Interoperability and harmonization analysis of the German Maritime Federation Object Model (GMF) in relation to SISO RPR FOM with possible extensions in a multinational forum

Kay Roos  
Modeling and Simulation Commissary  
German Navy  
German Navy Headquarters  
Kopernikusstraße 1  
18057 Rostock  
+49-(0) 381 – 882-52883  
Rostock, Germany  
kayroos@bundeswehr.org

Erik Solum  
Kongsberg Defence & Aerospace  
Kirkegårdsveien 45  
3616 Kongsberg, Norway  
+47 93 24 22 77  
erik.solum@kongsberg.com

Karsten Horn  
Commander  
Naval Warfare School  
Flådestationen 1  
9900 Frederikshavn  
+45 7285 5391  
khorn@mil.dk

Nathan Newton  
DSTL  
Porton Down  
Salisbury, United Kingdom  
+ 44 1980 658644  
nnewton@dsti.gov.uk

Jeppe Nyløkke, Brian Horn, IFAD TS A/S  
Østre Stationsvej 43  
5000 Odense C, Denmark  
+45 63 11 02 11  
jeppe.nylokkke@ifad.dk, brian.horn@ifad.dk

Niels Krarup-Hansen  
Danish Acquisition and Logistics Organization (DALO)  
Laurupbjerg 1-5  
2750 Ballerup, Denmark  
+45 7257 1671  
nkh@mil.dk

Simon Morris  
SE Section Lead  
Thales UK  
Manor Royal, Crawley  
West Sussex, UK  
RH10 9HA  
+44 1293 589476  
simon.morris@thalesgroup.com

Jeppe Nyløkke, Brian Horn, IFAD TS A/S  
Østre Stationsvej 43  
5000 Odense C, Denmark  
+45 63 11 02 11  
jeppe.nylokkke@ifad.dk, brian.horn@ifad.dk

Niels Krarup-Hansen  
Danish Acquisition and Logistics Organization (DALO)  
Laurupbjerg 1-5  
2750 Ballerup, Denmark  
+45 7257 1671  
nkh@mil.dk

Simon Morris  
SE Section Lead  
Thales UK  
Manor Royal, Crawley  
West Sussex, UK  
RH10 9HA  
+44 1293 589476  
simon.morris@thalesgroup.com

Keywords: HLA, RPR, FOM, Maritime

ABSTRACT: The German Navy has developed a German Navy Maritime Federation Object Model (GMF) with the goal to provide interoperable simulation capabilities for the German Naval Command and Control Systems. The GMF was developed from identified operational task requirements that have a need for simulation, using the most recent HLA 1516-2010 and based on the SISO RPR FOM. In parallel with this an international work focusing on possible extensions and improvements to the PRP FOM is taking place under the NATO Modelling and Simulation Group structure. The German Navy has chosen to bring the GMF to such a group in order to contribute to and strengthen maritime aspects of further FOM development and do this in coordination with ongoing extension and improvements initiatives.

This paper describes the work done and results achieved in the NATO MSG-106 group in relation to the GMF. In overall, the MSG-106 group deals with extensions to the RPR FOM that are related to NATO simulation issues and are collected in the NATO Educational Training Network FOM (NETN FOM) that builds on top of the RPR FOM. The first step for the GMF was an analysis of the list of tasks that the GMF supports in order to verify the general usefulness in a wider context than the national German level. This work was performed by a group led by operational users and supported by technical members. Four nations were supporting this activity. The next step was a technical analysis of the different GMF modules performed by a dedicated maritime sub-group in the project and supported by 5 nations.
The analysis output was a number of different recommendations for each GMF module. To maintain a clear division of responsibility it was not a purpose to do any changes to the GMF except some special conditions were fulfilled. The recommendations were two-fold: to the GMF owner and consisting of recommended changes to the GMF and recommendations that should be further delegated to SISO and to the NETN community addressing how to make NETN FOM and GMF compatible and interoperable.

The analysis of the German task list confirmed the general usefulness in a more international context. The overall conclusion of the technical analysis of the GMF was that the GMF was a good first step in the maritime field and that there is an interest in the field outside Germany. However, lack of documentation will prevent a wider use. Most of the modelling constructs are of such relevance that the group recommends incorporation in the RPR FOM after identified issues have been resolved. These issues are both on the GMF and RPR FOM. Until such full solutions are reached temporary solutions are also proposed.

1 Introduction

1.1 Background and Previous Work

The German Maritime FOM (GMF) was presented to the SISO community for the SISO SIW spring 2012 in the paper “A Multi-Faceted Approach to the Development of the HLA 1516-2010 German Maritime Federation Object Model (GMF)” [1]. The paper describes the task-driven approach to identify the simulation needs that in turn leads to the development of the GMF. Although, one of the goals was to achieve interoperability with other, non-German federations this was never tried in a wider context within the original project.

In NATO a NATO Educational and Training Network (NETN) was initiated with the purpose to establish a persistent, joint NETN capability by leveraging existing capabilities. An initial technical solution was developed in the NATO Modeling and Simulation Group (NMSG) MSG-068. For a closer link to operational support requirements and to establish a long term maintenance process for the initial NETN architecture, the MSG-106 was established and is currently in its ending phase [2].

Both the GMF and NETN are developed for HLA 1516 [3][4][5] and utilizing the SISO RPR FOM v2 [6]. With this in mind it seemed to be a relevant to bring the two together to investigate interoperability issues. This would add the extra value for international context for the GMF and the NETN would gain added operational value within the maritime context.

The German Navy then made the decision to bring in the GMF in the MSG-106 project [7].

1.2 Approach

When the interest for the GMF was verified among the nations participating in the MSG-106 project the approach taken to deal with the GMF was two-fold. The first step for the GMF was an analysis of the list of tasks that the GMF supports in order to verify the general usefulness in a wider context than the national German level. This work was performed by a group led by operational users and supported by technical members. Four nations, namely Germany, Denmark, France and Norway were supporting this activity. The next step was a technical analysis of the different GMF modules performed by a dedicated maritime sub-group in the project and supported by 5 nations, namely Denmark, Germany, Norway, England and France.

1.3 Maritime User Requirements

The purpose of the activity in the first step was to identify common operational tasks between nations for the maritime domain. To achieve this, a group led by operational users and supported by technical members was established. The main input to the work was the German Navy maritime Task List which contained tasks which are executed by navy within the maritime domain (sea, air and land). This list was originally used to create the GMF v1.1. Operational maritime members of the group from Denmark analyzed this Task List to determine the extent it meets the Danish Navy needs. Finally, the German and Danish members agreed on desired and/or necessary changes and implemented these in the Task List [8]. The activity concluded with a recommendation to use similar Task List for any other military service for the further development of the NETN FOM and a recommendation to establish a technical group in the MSG-106 project to conduct a technical analysis of the GMF both with respect to the agreed Task List and compatibility with the current version of the NETN FOM [9].

1.4 Technical Analysis

The technical maritime subgroup was established with the main objectives to provide recommendations for including the GMF modules in the NETN FOM and to provide recommendations for improving the GMF.

To achieve this, the analysis of each module in the GMF was distributed among the active participants in the group.

The following issues were taken into account:
• Initial, overall analysis – to identify any immediate and obvious issues
• Effect on module of change from RPR FOM v17 to v19
• Identify possible conflicts in integration with NETN, especially focus on inheritance but also on transfer of responsibility and aggregation
• Identify need for technical test if analysis isn’t sufficient.
• Identify need for technical support from GMF developers, in order to clarify design decisions.
• Identify need for test for operational need if analysis isn’t sufficient.
• Consider if a temporary solution can be recommended until full solution is reached.
• Review corresponding GMF GRIM documentation and identify changes.

Next issue to clarify was how to provide recommendations more specifically. It was agreed that the best approach would be to keep a clear division of responsibility for potential changes to the GMF. As the responsibility of the NETN FOM beyond the MSG-106 project is unclear, the best approach seemed to be to let the GMF responsibility stay on the German side with the possibility to further delegate the responsibility to SISO as a proposed standard. Based on this the following recommendation scheme was agreed:

• Recommendations to the GMF FOM owner should address both recommended changes to the GMF FOM and recommendations that should be delegated to SISO.

• Recommendations to NETN should address how to make the NETN FOM and GMF FOM compatible and how to make them interoperable.

The NETN recommendations should be described in the Federation Architecture and FOM Design (FAFD) [13] documentation while the GMF FOM recommendations should be part of the MSG-106 final report [10].

1.5 GMF Modules
The following GMF modules were analyzed:
• Acoustics module
• Sonobouy module
• Emitter module
• Transponder module
• Comm module
• METOC module
• Interchange base module

2 Acoustics module
The FOM module consists of a single Interaction class called AcousticPing which models a sonar ping.

2.1 Attributes
The attributes of the ping are fairly simple, they define which entity it originated from, the time it was sent, and the type of ping. The attributes seem to be a good match to the data that needs to be modelled for a ping event, with the exception of the lack of a geographical origin of the ping. It is possible to deduce the location from the position of the referenced host entity, but that seems unnecessarily cumbersome.

GMF has chosen to model the initial emission of the ping as an event instead of modelling the ping itself travelling through the water. Since it is complex to model the actual irregular sphere of the ping sound travelling through water, modelling the initial emission seems like a good approach. This allows federates to get access to the information that they need to deduce the data themselves, based on their level of fidelity and capability.

![Figure 1: AcousticPing interaction](image)

The sonar ping type is modelled as a single enumeration, combining several disparate properties. This reduces the flexibility of the class, and in combination with the inadequate semantics, makes it difficult to synchronize the expected behaviour between federates.

There are two possible resolutions to this. The sonar mode could remain as a simpler enumeration (CW (Continuous Wave), FM (Frequency Modulated),..), but change the secondary properties to more suitable types, such as an integer to hold milliseconds for the CW time property.

Alternatively, replace the coarse sonar property modelling enforced by this enumeration with a reference to the issuing RPR Underwater Acoustics (RPR-UA) v2.0 ActiveSonar and ActiveSonarBeam. Even though these RPR object classes are fairly limited in their modelling, they are more detailed than the enumeration, and would create a more consistent way to model sonar and sonic properties.
A sonar ping is a form of acoustic transient, which are defined in the RPR-UA FOM.

The Ping interaction class should thus inherit from the RP-UA FOM Object class. This would allow passive sonars to detect sonar pings in the same way as they detect other Acoustic transients.

A sonar ping is an instance/product of an active sonar beam, which is also defined in the RPR-UA FOM, but is not referenced by the GMF ping interaction.

It might be a good idea to provide a reference to an ActiveSonarBeam instance that produced the ping. This object class is defined in RPR-UAv2.0 and would provide a means for the receiver of the event to extract more information about the acoustic properties of the ping from the beam object.

To SISO:
It is worth bearing in mind that the SISO RPR FOM ActiveSonarBeam object is poorly designed, references a non-standard database, and should be revised.

2.2 Recommendations
For the GMF owner:
- Let the Ping interaction class inherit the RPR_UA AcousticTransient interaction class.
- Reuse the inherited ActivityCode and ActivityParameter to model the characteristics of the Ping.
- Add an attribute referencing the RPR_UA ActiveSonarBeam that produced the ping. This will allow other RPR federates to “see” the Pings as general acoustic transients.

To SISO:
- There is a need for a common way of representing Ping events across a federation in order for passive intercept sonars to correctly detect the usage of active sonars.
- Work with the GMF authority to extend the RPR-UA ActiveSonarBeam with an attribute to hold its acoustic frequency.

This will make it possible to model the ping events frequency through its reference to the ActiveSonarBeam that created it

Work with the GMF to incorporate the GMF Ping interaction class, after above mentioned modifications are implemented, into the RPR-UA standard

To NETN federates in order to support GMF_Acoustic v1.1 federates:
- Create a NETN/GMF bridge that maps between GMF Ping interactions and RPR_UA Transient interactions.
- Alternatively the NETN federates can load the GMF Acoustic module and use its Ping definition as it has no direct conflicts with NETN or RPR, but this solution will limit the NETN federates ability to display to correctly react and present the ping (as it will not be a proper acoustic transient).

3 Sonobuoy
The GMF_Sonobuoy module consists of a sonobuoy object class and two interaction classes with a common super-class for communication with the sonobuoy.

There is a real need for a way for federates to collaborate on the use of sonobuoys and the interaction objects seem to cover the need for communication between other entities and the sonobuoy well.

The Sonobuoy object class does a good job of providing a simple, but adequate, modelling of the sonobuoy's radio link and sonobuoy-specific properties, but seems to be lacking in the modelling of the attached sonar - as it only provide a very coarse sonar model. A recommendation for version 2 of the GMF would be to replace the sonar property attributes in the Sonobuoy object with references to instances of the RPR-UA v2.0 ActiveSonar and ActiveSonarBeam, which even though they are also poorly modeled and use non-standard databases, model the sonar properties much better. This would also provide a consistency to how sonar and sonar beams are modelled.
3.1 Attributes

The attributes of the sonobuoy command interaction models the radio channel for the communication, the command that is sent, and an associated parameter value for the command.

The attributes of the Sonobuoy object model the radio channel it listens for commands on, the number of ping charges left, and coarsely the sonar object itself.

The command type attribute in the CommandSignalGenerator interaction class is modeled as a single enumeration, a duplicate implementation of the AcousticPingType enumeration in the GMF Acoustics module. As in the Acoustics module it combines several properties different aspects of the sonar into one enumeration. This reduces the flexibility of the class, and combined with the lack of semantics, makes it difficult to synchronize the expected behaviour between federates.

![Figure 4: AcousticPingType enumeration](image)

![Figure 5: SonobuoyCSGTypeEnum8](image)

3.2 Recommendations

To the GMF owner:

- Make the module compatible with the RPR standard way of modeling radios and radio communication
  - Let the Sonobuoy object reference the RPR-Communication FOM’s RadioTransmitter and RadioReceiver object that represents the sonobuoy’s radio capability. Thus making it possible for other federates to threat the radio on the sonobuoy in the same way as other radios.
  - Let the SonobuoyCommand interaction class from inherit from the RPR-Communication_v2.0 RawBinaryRadioSignal interaction class. Thus enabling other federates the process and threat the radio emissions in the same way as all other radio emissions are threatened.

To SISO:

- Make the module compatible with the RPR standard way of modeling radios and radio communication
  - Work with GMF authority to incorporate the updated GMF_Sonobuoy object class into the RPR-UA standard.
  - Work with GMF authority to incorporate the two updated GMF_Sonobuoy interaction classes into the RPR-UA standard.

4 Emitter module

The GMF_Emitter module consists of a single Object class that extends the RPR Distributed Emission Regeneration (RPR-DER) v2.0 RadarBeam class. It models additional electromagnetic emission attributes for radar beams in order to support Electronic Support Measures (ESM) simulation. With these extensions ESM operators have the information needed for radar beam identification. The attributes, data types and enumerations used in the SupplementalEE class seems to match well with our experience of the needs of ESM simulation. Attributes such as ScanPeriod, Jitter, Sidelobes, scan and pulse type, are essential for federates to treat emitted radar beams correctly with their respective ESM simulations.

To SISO:

- There is a real need for a standard on the exchange of ESM data. Federations with multiple federates that emit electro-magnetic signals, such as radars, radios,
jammers etc., are very common and there is thus an obvious need for a standard on how that data should be modelled so federates are able to use that data in their ESM simulations.

4.1 Attributes
The attributes in the SupplementalEE Object class all model the properties of a radar beam to a higher fidelity than what is available through RPR FOM v2.0 [6].

4.2 Recommendations
The GMF_Emitter module is a well-designed module that could fill that role. Its only limitation is that it only extends the RadarBeam class, while all other electro-magnetic emitters such as jammer beams, radios and IFF, do not get extended, even though they also have many of these properties. The correct software engineering way to do this would have been to extend the EmitterBeam super class instead of the RadarBeam class, but as this is not possible, each of these classes should get their own extension

Work with SISO to include the SupplementalEE attributes in the RPR_DER module
• The attributes in the current GMF_Emitter object class SupplementalEE that are common to all electro-magnetic emitters, such as radio, jammers, transponders, should be added to the common RPR_DER superclass EmitterBeam.
• The attributes that are specific to radars, and not shared with other emitters, should be included into the RPR_DER RadarBeam object class.

5 Transponder module
As far as the Tacan object class is concerned, it is definitely a suitable implementation for modelling simple TACAN systems.

The GMF_Transponder module consists of a single object class Tacan, derived from the RPR2 EmbeddedSystem (RPR2-Base) class, as well as five enumeration data types used within the class attributes.

5.1 Attributes
As can be seen in Figure 8, the system type, mode, channel, range, station ID and operational state attributes exist within the class, which along with positional data from the EmbeddedSystem base class, enable modelling of a TACAN system. There is no documentation within the GMF GRIM or the GMF FOM semantics explaining operational use. It is assumed the receiver is meant to calculate the range and bearing themselves using their position relative to the TACAN location. However, it is not obvious how the (lack of) precision of the various TACAN systems should be modelled, if this is required. One way would
be for the receiver to have complete knowledge of the capability of every system it can communicate with, though this does not feel very scalable.

The number of system types specified in the TacanSystemTypeEnum32 enumeration, used for the TacanType attribute, may not be adequate to model non-naval TACAN systems. It does have a value of UNKNOWN, which could be used when a system that is not covered by the other four values is being modelled. Further clarity is needed as to how aware of the TACAN system type a receiver needs to be before it is able interact with it.

The TacanMode attribute allows the receiving federate to determine whether it is appropriate for it to process the object. The SensorsTacanModeEnum32 enumeration provides support for Air-to-Air functionality as well, which means all known TACAN modes are supported except those that only transmit to “on channel” receivers.

The TacanXYChannelSelect attribute provides facility for the receiver to know whether the TACAN system is transmitting / receiving on channel set X or Y.

As mentioned previously, the TacanRange attribute is of type SensorsTacanRangeEnum32, which appears to be restrictive. It is very likely that TACAN systems exist that do not match the small set of enumerated values provided. It is recommended that a numerical type such as Float32 (RPR2-Base) is used instead. In addition, the units for the range are not specified in the semantics. It is assumed that the enumeration value SENSORS_TACAN_RANGE_UNK would be specified for systems that do not have a range of 75, 100 or 300, but this is of no use unless the model is very simple.

The TacanChannel attribute is a 16bit integer field, so both transmitting and receiving federates need to have the same database of channel frequencies. This could be resolved by having a frequency attribute in addition to (or instead of) the TacanChannel attribute. The channel number is very likely required for user-interface modelling, so the best solution would be to add a new attribute for the frequency, as well as maintaining the existing TacanChannel attribute.

The attribute TacanStationID provides a facility to specify a TACAN system's identification, as a character array.

5.2 Recommendations
To the GMF owner:

- If the system type affects receipt, then the TacanSystemTypeEnum32 enumeration may need to be extended to provide support for more kinds of TACAN system.
- “On channel” support should be considered, either as a new operational state in the TacanState enumeration, or as a new attribute.
- The TacanRange attribute should be changed to a floating point type, such as Float32 (RPR2-Base), to remove the restriction of only having three ranges. The unknown range value (SENSORS_TACAN_RANGE_UNK) within the SensorsTacanRangeEnum32 enumeration is of little use unless the modelling is very simple.
- Add an additional attribute to provide frequency information to the TACan object class.
- Replace the use of GMF_Interchange types with their equivalents from RPR2-Base. If this is done for all GMF FOMs then the GMF_Interchange FOM would no longer be required.
- Every attribute in the TACan object class has HyperSpace_subspace, HyperSpace_one and HyperSpace_two dimensions, when none of them should have any.

To SISO:
The GMF_Transponder FOM is a good definition for use when modelling TACAN systems. If the recommendations outlined here are implemented, it is felt that simulation standards would benefit from the inclusion of the GMF_Transponder FOM.

6 COMM module
The GMF Comm module models Tactical Data Link communication as Link 11, Link 16 and Hawk Link.

Figure 9: Link 11 interactions
The Link 11 and Link 16 interaction classes are derived from the RawBinarySignal interaction; all other classes are inherited from the HLAInteractionRoot or HLAObjectRoot classes.

6.1 Findings
The implementation of Link 11/16 found in the GMF_Comm module does not break the RPR2 [6]. The SISO versions of the Link 11/16 BOMs provided in [11] and [12], both modify the RPR2 by adding attributes to existing RPR2 data types. For HawkLink no corresponding BOM exists.

6.2 Recommendations
To SISO:
- Revise/update the BOMs for Link 11 and Link 16 to the approach used in the GMF, as this approach ensures compatibility and interoperability with the RPR2, without modifying the FOM.
- Currently, the RPR2 FOM does not contain modules, objects, or interaction classes modelling HawkLink communication. Additionally, SISO does not seem to hold any references to BOMs specifying how to implement HawkLink. Therefore, SISO should look into adding HawkLink, either as an HLA Evolved module using the RPR2 FOM, or as a separate BOM.

7 METOC module
The GMF Meteorology and Oceanography module is a useful module with no counterpart in RPR or NETN FOM and with a wider interest also beyond maritime.

7.1 Recommendations
To the GMF owner:
- Remove weather server specific interactions
- The RPR location convention should be followed, using WoldLocationStruct geocentric coordinate system.
- The array data types require some refactoring.
- Dimensions should be removed.

8 Interchange module
The GMF_Interchange module defines some simple datatypes and some array datatypes.

It was agreed that the GMF_Interchange FOM should be checked to ensure that it had binary compatibility with NETN and RPR2, to ensure the GMF could be used in its current form within a NETN/RPR2 federation.

8.1 Binary Compatibility Review
It was discovered that not all of the primitive types defined in HLA-Evolved have an equivalent big-endian type in the RPR2-Base FOM. This is probably due to them not being used by RPR2 FOMs, but this is a significant deficiency in terms of extensibility.

Importantly, several of the GMF FOMs use types that do not have the RPR2-Base equivalents. This means that the advice for some of the GMF FOMs to use RPR2-Base types, in place of those defined in the GMF_Interchange FOM, cannot be followed.

Figure 10: Link 16 interactions

Figure 11: Interchange datatypes
The table below summaries the findings of the binary compatibility reviews, listing the problematic GMF_Interchange types,

<table>
<thead>
<tr>
<th>GMF_Interchange type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMFInteger64</td>
<td>No equivalent type exists in RPR2 for this. The GMF_Comm FOM uses this type.</td>
</tr>
<tr>
<td>GMFUnsignedShort</td>
<td>GMF_Interchange type derives from RPR2D17 UnsignedShort, which has been replaced by UnsignedInteger16 in D19. If the latest RPR2 FOM was to be used, this type would need to be derived from UnsignedInteger16.</td>
</tr>
<tr>
<td>GMFString</td>
<td>GMFString is actually an array of (HLAASCIIstring) strings, not an array of characters like the NullTerminatedASCIIstring. The GMF_Comm and SuTBw FOMs use this type.</td>
</tr>
<tr>
<td>GMFbooleanArray</td>
<td>No equivalent type exists in RPR2 for this. The GMF_Sonobuoy FOM uses this type.</td>
</tr>
<tr>
<td>GMFUnsignedShortArray</td>
<td>The base type for UnsignedInteger16Array1Plus is correct, however the array cannot be empty so this is not compatible with GMFUnsignedShortArray. The GMF_Sonobuoy FOM uses this type.</td>
</tr>
</tbody>
</table>

Table 1: GMF types

8.2 Recommendations

To the GMF owner:
- Several of the types defined in the GMF Interchange FOM are not used in any of the other GMF FOMs. These should be removed.
- There appear to be errors in some GMF Interchange types, which require correction before it can be declared to be binary compatible with the NETN FOMs.

Proposed updates to the Interchange module:
- Remove types from GMF_Interchange that have equivalents in RPR2-Base, and change the types appropriately in other GMF FOMs.
- Remove types from GMF_Interchange not used in other GMF FOMS.
- GMFInteger64: No equivalent type exists in RPR2-Base. Used by GMF_Comm FOM. Should remain in GMF_Interchange FOM.
- GMFString: Appears to be incorrectly defined as an array of strings. Used by GMF_Comm and SuTBw FOMs. RPR2-Base type NullTerminatedASCIIstring should be used instead, though this would require code changes.
- GMFBooleanArray, GMFUnsignedShortArray: No equivalent type exists in RPR2-Base. Used by GMF_Sonobuoy FOM. Should remain in GMF_Interchange FOM.

Recommendations to SISO:
- Basic and array data types used by GMF FOMs should be implemented in the RPR2-Base FOM. For proper extensibility, all basic and array types should exist.

9 Conclusion and Future Developments

During the MSG-106 project the German Navy Task List was analyzed by an operational led maritime group. The analysis of the task list confirmed the general usefulness in a more international context. After that a technical analysis of the GMF was performed by a technical maritime subgroup. The overall conclusion of the technical analysis of the GMF was that the GMF was a good first step in the maritime field and that there is an interest in the field outside Germany. However, lack of documentation will prevent a wider use. Most of the modelling constructs are of such relevance that the group recommends incorporation in the SISO RPR FOM after identified issues have been resolved. These issues are both on the GMF and RPR FOM. Until such full solutions are reached temporary solutions are also proposed.
10 References


[7] MSG-106 Minutes of meeting, Norfolk, USA, 29th May – 1st June 2012


11 Author Biographies

KAY ROOS is the Modeling and Simulation Auth. Rep. for the German Navy and is appointed to the German Navy Headquarters Plans & Policy Directory’s Methodology Branch. He has had a long distinguished career in the navy with leadership roles including being executive officer on the F124 frigate HAMBURG and key staff officer for the DEU flotilla 2 responsible for its Policies and Net Centric Warfare which included being the executive director German forces participation in large scale Joint and Combined NetOpFue Exercises and being the liaison for German participation in the US Navy Fleet Synthetic Training. Kay was the Project Lead for the GMF development and rendered support by providing the high level requirements, workshops, conferences, and data collection.

KARSTEN HORN is a Commander at the Danish Naval Warfare School in Frederikshavn.

JEPPÉ NYLØKKE has an MSc in Computer Science and is a Program Manager at IFAD TS A/S in Odense, Denmark. He has 20 years’ experience in software development and architecture, project management and quality assurance. He is project manager for a number of simulation software projects supporting the Danish Army and the Danish Navy.

BRIAN HORN is a Systems and Software Interoperability Engineer at IFAD TS A/S in Odense, Denmark. Brian is involved in multiple engineering and software activities in the area of modeling and simulation projects supporting the Danish Army and the Danish Navy.

SIMON MORRIS is the Section Lead for the Synthetic Environment Team within the Avionics Training Solutions division of Thales UK. Simon has over 10 years experience working as a principal software engineer on military training systems. He now plays a lead role in modelling and simulation research projects, as well as being involved in standards development on behalf of the UK Ministry of Defence.

ERIK SOLUM has a BEng. from HiBu, Norway, and a BSc and MSc from WSU, USA and is currently the System Architect for KONGSBERG Defence System’s (KDS) Simulation & Training (S&T) products where he is responsible for the common architecture for the products.

NATHAN NEWTON is a software engineer working in Modelling and Simulation for the UK MOD Dstl for
the past 6 years. During this time he has primarily worked on the development of Synthetic Environments for CBR Modelling and Simulation and the development of CBR interoperability standards.

**NIELS KRARUP-HANSEN** is Senior Advisor in the Danish Defence, member and former chairman of NATO Modelling and Simulation Group, chaired the updating of NATO M&S Master Plan, carried out many national and NATO studies, attended NATO Defense College Senior Course, was manager of and project coordinator in Danish Minister of Defence Advisory and Analysis Group etc.