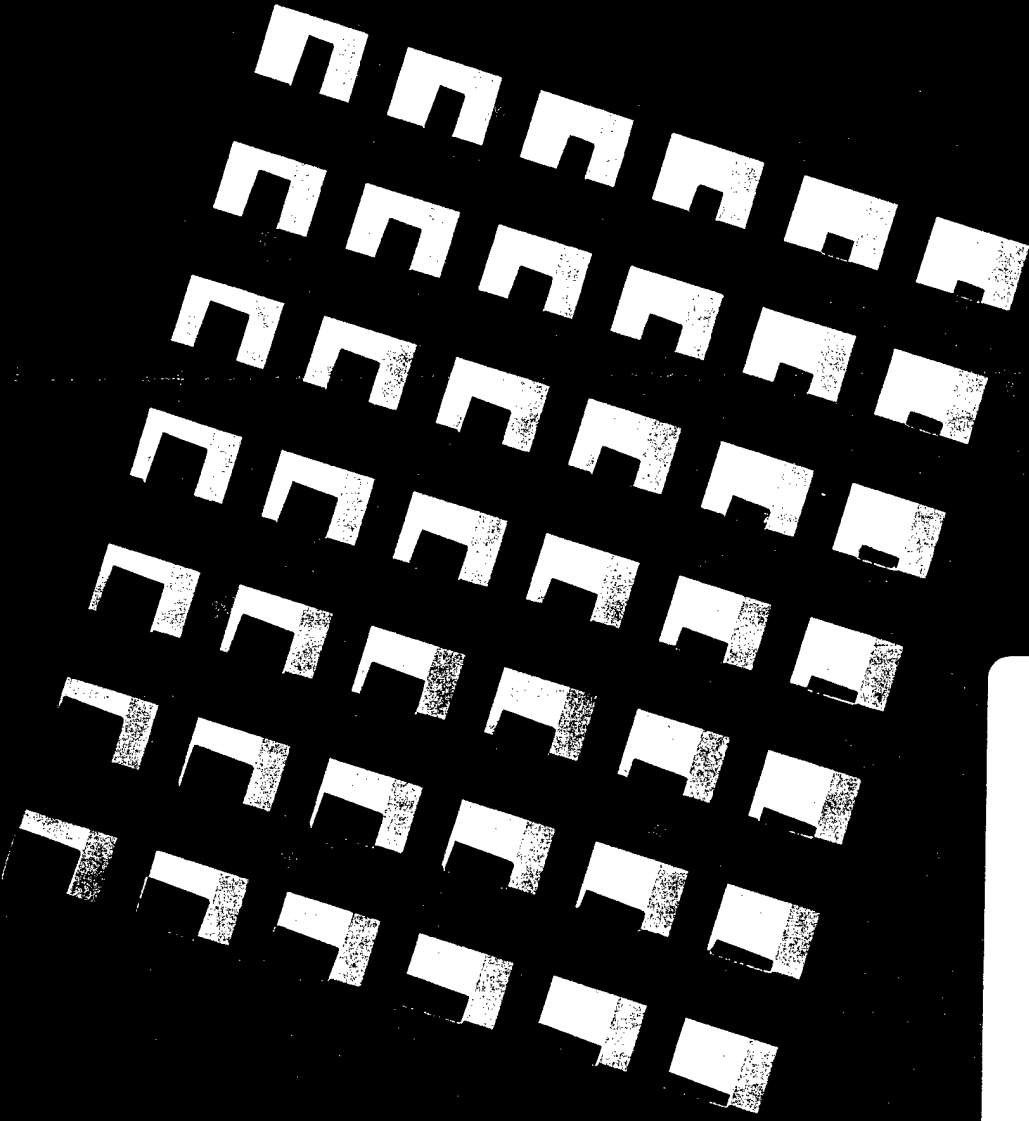


TNO-rapport
FEL-98-A014

Gedistribueerde Interactieve Simulatie: Van DIS naar HLA

TNO Fysisch en Elektronisch
Laboratorium



19980626 029



TNO-rapport
FEL-98-A014

Gedistribueerde Interactieve Simulatie: Van DIS naar HLA

TNO Fysisch en Elektronisch
Laboratorium

Oude Waalsdorperweg 63
Postbus 96864
2509 JG 's-Gravenhage

Telefoon 070 374 00 00
Fax 070 328 09 61

Datum
februari 1998

Auteur(s)
Drs. R.J.D. Elias
Ir. W.G. de Jong

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

Rubricering
Vastgesteld door : Ing. F.J. Remmerswaal
Vastgesteld d.d. : 6 februari 1998

Titel : Ongerubriceerd
Managementuittreksel : Ongerubriceerd
Samenvatting : Ongerubriceerd
Rapporttekst : Ongerubriceerd
Bijlage A : Ongerubriceerd

Alle rechten voorbehouden.
Niets uit deze uitgave mag worden
vermenigvuldigd en/of openbaar gemaakt
door middel van druk, fotokopie, microfilm
of op welke andere wijze dan ook, zonder
voorafgaande toestemming van TNO.

Indien dit rapport in opdracht werd
uitgebracht, wordt voor de rechten en
verplichtingen van opdrachtgever en
opdrachtnemer verwezen naar de
Algemene Voorwaarden voor onderzoeks-
opdrachten aan TNO, dan wel de
betreffende terzake tussen partijen
gesloten overeenkomst.
Het ter inzage geven van het TNO-rapport
aan direct belanghebbenden is toegestaan.

Exemplaar nr. : 12
Oplage : 42
Aantal pagina's : 68 (incl. bijlage,
excl. RDP & distributielijst)
Aantal bijlagen : 1

© 1998 TNO

DTIC QUALITY INSPECTED 1

TNO Fysisch en Elektronisch Laboratorium is onderdeel
van TNO Defensieonderzoek waartoe verder behoren:

TNO Prins Maurits Laboratorium
TNO Technische Menskunde



Nederlandse Organisatie voor toegepast-
natuurwetenschappelijk onderzoek TNO

Managementuittreksel

Titel : Gedistribueerde Interactieve Simulatie:
Van DIS naar HLA
Auteur(s) : Drs. R.J.D. Elias, Ir. W.G. de Jong
Datum : februari 1998
Opdrachtnr. : A95KL841
IWP-nr. : 758
Rapportnr. : FEL-98-A014

In 1995 is door het US Defense Modeling and Simulation Office (DMSO) het startschot gegeven voor de ontwikkeling van een generieke architectuur, de *High Level Architecture* (HLA), met als doel het bevorderen van interoperabiliteit tussen simulaties en van hergebruik van simulaties. Deze architectuur zal de basis gaan vormen voor alle simulatie activiteiten binnen de US DoD. De DIS IEEE 1278 standaarden, die de afgelopen jaren als basis hebben gediend voor gedistribueerde interactieve simulatie, zullen gaan wegebben en worden vervangen door de nieuwe HLA standaarden. De HLA vloeit voort uit de ervaringen die opgedaan zijn met bestaande standaarden als DIS en ALSP. Voor toekomstige simulatie systemen is het zaak om de blik op HLA te richten, o.a. om internationale interoperabiliteit op lange termijn te waarborgen.

De *Run-Time Infrastructure* (RTI) is de basis communicatie laag van de HLA die uitwisseling van informatie tussen simulatie applicaties verzorgt op basis van een gestandaardiseerde interface specificatie. Het data distributie mechanisme van de RTI biedt de gebruiker diverse vormen van informatie filtering om de communicatie van irrelevante data te voorkomen en de netwerk performance te maximaliseren. Een aantal implementaties van de RTI is reeds gerealiseerd, waaronder de DMSO RTI versie 1.0 die vrij verkrijgbaar is. Zowel qua functionaliteit als performance is deze versie een behoorlijk volwassen RTI implementatie, maar aangezien de source code niet vrij gegeven is en slechts een beperkt aantal computer platformen ondersteund worden, heeft het gebruik van de DMSO RTI zijn beperkingen.

HLA applicaties wisselen informatie uit in de vorm van *objects* en *interactions*. Een object representeert b.v. een voorwerp in de virtuele omgeving. Een interactie is een unieke, tijdgebonden gebeurtenis. Aangezien de HLA standaarden geen uitspraak doen over de semantiek van de uit te wisselen informatie, wordt deze vastgelegd in een *Federation Object Model* (FOM). De FOM is een 'contract' tussen de deelnemende applicaties met een specificatie van de uit te wisselen informatie. Omdat federates binnen een toepassingsdomein vaak dezelfde soort informatie uitwisselen, is het concept *Reference FOM* geïntroduceerd. Een *Reference FOM* beschrijft een generieke data structuur voor een bepaald applicatie-

domein. De FOM ontwikkelaar kan de Reference FOM verfijnen voor zijn specifieke eisen. De *Real-time Platform-level Reference Federation Object Model* (RPR-FOM) is een voorgestelde data structuur die de inhoud van de reeds bestaande DIS PDU's beschrijft in de vorm van een object en interaction class hiërarchie. De RPR-FOM zal een belangrijk hulpmiddel zijn bij de migratie van DIS naar HLA voor real-time, human-in-the-loop simulaties van platform entiteiten (zoals tanks en vliegtuigen).

Het *Advanced Simulation Framework* (ASF) vormt de basis voor de ontwikkeling van gedistribueerde simulatie applicaties binnen de TNO-FEL *Electronic Battlespace Facility* (EBF). De EBF biedt een infrastructuur voor onderzoek naar en toepassing van gedistribueerde interactieve simulatie technologieën. Het ASF schermt de applicatie ontwikkelaar af van standaarden zoals DIS en HLA d.m.v. een generiek interface. Het ASF vormt als het ware een laag ('*middleware layer*') tussen de simulatie applicatie en de onderliggende gedistribueerde simulatie standaarden. Deze opzet vergemakkelijkt migraties naar nieuwe standaarden zonder ingrijpende aanpassingen van de simulatie applicatie zelf.

Dit rapport is een tussenrapportage en beschrijft de recente ontwikkelingen op het gebied van de HLA, RTI en RPR-FOM. Tevens beschrijven we de integratie van deze standaarden en technologieën in het door TNO-FEL ontwikkelde ASF. De belangrijkste conclusie van het onderzoek is dat de HLA ontwikkeling op volle toeren draait en in de US als basis voor alle simulatie- en modelleer-activiteiten wordt geaccepteerd. Het onderzoek naar gedistribueerde simulatie zal worden voortgezet met o.a. een performance analyse van de RTI en de migratie van een DIS simulator prototype naar HLA.

Inhoud

1.	Introductie	5
2.	High Level Architecture	6
2.1	Inleiding	6
2.2	HLA Rules	7
2.3	HLA Object Model Templates	7
2.4	HLA Interface Specification	8
2.5	Data Distribution Management	9
3.	Run-Time Infrastructure	11
3.1	Inleiding	11
3.2	Implementaties	11
3.3	RTI 1.0 Architectuur	12
3.4	Multicasting	16
4.	Real-time Platform-level Reference FOM	17
4.1	Inleiding	17
4.2	RPR-FOM Object Class Structure	18
4.3	RPR-FOM Interaction Class Structure	20
4.4	RPR-FOM Ontwikkelingen	21
5.	Advanced Simulation Framework	22
5.1	Electronic Battlespace Facility	22
5.2	ASF Software Architectuur	22
5.3	DIS2 ObjectServer	23
5.4	HLA ObjectServer	24
5.5	Helicopter Simulatie Applicatie	25
6.	Conclusies	27
7.	Afkorting	30
8.	Referenties	31
9.	Ondertekening	32
	Bijlage	
A	RPR-FOM 0.1.7	

1. **Introductie**

Dit rapport is een (tussen)rapportage van het project 'Kennisonderhoud DIS/HLA' (A95KL841) en een vervolg op eerdere rapportages, presentaties en demonstraties over het onderwerp *Advanced Distributed Simulation Technology*.

Dit rapport is gericht op de ontwikkelingen van de *High Level Architecture* (HLA), met name op het technische vlak, gedurende het afgelopen jaar. Het HLA initiatief is medio 1995 gestart door het US Defense Modeling and Simulation Office (DMSO) en behelst de ontwikkeling van een generieke architectuur voor alle modelleer- en simulatie-activiteiten. HLA vloeit voort uit de ervaringen opgedaan met bestaande standaarden als *DIS* en *ALSP*. In de *Simulation Interoperability Workshop* (SIW) in Orlando, Florida worden twee keer per jaar de bevindingen en nieuwe ontwikkelingen rond HLA gepresenteerd. Deze workshop, de opvolger van de DIS workshop, wordt gehouden onder auspiciën van de *Simulation Interoperability Standards Organization* (SISO).

Hoofdstuk 2 geeft een beknopte beschrijving van de HLA standaarden en zijn componenten. In hoofdstuk 3 presenteren we de huidige stand van zaken rond de *Run-Time Infrastructure* (RTI), het 'hart' van HLA dat de communicatie tussen simulatie applicaties tot stand brengt. In hoofdstuk 4 schetsen we de activiteiten rond de migratie van de DIS standaard naar HLA. Het *Advanced Simulation Framework* (ASF) is een TNO-FEL software architectuur voor de ontwikkeling van gedistribueerde simulatie applicaties. De gerealiseerde koppeling tussen het ASF en de HLA RTI wordt beschreven in hoofdstuk 5. In hoofdstuk 6 trekken we een aantal conclusies en Bijlage A bevat de volledige *Real-time Platform Reference FOM* (RPR-FOM), versie 0.1.7.

2. High Level Architecture

2.1 Inleiding

Het doel van de HLA is tweeledig: *interoperabiliteit* tussen simulatie modellen en *hergebruik* van simulatie modellen.

De HLA standaard omvat drie onderdelen: de *HLA Rules*, de *HLA Interface Specification* en de *Object Model Templates*. Dit rapport zal deze onderdelen slechts beknopt beschrijven aangezien dit in de vorige rapportage ([FEL96A273]) reeds uitvoerig gedaan is (m.u.v. data distribution management). Alvorens verder in te gaan op de onderdelen van de HLA standaard, omschrijven we eerst een aantal termen:

Federate

HLA-compliant applicatie b.v. een simulator, data logger, semi-automatic forces generator, simulation management tool of presentatie gereedschap zoals 3D-Stealth of Audio server.

Federation

Verzameling van participerende *federates* die een applicatie-domein vertegenwoordigen. Deze federates moeten zich houden aan de Federation Object Model (FOM) die voor deze federation is opgesteld.

Federation Object Model (FOM)

Contract tussen federates die alle toegestane informatie uitwisseling tussen de federates vastlegt. De FOM is samengesteld uit delen van de SOM's van de participerende federates.

Simulation Object Model (SOM)

De SOM specificeert de 'capabilities' en 'requirements' van een individuele federate, d.w.z. welke data zal de federate genereren en welke data heeft de federate nodig.

Object

Entiteit met unieke identificatie en eigen status binnen de federation. De status van een object wordt bepaald door de huidige waarden van zijn attributes. Objecten worden formeel vastgelegd in de SOM en FOM.

Object Model Template (OMT)

Gestandaardiseerde formaten, in de vorm van tabellen, voor het beschrijven van de SOM en de FOM.

Attribute

Karakteristieke eigenschap van een object, b.v. positie of snelheid. De attribute waarden worden door de federates uitgewisseld conform de FOM.

Interaction

Unieke, tijdgebonden gebeurtenis (event) in de federation die uitgewisseld wordt door de federates conform de FOM. De interaction wordt beschreven aan de hand van een aantal parameter waarden.

Federation execution

Het verloop van de federation oefening. Zolang minstens 1 federate actief is, bestaat de federation execution.

2.2 HLA Rules

De *Rules* vormen een verzameling van technische principes en afspraken waaraan HLA deelnemers zich moeten houden om *HLA-compliant* te zijn. De Rules bestaan uit 5 *federation rules*, regels die slaan op de verzameling van applicaties die tot een bepaald simulatie applicatie-domein behoren, en 5 *federate rules*, regels die op de simulatie applicaties zelf slaan. Twee voorbeelden van Rules zijn:

Federation Rule 1:

Elke federation moet een Federation Object Model (FOM) hebben, conform de gestandaardiseerde formaten van de Object Model Templates (OMT).

Federate Rule 1:

Elke federate moet een Simulation Object Model (SOM) hebben, conform de gestandaardiseerde formaten van de Object Model Templates (OMT).

2.3 HLA Object Model Templates

De Object Model Templates zijn gestandaardiseerde formaten die gebruikt worden om de 'capabilities' en 'requirements' van alle deelnemende simulatie modellen te specificeren. De volgende vijf templates worden gebruikt om de SOM en de FOM op te stellen:

- Object Class Structure Table
Beschrijving van de object class-subclass relaties.
- Interaction Class Structure Table
Beschrijving van de interaction class-subclass relaties.

- Attribute Table
Specificatie van alle object attributen.
- Parameter Table
Specificatie van alle interaction parameters.
- FOM/SOM Lexicon
Definitie van alle termen in bovengenoemde tabellen.

Bijlage A bevat een formele beschrijving van de RPR-FOM 0.1.7 op basis van de HLA Object Model Templates.

2.4 HLA Interface Specification

De Interface Specification is een formele, functionele beschrijving van het interface tussen enerzijds de HLA applicatie en anderzijds de Run-Time Infrastructure (RTI). RTI versie 1.0 van DMSO ondersteunt Interface Specification 1.1. Deze specificatie biedt functies voor de volgende RTI *services*:

1. Federation Management
Functies voor het creëren, verwijderen, onderbreken en hervatten van federation executions.
2. Declaration Management (DM)
Een HLA federate deelt de federation mede welke *type* object- en interaction-informatie hij tijdens de oefening gaat produceren d.m.v. *publication*. Tevens abonneert elke federate zich op informatie die voor hem relevant is d.m.v. *subscription*. Zo zal de RTI alleen relevante informatie doorgeven aan de federate en irrelevante informatie negeren. Zowel publications als subscriptions kunnen tijdens de oefening dynamisch gewijzigd worden. Het doel van DM is om de benodigde communicatie bandbreedte te beperken, door alleen relevante data te distribueren.
3. Object Management
Functies voor het uitwisselen van object- en interaction data. De RTI bewaart object attributes en interaction parameters niet intern, maar fungeert als communicatie mechanisme tussen de HLA federates. De RTI bewaart wel sommige object informatie voor de interne boekhouding, zoals object ID's en ownership data.
4. Ownership Management
Elk object wordt in principe beheerd door de federate die het object geïnstanceerd heeft. Het is echter mogelijk de 'ownership' van object attributen over te dragen aan andere federates, of 'ownership' van bepaalde attributen aan te

vragen. Een praktisch voorbeeld hiervan is een tanksimulator die uitgerust wordt met een nieuw, geavanceerder bewegingsmodel. Dit bewegingsmodel is een aparte federate die eigenaar wordt van de positie-attribute van het tank object en zo de actuele positie kan specificeren.

5. Time Management

Aangezien HLA een breed scala van toepassingsgebieden moet ondersteunen, zowel event-driven (b.v. ALSP) als time-driven (b.v. DIS), biedt HLA meerdere time management mechanismen. Deze zijn gebaseerd op twee orthogonale factoren: "*time regulated*" en "*time constrained*". Time regulated wil zeggen dat *time advances* in de federation centraal gecoördineerd zijn. De federate bepaalt dus mede wat de simulatie tijd van de hele federation is. Time constrained, ook wel *paced* genoemd, wil zeggen dat de simulatie tijd van een federate gerelateerd is aan de muurklok (b.v. voor human-in-the-loop simulatie), terwijl unconstrained federates zelf hun voortgang bepalen (b.v. as-fast-as-possible). In het geval van een time constrained federate zal de RTI alleen berichten aan de federate doorgeven met een timestamp kleiner (dus ouder) dan de federate time. Time regulating zegt dus iets over de voortgang van de *federation time*, terwijl time constrained iets zegt over de voortgang van de *federate time*.

6. Data Distribution Management (DDM)

Behalve filtering op informatie *type* zoals beschreven in Declaration Management, biedt de Interface Specification ook de mogelijkheid te filteren op *object attribuut waarden*. Aangezien DDM niet in Interface Specification 1.0 zat en niet in de vorige rapportage is beschreven, gaan we er hier wat dieper op in.

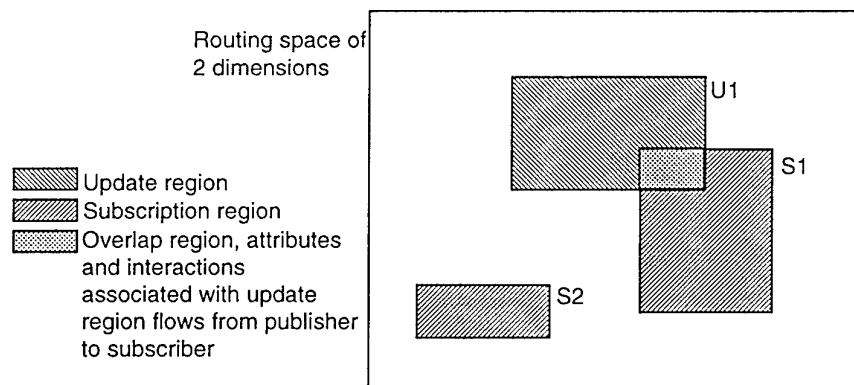
2.5 Data Distribution Management

Interface Specification 1.1 biedt een set van services om efficiënte data distributie binnen de federation te realiseren teneinde de benodigde communicatie bandbreedte te beperken. Deze data distributie is gebaseerd op het concept van *routing spaces*. Een routing space is een abstrakt, logisch multi-dimensionaal coördinaten systeem in welke de federates hun interesses beschrijven, voor het ontvangen of voor het zenden van bepaalde informatie. Binnen de federation kunnen een aantal routing spaces gedefinieerd worden met op de assen van het coördinaten systeem de variabelen. Elke routing space is uniek identificeerbaar.

Zoals eerder vermeld kan m.b.v. Declaration Management geselecteerd worden op informatie *type*. Routing spaces worden gebruikt om de distributie condities te specificeren voor het zenden of ontvangen van informatie (objects en interaction data) op basis van *object attribute* waarden en *interaction parameter* waarden. Elke federate bepaalt welke routing spaces voor hem van belang zijn en definieert *regions*, logische gebieden van de routing space die voor de federate interessant

zijn, door voor elke dimensie begrenzingen (*extents*) op te geven. Door een *subscription region* te koppelen aan een object instantie of interaction class geeft de federate aan dat hij alleen informatie wil ontvangen die voldoet aan de object attribute grenzen en interaction parameter grenzen zoals in de region gedefinieerd. Een *update region* geeft aan dat de federate informatie zal genereren binnen de gedefinieerde begrenzingen. Regions kunnen dynamisch worden aangepast en koppelingen met objecten/interactions kunnen worden veranderd. De RTI zal data distribueren op basis van overlapping van subscription- en update regions. Als er geen overlap is, betekent het dat er geen interesse is in de gegenereerde informatie, m.a.w. de data hoeft niet gedistribueerd te worden. Ook object attributen die zelf *niet* uitgewisseld worden kunnen dienst doen als routing space variabelen. Elke federate kan meerdere subscription- en update regions creëren. Een object attribute mag in maximaal 1 routing space voor komen, zodat geen conflicterende situaties kunnen ontstaan met verschillende extents voor dezelfde attribute in meerdere routing spaces.

Figuur 1 toont een voorbeeld van DDM met 1 update region (U1) en 2 subscription regions (S1 en S2). Aangezien U1 en S1 deels overlappen zullen attributen en interactions die met U1 geassocieerd zijn verstuurd worden naar de federate die subscription region S1 gecreëerd heeft. Omdat er geen overlap is tussen U1 en S2 zal de federate die S2 gecreëerd heeft geen data ontvangen.



Figuur 1 Routing space voorbeeld

DDM filtering op basis van routing spaces en Declaration Management filtering op basis van subscriptions en publications kunnen elkaar tegenspreken. In dat geval hebben de DDM filter criteria de hoogste prioriteit. Er wordt nog door de RTI ontwikkelaars onderzoek gedaan naar Data Distribution Management en de relatie met Time Management. Daarom doet de Interface Specification 1.1 nog geen uitspraak over het precieze tijdstip dat een DDM service effect zal hebben.

3. Run-Time Infrastructure

3.1 Inleiding

De Run-Time Infrastructure (RTI) is de general-purpose software module die de services, zoals vastgelegd in de HLA Interface Specification, implementeert. De RTI kan men vergelijken met een gedistribueerd operating system dat de applicatie bepaalde diensten biedt om informatie uit te wisselen met andere applicaties (federates).

De RTI fungeert als een doorgeefluik voor de uitwisseling van informatie tussen federates. De RTI bewaart zelf geen object attributes, dit is de taak van de federate. Het data distributie mechanisme van de RTI biedt de gebruiker diverse vormen van informatie filtering om de communicatie van irrelevante data te voorkomen en de netwerk performance te maximaliseren.

3.2 Implementaties

De eerste versie van de RTI (F.0) was een prototype gebaseerd op CORBA 1.0 [CORBA]. Vervolgens is er ook een prototype in C++ gebouwd. Deze prototypes bevatten een subset van de Interface Specification 1.0 en richtten zich voornamelijk op volledigheid van functionaliteit en nog niet op een maximale performance van de RTI.

De ontwikkeling van een RTI implementatie is nu onderverdeeld in twee fases. De eerste fase richt zich op de ontwikkeling van RTI 1.0 om de technische haalbaarheid van het RTI concept aan te tonen. Deze RTI implementatie wordt gesponsord door DMSO en is als library vrij verkrijgbaar op het internet. De source code van de RTI is niet vrij gegeven. RTI 1.0 implementeert alle HLA Interface Specification 1.1 services *behalve* Data Distribution Management (DDM). Implementatie van Interface Specification 1.2 is stop gezet en de volgende RTI implementatie, rond maart 1998, zal alle Interface Specification 1.3 services ondersteunen. De tweede fase behelst de ontwikkeling van RTI 2.0 door de industrie.

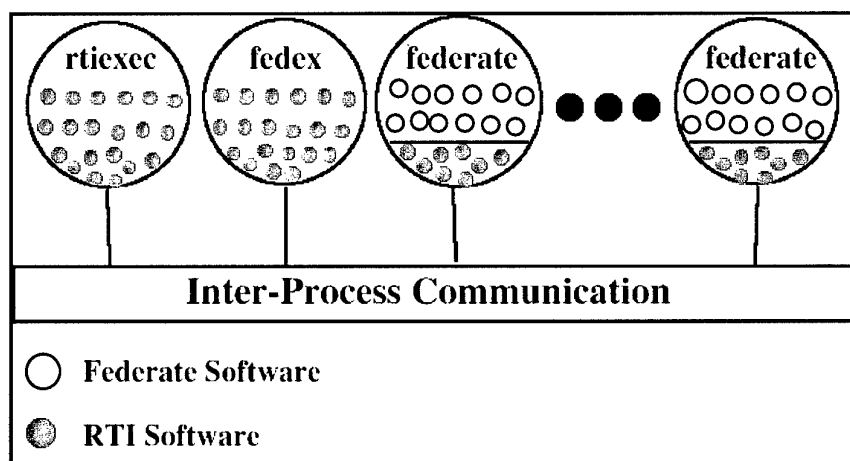
De RTI 1.0 is voor de volgende computer platforms verkrijgbaar:
Sun/Solaris 2.x, SGI/Irix 6.x, Windows NT, IBM AIX 4.1.5 en HP-UX B10.20.

We hebben geëxperimenteerd met RTI versie 1.0.2 met een C++ interface. Behalve C++ is er ook een Ada en Java versie. De Ada RTI versie is een schil om de C++ implementatie, de Java versie is geheel in Java geschreven. De distributie software van de C++ versie beslaat ca. 55 Mbyte inclusief demo programma's. Per federate is ca. 10 tot 15 Mbyte geheugen nodig voor de RTI federate library. De HLA federates draaiden op een Sun SPARCserver-1000 onder Solaris 2.5 [RTI].

RTI 1.0 is reeds door de HLA community uitvoerig getest met grote variëteit aan applicaties. Een federation met 500 objecten, gedistribueerd over 8 federates met ca. 200 updates/sec per federate is al met succes gedemonstreerd op basis van Ultra Sparc 2 processors in een LAN. T.o.v. de RTI F.0 is de performance enorm verbeterd. Over update latencies is nog niet veel bekend.

3.3 RTI 1.0 Architectuur

Figuur 2 geeft schematisch de componenten van de architectuur weer.

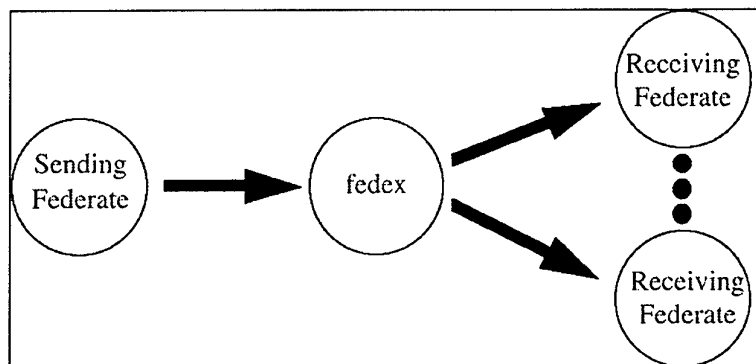


Figuur 2 RTI 1.0 Architectuur

RTI 1.0 is een gedistribueerd systeem dat uit 3 hoofdcomponenten bestaat:

- **RTI Executive (rtiexec)**
Globaal proces dat de creatie en destructie van federation executions beheert. De RTI Executive is het eerste aanspreekpunt voor een initialiserende federate en voorziet de federate met een *handle* naar de federation execution. Via deze handle kan de federate met het fedex process communiceren. De RTI Executive kent ook een *multicast group* toe aan een federation execution voor de communicatie van best-effort data op basis van unreliable UDP/IP.
- **Federation Executive (fedex)**
Globaal proces per federation execution dat 'joining' en 'resigning' van federates afhandelt. De Federation Executive fungeert als een 'information exploder' voor de communicatie van reliable data op basis van TCP/IP (Figuur 3) door de data van 1 ontvanger naar meerdere ontvangers te distribueren. Dit globale proces wordt automatisch opgestart door de eerste federate die een federation creëert. Het proces termineert als alle federates de federation hebben verlaten.

- RTI federate library
Deze wordt meegelinkt met de federate code en bevat de implementatie van de interface naar de HLA 1.1 services. De federate activeert de HLA services door calls in deze library.



Figuur 3 Reliable data communicatie

Het interface tussen de federate en de RTI bestaat uit twee classes: de *RTIambassador* en de *FedAmbassador*. Communicatie tussen de federate en RTI is bi-directioneel: data van federate naar RTI en data van RTI naar federate. De *RTIambassador* class bevat alle functionaliteit die de federate nodig heeft om met de RTI te communiceren b.v. voor het verzenden van object attribute updates. De *FedAmbassador* class bevat een interface voor de RTI om met de federate te communiceren b.v. voor het doorgeven van binnenkomende, door andere federates gegenereerde, attribute updates. Het is aan de federate om deze class methods verder te implementeren en uit te breiden met specifieke functionaliteit voor de verdere afhandeling. De federate is verplicht dit te doen, aangezien de compiler foutmeldingen zal genereren indien bepaalde *FederateAmbassador* class methods niet geïmplementeerd zijn.

3.3.1 Data representatie

Communicatie tussen heterogene computer platformen (Sun, SGI, PC) vereist duidelijke afspraken over de netwerk representatie van de uit te wisselen data. Om de computer platform-specifieke data representatie om te kunnen zetten naar de platform-onafhankelijke netwerk representatie (en vice versa), moet men het datatype weten. Aangezien de RTI 1.0 niet de datatypes kent van object attributen en interaction parameters, is deze conversie de verantwoordelijkheid van de federate zelf.

3.3.2 Federation Execution Data (FED)

De RTI 1.0 heeft twee configuratie files nodig om federation executions te kunnen uitvoeren: de *federation execution data* (FED) file en de *run-time initialization data* (RID) file. De FED file (Figuur 4) bevat de data structuur van de object classes en interaction classes, zoals in de FOM afgesproken. Ook is aangegeven

welke transport mechanisme ('best-effort' of 'reliable') en welke tijd-sorteer ('time-ordering') mechanisme ('receive-order' of 'timestamped-order') er per object class attribute of interaction class gebruikt gaat worden. Het sorteer mechanisme geeft aan of de RTI de informatie gesorteerd op timestamp doorgeeft aan de federate of dat de informatie in volgorde van ontvangst wordt doorgegeven. De FED file beschrijft *niet* de datatypes van de object attributen en interaction parameters.

```
;; Comments are any text after a semicolon.
;; basic syntax example
;; possible <transportation> =   FED_RELIABLE, FED_BEST_EFFORT
;;
;; possible <ordering> =         FED_RECEIVE,  FED_TIMESTAMP
;;
(fed
;; object, class, and attribute definitions follow
  (objects
    (class <name>
      (attribute <name> <transportation> <ordering>)
      (attribute <name> <transportation> <ordering>)
;; any other attributes must come before subclasses for same level
      (class <name>
        (attribute <name> <transportation> <ordering>)
        (attribute <name> <transportation> <ordering>)
      )
    )
  )
;; interactions, class, and parameter definitions follow
  (interactions
    (class <name> <transportation> <ordering>
      (parameter <name>)
      (parameter <name>)
;; any other parameters must come before subclasses for same level
      (class <name> <transportation> <ordering>
        (parameter <name>)
        (parameter <name>)
      )
    )
  )
) ; end of fed
```

Figuur 4 Federation Execution Data (FED) file formaat

3.3.3 Run-time Initialization Data (RID)

De run-time initialization data (RID) file (Figuur 5) bevat allerlei systeem informatie zoals b.v. op welke computer de RTI Executive draait en op welk portnummer de RTI Executive wacht op federate connecties. Ook specificeert deze configuratie file o.a. het maximale aantal objecten per federate.

```

#####
# FILE : RTI.rid
# PURPOSE: This file is the main configuration file for the RTI.
#####

#####
# VARIABLE: BEST_EFFORT_PORT
# UNITS : Positive integer
# PURPOSE : To specify the port number on which best-effort multicast
# addressing will be attempted.
#####
BEST_EFFORT_PORT 18134

#####
# VARIABLE: MAX_OBJECTS_PER_FEDERATE
# UNITS : Positive integer
# PURPOSE : To specify the maximum number of objects a federate may know
# about.
#####
MAX_OBJECTS_PER_FEDERATE 100000

#####
# VARIABLE: RTI_EXEC_HOST
# UNITS : Character string
# PURPOSE : To specify the hostname of machine on which the RTI Executive
# process is executing.
#####
RTI_EXEC_HOST s00sn1

#####
# VARIABLE: RTI_EXEC_PORT
# UNITS : Positive integer
# PURPOSE : To specify the port number on which the RTI Executive process is
# listening for connections.
#####
RTI_EXEC_PORT 3800
.
.
.

```

Figuur 5 RTI.rid voorbeeld file

3.3.4 Management Object Model

Behalve de uitwisseling van simulatie data tussen HLA federates, wisselen de federates ook data uit ter ondersteuning van *federation management* en *monitoring*. Deze data bevat o.a. gegevens over de identiteit van de federate, RTI settings, RTI versie en interne queue grootten. Uitwisseling van deze meta-data is nodig voor de interne boekhouding van de RTI. Tevens heeft de gebruiker toegang tot deze informatie. De structuur van deze informatie wordt beschreven in de *Management Object Model* (MOM). Deze structuur is qua formaat identiek aan de beschrijving van simulatie data. De MOM informatie die de RTI nodig heeft om te kunnen functioneren zal gestandaardiseerd worden. De federate kan zelf de MOM uitbreiden met federation-specifieke informatie.

3.4 Multicasting

Multicasting is het mechanisme waarbij data van 1 zender naar meerdere ontvangers wordt verstuurd. Op dit moment is multicasting alleen mogelijk binnen een LAN omgeving, omdat er nog geen gestandaardiseerde multicast routing protocols zijn voor multicasting op een WAN, zoals het internet. Routers weten nu niet hoe ze met de multicast packets om moeten gaan. Alhoewel er al meerdere voorstellen voor routing protocols zijn gedaan, laat een standaardisatie nog op zich wachten. Een multicast group wordt binnen het IP protocol gerepresenteerd door een Class D adres. Het aantal mogelijke multicast groups is beperkt, omdat de IP adressen in een vastgelegd interval moeten liggen. Er wordt een onderscheid gemaakt tussen permanente ('*permanent*') en tijdelijke ('*temporary*') multicast group adressen. Thans is multicasting gebaseerd op '*best-effort*' communicatie, maar er is ook een grote behoefte aan '*reliable multicasting*', waarbij de ontvangst van de data gegarandeerd is. Vanuit de HLA gemeenschap wordt er nauw gelet op de ontwikkelingen op het gebied van multicasting, omdat dit de basis vormt voor een goed functionerend data distributie mechanisme binnen HLA.

4. Real-time Platform-level Reference FOM

4.1 Inleiding

Voorafgaand aan de ontwikkeling van de High Level Architecture was het Distributed Interactive Simulation (DIS) protocol het meest gebruikte mechanisme om heterogene, real-time simulatie applicaties te koppelen (IEEE standaard 1278.1). DIS en HLA zijn twee verschillende mechanismen: DIS is een simpele data distributie architectuur, gebaseerd op UDP/IP broadcasting, voor berichten met een voorgedefinieerde data formaat (PDU's). Dit is aan de ene kant een voordeel omdat het network protocol gestandaardiseerd is, maar aan de andere kant een nadeel omdat de data formaten vastliggen en niet flexibel zijn. HLA daarentegen biedt een complex distributie mechanisme en laat de definitie van syntax en semantiek van de uit te wisselen berichten over aan de federation zelf (FOM). Dus meer flexibiliteit enerzijds, maar anderzijds moeten er meer afspraken worden gemaakt om simulatie applicaties te koppelen. In HLA wordt de netwerk performance geoptimaliseerd door filtering van data op verschillende niveaus (publication/subscription, DDM).

Het concept van de *Reference FOM* probeert de voordelen van DIS en HLA te combineren. Een Reference FOM is een soort 'basis' FOM voor een specifieke federation. Een Reference FOM bevat object en interaction class definities die veelvuldig voorkomen in de verschillende federation executions van deze federation. De federation execution ontwikkelaar(s) kan/kunnen een Reference FOM als basis nemen en veranderingen aanbrengen, die voor zijn/hun federation execution specifiek zijn. Op deze manier heeft de federation toch een 'standaard' definitie voor de uit te wisselen informatie, met de flexibiliteit om specifieke wensen te implementeren. Een speciale werkgroep houdt zich op dit moment bezig met het concept Reference FOM. Er wordt gedacht aan verschillende *Classes* van Reference FOM's:

- Class 1: Community Guidance FOM
Door ten minste een gebruikersgroep ondersteund;
- Class 2: Common Denominator (CD) FOM
In ten minste één programma/project gedemonstreerd en waarvoor een of andere vorm van stemming/accordering is uitgevoerd;
- Class 3: Procurement FOM
Gebruikt om federates bij aanschaf en ontwikkeling te specificeren.
- Class 4: Base Object Model
Een pakket van fundamentele specificaties die gebruikt kunnen worden om FOMs te construeren.
- Class 5: Hierarchical Object Model
Geordende verzameling van fundamentele specificaties.

Deze verschillende klassen van Reference FOM's vereisen ook verschillende beheerprocedures. Vooralsnog is de discussie over het concept Reference FOM niet afgerond.

De *Real-time Platform-level Reference FOM* (RPR-FOM, uitgesproken als 'reaper fom') is een voorbeeld van een Reference FOM die de inhoud van de DIS PDU's beschrijft in de vorm van een robuuste object en interaction class hiërarchie. De RPR-FOM kan als basis dienen voor (DIS-achtige) real-time, human-in-the-loop simulaties van fysieke entiteiten (zoals tanks en vliegtuigen) en zal een belangrijk hulpmiddel zijn voor de migratie van DIS naar HLA. Aangezien het concept Reference FOM nog nader onderzocht moet worden, wordt de RPR-FOM nog niet als HLA data standaard erkend, nog afgezien van de interim versie van dit moment.

In het vervolg van dit hoofdstuk bekijken we de object en interaction classes van de RPR-FOM versie 0.1.7 in meer detail. Bijlage A bevat de volledige RPR-FOM.

4.2 RPR-FOM Object Class Structure

De RPR-FOM object classes zijn georganiseerd in een 4-niveau diepe hiërarchie. Elke subclass erft automatisch alle attributen van zijn superclasses. Dus b.v. *MilitaryEntity* heeft alle attributen van zowel de *PhysicalEntityClass* als de *BaseEntity* class. Een subclass is een specifiekere vorm van de superclass.

Sommige classes hebben geen attributen maar zijn toch toegevoegd met het oog op Declaration Management (DM) filtering d.w.z. filtering op basis van informatie type. Voor optimale DM filtering moet elke federate *publications* op 'leaf-node' niveau doen (d.w.z. op het laagste niveau in de hiërarchie) en *subscriptions* op een zo laag mogelijk niveau als nodig. Stel federate A gaat tanks simuleren in de federation en publiceert dus de class *MilitaryLandPlatform*. Federate B is geïnteresseerd in alle militaire voertuigen en abonneert zich op de *MilitaryPlatformEntity* class (subscription). Federate C is een maritieme applicatie en heeft alleen belangstelling voor schepen en abonneert zich dus op de class *MilitarySeaSurfacePlatform*. Attribute updates van federate A zullen dan alleen bij federate B aankomen, en voor federate C eruit gefilterd worden. Federate B ontvangt alleen de attributen van *MilitaryPlatformEntity* en zijn superclasses, want de binnenkomende *MilitaryLandPlatform* instantie wordt automatisch 'gepromoveerd' tot *MilitaryPlatformEntity* omdat daar een subscription voor is. Wil Federate B ook de attributen van de 'leaf-node' classes ontvangen dan moet hij zich op de individuele 'leaf-node' classes abonneren.

Base Class	1 st Subclass	2 nd Subclass	3 rd Subclass	4 th Subclass
BaseEntity	PhysicalEntity	MilitaryEntity	MilitaryPlatformEntity	MilitaryAirLandPlatform
				MilitaryAmphibiousPlatform
				MilitaryLandPlatform
				MilitarySpacePlatform
				MilitarySeaSurfacePlatform
				MilitarySubmersiblePlatform
				MilitaryMultiDomainPlatform
			MunitionEntity	
			Soldier	
		CivilPlatformEntity	CivilAirLandPlatform	
			CivilAmphibiousPlatform	
			CivilLandPlatform	
			CivilSpacePlatform	
			CivilSeaSurfacePlatform	
			CivilSubmersiblePlatform	
			CivilMultiDomainPlatform	
		Civilian		
		AggregateEntity		
		EnvironmentEntity		
EmbeddedSystem	Designator			
	EmitterSystem			
	RadioReceiver			
	RadioTransmitter			
EmitterBeam	TrackJamBeam			
SimulationManager				

Figuur 6: RPR-FOM Object Class Structure Table

De RPR-FOM specificeert de volgende base object classes:

- **BaseEntity**

De *BaseEntity* class vertegenwoordigt alle fysieke entiteiten zowel individueel als geaggregeerd, zoals voertuigen, personen en peletons. De class bevat attributen die betrekking hebben op de locatie en bewegingen van de entiteit in de virtuele wereld, zoals positie, oriëntatie, snelheid en acceleratie.

- **EmbeddedSystem**
Deze class representeert alle fysieke *subsystemen* die onderdeel zijn van een entiteit, maar zich zelf niet als entiteit in de federation presenteren. Voorbeelden van de *EmbeddedSystem* class zijn radars, radio en sensoren. De attributen van deze class specificeren de relatie met de entiteit b.v. de entity identificatie.
- **EmitterBeam**
De *EmitterBeam* class beschrijft de karakteristieken van een *emissie* zoals b.v. scan volume, frequentie en vermogen.
- **SimulationManager**
Deze class wordt gebruikt als de initiator van simulatie management berichten en is dus *geen* onderdeel van de virtuele wereld zoals de bovenstaande base classes. Een simulation manager federate correspondeert met één instantie van de class *SimulationManager*. De enige attribute van deze class is een string die de naam van de simulation manager federate identificeert.

4.3 RPR-FOM Interaction Class Structure

BaseClass
ActionRequest
ActionResult
AttributeChangeRequest
AttributeChangeResult
Collision
CreateObjectRequest
CreateObjectResult
MunitionDetonation
RadioSignal
RemoveObjectRequest
RemoveObjectResult
WeaponFire

Figuur 7: Interaction Class Structure Table

Interactions bieden een mechanisme voor federates om discrete events te versturen naar andere federates. In de RPR-FOM zijn een aantal interactions gespecificeerd voor:

- Simulatie management taken (*ActionRequest/Result*, *AttributeChangeRequest/Result*, *CreateObjectRequest/Result*, *RemoveObjectRequest/Result*)
- Botsingen tussen entiteiten (*Collision*)

- Munitie interactie (*MunitionDetonation, WeaponFire*)
- Inter-object communicatie (*RadioSignal*)

Zoals Figuur 7 laat zien, heeft de huidige class structure table geen hiërarchie en dus alleen maar base classes.

4.4 RPR-FOM Ontwikkelingen

De RPR-FOM ontwikkeling wordt gedreven door een werkgroep welke zijn wortels heeft in de twee laatste DIS Workshops (mrt/sep 1996). De RPR-FOM kent sinds kort ook een Versie Plan.

Versie 1 dient alle functionaliteit te bevatten van de (DIS) IEEE 1278.1-1995 standaard. In Versie 2 zal daaraan de functionaliteit van IEEE 1278.1A-1998 toegevoegd worden (o.a. *Collision-Elastic, Underwater Acoustics, Intercom Communication, Entity Management, Minefield*).

Versie 3 wordt genoemd 'Next Generation RPR-FOM'. Waarschijnlijk wordt daarin de relatie met de DIS- IEEE 1278 standaarden meer losgelaten en de mogelijkheden van HLA wat dieper geëxploreerd.

De belangrijkste discussiepunten rond Versie 1 zijn nog de 'vertaling' van de Radio Signal PDU en of al dan niet *padding fields* toegevoegd moeten worden aan Complex Data Types (conform DIS). Ook komen er vanuit de C3I-hoek vragen om meer functionaliteit dan in de DIS-standaarden is gedefiniëerd.

5. Advanced Simulation Framework

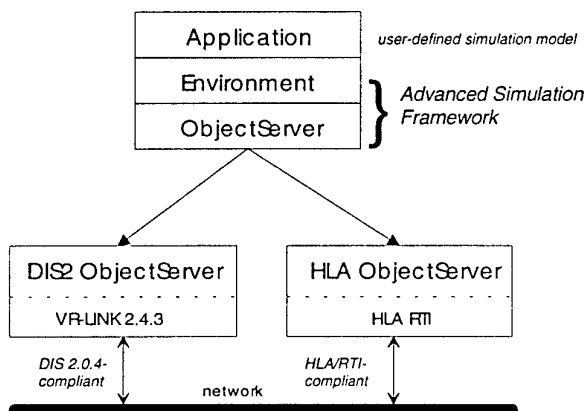
5.1 Electronic Battlespace Facility

De *Electronic Battlespace Facility* (EBF) is een TNO-FEL infrastructuur, bestaande uit hardware en software, voor onderzoek naar en toepassing van gedistribueerde interactieve simulatie technologieën. Er is gekozen voor een flexibele en uitbreidbare opzet van het EBF, zodat ingesprongen kan worden op nieuwe ontwikkelingen op het gebied van gedistribueerde simulatie (zoals HLA) en herbruikbaarheid van beschikbare componenten (zowel hardware als software) maximaal is. Het *Advanced Simulation Framework* ([ASF]) vormt de basis software laag van de EBF.

5.2 ASF Software Architectuur

Om flexibiliteit en herbruikbaarheid te garanderen is gekozen voor een object-georiënteerde aanpak van het ASF. Het ASF biedt de applicatie toegang tot de virtuele omgeving aan de hand van een gestandaardiseerd interface. Het ASF schermt de applicatie zo veel mogelijk af van gedistribueerde simulatie standaarden, zoals DIS en HLA. Deze opzet vergemakkelijkt migratie naar nieuwe standaarden omdat de applicatie zelf minimaal veranderd hoeft te worden. In principe zou de migratie van DIS naar HLA door een hercompilatie van de applicatie gerealiseerd kunnen worden, mits er geen DIS-specifieke functionaliteit in de applicatie zit. Het ASF vormt als het ware een tussenlaag ('*middleware layer*') tussen de applicatie en de onderliggende gedistribueerde simulatie standaarden.

Figuur 8 illustreert de twee lagen van het ASF: *Environment* en *ObjectServer*.



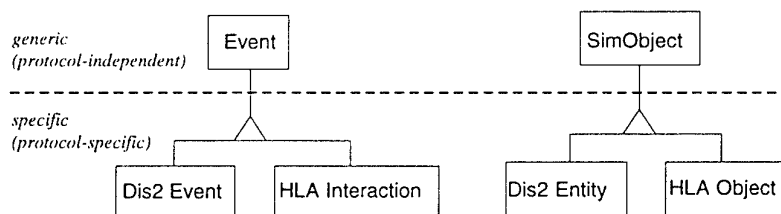
Figuur 8 Advanced Simulation Framework Software Architectuur

Environment biedt de gebruiker een protocol-onafhankelijk interface tot de virtuele omgeving, m.a.w. zonder DIS- of HLA specifieke functionaliteit. Via Envi-

ronment heeft de applicatie toegang tot alle *simulation objects* (b.v. battlefield entities) en *simulation events* (b.v. fire en detonation events) in de virtuele omgeving.

ObjectServer representeert het onderliggende data transport mechanisme dat zorgt voor de uitwisseling van informatie met andere applicaties. *ObjectServer* is deels generiek en deels protocol-specifiek: voor verschillende gedistribueerde simulatie standaarden zijn verschillende *ObjectServer* specialisaties nodig, afgeleid van de generieke *ObjectServer*. Figuur 8 toont twee *ObjectServer* specialisaties voor DIS en HLA. Voor de DIS variant dient de commercieel verkrijgbare VR-LINK Toolkit ([VRLINK]) als communicatie laag, voor de HLA variant wordt de shareware DMSO HLA/RTI als basis gebruikt.

De object-georiënteerde methode van specialisatie (ook wel *subclassing* of *inheritance* genoemd) wordt gebruikt voor de data organisatie van simulation objects en simulation events. Figuur 9 toont deze data structuur in de *Object Modeling Technique* ([OMT]) notatie. *SimObject* bevat functionaliteit en attributen die generiek zijn voor alle simulation objects b.v. object ID. *Event* bevat functionaliteit en attributen gemeenschappelijk voor alle events zoals sender ID. Beide generieke classes hebben specialisaties voor zowel DIS als HLA. Op deze wijze kan de protocol-onafhankelijke Environment gebruik maken van de generieke representaties (*Event* en *SimObject*) terwijl *ObjectServer* de afgeleide classes (b.v. *Dis2Event* of *HlaObject*) kan gebruiken voor protocol-specifieke functionaliteit.



Figuur 9 ASF Data Structures

Het ASF is beschikbaar voor zowel Sun als SGI computer platformen.

5.3 DIS2 ObjectServer

De DIS2 ObjectServer stelt de gebruiker in staat DIS 2.0.4-compliant applicaties te ontwikkelen. Deze implementatie is gebaseerd op COTS software van Mäk Technologies Inc., namelijk de VR-LINK Toolkit. Op dit moment wordt binnen het ASF gewerkt met versie 2.4.3 en 2.4.6 van de toolkit. VR-LINK 2.4.3 ondersteunt

DIS versies 2.0.3 en 2.0.4. VR-LINK 2.4.6 [VRLINK] ondersteunt ook de officiële DIS-IEEE 1278.1-1995 standaard.

5.4 HLA ObjectServer

De huidige HLA ObjectServer is een eerste implementatie van een RPR-FOM 0.1.7-compliant interface. De HLA ObjectServer maakt gebruik van de shareware RTI 1.0.2 van DMSO. Het doel van deze ObjectServer implementatie was aan te tonen dat een integratie van het ASF met de RTI conceptueel mogelijk is en de gebruiker in staat stelt HLA federates te ontwikkelen op basis van de gestandaardiseerde ASF Application Programmer's Interface (API), dezelfde interface die ook voor de ontwikkeling van DIS-applicaties wordt gebruikt.

Omdat niet de nadruk is gelegd op volledigheid van de HLA ObjectServer maar meer op haalbaarheid van het concept, is slechts een subset van de RTI services van Interface Specification 1.1 geïntegreerd in het ASF. Ook wordt slechts een subset van de RPR-FOM 0.1.7 ondersteund. Figuur 10 beschrijft de RTI services die thans geïntegreerd zijn. Van Federation Management, Declaration Management, Object Management en Time Management is slechts een subset van de services geïntegreerd. Ownership Management en Data Distribution Management worden nog niet ondersteund door het ASF. In een later stadium zullen de ontbrekende services toegevoegd worden.

Interface Specification 1.1 Service	Service Category
Create Federation Execution	Federation Management
Destroy Federation Execution	Federation Management
Join Federation Execution	Federation Management
Resign Federation Execution	Federation Management
Publish Object Class	Declaration Management
Publish Interaction Class	Declaration Management
Subscribe Object Class Attribute	Declaration Management
Subscribe Interaction Class	Declaration Management
Request ID	Object Management
Register Object	Object Management
Update Attribute Values	Object Management
Discover Object	Object Management
Reflect Attribute Values	Object Management
Send Interaction	Object Management
Receive Interaction	Object Management
Delete Object	Object Management
Remove Object	Object Management
Request Federate Time	Time Management
Time Advance Request	Time Management
Time Advance Grant	Time Management

Figuur 10 RTI Services in ASF

De volgende RPR-FOM 0.1.7 object classes zijn deels geïmplementeerd in het ASF voor de simulatie van battlespace objecten zoals tanks, vliegtuigen en militair personeel.

Object Class	Super Class	Representatie voor
BaseEntity	-	Alle <i>individuele</i> en <i>geaggregeerde</i> entiteiten zoals voertuigen en personen. De class bevat attributen die betrekking hebben op de locatie en bewegingen van de entiteit in de virtuele wereld, zoals positie, oriëntatie, snelheid en acceleratie.
PhysicalEntity	BaseEntity	Alle <i>individuele</i> , fysische platform entiteiten zoals militaire en civiele voertuigen en personen. De attributen specificeren de karakteristieken van het platform b.v. welke bewegende delen het platform heeft.
MilitaryEntity	PhysicalEntity	Alle <i>militaire</i> , individuele platform entiteiten zoals tanks en militair personeel. De attributen bepalen karakteristieken als force ID en camouflage.

Figuur 11 ASF RPR-FOM Object Classes

De volgende RPR-FOM 0.1.7 interaction classes zijn deels geïmplementeerd in het ASF voor de simulatie van munitie lancering en detonatie.

Object Class	Super Class	Representatie voor
Munition Detonation	-	Detonatie van ballistische of geleide munitie in de battlespace. Parameters specificeren o.a. detonatie locatie, doel en type munitie.
WeaponFire	-	Lancering van ballistische of geleide munitie. Parameters specificeren o.a. lanceer positie, doel en type munitie.

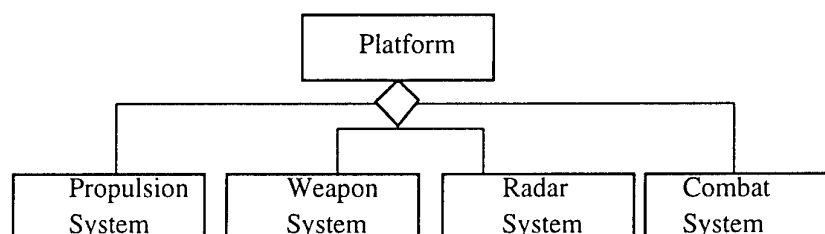
Figuur 12 ASF RPR-FOM Interaction Classes

Een verdere beperking van de huidige HLA-variant van het ASF is dat er nog geen deadreckoning is geïmplementeerd.

5.5 Helicopter Simulatie Applicatie

Het ASF wordt reeds in een aantal simulator prototypes toegepast: b.v. Fennek Reconnaissance Vehicle, Leopard 2 A5, F-16 en Forward Air Controller (FAC). Tevens zijn een aantal tools op het ASF ontwikkeld zoals een scenario manager en een tactical display. Reeds is aangetoond dat m.b.v. ASF vrij eenvoudig gedistribueerde simulatie applicaties kunnen worden gebouwd zonder dat de applicatie bouwer veel van de onderliggende gedistribueerde simulatie standaard hoeft af te weten

Om de migratie mogelijkheden van DIS naar HLA m.b.v. het ASF te onderzoeken, is een eenvoudige helicopter simulatie applicatie ontwikkeld. Deze applicatie is in staat een aantal autonome helicopter platformen te simuleren. Figuur 13 illustreert de object-georiënteerde architectuur van een helicopter met zijn subsystemen.



Figuur 13 Helicopter Platform Architectuur

Elk helicopter platform heeft een *propulsion system*, *weapon system*, *radar system* en een *combat system*. Het propulsion system zorgt voor de voortstuwing van het platform. Het weapon system vertegenwoordigt de bewapening. Het radar system is een sensor waarmee doelen (andere helicopters) kunnen worden opgespoord. De elektronische emissie van de radar is niet gesimuleerd. Het combat system is een semi-intelligent systeem dat de volgende taken sequentieel uitvoert:

- detecteer doelen m.b.v. radar systeem;
- selecteer doel
- achtervolg doel totdat platform binnen schietbereik is;
- schiet op doel totdat doel uitgeschakeld is.

De helicopter simulatie applicatie kan op een willekeurig aantal computer platforms opgestart worden. Per applicatie kan een willekeurig aantal helicopters worden gesimuleerd. Het ASF zorgt voor de uitwisseling van data tussen de applicaties (locatie gegevens en munitie interactie).

Voor migratie van DIS naar HLA van deze applicatie was slechts een hercompilatie nodig van de applicatie. Wijziging van de source code was niet nodig. Zowel de DIS-variant als de HLA-variant bleken te draaien. Er was één verschil: helicopters gesimuleerd door dezelfde HLA federate (applicatie) konden *elkaar* niet uitschakelen. De reden hiervoor bleek te zijn dat *interactions* verzonden door een federate, niet door dezelfde federate ontvangen werden. M.a.w. de detonations interactions, die communiceren dat een helicopter geraakt is, werden niet gedetecteerd door dezelfde federate en dus niet afgehandeld: alleen andere federates konden zijn helicopters uitschakelen. Met DIS is dit probleem er niet omdat PDU's gebroadcast worden en door de zendende applicatie ook zelf weer ontvangen worden. Aangezien deze kwestie ("moet verzonden informatie door de zendende federate zelf gereflecteerd worden?") nog een discussiepunt is binnen de HLA community, is er voor gekozen dit 'probleem' niet in het ASF op te lossen, maar het resultaat van deze discussie af te wachten.

6. Conclusions

De belangrijkste conclusies van het verrichte onderzoek zijn:

- De ontwikkeling van de High Level Architecture draait op volle toeren. De architectuur wordt in de US geaccepteerd als basis voor alle simulatie- en modelleer-activiteiten.
- De RPR-FOM is een belangrijk hulpmiddel voor de migratie van DIS naar HLA voor real-time, human-in-the-loop simulaties. Dit Object Model is een voorstel voor een data standaard die de informatie uit DIS PDU's beschrijft in de vorm van HLA object en interaction classes.
- De RTI is niet veel meer dan een doorgeefluik van type-loze federate data. De federate is zelf verantwoordelijk voor de opslag van deze data en voor de conversie tussen computer platform specifieke data en de netwerk representatie van de data. Veel mensen verwachten veel meer functionaliteit in de RTI ter ondersteuning van de gebruiker.
- De RTI implementatie van DMSO is vrij verkrijgbaar en biedt, behalve Data Distribution Management, alle functionaliteit uit de HLA Interface Specification met een zeer behoorlijke performance. Aangezien de source code niet beschikbaar is en slechts een beperkt aantal computer platformen ondersteund worden, heeft de toepasbaarheid van deze RTI implementatie zijn beperkingen.
- Het ASF biedt een vrij eenvoudig migratie pad voor de TNO DIS-applicaties naar HLA via een '*middleware layer*' approach. Deze software laag schermt de applicatie ontwikkelaar af van de onderliggende gedistribueerde simulatie standaarden d.m.v. een generiek interface. Een HLA-DIS gateway zou een andere eenvoudige oplossing zijn voor een migratie van DIS naar HLA: de gateway converteert DIS PDU's naar RTI calls en vice versa. Deze aanpak heeft echter als nadeel dat de conversie slag de latency van data verhoogt en dat de applicaties nog steeds op DIS gebaseerd zijn en dus niet de voordelen van HLA gebruiken.

Behalve deze conclusies resulteerde het onderzoek tot een aantal bevindingen:

- Object attributes en interactions zijn de atomaire eenheden binnen HLA in tegenstelling tot DIS waar de Protocol Data Unit (PDU) de eenheid is. Na ontdekking van een nieuw object binnen de federation (*discovery*) worden niet automatisch alle object attributes aan de federation bekend gemaakt. Een federate kan daardoor een onvolledig beeld hebben van het object b.v. wel de positie maar niet de orientatie. De federate heeft wel de mogelijkheid om de gemiste attributen op te vragen om zo een volledige status van het object te ver-

krijgen. Een 3D visual heeft b.v. zowel de positie- als orientatie data nodig om het object te kunnen visualiseren.

- HLA staat toe dat per attribute een deadreckon algorithme gespecificeerd wordt voor de extrapolatie van de attribuut waarde. DIS biedt alleen de mogelijkheid voor de extrapolatie van positie- en orientatie data.
- De DMSO RTI maakt gebruik van *exception handling*. Dit is een mechanisme dat de applicatie ontwikkelaar in staat stelt run-time abnormaliteiten af te vangen en af te handelen. Denk b.v. aan een division-by-zero situatie. C++ biedt een uniforme syntax voor exception handling a.h.v. de `try`, `throw` en `catch` keywords. De gebruiker heeft de mogelijkheid dit mechanisme te fine-tunen met zelf-gegenereerde exceptions. Binnen de RTI wordt veel gebruik gemaakt van exception throwing en het blijkt een zeer nuttig hulpmiddel bij de ontwikkeling van HLA federates. De programmeur is in staat exceptions af te vangen om de abnormaliteit te herstellen, zoniet dan termineert de federate met een duidelijke melding van de oorzaak. B.v. als een federate zich aanmeldt bij een federation ('join') en dit mislukt, dan zal de RTI een exception genereren. De federate kan deze exception afvangen en b.v. nog een poging doen zich aan te melden. Als de federate de exception niet afvangt, termineert de applicatie met een duidelijke weergave van de oorzaak.
- De VR-LINK 3.0 Toolkit van Mäk Technologies is in staat applicaties op basis van DIS en HLA te ontwikkelen. Helaas is er geen *backward-compatibility* naar oude versies doordat de VR-LINK API aanzienlijk veranderd is. Tevens ondersteunt VR-LINK een nu al verouderde RPR-FOM. De toolkit biedt dus een migratie pad, maar met de volgende nadelen: licentiekosten, weinig flexibiliteit en afhankelijkheid van Mäk m.b.t. ondersteuning van nieuwe RPR-FOM releases. Het ASF heeft deze nadelen niet.
- Attribute updates en interactions worden niet gereflecteerd door de federate die de data heeft gegenereerd. B.v. als een federate de 'send interaction' service aanroept, zal dat niet leiden tot een 'receive interaction' callback binnen dezelfde federate. Dit betekent dat als een federate een interaction aan een bepaald object richt (b.v. een munitie detonatie event met een target ID) de federate zich er terdege van bewust moet zijn of het object lokaal of extern gesimuleerd wordt. Er is discussie gaande binnen de HLA gemeenschap of reflectie van eigen data misschien toch gewenst is.
- Datalogging en exercise replay in HLA is verre van triviaal. In DIS is dit relatief eenvoudig omdat DIS gebaseerd is op *broadcasting* en de entity heartbeat een continue stroom van berichten garandeert. De HLA logger moet tijdens replay rekening houden met o.a. routing spaces, subscriptions van andere federates en data requests van andere federates. Ook moet de HLA logger de transport mode (best-effort versus reliable) en de ordering mode (receive-order

versus timestamp-order) van de gelogde data via de RTI te weten komen om ze op dezelfde manier te kunnen afspelen. Dit onderwerp benodigt nog verder onderzoek.

Voor de komende periode staan o.a. de volgende vervolgactiviteiten op het programma:

- Ervaring opdoen met Object Model development tools. Deze tools ondersteunen de gebruiker bij het opstellen van SOM's en FOM's.
- Performance en latency metingen van de RTI en het ASF.
- Migratie van DIS simulator prototype naar HLA a.h.v. ASF.
- Uitbreiden ASF voor wat betreft RTI services support en RPR-FOM classes.
- Volgen van en rapporteren over de ontwikkelingen op het gebied van HLA, RTI en RPR-FOM.

7. Afkortingen

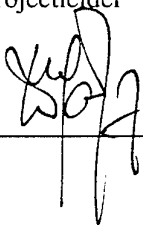
ALSP	Aggregate Level Simulation Protocol
API	Application Programmer's Interface
ASF	Advanced Simulation Framework
CORBA	Common Object Request Broker Architecture
COTS	Commercial Of The Shelf
DDM	Data Distribution Management
DIS	Distributed Interactive Simulation
DMSO	Defense Modeling and Simulation Office (US)
DoD	Department of Defense (US)
EBF	Electronic Battlespace Facility
FED	Federation Execution Data
FOM	Federation Object Model
HLA	High Level Architecture
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol (RFC 791)
LAN	Local Area Network
MOM	Management Object Model
OMT	Object Modeling Technique
PDU	Protocol Data Unit
RID	RTI Initialization Data
RPR-FOM	Real-time Platform level Reference Federation Object Model
RTI	Run-Time Infrastructure
SISO	Simulation Interoperability Standards Organization
SIW	Simulation Interoperability Workshop
SOM	Simulation Object Model
TCP	Transmission Control Protocol (RFC 793)
UDP	User Datagram Protocol (RFC 768)
WAN	Wide Area Network

8. Referenties

- [ASF] Elias and Huiskamp, Advanced Simulation Framework: A Generic Approach to Distributed Simulation. Proc. ITEC '97, The 8th International International Training and Education Conference, Lausanne, Switzerland, 1997.
- [CORBA] Object Management Group, The Common Object Request Broker Architecture and Specification, OMG Document Number 91.12.1, Revision 1, 1992.
- [FEL96A273] Elias, de Jong en Luijff, DIS, DIS++ en de High Level Architecture, TNO-rapport FEL-96-A273, december 1996.
- [OMT] Rumbaugh et.al, Object-Oriented Modeling and Design, Prentice Hall International Editions, 1991.
- [RTI] DoD, High Level Architecture Run-Time Infrastructure Programmer's Guide, version 1.0, May 1997.
- [VRLINK] Granowetter, VR-LINK Release 2.4.6, Mäk Technologies Inc., 1995.

9. Ondertekening

W.G. de Jong
Projectleider

A handwritten signature in black ink, consisting of stylized, overlapping letters, positioned above a horizontal line.

R.J.D. Elias
Auteur

A handwritten signature in black ink, featuring a long, sweeping horizontal stroke that extends to the right, positioned above a horizontal line.

Bijlage A RPR-FOM 0.1.7

Name	Real-time Platform Reference FOM (RPR-FOM)
Version	0.1.7
Modification Date	10 th September, 1997
Type	ReferenCe FOM
Class	Hybrid
Other information	This is a Reference FOM intended for real-time, platform level simulations. It was created to ease the transition of DIS simulations to HLA, although it is not restricted to such simulations.
Sponsor	Simulation Interoperability Standards Organisation (SISO)
Developer	GEC-Marconi RDS Ltd., Simulation and Training Division
Point of Contact	Graham Shanks
POC Phone	+44 1383 828062
POC e-mail	graham.shanks@gecm.com
POC Address	John Sutcliffe Building, Fulmar Way, Donibristle Industrial Park, Fife, SCOTLAND KY11 5JX
Time Management	Real Time
Application Domain	Real time, platform level simulations
Security	Unrestricted
Document references	IEEE 1278.1-1995

Class1	Class2	Class3	Class4	Class5
BaseEntity (S)	AggregateEntity (PS)			
	EnvironmentEntity [1] (PS)			
	PhysicalEntity [1] (PS)	MilitaryEntity (S)	MilitaryPlatformEntity (PS)	MilitaryAirLandPlatform (PS)
				MilitaryAmphibiousPlatform (PS)
				MilitaryLandPlatform (PS)
				MilitarySpacePlatform (PS)
				MilitarySeaSurfacePlatform (PS)
				MilitarySubmersiblePlatform (PS)
				MilitaryMultiDomainPlatform (PS)
			MunitionEntity (PS)	
EmbeddedSystem		Soldier (PS)		
		CivilPlatform (PS)	CivilAirLandPlatform (PS)	
			CivilAmphibiousPlatform (PS)	
			CivilLandPlatform (PS)	
			CivilSpacePlatform (PS)	
			CivilSeaSurfacePlatform (PS)	
			CivilSubmersiblePlatform (PS)	
			CivilMultiDomainPlatform (PS)	
		Civilian (PS)		
	Designator (PS)			
EmitterBeam (PS) SimulationManager (PS)	EmitterSystem (PS)			
	RadioReceiver (PS)			
	RadioTransmitter (PS)			
	TrackJamBeam (PS)			

Interaction Structure	Initiating Object		Receiving Object/Area		Interaction Parameters	Init/ Sense/ React
	Class	Affected Attributes	Class	Affected Attributes		
ActionRequest	SimulationManager	None	AggregateEntity [4]	None	ObjectCount ObjectIDs Action	IR
ActionResult	AggregateEntity [4]	None	SimulationManager	None	ActionResult	IR
AttributeChangeRequest	SimulationManager	None	AggregateEntity [4]	None	ObjectCount ObjectIDs AttributeValueSet	IR
AttributeChangeResult [3]	AggregateEntity [4]	None	SimulationManager	None	ObjectID AttributeChangeResult AttributeValueSet	IR
Collision	PhysicalEntity	AccelerationVector AngularVelocityVector DamageState Immobilized Orientation Position VelocityVector	PhysicalEntity	AccelerationVector AngularVelocityVector DamageState Immobilized Orientation Position VelocityVector	CollidingObjectID CollidingObjectMass CollidingObjectVelocity CollisionType CollisionLocation EventID IssuingObjectID ObjectClass AttributeValueSet	IR
CreateObjectRequest	SimulationManager	None	AggregateEntity [4]	None	ObjectClass AttributeValueSet	IR
CreateObjectResult	AggregateEntity [4]	None	SimulationManager	None	CreateObjectResult	IR
MunitionDetonation	MilitaryPlatformEntity Soldier	None	PhysicalEntity	AccelerationVector AngularVelocityVector DamageState Immobilized Orientation Position VelocityVector	ArticulatedPartsArray ArticulatedPartsCount DetonationLocation DetonationResult EventID FiringObjectID FinalVelocityVector Fuse Type MunitionObjectID MunitionType QuantityFired RateOfFire RelativeDetonationLocation TargetObjectID	IR

Interaction Structure	Initiating Object		Receiving Object/Area		Interaction Parameters	Init/Sense/React
	Class	Affected Attributes	Class	Affected Attributes		
RadioSignal	RadioTransmitter	None	RadioReceiver	None	WarheadType	IR
					EncodingScheme	
					HostRadioID	
					SampleCount	
					SampleRate	
					SignalData	
					SignalDataLength	
RemoveObjectRequest	SimulationManager	None	AggregateEntity [4]	None	TacticalDataLinkType	IR
					ObjectCount	
RemoveObjectResult WeaponFire	AggregateEntity [4] MilitaryEntity	None None	SimulationManager None	None None	ObjectIDs	IR
					RemoveObjectResult	
					EventID	
					FireControlSolutionRange	
					FireMissionIndex	
					FiringLocation	
					FiringObjectID	
					FuseType	
					InitialVelocityVector	
					MunitionObjectID	
					MunitionType	
					QuantityFired	
					RateOfFire	
					TargetObjectID	
					WarheadType	

Object/Interaction	Attribute/Parameter	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condition	Update Type	Update Condition
AggregateEntity	AggregateMarking	MarkingStruct	1	N/A	N/A	N/A	N/A	Static	N/A
	AggregateState	AggregateStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	Dimensions	DimensionStruct	1	N/A	N/A	N/A	N/A	Conditional	AggSizeChange
	EntityIDs	unsigned long [5]	0+	N/A	N/A	perfect	always	Conditional	On change
	ForceID	ForceIDEnum	1	N/A	N/A	N/A	N/A	Static	N/A
	Formation	FormationEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	NumberOfEntities	unsigned short	1	N/A	1	perfect	always	Conditional	On change
	NumberOfSilentAggregates	unsigned short	1	N/A	1	perfect	always	Conditional	On change
	NumberOfSilentEntities	unsigned short	1	N/A	1	perfect	always	Conditional	On change
	NumberOfSubAggregates	unsigned short	1	N/A	1	perfect	always	Conditional	On change
	NumberOfVariableDatums	unsigned short	1	N/A	1	perfect	always	Conditional	On change
	SilentAggregates	SilentEntityStruct	0+	N/A	N/A	N/A	N/A	Conditional	On change
	SilentEntities	SilentEntityStruct	0+	N/A	N/A	N/A	N/A	Conditional	On change
	SubAggregateIDs	unsigned long [5]	0+	N/A	N/A	perfect	always	Conditional	On change
	VariableDatums	VariableDatumStruct	0+	N/A	N/A	N/A	N/A	Conditional	On change
BaseEntity	AccelerationVector	AccelerationStruct	1	N/A	N/A	N/A	N/A	Conditional	AccelerationChange
	AngularVelocityVector	AngVelocityStruct	1	N/A	N/A	N/A	N/A	Conditional	AngVelocityChange
	DRAAlgorithm	DRAAlgorithmEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	EntityType	EntityTypeStruct	1	N/A	N/A	N/A	N/A	Static	N/A
	FederateID	FederateIDStruct	1	N/A	N/A	N/A	N/A	Static	N/A
	IsFrozen	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	Orientation	OrientationStruct	1	N/A	N/A	N/A	N/A	Conditional	OrientationChange
	Position	PositionStruct	1	N/A	N/A	N/A	N/A	Conditional	PositionChange
	VelocityVector	VelocityStruct	1	N/A	N/A	N/A	N/A	Conditional	VelocityChange
	CodeName	CodeNameEnum	1	N/A	N/A	N/A	N/A	Static	N/A
Designator	DesignatedObject	unsigned long [5]	1	N/A	N/A	perfect	always	Conditional	On change
	DesignatorCode	DesignatorCodeEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	DesignatorID	unsigned long [5]	1	N/A	N/A	perfect	always	Static	N/A
	DesignatorPower	float	1	Watts	N/A	perfect	always	Conditional	> DS_PWR_THRSH
	DesignatorSpotLocation	PositionStruct	1	N/A	N/A	N/A	N/A	Conditional	DesignPositionChange
	DesignatorWavelength	float	1	Microns	N/A	perfect	always	Static	N/A
	DRAAlgorithm	DRAAlgorithmEnum	1	N/A	N/A	N/A	N/A	Static	N/A
	RelativeSpotLocation	RelativePositionStruct	1	N/A	N/A	N/A	N/A	Conditional	RelDesigPositionChange

RPR-FOM
Version 0.1.7

Object/Interaction	Attribute/Parameter	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condition	Update Type	Update Condition
EmbeddedSystem	SpotLinearAcceleration	AccelerationStruct	1	N/A	N/A	N/A	N/A	Conditional	DesignAccelerationChange
	FederateID	FederateIDStruct	1	N/A	N/A	N/A	N/A	Static	N/A
	HostObjectID	unsigned long [5]	1	N/A	N/A	perfect	always	Static	N/A
	RelativePosition	RelativePositionStruct	1	N/A	N/A	N/A	N/A	Conditional	On change
	AzimuthCenter	float	1	radians		perfect	always	Conditional	> EE AZ THRSH
EmitterBeam	AzimuthSweep	float	1	radians		perfect	always	Conditional	> EE AZ THRSH
	BeamFunction	BeamFunctionEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	BeamID	octet	1	N/A	N/A	perfect	always	Static	N/A
	BeamParameterIndex	unsigned short	1	N/A	1	perfect	always	Conditional	On change
	ElevationCenter	float	1	radians		perfect	always	Conditional	> EE EL THRSH
	ElevationSweep	float	1	radians		perfect	always	Conditional	> EE EL THRSH
	EmittingSystemID	unsigned long [5]	1	N/A	N/A	perfect	always	Static	N/A
	ERP	float	1	dBm		perfect	always	Conditional	EmiERPChange
	FederateID	FederateIDStruct	1	N/A	N/A	N/A	N/A	Static	N/A
	Frequency	float	1	Hz		perfect	always	Conditional	> EE FREQ THRSH
	FrequencyRange	float	1	Hz		perfect	always	Conditional	> EE FRNG THRSH
	PRF	float	1	Hz		perfect	always	Conditional	> EE PRF THRSH
	PulseWidth	float	1	microsec		perfect	always	Conditional	> EE PW THRSH
	SweepSynch	float	1	%		perfect	always	Periodic	HRT BEAT TIMER secs
	EmitterFunction	EmitterFunctionEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
EmitterSystem	EmitterName	EmitterNameEnum	1	N/A	N/A	N/A	N/A	Static	N/A
	EmitterNumber	octet	1	N/A	1	perfect	always	Static	N/A
	BoundingVolume	BoundingVolumeStruct	1	N/A	N/A	N/A	N/A	Conditional	EnvBVChange
EnvironmentEntity [1]	GeometryDefinitionArray	GeometryShapeStruct	0+	N/A	N/A	N/A	N/A	Conditional	EnvGeoChange
	NumberOfGeometries	unsigned short	1	N/A	1	perfect	always	Conditional	On change
	AlternateEntityType	EntityTypeStruct	1	N/A	N/A	N/A	N/A	Static	N/A
MilitaryEntity	CamouflageType	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	FirePowerDisabled	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	ForceID	ForceIDEnum	1	N/A	N/A	N/A	N/A	Static	N/A
	IsConcealed	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
MilitaryPlatformEntity	AfterburnerOn	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	HasAmmunitionSupplyCap	boolean	1	TRUE/FALSE	N/A	perfect	always	Static	N/A
	LauncherRaised	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change

RPR-FOM

Version 0.1.7

Attribute/Parameter Table

Object/Interaction	Attribute/Parameter	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condition	Update Type	Update Condition
MunitionEntity PhysicalEntity [1]	LauncherFlashPresent	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	ArticulatedParametersArray	ArticulatedParameterStruct	0+	N/A	N/A	N/A	N/A	Conditional	On change
	ArticulatedParametersCount	unsigned short	1	N/A	1	perfect	always	Static	N/A
	DamageState	DamageStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	EngineSmokeOn	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	FlamesPresent	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	HasFuelSupplyCap	boolean	1	TRUE/FALSE	N/A	perfect	always	Static	N/A
	HasRecoveryCap	boolean	1	TRUE/FALSE	N/A	perfect	always	Static	N/A
	HasRepairCap	boolean	1	TRUE/FALSE	N/A	perfect	always	Static	N/A
	HatchState	HatchStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	Immobilized	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	LifformState	LifformStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	LightsState	LightStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	Marking	MarkingStruct	1	N/A	N/A	N/A	N/A	Static	N/A
	PowerPlantOn	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	RampDeployed	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	SmokePlumePresent	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
RadioReceiver	TentDeployed	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	TrailState	TrailStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	RadioNumber	unsigned short	1	N/A	N/A	perfect	always	Static	N/A
	ReceivedPower	float	1	dB-milliwatts		perfect	always	Conditional	> RX PWR THRESH
	ReceivedTransmitter	unsigned long [5]	1	N/A	N/A	perfect	always	Static	N/A
	ReceiverState	ReceiveStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	AntennaPattern	AntennaPatternStruct	1	N/A	N/A	N/A	N/A	Conditional	On change
	AntennaPatternType	AntennaPatternEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	CryptoKeyID	unsigned short	1	N/A	N/A	perfect	always	Conditional	On change
	CryptoSystem	CryptoSystemEnum	1	N/A	N/A	N/A	N/A	Static	N/A
RadioTransmitter	DetailedModulationType	DetailedModulationEnum	1	N/A	N/A	N/A	N/A	Static	N/A
	Frequency	UnsignedInteger64	1	N/A	N/A	N/A	N/A	Conditional	> RADIO FREQ THRESH
	FrequencyBandwidth	float	1	Hz		perfect	always	Conditional	> RADIO FBAND THRESH
	FrequencyHopInUse	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	InputSource	InputSourceEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	ModulationParameterCount	octet	1	N/A	N/A	perfect	always	Conditional	On change

RPR-FOM

Version 0.1.7

Object/Interaction	Attribute/Parameter	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condition	Update Type	Update Condition
	ModulationParameters	ModulationStruct	0+	N/A	N/A	N/A	N/A	Conditional	On change
	MajorModulationType	MajorModulationEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	ModulationSystemType	ModulationSystemEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	Power	float	1	dB-milliwatts	N/A	perfect	always	Conditional	> RADIO PWR THRESH
	PseudoNoiseSpectrumInUs	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	RadioNumber	unsigned short	1	N/A	N/A	perfect	always	Static	N/A
	RadioTransmitterID	unsigned long [5]	1	N/A	N/A	perfect	always	Static	N/A
	RadioType	RadioTypeStruct	1	N/A	N/A	N/A	N/A	Static	N/A
	TimeHopInUse	boolean	1	TRUE/FALSE	N/A	perfect	always	Conditional	On change
	TransmitState	TransmitStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
Soldier	PrimaryWeaponState	WeaponStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
	SecondaryWeaponState	WeaponStateEnum	1	N/A	N/A	N/A	N/A	Conditional	On change
TrackJamBeam	JammingModeSequence	unsigned long	1	N/A	N/A	perfect	always	Conditional	On change
	NumberOfTrackJamObjects	octet	1	N/A	N/A	perfect	always	Conditional	On change
	TrackJamObjectIDs	unsigned long [5]	1+	N/A	N/A	perfect	always	Conditional	On change
	TrackOrJammer	boolean	1	TRUE/FALSE	N/A	perfect	always	Static	N/A
	SimulationManagerName	string	1	N/A	N/A	perfect	always	N/A	N/A
SimulationManager ActionRequest	ObjectCount	unsigned short	1	N/A	N/A	perfect	always	N/A	N/A
	ObjectIDs	unsigned long	1+	N/A	N/A	perfect	always	N/A	N/A
ActionResult	Action	ActionEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	ActionResult	ActionResultEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
AttributeChangeRequest	ObjectCount	unsigned short	1	N/A	N/A	perfect	always	N/A	N/A
	ObjectIDs	unsigned long	1+	N/A	N/A	perfect	always	N/A	N/A
AttributeChangeResult [3]	AttributeValueSet	AttributeValueSetStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	ObjectID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
Collision	AttributeChangeResult	AttributeChangeResultEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	AttributeValueSet	AttributeValueSetStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	CollidingObjectID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	CollidingObjectMass	float	1	N/A	N/A	perfect	always	N/A	N/A
	CollidingObjectVelocity	VelocityStruct	1	kilograms	N/A	perfect	always	N/A	N/A
	CollisionType	CollisionTypeEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	CollisionLocation	PositionStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	EventID	EventIDStruct	1	N/A	N/A	N/A	N/A	N/A	N/A

Attribute/Parameter Table

Object/Interaction	Attribute/Parameter	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condition	Update Type	Update Condition
CreateObjectRequest	IssuingObjectID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	ObjectClass	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	AttributeValueSet	AttributeValueSetStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	CreateObjectResult	CreateObjectResultEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	ArticulatedPartsArray	ArticulatedParameterStruct	0+	N/A	N/A	N/A	N/A	N/A	N/A
	ArticulatedPartsCount	octet	1	N/A	N/A	perfect	always	N/A	N/A
	DetonationLocation	PositionStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	DetonationResult	DetonationResultEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	EventID	EventIDStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	FiringObjectID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	FinalVelocityVector	VelocityStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	Fuse Type	FuseTypeEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	MunitionObjectID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	MunitionType	EntityTypeStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	QuantityFired	unsigned short	1	N/A	N/A	perfect	always	N/A	N/A
RadioSignal	RateOfFire	unsigned short	1	N/A	N/A	perfect	always	N/A	N/A
	RelativeDetonationLocation	RelativePositionStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	TargetObjectID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	WarheadType	WarheadTypeEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	EncodingScheme	EncodingSchemeEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	HostRadiolID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	SampleCount	unsigned short	1	N/A	1	perfect	always	N/A	N/A
	SampleRate	unsigned long	1	bits/second	1	perfect	always	N/A	N/A
	SignalData	octet	1+	N/A	N/A	perfect	always	N/A	N/A
	SignalDataLength	unsigned short	1	N/A	1	perfect	always	N/A	N/A
	TacticalDataLinkType	TacticalDataLinkTypeEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	ObjectCount	unsigned short	1	N/A	N/A	perfect	always	N/A	N/A
	ObjectIDs	unsigned long	1+	N/A	N/A	perfect	always	N/A	N/A
	RemoveObjectResult	RemoveObjectResultEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	WeaponFire	EventID	EventIDStruct	1	N/A	N/A	N/A	N/A	N/A
FireControlSolutionRange		float	1	metres	N/A	perfect	always	N/A	N/A
FireMissionIndex		unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	FireInLocation	PositionStruct	1	N/A	N/A	N/A	N/A	N/A	N/A

Object/Interaction	Attribute/Parameter	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condition	Update Type	Update Condition
	FiringObjectID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	Fuse Type	FuseTypeEnum	1	N/A	N/A	N/A	N/A	N/A	N/A
	InitialVelocityVector	VelocityStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	MunitionObjectID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	MunitionType	EntityTypeStruct	1	N/A	N/A	N/A	N/A	N/A	N/A
	QuantityFired	unsigned short	1	N/A	N/A	perfect	always	N/A	N/A
	RateOfFire	unsigned short	1	N/A	N/A	perfect	always	N/A	N/A
	TargetObjectID	unsigned long	1	N/A	N/A	perfect	always	N/A	N/A
	WarheadType	WarheadTypeEnum	1	N/A	N/A	N/A	N/A	N/A	N/A

RPR-FOM

Version 0.1.7

Identifier	Enumerator	Representation
AggregateStateEnum	Other	0
	Aggregated	1
	Disaggregated	2
	FullyDisaggregated	3
	PseudoDisaggregated	4
AntennaPatternEnum	PartiallyDisaggregated	5
	Other	0
	OmniDirectional	1
	Beam	2
	SphericalHarmonic	3
ArticulatedTypeEnum	Other	0
	Rudder	1024
	LeftFlap	1056
	RightFlap	1088
	LeftAileron	1120
	RightAileron	1152
	HelicopterMainRotor	1184
	HelicopterTailRotor	1216
	OtherAircraftControlSurfaces	1248
	Periscope	2048
	GenericAntenna	2080
	Snorkel	2112
	OtherExtendableParts	2144
	LandingGear	3072
	TailHook	3104
	SpeedBrake	3136
	LeftWeaponBayDoors	3168
	RightWeaponBayDoors	3200
	TankOrAPChatch	3232
	Wingsweep	3264
BridgeSectionEnum	BridgeLauncher	3296
	BridgeSection1	3328
	BridgeSection2	3360
	BridgeSection3	3392
	PrimaryBlade1	3424
	PrimaryBlade2	3456

RPR-FOM

Version 0.1.7

Identifier	Enumerator	Representation
	PrimaryBoom	3488
	PrimaryLauncherArm	3520
	OtherFixedPositionParts	3552
	PrimaryTurretNumber1	4096
	PrimaryTurretNumber2	4128
	PrimaryTurretNumber3	4160
	PrimaryTurretNumber4	4192
	PrimaryTurretNumber5	4224
	PrimaryTurretNumber6	4256
	PrimaryTurretNumber7	4288
	PrimaryTurretNumber8	4320
	PrimaryTurretNumber9	4352
	PrimaryTurretNumber10	4384
	PrimaryGunNumber1	4416
	PrimaryGunNumber2	4448
	PrimaryGunNumber3	4480
	PrimaryGunNumber4	4512
	PrimaryGunNumber5	4544
	PrimaryGunNumber6	4576
	PrimaryGunNumber7	4608
	PrimaryGunNumber8	4640
	PrimaryGunNumber9	4672
	PrimaryGunNumber10	4704
	PrimaryLauncher1	4736
	PrimaryLauncher2	4768
	PrimaryLauncher3	4800
	PrimaryLauncher4	4832
	PrimaryLauncher5	4864
	PrimaryLauncher6	4896
	PrimaryLauncher7	4928
	PrimaryLauncher8	4960
	PrimaryLauncher9	4992
	PrimaryLauncher10	5024
	PrimaryDefenseSystems1	5056
	PrimaryDefenseSystems2	5088
	PrimaryDefenseSystems3	5120

RPR-FOM

Version 0.1.7

Identifier	Enumerator	Representation
	PrimaryDefenseSystems4	5152
	PrimaryDefenseSystems5	5184
	PrimaryDefenseSystems6	5216
	PrimaryDefenseSystems7	5248
	PrimaryDefenseSystems8	5280
	PrimaryDefenseSystems9	5312
	PrimaryDefenseSystems10	5344
	PrimaryRadar1	5376
	PrimaryRadar2	5408
	PrimaryRadar3	5440
	PrimaryRadar4	5472
	PrimaryRadar5	5504
	PrimaryRadar6	5536
	PrimaryRadar7	5568
	PrimaryRadar8	5600
	PrimaryRadar9	5632
	PrimaryRadar10	5664
	SecondaryTurretNumber1	5696
	SecondaryTurretNumber2	5728
	SecondaryTurretNumber3	5760
	SecondaryTurretNumber4	5792
	SecondaryTurretNumber5	5824
	SecondaryTurretNumber6	5856
	SecondaryTurretNumber7	5888
	SecondaryTurretNumber8	5920
	SecondaryTurretNumber9	5952
	SecondaryTurretNumber10	5984
	SecondaryGunNumber1	6016
	SecondaryGunNumber2	6048
	SecondaryGunNumber3	6080
	SecondaryGunNumber4	6112
	SecondaryGunNumber5	6144
	SecondaryGunNumber6	6176
	SecondaryGunNumber7	6208
	SecondaryGunNumber8	6240
	SecondaryGunNumber9	6272

RPR-FOM
Version 0.1.7

Identifier	Enumerator	Representation
	SecondaryGunNumber10	6304
	SecondaryLauncher1	6336
	SecondaryLauncher2	6368
	SecondaryLauncher3	6400
	SecondaryLauncher4	6432
	SecondaryLauncher5	6464
	SecondaryLauncher6	6496
	SecondaryLauncher7	6528
	SecondaryLauncher8	6560
	SecondaryLauncher9	6592
	SecondaryLauncher10	6624
	SecondaryDefenseSystems1	6656
	SecondaryDefenseSystems2	6688
	SecondaryDefenseSystems3	6720
	SecondaryDefenseSystems4	6752
	SecondaryDefenseSystems5	6784
	SecondaryDefenseSystems6	6816
	SecondaryDefenseSystems7	6848
	SecondaryDefenseSystems8	6880
	SecondaryDefenseSystems9	6912
	SecondaryDefenseSystems10	6944
	SecondaryRadar1	6976
	SecondaryRadar2	7008
	SecondaryRadar3	7040
	SecondaryRadar4	7072
	SecondaryRadar5	7104
	SecondaryRadar6	7136
	SecondaryRadar7	7168
	SecondaryRadar8	7200
	SecondaryRadar9	7232
	SecondaryRadar10	7264
BeamFunctionEnum	Other	0
	Search	1
	HeightFinder	2
	Acquisition	3
	Tracking	4

Enumerated Datatype Table

Identifier	Enumerator	Representation
	AcquisitionAndTracking	5
	CommandGuidance	6
	Illumination	7
	RangeOnlyRadar	8
	MissileBeacon	9
	MissileFuze	10
	ActiveRadarMissileSeeker	11
	Jammer	12
	Other	0
	Other	0
	TBD	1
	CollisionTypeEnum	Inelastic
Elastic		1
Other		2
CryptoSystemEnum	Other	0
	KY_28	1
	KY_58	2
	NarrowSpectrumSecureVoice_NSVE	3
DamageStateEnum	WideSpectrumSecureVoice_WSVE	4
	SINGARS_ICOM	5
	NoDamage	0
	SlightDamage	1
DatumIDEnum	ModerateDamage	2
	Destroyed	3
	Other	0
	Other	0
DesignatorCodeEnum	TBD	1
	Other	0
	Other	0
	Other	0
DetailedModulationEnum	EntityImpact	1
	EntityProximateDetonation	2
	GroundImpact	3
	GroundProximateDetonation	4
DetonationResultEnum	Detonation	5
	None	6
	HF bit Small	7

RPR-FOM

Version 0.1.7

Identifier	Enumerator	Representation
DRAAlgorithmEnum	HE_hit_Medium	8
	HE_hit_Large	9
	ArmorPiercingHit	10
	DirtBlast_Small	11
	DirtBlast_Medium	12
	DirtBlast_Large	13
	WaterBlast_Small	14
	WaterBlast_Medium	15
	WaterBlast_Large	16
	AirHit	17
	BuildingHit_Small	18
	BuildingHit_Medium	19
	BuildingHit_Large	20
	MineClearingLineCharge	21
	EnvironmentObjectImpact	22
	EnvironmentObjectProximateDetonation	23
	WaterImpact	24
	AirBurst	25
	Other	0
	Static	1
	DRM_FPW	2
	DRM_RPW	3
	DRM_RVW	4
	DRM_FVW	5
	DRM_FPB	6
	DRM_RPB	7
	DRM_RVB	8
	DRM_FVB	9
EmitterFunctionEnum	Other	0
	MultiFunction	1
	EarlyWarningSurveillance	2
	HeightFinding	3
	FireControl	4
	AcquisitionDetection	5
	Tracking	6
	GuidanceIllumination	7

RPR-FOM
Version 0.1.7

Identifier	Enumerator	Representation
FiringPointLaunchPointLocation	FiringPointLaunchPointLocation	8
	Ranging	9
	RadarAltimeter	10
	Imaging	11
	MotionDetection	12
	Navigation	13
	Weather	14
	Instrumentation	15
	IdentificationClassification	16
	JammingNoise	64
	JammingDeception	65
	Decoy	66
	WeaponNonLethal	96
	WeaponLethal	97
	Other	0
	Other	0
	Other	0
EntityNameEnum	EntityNameEnum	0
	EntityCategoryEnum	0
	EntityCountryEnum	0
	EntityDomainEnum	0
	EntityExtraEnum	0
	EntityKindEnum	0
	EntitySpecificEnum	0
	EntitySubcategoryEnum	0
	EntityStateEnum	0
	Idle	0
FederationStateEnum	Running	1
	Saving	2
	Restoring	3
	Damaged	4
	Joining	5
	Resigning	6
	Resigned	7
	Idle	0
	Running	1
	Saving	2
ForceIdEnum	Restoring	3
	Other	0
	Friendly	1

RPR-FOM

Version 0.1.7

Identifier	Enumerator	Representation
FormationEnum	Opposing	2
	Neutral	3
	Other	0
	Assembly	1
	Vee	2
	Wedge	3
	Line	4
	Column	5
	Other	0
	IntelligentInfluence	10
FuseTypeEnum	Sensor	20
	SelfDestruct	30
	UltraQuick	40
	Body	50
	DeepIntrusion	60
	Multifunction	100
	PointDetonation_PD	200
	BaseDetonation_BD	300
	Contact	1000
	ContactInstantImpact	1100
	ContactDelayed	1200
	ContactElectronicObliqueContact	1300
	ContactGraze	1400
	ContactCrush	1500
	ContactHydrostatic	1600
	ContactMechanical	1700
	ContactChemical	1800
	ContactPiezoelectric	1900
	ContactPointInitiating	1910
	ContactPointInitiatingBaseDetonating	1920
	ContactBaseDetonating	1930
	ContactBallisticCapAndBase	1940
	ContactBase	1950
	ContactNose	1960
	ContactFittedInStandoffProbe	1970
	ContactNonAligned	1980

RPR-FOM
Version 0.1.7

Identifier	Enumerator	Representation
	Timed	2000
	TimedProgrammable	2100
	TimedBurnout	2200
	TimedPyrotechnic	2300
	TimedElectronic	2400
	TimedBaseDelay	2500
	TimedReinforcedNoseImpactDelay	2600
	TimedShortDelayImpact	2700
	TimedNoseMountedVariableDelay	2800
	TimedLongDelaySide	2900
	TimedSelectableDelay	2910
	TimedImpact	2920
	TimedSequence	2930
	Proximity	3000
	ProximityActiveLaser	3100
	ProximityMagneticMagnetpolarity	3200
	ProximityActiveDopplerRadar	3300
	ProximityRadioFrequencyRF	3400
	ProximityProgrammable	3500
	ProximityProgrammablePrefragmented	3600
	ProximityInfrared	3700
	Command	4000
	CommandElectronicRemotelySet	4100
	Altitude	5000
	AltitudeRadioAltimeter	5100
	AltitudeAirBurst	5200
	Depth	6000
	Acoustic	7000
	Pressure	8000
	PressureDelay	8010
	Inert	8100
	Dummy	8110
	Practice	8120
	PlugRepresenting	8130
	Training	8150
	Pyrotechnic	9000

RPR-FOM

Version 0.1.7

Identifier	Enumerator	Representation
HatchStateEnum	PyrotechnicDelay	9010
	ElectroOptical	9100
	ElectroMechanical	9110
	ElectroMechanicalNose	9120
	Strikerless	9200
	StrikerlessNoseImpact	9210
	StrikerlessCompressionIgnition	9220
	CompressionIgnition	9300
	CompressionIgnitionStrikerlessNoseImpact	9310
	Percussion	9400
	PercussionInstantaneous	9410
	Electronic	9500
	ElectronicInternallyMounted	9510
	ElectronicRangeSetting	9520
	ElectronicProgrammed	9530
	Mechanical	9600
	MechanicalNose	9610
	MechanicalTail	9620
	NotApplicable	0
	PrimaryHatchIsClosed	1
	PrimaryHatchIsPopped	2
LifeformStateEnum	PrimaryHatchIsPoppedAndPersonIsVisibleUnderHatch	3
	PrimaryHatchIsOpen	4
	PrimaryHatchIsOpenAndPersonIsVisible	5
	NotApplicable	0
	UprightStandingStill	1
	UprightWalking	2
	UprightRunning	3
	Kneeling	4
	Prone	5
	Crawling	6
	Swimming	7
	Parachuting	8
	Jumping	9
	Sitting	10
	Squatting	11

RPR-FOM

Version 0.1.7

Identifier	Enumerator	Representation
LightStateEnum	Crouching	12
	Wading	13
	Other	0
	Other	0
MajorModulationEnum	Amplitude	1
	AmplitudeAndAngle	2
	Angle	3
	Combination	4
	Pulse	5
	Unmodulated	6
	Other	0
ModulationSystemEnum	Generic	1
	HQ	2
	HQII	3
	HQIIA	4
	SINGARS	5
	CCTT_SINGARS	6
	Other	0
	Other	0
	Other	0
	Dummy	0
MunitionTypeEnum	Off	0
	OnButNotReceiving	1
	OnAndReceiving	2
	Dummy	0
ReferenceSystemEnum	Off	0
	OnButNotTransmitting	1
	OnAndTransmitting	2
	Other	0
WarheadTypeEnum	Other	0
	CargoVariableSubmunitions	10
	FuelAirExplosive	20
	GlassBlads	30
	Warhead_1um	31
	Warhead_5um	32
	Warhead_10um	33
	Other	0
	Other	0
	Other	0

RPR-FOM
Version 0.1.7

Identifier	Enumerator	Representation
	HighExplosive	1000
	HE_Plastic	1100
	HE_Incendiary	1200
	HE_Fragmentation	1300
	HE_Antitank	1400
	HE_Bomblets	1500
	HE_ShapedCharge	1600
	HE_ContinuousRod	1610
	HE_TungstenBall	1615
	HE_BlastFragmentation	1620
	HE_SteerableDartswithHE	1625
	HE_Darts	1630
	HE_Flechettes	1635
	HE_DirectedFragmentation	1640
	HE_SemiArmorPiercing	1645
	HE_ShapedChargeFragmentation	1650
	HE_SemiArmorPiercingFragmentation	1655
	HE_HollowCharge	1660
	HE_DoubleHollowCharge	1665
	HE_GeneralPurpose	1670
	HE_BlastPenetrator	1675
	HE_RodPenetrator	1680
	HE_Antipersonnel	1685
	Smoke	2000
	Illumination	3000
	Practice	4000
	Kinetic	5000
	Mines	6000
	Nuclear	7000
	NuclearMT	7010
	ChemicalGeneral	8000
	ChemicalBlisterAgent	8100
	HD_Mustard	8110
	ThickenedHD_Mustard	8115
	DustyHD_Mustard	8120
	ChemicalBloodAgent	8200

RPR-FOM

Version 0.1.7

Identifier	Enumerator	Representation
	AC_HCN	8210
	CK_CNCl	8215
	CG_Phosgene	8220
	ChemicalNerveAgent	8300
	VX	8310
	ThickenedVX	8315
	DustyVX	8320
	GA_Tabun	8325
	ThickenedGA_Tabun	8330
	DustyGA_Tabun	8335
	GB_Sarin	8340
	ThickenedGB_Sarin	8345
	DustyGB_Sarin	8350
	GD_Soman	8355
	ThickenedGD_Soman	8360
	DustyGD_Soman	8365
	GF	8370
	ThickenedGF	8375
	DustyGF	8380
	Biological	9000
	BiologicalVirus	9100
	BiologicalBacteria	9200
	BiologicalRickettsia	9300
	BiologicalGeneticallyModifiedMicroOrganisms	9400
	BiologicalToxin	9500
WeaponStateEnum	NoWeapon	0
	Stowed	1
	Deployed	2
	FiringPosition	3
ActionEnum	Dummy	0
ActionResultEnum	Dummy	0
AttributeChangeResultEnum	Dummy	0
CreateObjectResultEnum	Dummy	0
RemoveObjectResultEnum	Dummy	0
EncodingSchemeEnum	Dummy	0
TacticalDataLinkTypeEnum	Dummy	0

Identifier	Enumerator	Representation
InputSourceEnum	Dummy	0

Complex Datatype	Field Name	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condition
AccelerationStruct	XAcceleration	float	1	m/s/s		perfect	always
	YAcceleration	float	1	m/s/s		perfect	always
	ZAcceleration	float	1	m/s/s		perfect	always
AngVelocityStruct	XAxisRate	float	1	radians/s		perfect	always
	YAxisRate	float	1	radians/s		perfect	always
	ZAxisRate	float	1	radians/s		perfect	always
AntennaPatternStruct	BeamAntenna	BeamAntennaStruct	0-1	N/A	N/A	N/A	N/A
	SphericalHarmonicAntenna	SphericalHarmonicAntennaStruct	0-1	N/A	N/A	N/A	N/A
	ArticulatedParameterType	ArticulatedTypeEnum	1	N/A	N/A	N/A	N/A
ArticulatedParameterStruct	ArticulatedParameterChange	octet	1	N/A	1	perfect	always
	PartAttachedTo	unsigned short	1	N/A	1	perfect	always
	ParameterType	ParameterTypeEnum	1	N/A	N/A	N/A	N/A
BeamAntennaStruct	ParameterValue	ParameterValueType	1	N/A	N/A	N/A	N/A
	AzimuthBandwidth	float	1	radians		perfect	always
	BeamDirection	OrientationStruct	1	N/A	N/A	N/A	N/A
	ElevationBeamwidth	float	1	radians		perfect	always
	Ex	float	1			perfect	always
	Ez	float	1			perfect	always
	Phase	float	1			perfect	always
	ReferenceSystem	ReferenceSystemEnum	1	N/A	N/A	N/A	N/A
BoundingVolumeStruct	Dummy	any	1			perfect	always
DimensionStruct	XAxisLength	float	1	metres		perfect	always
	YAxisLength	float	1	metres		perfect	always
	ZAxisLength	float	1	metres		perfect	always
EntityTypeStruct	EntityKind	EntityKindEnum	1	N/A	N/A	N/A	N/A
	Domain	EntityDomainEnum	1	N/A	N/A	N/A	N/A
	Country	EntityCountryEnum	1	N/A	N/A	N/A	N/A
	Category	EntityCategoryEnum	1	N/A	N/A	N/A	N/A
	Subcategory	EntitySubcategoryEnum	1	N/A	N/A	N/A	N/A
	Specific	EntitySpecificEnum	1	N/A	N/A	N/A	N/A
EventIDStruct	Extra	EntityExtraEnum	1	N/A	N/A	N/A	N/A
	EventCount	unsigned short	1	N/A	N/A	perfect	always
	IssuingObjectID	unsigned long	1	N/A	N/A	perfect	always
FederateIDStruct	SiteID	unsigned short	1	N/A	N/A	perfect	always
	ApplicationID	unsigned short	1	N/A	N/A	perfect	always

RPR-FOM

Version 0.1.7

Complex Datatype	Field Name	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condition
GeometryShapeStruct	EntityID	unsigned short	1	N/A	N/A	perfect	always
	DummyGeometry	any	1			perfect	always
	CharacterSet	CharacterSetEnum	1	N/A	N/A	N/A	N/A
MarkingStruct	MarkingString	octet	1	[2]		perfect	always
	DummyModulation	any	1			perfect	always
	Psi	float	1	radians		perfect	always
ModulationStruct	Theta	float	1	radians		perfect	always
	Phi	float	1	radians		perfect	always
	Dummy	any	1			perfect	always
ParameterValueTypePositionStruct	X	double	1	metres		perfect	always
	Y	double	1	metres		perfect	always
	Z	double	1	metres		perfect	always
RadioTypeStruct	EntityKind	EntityKindEnum	1	N/A	N/A	N/A	N/A
	Domain	EntityDomainEnum	1	N/A	N/A	N/A	N/A
	Country	EntityCountryEnum	1	N/A	N/A	N/A	N/A
	Category	EntityCategoryEnum	1	N/A	N/A	N/A	N/A
	Subcategory	EntitySubcategoryEnum	1	N/A	N/A	N/A	N/A
	NomenclatureVersion	NomenclatureVersionEnum	1	N/A	N/A	N/A	N/A
	Nomenclature	NomenclatureEnum	1	N/A	N/A	N/A	N/A
	BodyX	float	1	metres		perfect	always
	BodyY	float	1	metres		perfect	always
	BodyZ	float	1	metres		perfect	always
SilentAggregateStruct	AggregateType	EntityTypeStruct	1	N/A	N/A	N/A	N/A
	NumberOfAggregatesOfThisType	unsigned short	1	N/A	1	perfect	always
SilentEntityStruct	EntityType	EntityTypeStruct	1	N/A	N/A	N/A	N/A
	NumberOfEntitiesOfThisType	unsigned short	1	N/A	1	perfect	always
SphericalHarmonicAntennaStruct	Coefficients	float	1+			perfect	always
	Order	unsigned short	1	N/A	1	perfect	always
UnsignedInteger64	ReferenceSystem	ReferenceSystemEnum	1	N/A	N/A	N/A	N/A
	Dummy64	any	1			perfect	always
VariableDatumStruct	DatumID	DatumIDEnum	1	N/A	N/A	N/A	N/A
	DatumLength	unsigned long	1	N/A	1	perfect	always
	DatumValue	any	1			perfect	always
	XVelocity	float	1	m/s		perfect	always
VelocityStruct	YVelocity	float	1	m/s		perfect	always

Complex Datatype	Field Name	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condition
AttributeValueSetStruct	ZVelocity	float	1	m/s		perfect	always
	AttributeSetCount	any	1			perfect	always

Term	Definition
AggregateEntity	An object which is the aggregate of one or more other objects. The objects can be discrete or may be other aggregate objects.
BaseEntity	A base class of all scenario domain participants, both aggregate and discrete. The BaseEntity class is intended to be a container for common attributes for entities of all classes. Since it lacks sufficient class specific attributes that are required for simulation purposes it is not expected that any federate shall publish objects of this class. Certain simulation management federates, e.g. viewers, may subscribe to this class. Simulation federates will normally subscribe to one of the subclasses, to gain the extra information required to properly simulate the entity.
Designator	A designating function, such as a lasing function to support a laser-guided weapon engagement.
EmbeddedSystem	A base class used to associate sensor and emitting systems with their parent entity object.
EmitterBeam	
EmitterSystem	
EnvironmentEntity	
MilitaryEntity	
MilitaryPlatformEntity	
MilitaryAirLandPlatform	A military platform entity that operates mainly in the air, such as aircraft, balloons, etc.. This includes the entities when they are on the ground
MilitaryAmphibiousPlatform	A military platform entity that can operate both on the land and the sea
MilitaryLandPlatform	A military platform entity that operates wholly on the surface of the earth
MilitarySpacePlatform	A military platform entity that operates mainly in space
MilitarySeaSurfacePlatform	A military platform entity that operates wholly on the surface of the sea
MilitarySubmersiblePlatform	A military platform entity that operates either on the surface of the sea, or beneath it
MilitaryMultiDomainPlatform	A military platform entity that operates in more than one domain (excluding those combinations explicitly defined as subclasses of the superclass of this class)
MunitionEntity	
PhysicalEntity	A base class of all discrete platform scenario domain participants.
CivilPlatform	A civilian platform entity
Civilian	A civilian (human)
CivilAirLandPlatform	A civilian platform entity that operates mainly in the air, such as aircraft, balloons, etc.. This includes the entities when they are on the ground
CivilAmphibiousPlatform	A civilian platform entity that can operate both on the land and the sea
CivilLandPlatform	A civilian platform entity that operates wholly on the surface of the earth
CivilSpacePlatform	A civilian platform entity that operates mainly in space
CivilSeaSurfacePlatform	A civilian platform entity that operates wholly on the surface of the sea
CivilSubmersiblePlatform	A civilian platform entity that operates either on the surface of the sea, or beneath it
CivilMultiDomainPlatform	A civilian platform entity that operates in more than one domain (excluding those combinations explicitly defined as subclasses of the superclass of this class)
RadioReceiver	
RadioTransmitter	

Term	Definition
Soldier	
TrackJamBeam	
SimulationManager	The simulation manager object. This object is not part of the virtual world, rather it is used as the initiating object for simulation management purposes. Simulation management federates should instantiate a single instance of this object class.

Term	Definition
ActionRequest	
ActionResult	
AttributeChangeRequest	
AttributeChangeResult	
Collision	
CreateObjectRequest	
CreateObjectResult	
MunitionDetonation	Communicates information associated with the impact or detonation of a munition
RadioSignal	
RemoveObjectRequest	
RemoveObjectResult	
WeaponFire	

Class/Interaction	Term	Definition
AggregateEntity	AggregateMarking	
	AggregateState	
	Dimensions	
	EntityIDs	
	ForceID	
	Formation	
	NumberOfEntities	
	NumberOfSilentAggregates	
	NumberOfSilentEntities	
	NumberOfSubAggregates	
BaseEntity	NumberOfVariableDatums	
	SilentAggregates	
	SilentEntities	
	SubAggregateIDs	
	VariableDatums	
	AccelerationVector	Linear acceleration vector
	AngularVelocityVector	
	DRAAlgorithm	Dead reckoning algorithm used by the issuing object
	EntityType	
	FederateID	
Designator	IsFrozen	
	Orientation	
	Position	
	VelocityVector	
	CodeName	
	DesignatedObject	
	DesignatorCode	
	DesignatorID	
	DesignatorPower	
	DesignatorSpotLocation	
EmbeddedSystem	DesignatorWavelength	
	DRAAlgorithm	
	RelativeSpotLocation	
	SpotLinearAcceleration	
	FederateID	
	HostObjectID	

Class/Interaction	Term	Definition
EmitterBeam	RelativePosition	
	AzimuthCenter	
	AzimuthSweep	
	BeamFunction	
	BeamID	
	BeamParameterIndex	
	ElevationCenter	
	ElevationSweep	
	EmittingSystemID	
	ERP	
	FederateID	
	Frequency	
	FrequencyRange	
	PRF	
	PulseWidth	
	SweepSynch	
	EmitterFunction	
	EmitterName	
EmitterSystem	EmitterNumber	
	BoundingVolume	
EnvironmentEntity	GeometryDefinitionArray	
	NumberOfGeometries	
MilitaryEntity	AlternateEntity Type	
	Camouflage Type	
	FirePowerDisabled	
	ForceID	
	IsConcealed	
	AfterburnerOn	
MilitaryPlatformEntity	HasAmmunitionSupplyCap	
	LauncherRaised	
MunitionEntity PhysicalEntity	LauncherFlashPresent	
	ArticulatedParametersArray	
	ArticulatedParametersCount	
	DamageState	
	EngineSmokeOn	
	FlamesPresent	

Class/Interaction	Term	Definition
	HasFuelSupplyCap	
	HasRecoveryCap	
	HasRepairCap	
	HatchState	
	Immobilized	
	LifeformState	
	LightsState	
	Marking	
	PowerPlantOn	
	RampDeployed	
	SmokePlumePresent	
	TentDeployed	
	TrailState	
	RadioNumber	
RadioReceiver	ReceivedPower	
	ReceivedTransmitter	
	ReceiverState	
	AntennaPattern	
	AntennaPatternType	
	CryptoKeyID	
	CryptoSystem	
	DetailedModulationType	
	Frequency	
	FrequencyBandwidth	
	FrequencyHopInUse	
	InputSource	
	ModulationParameterCount	
	ModulationParameters	
	MajorModulationType	
	ModulationSystemType	
	Power	
	PseudoNoiseSpectrumInUse	
	RadioNumber	
	RadioTransmitterID	
	RadioType	
	TimeHopInUse	

Class/Interaction	Term	Definition
Soldier	TransmitState	
	PrimaryWeaponState	
	SecondaryWeaponState	
	JammingModeSequence	
	NumberOfTrackJamObjects	
TrackJamBeam	TrackJamObjectIDs	
	TrackOrJammer	
	SimulationManagerName	
	ObjectCount	
	ObjectIDs	
SimulationManager	Action	
	ActionResult	
	ObjectCount	
	ObjectIDs	
	AttributeValueSet	
AttributeChangeRequest	ObjectID	
	AttributeChangeResult	
	AttributeValueSet	
	CollidingObjectID	
	CollidingObjectMass	
Collision	CollidingObjectVelocity	
	CollisionType	
	CollisionLocation	
	EventID	
	IssuingObjectID	
CreateObjectRequest	ObjectClass	
	AttributeValueSet	
	CreateObjectResult	
	ArticulatedPartsArray	
	ArticulatedPartsCount	
CreateObjectResult	DetonationLocation	
	DetonationResult	
	EventID	
	FiringObjectID	
	FinalVelocityVector	
MunitionDetonation	FuseType	

Class/Interaction	Term	Definition
RadioSignal	MunitionObjectID	
	MunitionType	
	QuantityFired	
	RateOfFire	
	RelativeDetonationLocation	
	TargetObjectID	
	WarheadType	
	EncodingScheme	
	HostRadioID	
	SampleCount	
RemoveObjectRequest	SampleRate	
	SignalData	
	SignalDataLength	
	TacticalDataLinkType	
	ObjectCount	
RemoveObjectResult WeaponFire	ObjectIDs	
	RemoveObjectResult	
	EventID	
	FireControlSolutionRange	
	FireMissionIndex	
	FiringLocation	
	FiringObjectID	
	FuseType	
	InitialVelocityVector	
	MunitionObjectID	
	MunitionType	
	QuantityFired	
	RateOfFire	
	TargetObjectID	
	WarheadType	

RPR-FOM
Version 0.1.7

ID	Text
1	The choice of whether an entity is a physical entity rather than an environmental entity, boils down to whether the entity has a fixed size and shape (i.e. can simulations which need a model of the entity as seen by a sensor, for instance the human eyeball, perform a look up into a model database). If the entity does have a fixed size and shape then it is a PhysicalEntity. If it doesn't, then further information is required to define its size and shape and it is an environment entity. Articulated and attached parts should not be taken into account when determining whether the entity has a fixed size. The terms Physical and Environment are not exclusive. They have been chosen because they best describe the majority of entities that they contain.
2	The units for the MarkingString are specified by the value of the CharacterSet.
3	The AttributeChangeResult interaction should be sent in response to an AttributeChangeRequest interaction by any object that is mentioned in the associated object list. The AttributeChangeResult parameter should indicate the success or failure of the object to implement the requested changes. If the object can implement the changes then the AttributeChangeResult parameter should be set to successful and the AttributeValueSet shall be empty. If the object cannot immediately comply with the request, but is likely to be able to comply in the future then the AttributeChangeRequest parameter shall be set to pending and the AttributeValueSet shall be empty. When the object complies with the request (or fails to implement the request) then another AttributeChangeResult interaction shall be sent informing the initiating federate of the result of the request. If the object cannot comply with the request then the AttributeChangeResult parameter shall indicate the reason why the object cannot comply with the request. The AttributeValueSet shall contain the attributes which have been not been set and the current values of those attributes.
4	Or any other publishable class
5	This is an RTI Object ID

REPORT DOCUMENTATION PAGE

(MOD-NL)

1. DEFENCE REPORT NO (MOD-NL) TD97-0394	2. RECIPIENT'S ACCESSION NO A95KL841	3. PERFORMING ORGANIZATION REPORT NO FEL-98-A014
4. PROJECT/TASK/WORK UNIT NO 25897	5. CONTRACT NO A95KL841	6. REPORT DATE February 1998
7. NUMBER OF PAGES 68 (incl 1 appendix, excl RDP & distribution list)	8. NUMBER OF REFERENCES 6	9. TYPE OF REPORT AND DATES COVERED
10. TITLE AND SUBTITLE Gedistribueerde Interactieve Simulatie: Van DIS naar HLA (Distributed Interactive Simulation: From DIS to HLA)		
11. AUTHOR(S) R.J.D. Elias W.G. de Jong		
12. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) TNO Physics and Electronics Laboratory, PO Box 96864, 2509 JG The Hague, The Netherlands Oude Waalsdorperweg 63, The Hague, The Netherlands		
13. SPONSORING AGENCY NAME(S) AND ADDRESS(ES) RNLA/DMKL/T&WO, Van der Burchlaan 31, 2597 PC The Hague, The Netherlands		
14. SUPPLEMENTARY NOTES The classification designation Ongerubriceerd is equivalent to Unclassified, Stg. Confidentieel is equivalent to Confidential and Stg. Geheim is equivalent to Secret.		
15. ABSTRACT (MAXIMUM 200 WORDS (1044 BYTE)) In this report an outline is given of the ongoing developments of distributed simulation and its standardisation for interoperability of simulations. It highlights the High Level Architecture (HLA) initiative of the DoD DMSO, with its base module, the Run-Time Infrastructure (RTI), which provides for communication between simulation applications. It also discusses the Real-time Platform-level Reference Federation Object Model (RPR-FOM), a proposal for a data standardisation for DIS-like, real-time, human-in-the-loop physical platform simulation in HLA. Finally the Advanced Simulation Framework (ASF) is presented. The ASF is a software infrastructure for the development of distributed simulation applications and provides for a smooth transition from DIS to HLA by shielding the application developer from the underlying distributed simulation standards through a generic application programmer's interface.		
16. DESCRIPTORS Distributed Simulation Standards Interoperability		IDENTIFIERS High Level Architecture Run-Time Infrastructure Real-time Platform-level Reference FOM Distributed Interactive Simulation Protocol Data Units
17a. SECURITY CLASSIFICATION (OF REPORT) Ongerubriceerd	17b. SECURITY CLASSIFICATION (OF PAGE) Ongerubriceerd	17c. SECURITY CLASSIFICATION (OF ABSTRACT) Ongerubriceerd
18. DISTRIBUTION AVAILABILITY STATEMENT Unlimited Distribution		17d. SECURITY CLASSIFICATION (OF TITLES) Ongerubriceerd

Distributielijst

1. DWOO
2. HWO-KM*
3. HWO-KL
4. HWO-KLu*
5. HWO-CO*
6. DS/CZ, t.a.v. Ir. J.B.J. Orbons
7. DMKL/B&B/T&WO, t.a.v. Ing. F.J. Remmerswaal
8. DM&P TNO-DO
9. Directie TNO-PML*
10. Directie TNO-TM*
- 11 t/m 13. Bibliotheek KMA
14. DMKL/INFO/SIM, t.a.v. Ing. J.J. Smit
15. DMKL/INFO/SIM, t.a.v. Ing. R.W.G. Struiwig
16. Cdt OC Ede, t.a.v. Kol. N.P. le Grand
17. LAS/DB&P/BO/OB, t.a.v. LKol. M.C. de Kruif, vz. ACGOLM
18. LAS/DB&P/BO/OB, t.a.v. Maj. B. Sundquest
19. NATCO/WBU, t.a.v. C.J. Zuurveld
20. DMKM/pHWO-KM, t.a.v. Ir. A. de Jong
21. TNO-TM, t.a.v. Prof.dr.ir. H. Schuffel
22. Directie TNO-FEL, t.a.v. Dr. J.W. Maas
23. Directie TNO-FEL, t.a.v. Ir. J.A. Vogel, daarna reserve
24. Archief TNO-FEL, in bruikleen aan MPC*
25. Archief TNO-FEL, in bruikleen aan Ir. M.J. van de Scheur
26. Archief TNO-FEL, in bruikleen aan Dr. G.J. Jense
27. Archief TNO-FEL, in bruikleen aan Ir. R.C. van Rijnsoever
28. Archief TNO-FEL, in bruikleen aan Dr. H.F.R. Arciszewski
29. Archief TNO-FEL, in bruikleen aan Ing. D.J. Coetsier
30. Archief TNO-FEL, in bruikleen aan Ir. P.C.A. van Gool
31. Archief TNO-FEL, in bruikleen aan Drs. R.G.W. Gouweleeuw
32. Archief TNO-FEL, in bruikleen aan Ir. W. Huiskamp
33. Archief TNO-FEL, in bruikleen aan Ir. N.H.L. Kuijpers
34. Archief TNO-FEL, in bruikleen aan Ir. H.A.M. Luijff
35. Archief TNO-FEL, in bruikleen aan Ir. W.G. de Jong
36. Archief TNO-FEL, in bruikleen aan Drs. R.J. Elias
37. Documentatie TNO-FEL
- 38 t/m 42. Reserve

Indien binnen de krijgsmacht extra exemplaren van dit rapport worden gewenst door personen of instanties die niet op de verzendlijst voorkomen, dan dienen deze aangevraagd te worden bij het betreffende Hoofd Wetenschappelijk Onderzoek of, indien het een K-opdracht betreft, bij de Directeur Wetenschappelijk Onderzoek en Ontwikkeling.

* Beperkt rapport (titelblad, managementuitreksel, RDP en distributielijst).