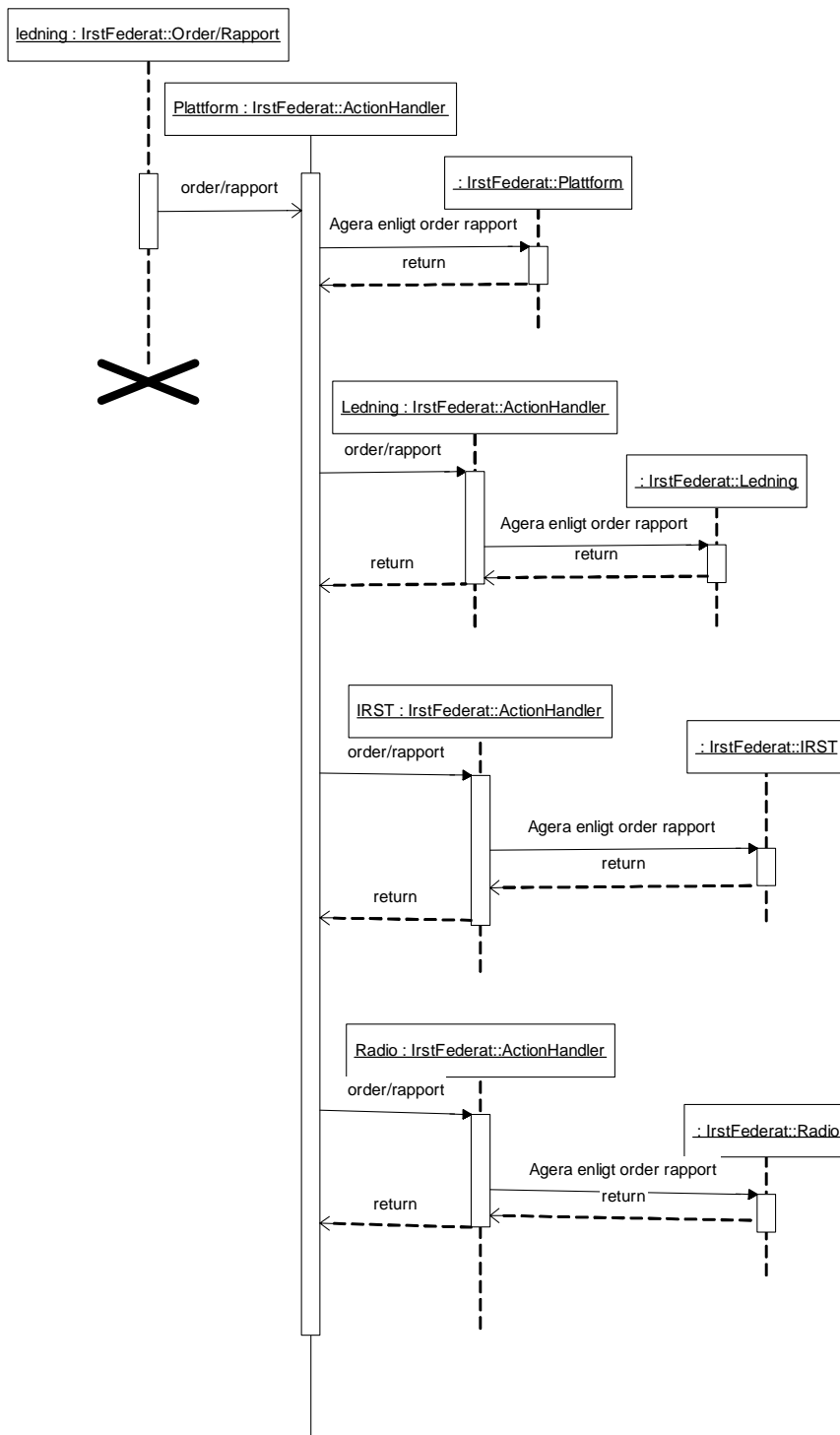


Figur 29 Hur IRST-sensorn arbetar internt

Egenskaper som IRST:n utnyttjar för att karakterisera mål bygger på ifall:

- Målet är upplöst (kan identifieras) - bestäms av att bilden av målet i sensorn blir ett visst antal bildelement.
- Målet befinner sig ovanför horisonten (målet är en flygande farkost - flygplan, helikopter, UAV - och inte ett fartyg eller ett markfordon) - beräkningar utnyttjar information om horisontens läge, målets position i höjddled, målets utbredning i höjddled och sensorns vinkelupplösning.
- Målets intensitet ökar - tyder på att målet närmar sig (om ökningen inte är alltför snabb vilket kan tyda på att signaturen ändrar sig).
- Målets vinkel ändras inte över tiden - tyder vid intensitetsökning på att målet närmar sig den egna plattformen.

En åtgärd som skapas skickas till *ActionHandlern* i bärande plattform (t.ex. *osgGroundVehicle*) som hanterar åtgärder som den skall hantera och i annat fall skickar åtgärden vidare till sina underobjekt som i detta fall är en *CommandnControl*, en *osgIrst* och en *osgRadio*. Figur 30 visar ett sekvensdiagram för åtgärdshantering i IRST-plattformen.



Figur 30 Åtgärdshantering i IRST-plattformen

en annan federat fungerar som ledning/stab till radarn och man vill kunna styra radarn genom att beordra olika hantering av radarn. Samma resonemang gäller både för spaningsradarn och för radarstöraren varför endast spaningsradarn beskrivs här.

Interaktioner mellan radar och annan federat

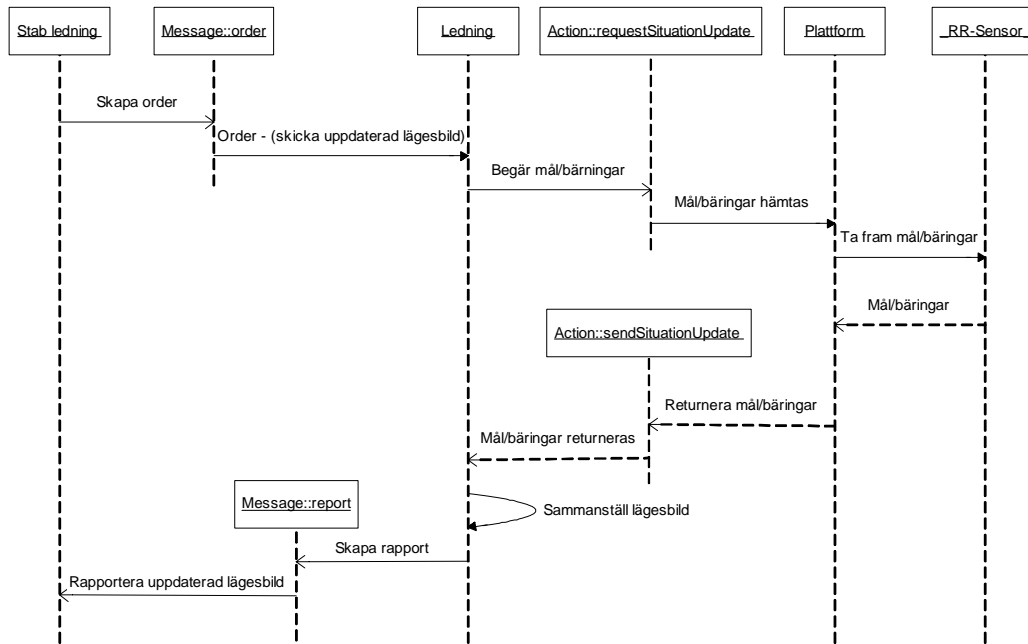
För att en spaningsradarfederat från EWSim ska kunna se mål som är skapade på andra federater måste dessa ärva från *PhysicalEntity*, se Figur 31, enligt [RPR-FOM]. *PhysicalEntity* tillhandahåller *RadarCrossSectionSignatureIndex* som behövs för att målet ska få en radarmålära i EWSim, vilket i sin tur gör det synligt för radarn. Radarn gör sedan beräkningar med radarsignaler internt eftersom beskrivningen av radarns signaler i RPR-FOM inte är tillräckligt ingående. Detta betyder att ingen information om vad radarn sänder ut skickas med hjälp av HLA. Det enda som skickas är radarns inställningar.

Inställningar i radarn såsom radarmod (på/av etc.), frekvens, sändeffekt m.m. hanteras i radarklasserna (*EWRadar*, *EWTAR*). Denna interaktion är en utökning av RPR-FOM:en och är gjord för få en generell interaktion för att uppdatera parametrar på ett godtyckligt objekt. I interaktionen anges objektets namn (i simuleringen) som sträng, parameter-ID som heltal och nytt parametervärde som sträng. Det är upp till varje objekt att implementera funktioner för att skicka och ta emot parametrar med hjälp av denna interaktion och att implementera parameter-ID så att avkodning vid mottagande av interaktionen kan ske.

Sammanfattat kan man säga att spaningsradarn i EWSim kräver att objekt som skapas av andra federater ärver av *PhysicalEntity* från RPR-FOM och att övriga federater som har en egen radarmodell måste använda *ObjectParameterUpdate*-interaktionen för att publicera och prenumerera på sina radarspecifika parametrar.

Interaktioner för att kunna styra radar

En spaningsradarplattform har en egen ledningsfunktion genom att fordonet ärver av klassen *Platform*, se Figur 2. För att styra radarn ger man en order till dess ledning som sedan hanterar radarn genom olika händelser, *Actions*. Händelser hanteras i *ActionHandler* vilken omvandlar händelserna till funktionsanrop i radarplattformen och ordern utförs. För att illustrera vägen från att order ges tills det att den exekveras visas här ett exemplifierande sekvensdiagram för fallet då staben begär uppdaterad lägesbild från radarplattformen (radarplattformen uppdaterar i detta fall sin egen lägesbild innan den skickas), se Figur 32.



Figur 32 Sekvensdiagram för fallet då spaningsradarplattformen tar emot och exekverar order om att skicka uppdaterad lägesbild

Staben skickar en order till ledningen i spaningsradarplattformen och begär en uppdaterad lägesbild. Ledningen omvandlar ordern till en händelse (*Action*) som hanteras (av *ActionHandler*). Spaningsradarplattformen tar fram aktuella målobjekt och/eller bärningar som ledningen sedan sammanställer till en lägesbild, vilken i sin tur rapporteras till staben.

Motsvarande förlopp sker för övriga order som kan ges till spaningsradarplattformen. Det som skiljer mot vad som visas i Figur 32 är typen av order och vad som returneras. I vissa fall sker ingen retur utan ordern kan vara av typen att spaningsradarplattformen ska ändra en viss inställning, t.ex. att sända intermittent.

4.8. SiS-plattformsfederat

Denna federat simulerar ett markgående fordon med radio samt signalspanings-/pejlutrustning. Signalspaningsutrustningen beskrivs i 6.1.

4.8.1. SiS-plattformsfederatens simulerade objekt.

De objekt som inte är gråmarkerade simuleras i denna federat.

SiS plattform

Markgående fordon med SiS-kapacitet

- Fordon
 - Radio
 - NätNod Simuleras i nätverksfederaten
 - Radio SiS/Pejl
 - Ledning
 - CNOApp Simuleras i nätverksfederaten

4.9. Federation Management federat

Federationskontrollfederat som styr scenariots exekvering genom att hantera tiden och händelser som start, stopp och paus, samt att specificera i vilket terrängavsnitt aktuell simulering skall köras se [MOSART].

4.10. Spelledningsfederat

Distribuerar scenariokomponenter (centralt) till respektive federat vid uppstart samt visar verkligt mark läge och uppfattat läge för respektive sida [AGR].

4.11. Loggerfederat

Logger kör hela tiden i bakgrunden och loggar kontinuerligt de data som utbyts över HLA. Den fungerar snarlikt en ”bandspelare” med funktioner som play, rec och pause. Record under en simulering sparar ner simulerad data (HLA objekt) till fil. Play spelar upp inspelade data från en logfil i federater som prenumererar på de HLA objekt som finns i logfilen [EW].

5. Infrastruktur och infrastrukturgränssnitt

Infrastrukturen består av HLA IEEE 1516, pRTI samt ett GB nätverk.

Till nätverket är datorer som har Windows XP med Nvidia grafikprocessorer anslutna. För att slippa starta upp alla federater manuellt från den dator de ska köras på finns hjälpprogram kallat remProcMgr samt remProcSrv. Genom att lägga in remProcSrv på de datorer som ska vara med i federationen kan hela federationen startas från en dator (den med remProcMgr). Vilka datorer och program som ska vara med i federationen sparas i en konfigurationsfil som kan öppnas med remProcMgr.

6. Funktionsbeskrivningar

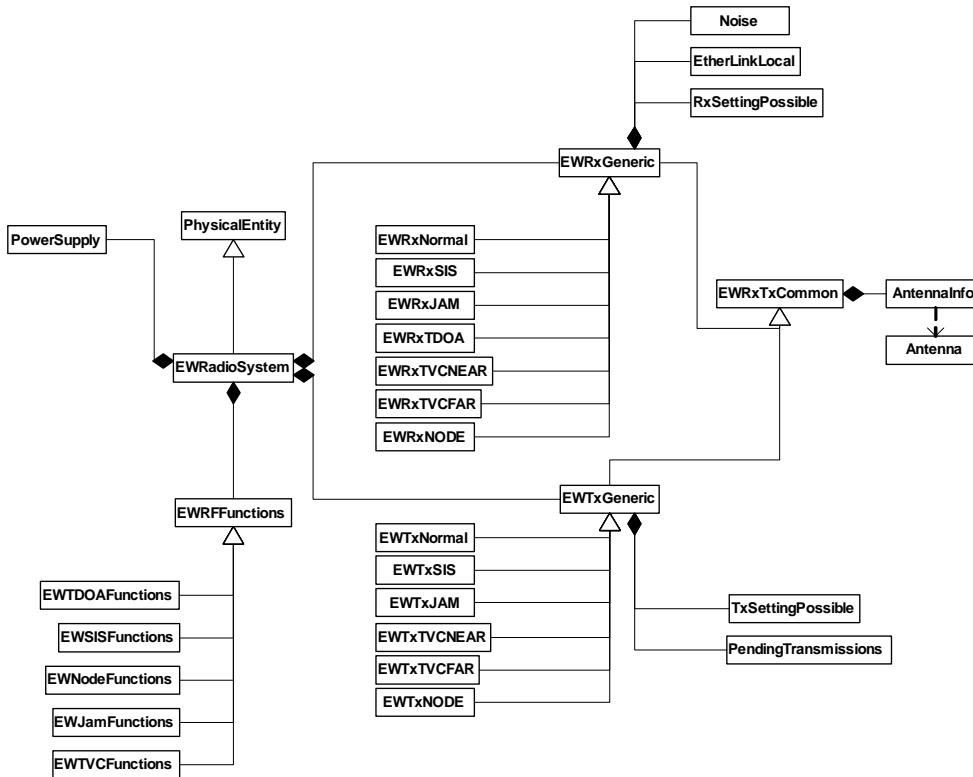
I detta kapitel beskrivs funktioner som involverar mer än enstaka objekt. Till exempel beskrivs här spaning vilket är en funktion som involverar både det som ska spanas på samt spaningsutrustningen.

6.1. Signalspanings- och störsystemsfunktion

I grunden är en SiS-mottagare en vanlig radiomottagare med en speciell slags antenn och diverse extra funktioner för riktungs- och tidsbestämning. En störsändare kan betraktas som en vanlig radiosändare med i vissa fall en speciell slags antenn och förmåga att sända brus. Både SiS-mottagare och störsändare byggs som speciella fall av *EWRadioSystem*, se Figur 33.

6.1.1. Klassdiagram

Figur 33 visar ett klassdiagram över *EWRadioSystem* och dess beståndsdelar. Varje klass beskrivs även textmässigt på ett översiktligt sätt.



Figur 33 Klassdiagram av *EWRadioSystem*.

EWRadioSystem

Detta är den sammanhållande klass som utgör själva modellen av kommunikationssystemet. *EWRadioSystem* kan konfigureras på olika sätt beroende på hur det scenario som skall köras är definierat.

EWRadioTxRxCommon

Klassen är virtuell och innehåller funktioner och attribut som är gemensamma för sändare och mottagare, t ex antenn, modulation, frekvensband.

EWRadioTxGeneric

Klassen ärver från *EWRadioRxTxCommon* och innehåller därutöver funktioner och attribut som är unika för en generell sändare, t ex uteffekt. Från *EWRadioTxGeneric* ärvs sedan beroende på hur *EWRadioSystem* har definierats vid scenariokapandet till specifika sändare som t ex *EWTxJam* (sändaren i en störsändare) eller *EWTxNormal* (sändaren i en normal kommunikationsradio, t ex Ra180).

EWRadioRxGeneric

Klassen ärver från *EWRadioRxTxCommon* och innehåller därutöver funktioner och attribut som är unika för en generell mottagare, t ex lägsta detekterbara mottagen effekt. Från *EWRadioRxGeneric* ärvs sedan beroende på hur *EWRadioSystem* har definierats vid scenarioskapandet till specifika mottagare som t ex *EWRxJam* (mottagare för störsändare) eller *EWRxNormal* (mottagare i en normal kommunikationsradio, t ex Ra180).

AntennaInfo

Klassen innehåller data om antennen, såsom antennriktning, antennamn och antennhöjd. Själva antenndatat, som t ex strålningsdiagram finns i *Antenna*. Klassen innehåller funktioner för att kunna läsa antennvinst från *Antenna*.

Antenna

Antenndata. Nyttjar en tidigare utvecklad klass som hanterar läsning av antenndata på olika format, t ex i form av strålningsdiagram, formler och vektorer.

EWRFFunctions

Detta är ett funktionspaket som innehåller generella funktioner/algorithmerna som är tillämpliga i olika radiosammanhang, t ex:

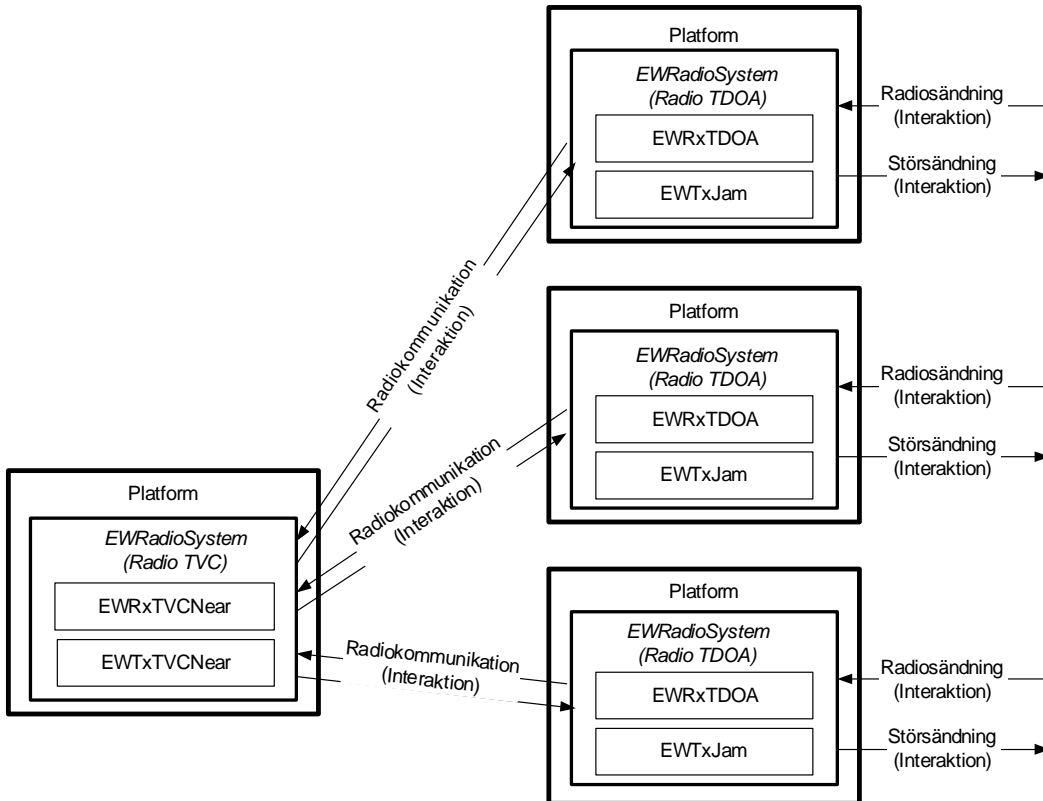
- Funktioner för att beräkna sträckdämpning.
- Funktioner för att beräkna brusnivån i mottagaren.

Mer specifik funktionalitet finns i ett antal klasser som ärver från *EWFunctions*, t ex *EWSISFunctions* och *EWJAMFunctions* som tilhandahåller funktionalitet för att hantera signalspanings- och störsändar specifika funktioner.

6.1.2. Olika typer av SiS-/störsystem

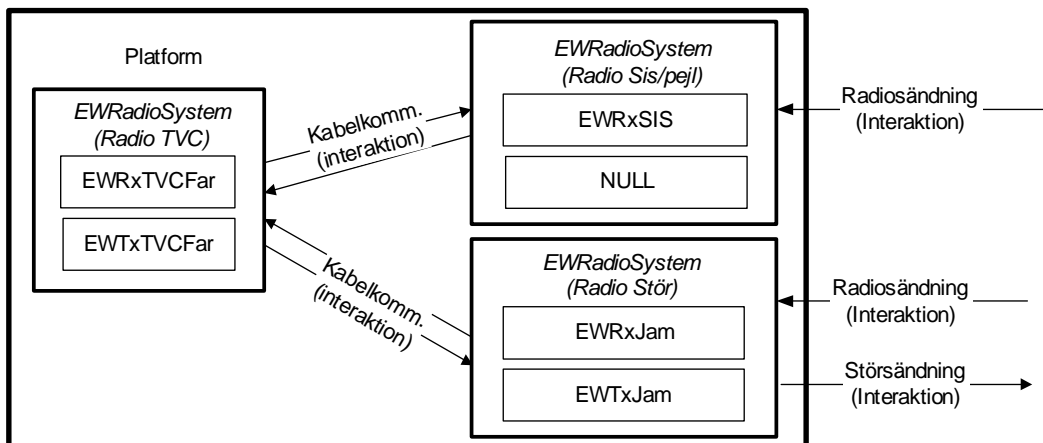
I Figur 34 ges exempel på ett telekrigsystem (TKS) som består av tre underenheter och en huvudenhet. Alla enheter är instanser av klassen *EWRadioSystem*. Funktionen styrs av hur enheterna konfigureras. Vi benämner detta system *närssystem*. I detta system har underenheterna förmågan att både utföra ”pejlning” via tidmätning och att störsända. Huvudenheten (TVC = Televapencentral) kan utföra positionsberäkning genom att sammanställa data från underenheterna. Positioneringen sker med en metod som kallas TDOA (Time Difference of Arrival). Underenheterna utför störsändning med rundstrålande antenn.

Kommunikationen från underenheter till huvudenheten sker via interaktioner. Andra SiS-system har därmed möjlighet att upptäcka trafiken. Kommunikationen med yttrevärlden sker via TVC.



Figur 34. Principskiss av närsystem (benämns senare TKS1), en huvudenhet (EWRxTVCNear + EWTxTVCNear) och tre underenheter (EWRxTDOA + EWTxJam).

Figur 35 visar exempel på en annan typ av system. Systemet består av tre *EWRadioSystem*. Huvudenheten styr systemet och är kommunikationslänk mot omvärlden (TVC), en underenhet hanterar SiS-funktionaliteten och den andra underenheten hanterar störsändningar. Detta system arbetar typiskt på långt avstånd från målet och kallas *fjärrsystem*. SiS består av att signalen upptäcks och riktning bestäms med en viss osäkerhet. Störning sker med riktantenn. Att störsändaren även har en mottagare (EWRxJam) beror på att i vissa störmoder skall störningen kunna triggas av mottagen effekt. Kommunikationen till huvudenheten sker i detta fall via kabel. Kommunikationen med yttervärlden sker via TVC.

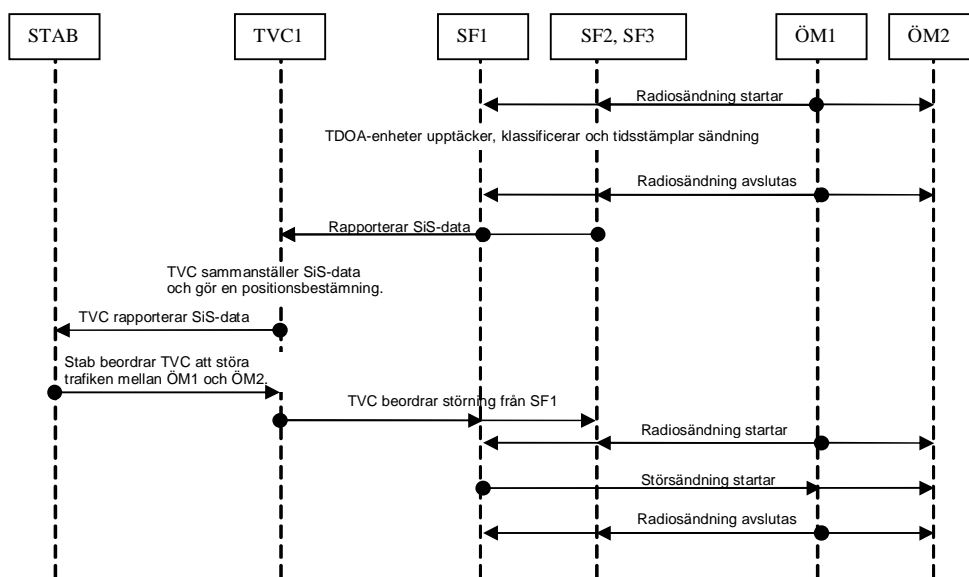


Figur 35 Principskiss av fjärrsystem (benämns senare TKS2), en huvudenhet (EWRxTVCFar + EWTxTVCFar), en SiS/pejl (EWRxSIS) och en störsändare (EWRxJam + EWTxJam).

6.1.3. Sekvensdiagram

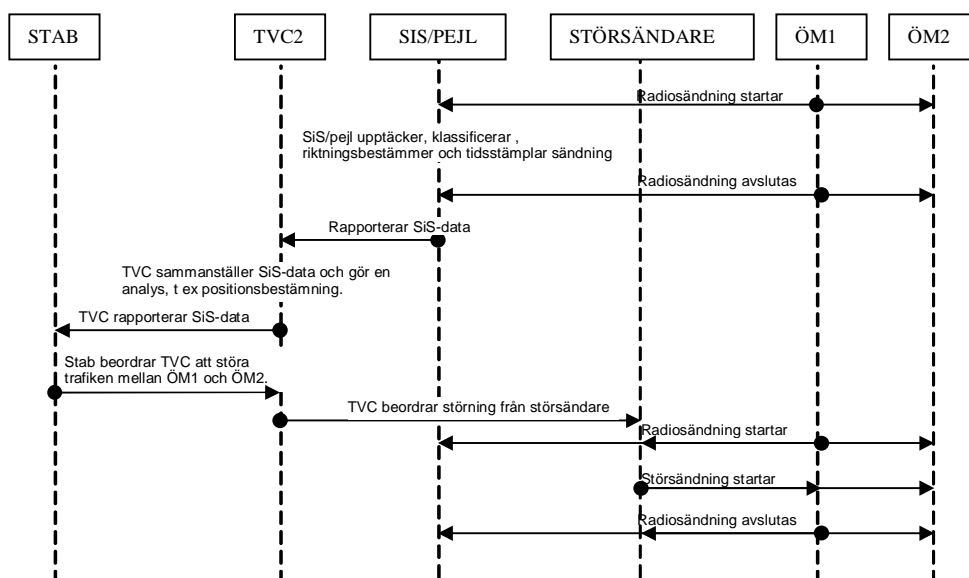
I Figur 36 till Figur 41 visas först fyra principfall för pejling och störning av enskilda sändningar och nätverkstrafik med när- och fjärrsystem. Därefter visas två testscenarion med två SiS/störsystem inblandade. Figureerna visar vilka interaktioner som sänds.

I Figur 36 visas ett typförlopp där en enskild sändning upptäcks och behandlas via ett närsystem. Ingående enheter är en stab, ett närsystem bestående av en huvudenhet (TVC1) och tre underenheter (SF1, SF2 & SF3), samt två fiendliga enheter (ÖM1 & ÖM2, ÖM = Övrig Markplattformar). Enheterna i närsystemet är utgrupperade och kommunikation sker via radio. En fiendlig sändning startar. Sändningen upptäcks, tidsbestäms (för TDOA-analys) och klassificeras av närsystemets underenheter. Dessa rapporterar insamlat SiS-data till den styrande enheten, TVC1. TVC1 sammanställer datat från underenheterna och beräknar sändarens position via TDOA. Resultatet rapporteras till staben som i sin tur beordrar TVC1 att utföra störsändning mot den pågående radiotrafiken. Nästa gång trafiken aktiveras störs det fiendliga sambandet ut.



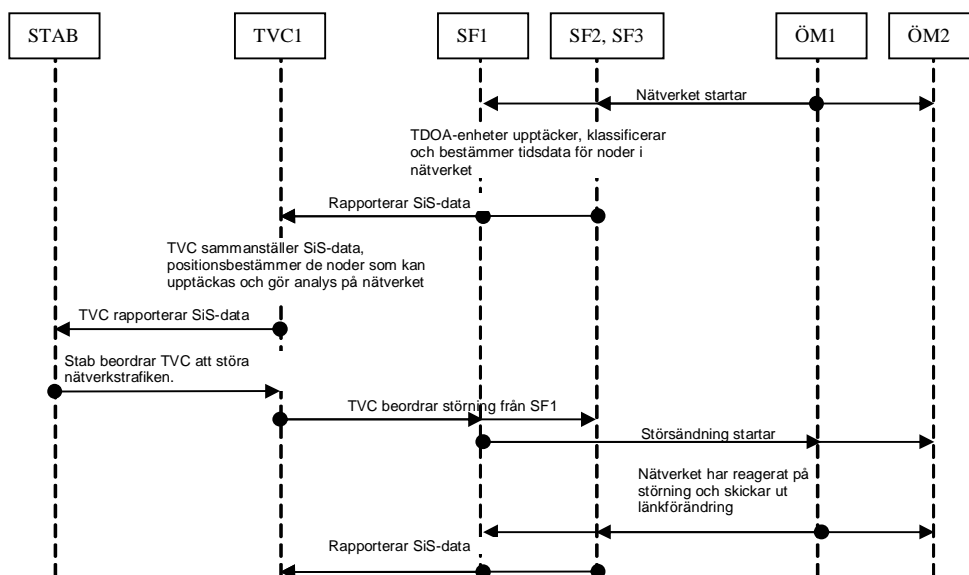
Figur 36 Upptäckt, pejling och störning av enskild sändning i ett närsystem, TVC1.

I Figur 37 visas ett typförlopp där en enskild sändning upptäcks och behandlas via ett fjärrsystem. Ingående enheter är en stab, ett fjärrsystem bestående av en huvudenhet (TVC2) och två underenheter (pejl- och störenhet), samt två fiendliga enheter (ÖM1 & ÖM2). Enheterna i fjärrsystemet är samgrupperade och kommunikation sker via kabel. En fiendlig sändning startar. Sändningen upptäcks, riktningbestäms och klassificeras av pejl-enheten som rapporterar in SiS-data till TVC2. TVC2 rapporterar in SiS-data till staben som i sin tur beordrar störning av den fiendliga signaleringen. När nästa sändning startar, hör störenheten denna och sätter in en störsändning.



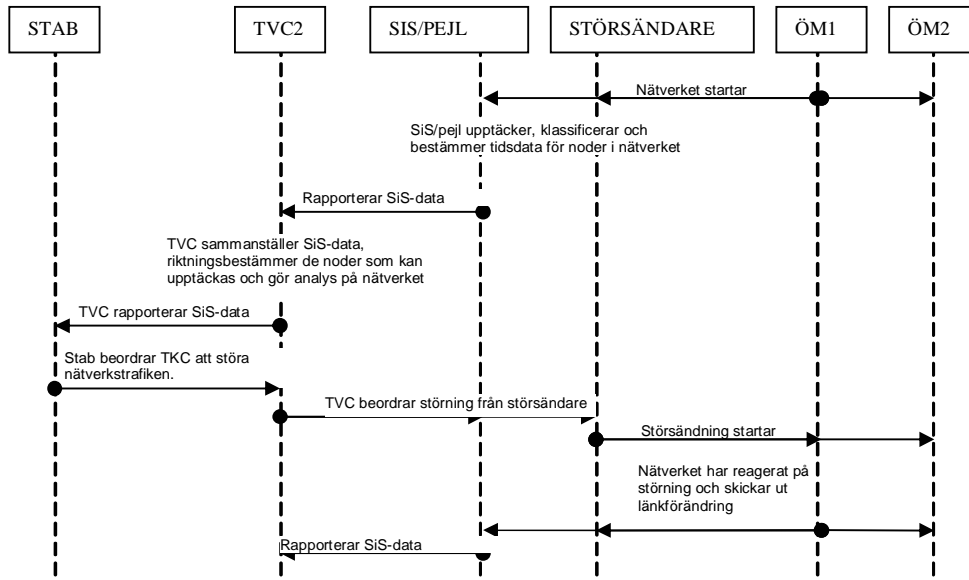
Figur 37 Upptäckt, pejlning och störning av enskild sändning i ett fjärrsystem, TVC2.

I Figur 38 visas ett typförlopp där nätverkssändning upptäcks och behandlas via ett närsystem. Ingående enheter är en stab, ett närsystem bestående av en huvudenhet (TVC1) och tre underenheter (SF1, SF2 & SF3), samt två fiendliga enheter (ÖM1 & ÖM2). Enheterna i närsystemet är utgrupperade och kommunikation sker via radio. De fiendliga enheterna kommunicerar via ett nätverk. Det fiendliga nätverket startas och upptäcks av närsystemets underenheter. Då nätverket startas distribueras en routingtabell inkluderande ingående nätnoder och trafiklast på respektive länk i nätverket. SiS-enheterna plockar ut den delmängd de skulle ha upptäckt baserat på vågutbredningsberäkningar ur detta "facit" och rapporterar detta till huvudenheten. Huvudenheten sammanställer data från underenheterna och gör en analys på nätverket, t ex bestämmer vilken nod som är viktigast för det fiendliga systemet. TVC1 rapporterar data till staben som beordrar störning av nätverket. Nätverket reagerar på störningen genom att ändra sin routingtabell. Den uppspanade nätverksbilden kommer då att ändras. Detta kan av signalspanaren ses som ett kvitto på lyckad störinsats.



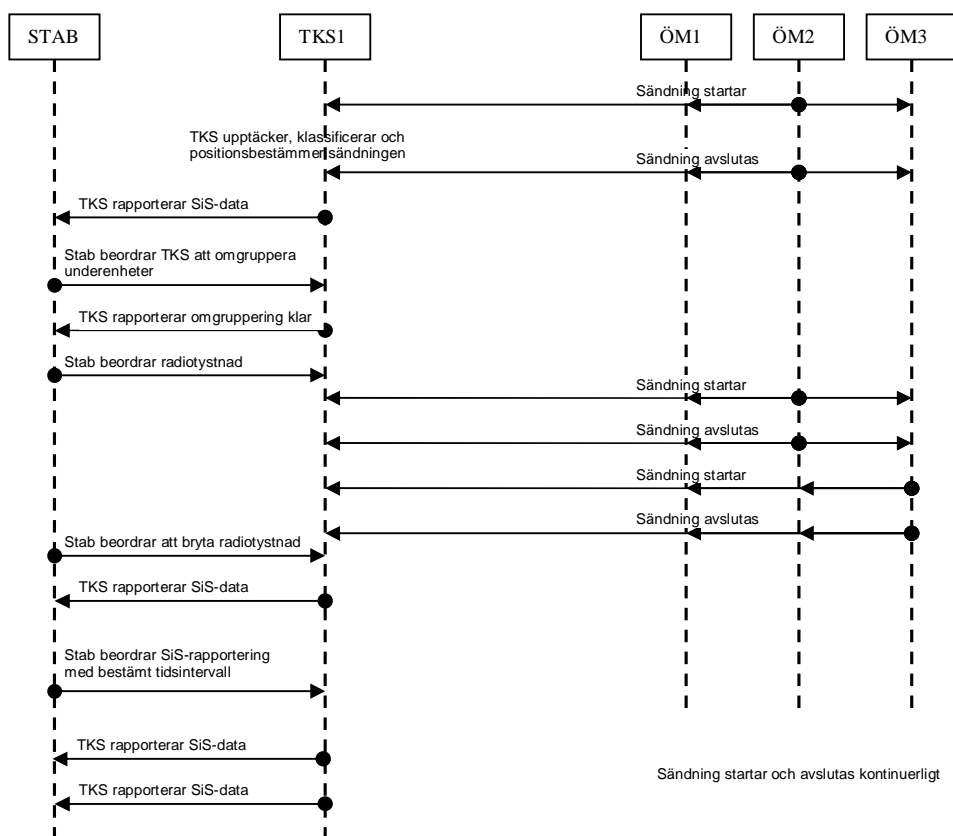
Figur 38 Upptäckt, pejlning och störning av nätverkssändning i ett närsystem, TVC1.

I Figur 39 visas ett typförlopp där nätverkssändning upptäcks och behandlas via ett fjärrsystem. Ingående enheter är en stab, ett fjärrsystem bestående av en huvudenhet (TVC2) och två underenheter (SiS/pejl och störsändare), samt två fiendliga enheter (ÖM1 & ÖM2). Enheterna i fjärrsystemet är samgrupperade och kommunikation sker via kabel. De fiendliga enheterna kommunicerar via ett nätverk. Det fiendliga nätverket startas och upptäcks av SiS/pejl. Nätverksdata behandlas på samma sätt som i det förra exemplet för närsystemet. TVC2 rapporterar data till staben som beordrar störning av nätverket. Nätverket reagerar på störningen genom att ändra sin routingtabell. Den uppspanade nätverksbilden kommer då att ändras. Detta kan av signalspanaren ses som ett kvitto på lyckad störinsats.



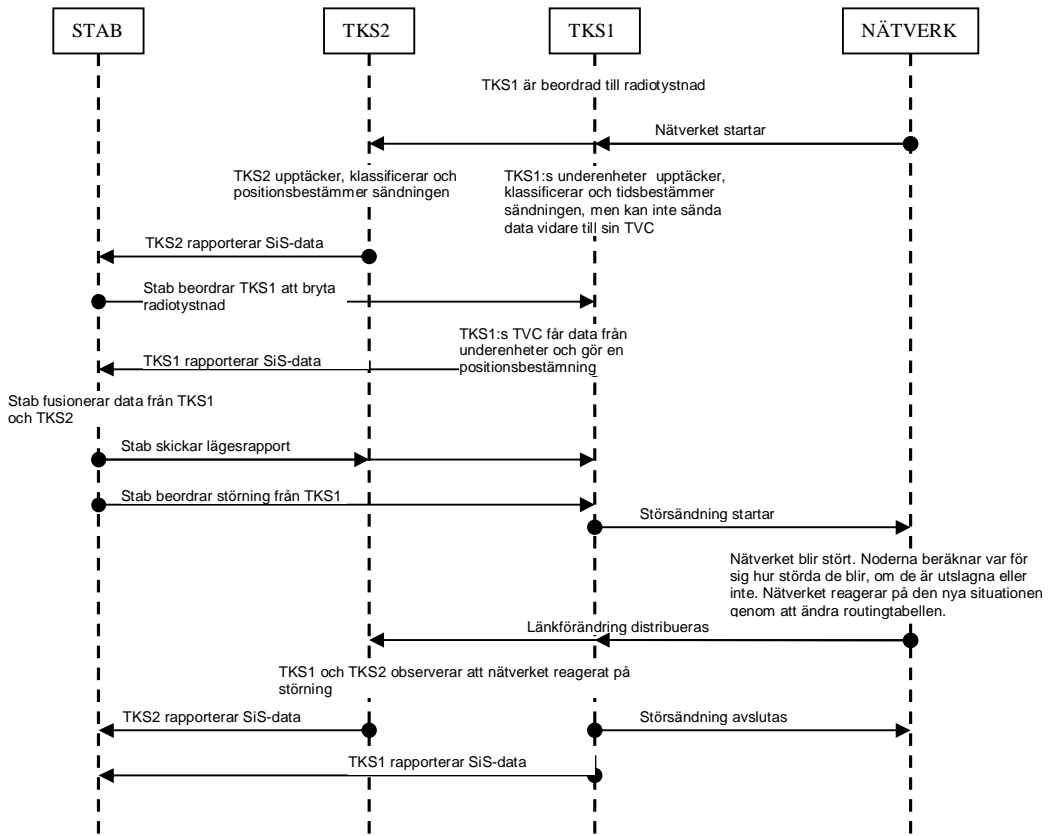
Figur 39 Upptäckt, pejling och störning av nätverkssändning i ett fjärrsystem, TVC2.

Figur 40 visar en testsekvens som testar förmåga m a p SiS mot enskilda sändningar. I detta fall är hela telekrigssystemet inorporerat i TKS1, dvs figuren visar inte den interna kommunikationen i telekrigssystemet. TKS1 är ett närsystem. Det fiendliga nätet börjar sända och upptäcks av TKS1. TKS1 rapporterar till staben som beordrar omgruppering av underenheter för att om möjligt få en noggrannare positionering av de fiendliga sändarna. TKS1 rapporterar till staben att omgruppering är klar, varpå staben beordrar radiotystnad. De fiendliga enheterna utbyter radiotrafik som loggas i TKS1:s underenheter. Då staben beordrar att radiotystnad skall brytas, rapporterar TKS1 in uppsnappat SiS-data. Därpå beordrar staben att TKS1 skall rapportera in SiS-data med givet tidsintervall. De fiendliga enheterna sänder nu kontinuerligt till varandra och TKS1 rapporterar SiS-data till staben med det givna tidsintervallet.



Figur 40 Sekvens som testar förmåga mot enskilda sändningar m.a.p. SiS och störutrustning. TKS1 innefattar ett komplett närsystem med TVC och 3 TDOA-enheter.

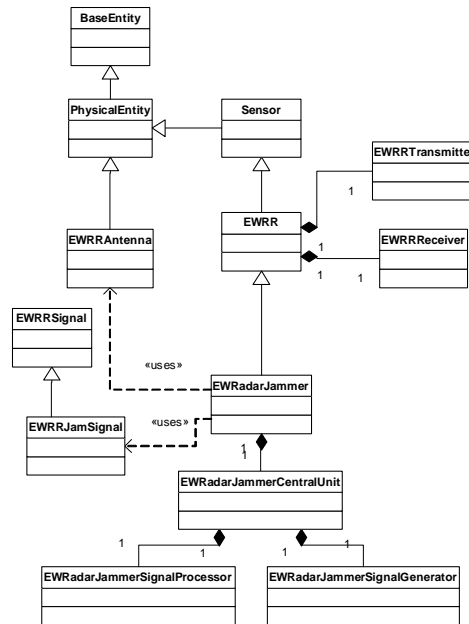
Figur 41 visar även den en testsekvens som testar förmåga m a p SiS och störning mot nätverkssändningar. Här testas möjligheten att samutnyttja data från de två telekrigssystemen TKS1 (ett närsystem) respektive TKS2 (ett fjärrsystem). Figuren visar inte den interna kommunikationen i telekrigssystemen. TKS1 är beordrad radiotystnad. Det fiendliga nätet tänder upp och upptäcks av både TKS1 och TKS2. TKS1:s underenheter kan dock inte rapportera vidare till TVC1, eftersom radiotystnad råder. Däremot kan TKS2 rapportera SiS-data till staben, eftersom den är kabelansluten. Staben finner dessa data intressant och vill ha mer information från TKS1 och beordrar därför att radiotystnad brytes. TVC1 får data från underenheter och gör en positionsbestämning samt rapporterar SiS-data till staben. Staben fusionerar data från TKS1 och TKS2 för att få bästa möjliga lägesinformation från tillgängligt SiS-data. Staben skickar en lägesrapport till TKS1 och TKS2 och beordrar därefter störinsats från TKS1. Nätverket blir stört och reagerar med att ändra routingtabell. Länkförändringar går ut via radio och snappas upp av TKS1 och TKS2 som därmed får bekräftelse på att störningen fått effekt.



Figur 41 Sekvens som testar nätverksfunktionalitet i SiS och störutrustning. Sekvensen testar även fusion av data från två olika telekrigsystem, ett när- och ett fjärrsystem.

6.2. Radarstörare

Strukturen för radarstöraren är väldigt lik den för spaningsradarn. Radarstöraren är dock inte en egen plattform utan själva systemet som sätts på en enhet. Figur 42 visar strukturen för radarstöraren. Radarstöraren använder sig av sändare och mottagare som hanteras av samma radarkanaladministratör som beskrivs ovan. Antennerna har samma funktion som för spaningsradarplattformen. Signalprocessorn analyserar mottagen radarsignal i det fall radarstöraren är beroende av inkommande signaler för att generera utsignaler (t.ex. DRFM-störning). Signalgeneratoren genererar en utsignal från de inställningar som är gjorda samt eventuella data från analysen av mottagna signaler.



Figur 42 Klassdiagram för radarstörare

6.3. Visualisering av nät

Både stab och netScene har möjlighet att visualisera nätverk. Nätverken åskådliggörs med hjälp av grafer. I detta avsnitt beskrivs hur dessa grafer skapas.

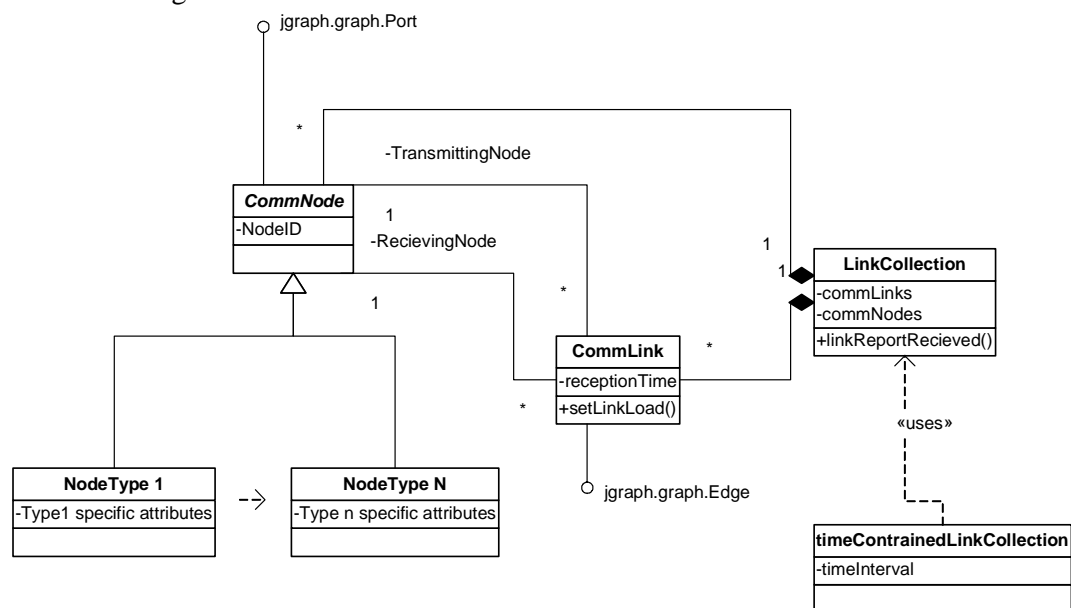
6.3.1. Vägen från simulering till visualisering

Information om nätets uppbyggnad kan komma dels som en guds-vy av verkligt läge och dels som en uppspanad vy. För staben gäller att man för sitt eget nät antas känna den sanna bilden medan man för motståndarens nät bara kan få tillgång till det uppspanade läget. Spaning kan endast ske mot nätets länkar. Routinginformation kan endast fås som det sanna läget. Till skillnad mot den spelade staben kan spelledningen få tillgång till båda sidors sanna nätbild, både routing och länkstatus.

När simuleringen genererar en länkuppdatering (UpdateEther) kommer radiosystemet att generera ett meddelande som vandrar uppåt i nätverksstacken. Det fysiska lagret kommer då att uppdatera länkdatan och senare kommer även routingtabellen att anpassas till de nya länkförhållandena. Dessa båda ändringar skickas som interaktioner via HLA (logpoint) för andra federater att lyssna på. Staben tillåts bara lyssna om informationen gäller egna sidans nät medan spelledningen däremot får tillgång till all information. Möjliga routing vägar visas som linjer mellan noder medan länkarnas status visas i ett stapeldiagram som beskriver länkarnas signal brus förhållande.

6.3.2. Klassdiagram

För visualiseringen av grafen används javapaketen jgraph, vilket är baserat på MVC-paradigmen (Model, View, Controller). Man har en modell som innehåller en abstrakt beskrivning av ens graf, en vy som enbart har till uppgift att visualisera grafen samt en kontrollfunktion som har till uppgift att modifiera grafen.



Figur 43 Modell beskrivande grafen.

6.3.3. CommNode

Abstrakt basclass beskrivande en nod, identifierad av SiS-systemet. Den har en identitet beskriven av dess NodeID. Vilken konkret implementation som används är beroende på typidentifikationen på kommunikationsenheten av SiS-systemet.

6.3.4. NodeType1 – N

Innehåller specifika attribut för just den här typen av nod. Exempelvis frekvenshopsinformation för en frekvenshoppare.

6.3.5. CommLink

En association mellan två noder som beskriver en uppspanad radiolänk mellan dem. Innehåller bland annat information om länkens belastning, samt tidsstämplar för när den är observerad.

6.3.6. LinkCollection

En samling av alla uppspanade länkar som rapporterats till stabsverktyget.

6.3.7. TimeConstrainedLinkCollection

Ett filter över en LinkCollection som bara släpper igenom observerade länkar inom ett visst tidsintervall, används för att enbart visualisera i någon mån färska länkar.

7. Akronymmer och begrepp

Begrepp	Definition
API	Application Interface
C2, CnC	Command and Control, Ledningssystem
CIM	ComNet Internal Message
CNO	Computer Network Operations
CND	Computer Network Defense
CNA	Computer Network Attack
FFM	FOI Fast Message
FOM	Federation Object Model
Gudsvy	Möjlighet att kunna se hela det sanna läget i simuleringen (alla enheters positioner, rörelser, mm)
HIL	Human in the loop
HLA	High Level Architecture
IPS	Intrusion Prevention System
IRST	Infrared search and track
LKS	Ledningskrigföringssimulator
RoRo	Fartyg med möjlighet till Roll On – Roll off, dvs fordon lastas genom att de kör ombord och lastas av genom att de kör av fartyget.
RPR-FOM	Real-time Platform Reference Federation Object Model. En de facto standard Federation Object Model, som ofta används som utgångspunkt vid federationsutvecklingar.
RTI	Run Time Infrastructure
SF	Specialförband
SiS	Signalspaning
SnR	Signal noise ratio
SRR	Spaningsradar
TDMA	Time Division Multiple Access, accessprotokoll som används av t ex GSM
TDOA	Time Difference of Arrival
TK	Telekrig
TKS	Telekrigssystem
TVC	Televapentral
UAV	Unmanned Aerial Vehicle

Bilaga 1 FOM LKS

FOM:en vi använder är MOSART Reference FOM v1.0. Denna består av RPR-FOM v2d17 med tilläggen i MOSART FOM Extension v1.0.

Information om MOSART FOM Extension v1.0:

Object Model Identification Table

Category Information

Name MOSART FOM Extension

Type FOM

Version 1.0

Date 2006-01-23

Purpose Extend the RPR-FOMv2d17 with MOSART FOM additions. This FOM must actually be merged with RPR-FOMv2 before use.

Domain MOSART simulations

POC Name Tobias Horney

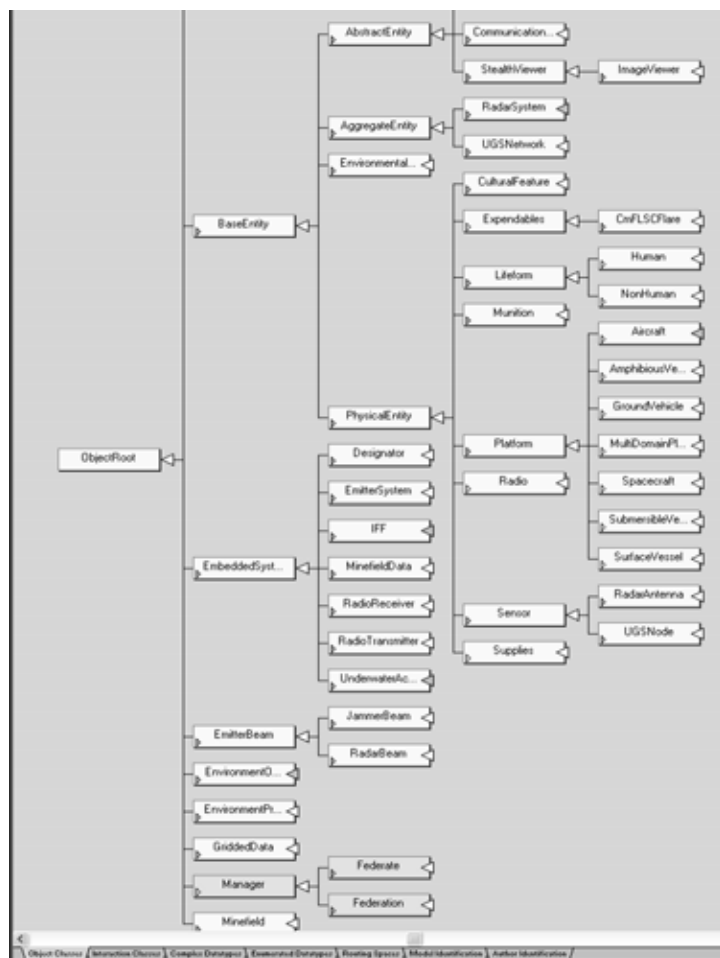
POC Organization FOI

POC Phone +46-13-378554

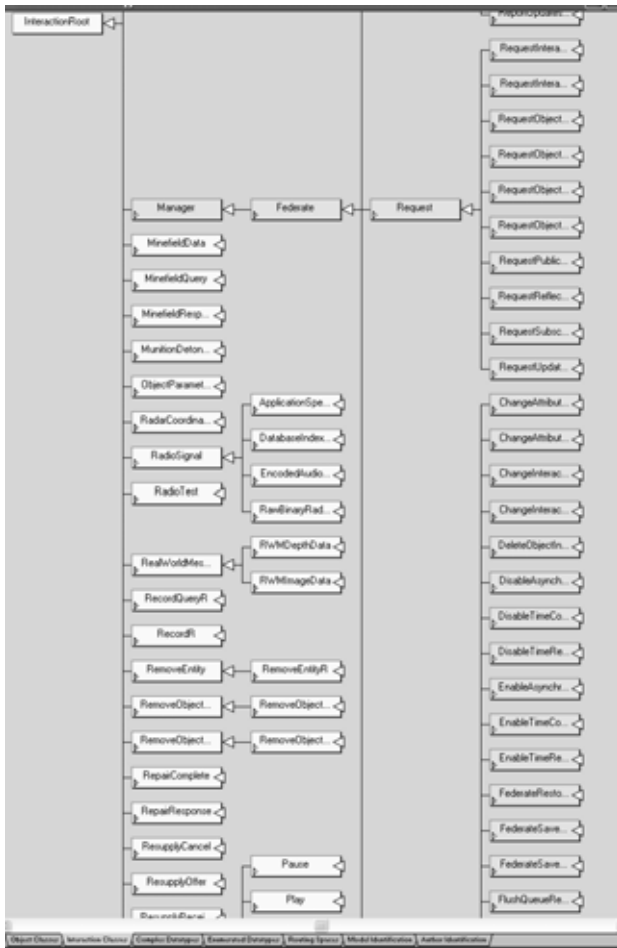
POC Email tobias.horney@foi.se <<mailto:tobias.horney@foi.se>>

References <http://www-int.foi.se/MOSART> (only available within FOI)

Other Created with Visual OMT 1516

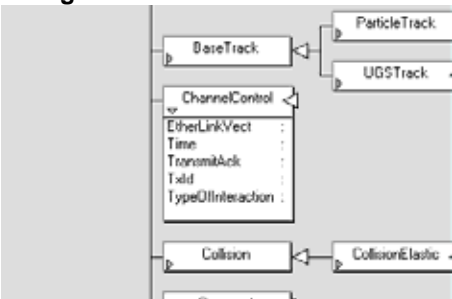


Figur 44 Utdrag från Visual OMT över de mest använda objekten i FOM:ens Object Classes



Figur 45 Utdrag från Visual OMT över några av klasserna i FOM:ens Interaction Classes.

Tillagda interaktioner: Utöver RPR-FOM v2d17



ChannelControl



RadioTest

StartRadioTx

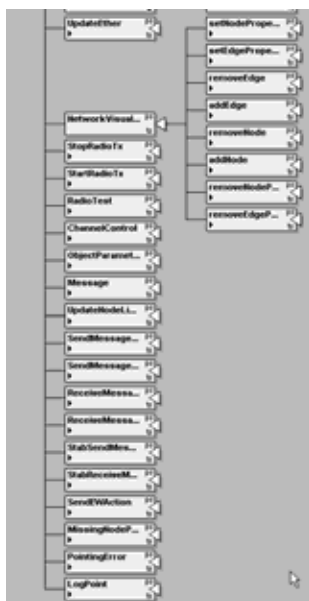
Används för att tala om att en radiosändning startat.

StopRadioTx

Används för att tala om att en radiosändning avslutats.

UpdateEther

Används för att meddela länkförändringar.



Figur 46 Utdrag från Visual OMT över några av klasserna i FOM:ens Interaction Classes som används.

Bilaga 2 Publish Subscribe för federater I LKS V2

”Interaction Classes” är HLA objekt i FOM:en som beskriver dynamiska relationer mellan fysiska objekt som t ex ett radio meddelande.

P= Publish, S= Subscribe.

Interaction Classes	LKS Federat							
	Fartyg	Flygplan	Radar -plattform federat Federat	IRST - plattform	SIS - plattform	Övriga mark plattform	Stab blå och röd	Nätverk
remoteCreateObject	S	S	S	S	S	S	S	S
remoteCreateObjectResult	P	P	P	P	P	P	P	P
Collision	PS	PS	PS	PS	PS	PS	PS	
BaseTrack	PS	PS	PS					
RadioTest	PS	PS	PS	PS	PS	PS	PS	PS
ChannelControl	PS	PS	PS	PS	PS	PS	PS	PS
ObjectParameterUpdate	PS	PS	PS	PS	PS	PS	PS	PS
Message	PS	PS	PS	PS	PS	PS	PS	PS
StartRadioTx								PS
StopRadioTx								PS
UpdateEther								PS
SendMessageToNode	P	P	P	P	P	P	P	S
SendMessageFromNode	S	S	S	S	S	S	S	P
ChannelControl								PS
MissingNodeProperties								PS
StabSendMessage	P	P	P	P	P	P	P	S

”Object Classes” är HLA objekt i FOM:en som beskriver fysiska objekt som t ex flygplan.

Object Classes	LKS Federat							
	Fartyg	Flygplan	Radar - plattform	IRST - plattform	SIS - plattform	Övriga mark	Stab blå och röd	Nätverk
Platform							P	P
PhysicalEntity	P		S					
BaseEntity	S	S	S	S	S	S	S	S
EmbeddedSystem	S	S	S	S	S	S	S	S
Groundvehicle			P	P	P	P	P	
Munition	P	P	P	P	P	P	P	
RadioNormal	PS	PS	PS	PS	PS	PS	PS	PS
Aircraft		P						
NetNode	S	S	S	S	S	S	S	PS
SurfaceVessel	P							
Human						P		
SurveillanceRadar	PS	PS	PS	S	S	S	S	
RadarWarner	PS	PS	PS	S	S	S	S	
RadarNoiseJammer	PS	PS	PS	S	S	S	S	
RadarDRFMJammer	PS	PS	PS	S	S	S	S	
RadarAntenna	PS	PS	PS	PS	PS	PS	PS	

Federaten Logger Prenumererar (Subscribe) på alla objekt under simuleringen och Publicerar dem vid återuppspelning (Publish).

Bilaga 3 Publish and Subscribe för LKS objekt

Simulerade objekt i LKS V2 och deras HLA objekt i FOM:en

Med simulerade objekt avses objekt som kommer att simuleras i LKS-federationen version 2.

- Fordon
 - Subscribe:*
 - remoteCreateObject*
 - Collision*
 - BaseEntity*
 - Publish:*
 - remoteCreateObjectResult*
 - GroundVehicle*
 - BaseEntity*
- Flyg
 - Subscribe:*
 - remoteCreateObject*
 - Collision*
 - BaseEntity*
 - Publish:*
 - RemoteCreateObjectResult*
 - AirCraft*
- Fartyg
 - Subscribe:*
 - remoteCreateObject*
 - Collision*
 - BaseEntity*
 - Publish:*
 - RemoteCreateObjectResult*
 - SurfaceVessel*
- Människa
 - Subscribe:*
 - remoteCreateObject*
 - Collision*
 - BaseEntity*
 - Publish:*
 - RemoteCreateObjectResult*
 - Human*
- Antenn - Radarantenn som simulerar antenndiagram
 - Subscribe:*
 - remoteCreateObject*
 - Publish:*
 - RemoteCreateObjectResult*
 - BaseEntity*
- Radar – Parametersatt radarsystem
 - Subscribe:*
 - remoteCreateObject*
 - Track*
 - PhysicalEntity*
 - Publish:*
 - RemoteCreateObjectResult*
 - Collision*
 - Track*
- IRST – Parametersatt IRST-system (*Infrared Search and Track*)
 - Publish:*

*BaseEntity**RemoteCreateObjectResult**Subscribe:**remoteCreateObject**PhysicalEntity*

- Radio – Kommunikation via radio. SiS/Pejl/Stör och NätNod är specialfall av Radio

*Subscribe:**RemoteCreateObject**PhysicalEntity**StartRadioTx**StopRadioTx**UpdateEther**Message**Publish:**RemoteCreateObjectResult**StartRadioTx**StopRadioTx**UpdateEther**Message*

- Radio SiS/Pejl – Pejl och signalspaning mot kommunikation via radio, är ett specialfall av Radio.

*Subscribe:**RemoteCreateObject**PhysicalEntity**StartRadioTx**StopRadioTx**UpdateEther**Message**Publish:**RemoteCreateObjectResult**StartRadioTx**StopRadioTx**UpdateEther**Message*

- Radio TDOA – Pejl-, signalspaning och störning mot kommunikation via radio, är ett specialfall av Radio.

*Subscribe:**RemoteCreateObject**PhysicalEntity**StartRadioTx**StopRadioTx**UpdateEther**Message**Publish:**RemoteCreateObjectResult**StartRadioTx**StopRadioTx**UpdateEther**Message*

- Radio Stör – Radio störsändning mot kommunikation via radio, är ett specialfall av Radio.

*Subscribe:**RemoteCreateObject*

PhysicalEntity
StartRadioTx
StopRadioTx
UpdateEther
Message

Publish:

RemoteCreateObjectResult
StartRadioTx
StopRadioTx
UpdateEther
Message

- Radio TVC – *Kontroll- och sammanställningsenhet i ett telekrigsystem, är ett specialfall av Radio.*

Subscribe:

RemoteCreateObject
PhysicalEntity
StartRadioTx
StopRadioTx
UpdateEther
Message

Publish:

RemoteCreateObjectResult
StartRadioTx
StopRadioTx
UpdateEther
Message

- Radio Nod – *interface i NätNod, är ett specialfall av Radio.*

Subscribe:

RemoteCreateObject
PhysicalEntity
StartRadioTx
StopRadioTx
UpdateEther

Publish:

RemoteCreateObjectResult
StartRadioTx
StopRadioTx
UpdateEther

- NätNod – *Kommunicerande nod i nätverket. Nätnoden är en del "ovanpå" Radio. Nätnoden innehåller all funktionalitet för att beräkna nätverkskopplingar.*

Subscribe:

remoteCreateObject
Message

Publish:

RemoteCreateObjectResult
Message

- Radarstör – *Parametersatt radarstörare*

Subscribe:

remoteCreateObject
Radar
PhysicalEntity

Publish:

RemoteCreateObjectResult

- PhysicalEntity*
- Radarvarnare – *Parametersatt radarvarnare*
 - Subscribe:*
 - remoteCreateObject*
 - radar*
 - PhysicalEntity*
 - Publish:*
 - RemoteCreateObjectResult*
 - PhysicalEntity*
- Raketartilleri – *Urustning för att kunna avfyra en raket*
 - Publish*
 - RemoteCreateObjectResult*
 - Munition*
 - Subscribe*
 - remoteCreateObject*
 - Platform*

Referenser

- [FS] FMV (2004) Förstudie av Demonstrator Ledningskrigföringssimulator. Delrapport. FMV beteckning: VO FoT 23 321:16418/2004
- [FEDEP1v2] Ledningskrigföringssimulator version 2 (LKS v2) FEDEP 1, Måldefinition, FMV beteckning: VO FoT 23 321:57061/2006
- [FEDEP2v2] Ledningskrigföringssimulator version 2 (LKS v2) FEDEP 2, Konceptuell analys, FMV beteckning: VO FoT 23 321:57062/2006.
- [DESIGNv1] FOI (2006). *Design LKS Version 1*. FMV beteckning: VO FoT 23321:22650/2006
- [AGR] Tobias Horney m fl, MOSART Federation Manager Documentation, FOI Memo 1682, 2004-06-21
- [MOSART] Tobias Horney m fl, Slutrapport MOSART, FOI-R--1814—SE, 2006
- [EW] Lars Tydén m fl, Slutrapport Duellsimulering Telekrig, FOI-R--1724—SE, 2006
- [RPR-FOM] www.sisostds.org, SISO-STD-001.1-1999: Real-time Platform Reference Federation Object Model (RPR FOM 1.0)

