21$^{\text{ST}}$ ICCRTS

21st International Command and Control Research and Technology Symposium

C2 in a Complex Connected Battlespace

September 6 – 8, 2016

London, UK

Topic: Interoperability/Integration and Security

Title: MIP Information Model 4.0 –
Semantic Interoperability in Multinational Missions

Authors:

Names: **Michael Gerz (POC)**
Nico Bau

Organization: Fraunhofer FKIE

Address: Fraunhoferstraße 20
53343 Wachtberg-Werthhoven, Germany

Phone: +49 228 9435 414

E-Mail: {michael.gerz | nico.bau}@fkie.fraunhofer.de

MIP Information Model 4.0
Semantic Interoperability in Multinational Missions

Michael Gerz, Nico Bau

Abstract

Semantic interoperability in combined and joint operations requires a commonly accepted standard that clearly specifies the information elements exchanged between the participating partners. The *Multilateral Interoperability Programme* (*MIP*) is developing the *MIP Information Model* (*MIM*) as a semantic reference for C2 information exchange. The model defines detailed taxonomies for key concepts such as objects and actions as well as the relationships between them. The 24 MIP member nations, NATO, and EDA continuously enhance the model. The latest version, MIM 4.0.1, has been released in July 2016. It builds the foundation for the MIP 4 Information Exchange Specification (IES).

In this paper, we start with a brief summary of the concepts of the MIP Information Model. This includes the modeling approach to capture semantic information and an overview of the tools and services complementing the model. Thereafter, we highlight enhancements introduced in the MIM 4.0. The latest model features a revised taxonomy of objects, especially for military units. The MIM 4.0 also applies a generic approach to deal with unknown information. Finally, we discuss various techniques for how Communities of Interest (COIs) may build their own interoperability solution based on the MIM and the tools that accompany the model. We describe how the elements of the MIM can be grouped logically to form larger, semantically complete messages.

1 Introduction

The *Multilateral Interoperability Programme* (MIP; [3]) is a standardization forum that defines specifications for the interoperability of command and control (C2) information systems in coalition networks. Currently, the programme is supported by 24 member nations, NATO, and the European Defence Agency (EDA).

One of the key products of MIP is the *MIP Information Model* (*MIM*). It defines concepts needed for C2 information exchange in joint and combined operations. The focus of the MIM is on capturing the semantics rather than mandating a specific syntax for exchanging data.

In this paper, we present and discuss two aspects:

- **Model improvements:** The model is continuously enhanced to meet new requirements. A new version, MIM 4.0.1, has been released in July 2016. It builds the foundation for the new MIP 4 Information Exchange Specification (IES), which has entered a one-year validation phase recently.
- **Adoption of the MIM by COIs:** The MIM itself does not define an interoperability solution. Instead, it is considered a semantic reference from which multiple information exchange specifications can be derived. The MIM harmonizes/unifies information exchange requirements coming from many different Communities of Interest (COIs). Several standardization communities in the military domain have adopted concepts from the MIM (or its predecessor, the JC3IEDM) to build their own exchange specifications. With modern tools at hand, the adoption of the MIM can be automated and it is possible to establish a process that improves cross-COI interoperability.

This paper is structured accordingly: in section 2, a brief overview of the key concepts of the MIP Information Model is given. We sketch the modeling approach taken to capture semantic information. Moreover, we present the tools and services complementing the model. Section 3 presents some of the advancements achieved for the MIM 4.0. These include a better taxonomy of objects, handling of unknown information, and traceability through Semantic IDs. In Section 4, we discuss techniques for how COIs may build their own interoperability solution based on the MIM and the tools available. We describe how the elements of the MIM can be grouped logically to form larger, semantically complete messages. The paper ends with a short summary in section 5.

Acknowledgment The MIP Information Model is the result of a long-term effort of the Multilateral Interoperability Programme. As members of the MIP community, the authors have taken an active role in shaping the new model for several years. However, there are many other people – operational subject matter experts, data modelers, and system engineers – that have provided valuable input to what has become the MIP Information Model. The authors would like to thank all co-workers for their contributions.

2 Model Overview

The MIP Information Model is specified in the Unified Modeling Language (UML). It is a platform-independent model, i.e., it is agnostic of any particular exchange technology. Version 3.0 of the MIM was published in March 2015. Its concepts are described in great detail in [2]. In the following, only a short summary is given.

The basic concepts of the MIM are depicted in Figure 1. The backbone of the model is formed by a taxonomy of Objects and Actions. In total, the MIM defines about 2,300 different types of objects and more than 500 types of actions.

One of the main benefits of the MIM is its semantic annotations. UML stereotypes are assigned to all model elements in order to enrich them semantically. For instance, each attribute is classified with regard to representation terms such as *identifier, name, text, indicator, speed*,

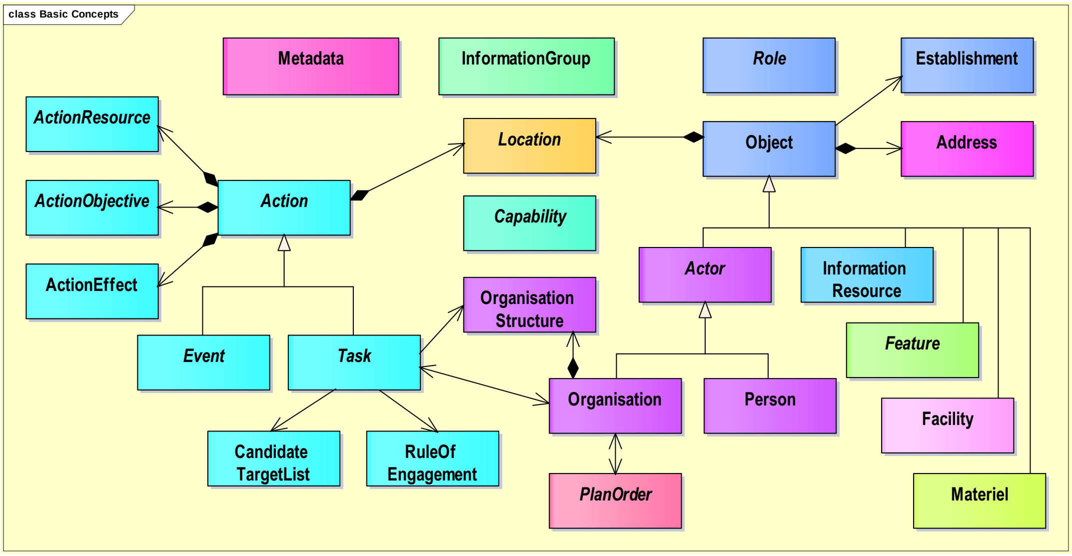

Figure 1: Basic Concepts

dimension, *duration*, or *quantity* to name a few (see Figure 2, left side). Depending on the representation term, specific meta data are provided for the attributes, e.g., the unit of measure, the minimum and maximum value, or the scheme for a given identifier. The representation terms and their meta data have been adopted from the UN/CEFACT Core Components Data Type Catalogue Version 3.1 [6], with minor extensions to disambiguate attributes for which more than one representation term seemed to be applicable.

Similarly, the code lists of the MIM (modeled as UML enumerations) have several meta data (tagged values) that define their intended use (cf. Figure 2, right side): Codes may be defined as "complete" if the provided list of values is considered operationally exhaustive. In that case, the end user of a C2 information system must always pick one of the available values and is not allowed to introduce a new value. Codes may also be "managed", i.e. they may be defined "at run-time" by a military authority rather than at "design time" by the MIM modelers. Codes may also be ordered, in which case the MIM defines the ordering relationship.

The MIM is provided as a UML model in Sparx Enterprise Architect format. It is complemented by a variety of products that are auto-generated from the model. Documentation is exported in both HTML and OpenXML (MS Word) format. Exchange schemas for the entire model are provided as OWL ontology and XML schemas. Finally, the MIM is published as a web of Linked Data (see `https://www.mimworld.org/mim/`).

The MIM comes with a suite of open source tools for model tailoring (see section 4.3), schema generation, and model management. The latter set of tools is used internally by the MIM team to make sure that all parts of the MIM – class model, constraints, diagrams, examples, free-text documentation – are consistent.

The MIM ecosystem is depicted in Figure 3. All models, products, and services can be downloaded from the MIM portal [4].

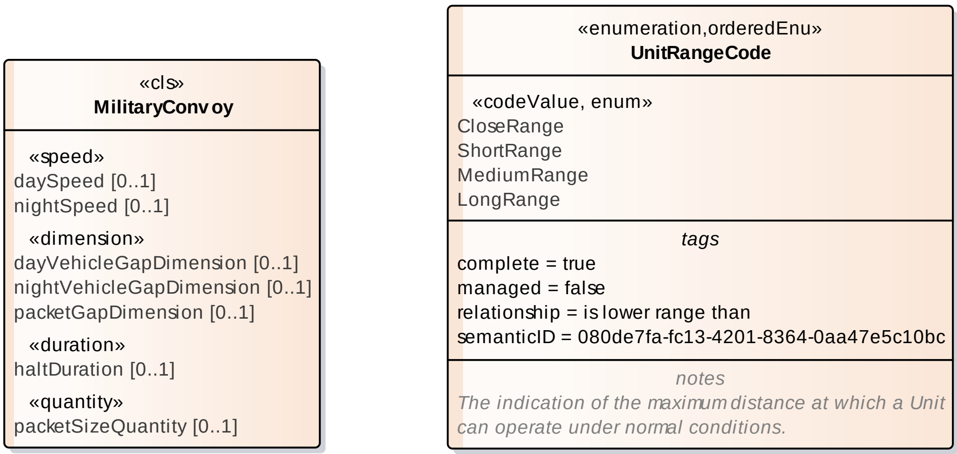

Figure 2: Semantic Annotations

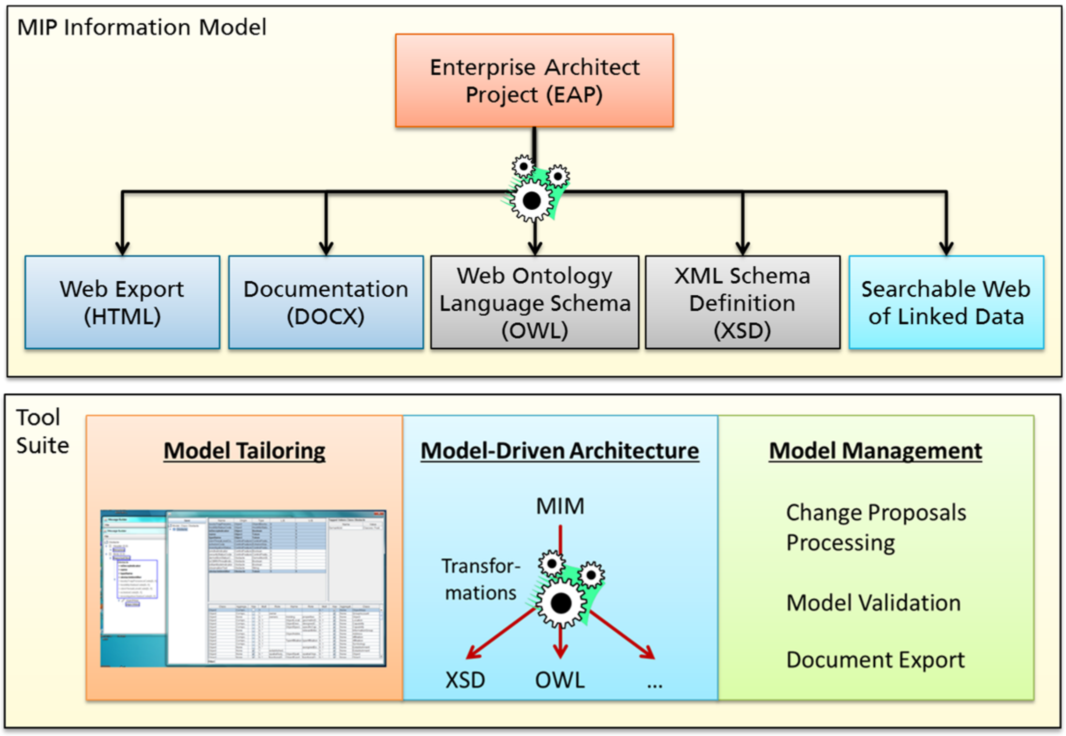

Figure 3: The MIM Ecosystem

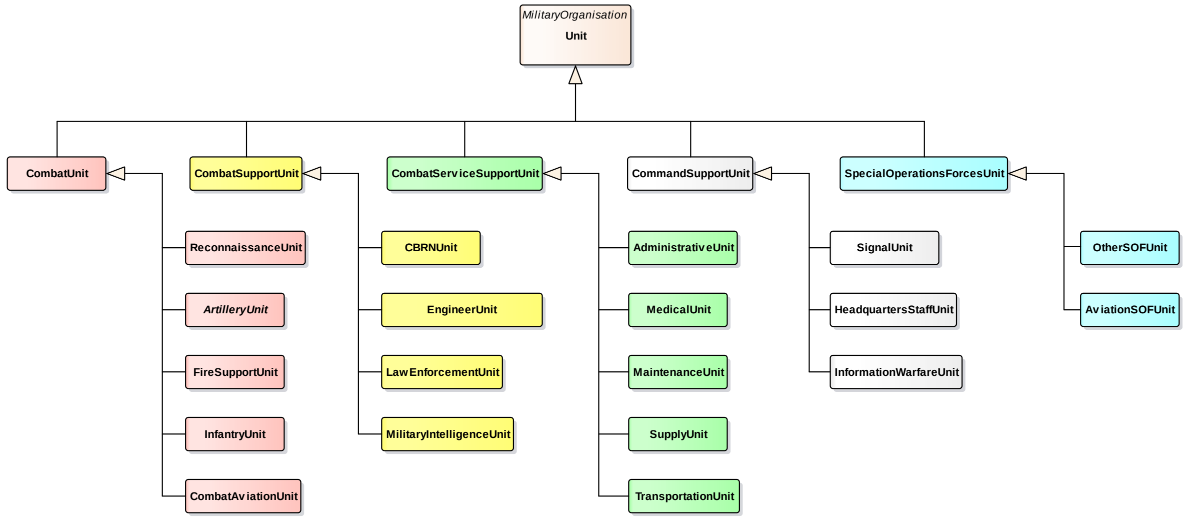

Figure 4: Unit Taxonomy (Excerpt)

3 MIM 4.0 Improvements

The MIP Information Model 4.0 includes several new modeling concepts and many smaller improvements.

3.1 Object Taxonomy

The core of the MIM is a taxonomy of objects. Over time, the taxonomy has been extended with new concepts but its overall structure has not changed much.

In the MIM 4.0, much effort was spent on cleaning up the Object hierarchy. This includes deleting duplicate concepts, moving specific classes to a more suitable branch in the taxonomy, and clarifying names and definitions.

The focus was on the proper modeling of Units. In the past, units were mainly characterized by six attributes that captured concepts from APP-6(a) and APP-11. To rule out invalid value combinations, complicated constraints had to be defined. Even worse, this way of modeling resulted in different representations for the same type of unit.

The latest version of the MIM defines a new Unit taxonomy based on the functional aspects of military units. An excerpt of the taxonomy is shown in Figure 4. At the top level, the MIM distinguishes between combat units, combat support units, combat service support units, command support units, and special operations forces units. Each subclass spans its own subtree of units according to the function they perform.

Additional attributes characterize the military units, e.g., their type of mobility, operation level, support level of command, missile range, equipment scaling (light, medium, heavy), range (close, short, medium, long), and air defense altitude. Not all of these attribute are applicable to all types of units. Therefore, they have been assigned to specific subclasses in the `Unit` taxonomy, where needed.

3.2 Semantic ID

Each model element is uniquely identified by a qualified name. It maps to a Unique Resource Locator (URL) that does not only take into account the package structure of the UML model but also includes a prefix for the MIM in general and the specific model version. For instance, the URL for `Line` in the MIM 4.0.1 is `https://www.mimworld.org/mim/4.0.1/Classifiers/Location/Line`.[1] Since the MIM defines its own namespace, concepts in the model are not confused with concepts of other standards that may have the same name. The same holds for concepts defined in different versions of the MIM.

While two concepts are probably semantically the same if their qualified names only differ with regard to the model version, there is no guarantee that this is in fact true. The definition of `Unit` may have changed from one version of the model to another, thereby changing the set of instances that are considered Units.

The opposite case may occur when the model is restructured over time:

- Model elements may be renamed to follow naming conventions.
- Model elements may be moved from one package to another.
- Literals may be turned into classes to specify their taxonomic relationships.

Although their qualified names have changed, the modified or new model elements still represent the same semantic concept.

In order to capture semantic equivalence of model elements across multiple versions of the MIM, each class, data type, attribute, literal, and association has a *Semantic ID*.

There are several use cases for Semantic IDs:

- Semantic IDs can be used to trace concepts between different model versions, thereby reducing the work to support backwards compatibility.
- Semantic IDs can be used to specify mappings to other standards such as APP-6. Since the Semantic ID of a given concept is stable whereas its qualified name may change, mappings defined for a previous model version can be mostly preserved for a new model version.

Technically, Semantic IDs are defined according to IETF RFC 4122, ”A Universally Unique IDentifier (UUID) URN Namespace”. The exact value of a Semantic ID is meaningless. It only serves the purpose of comparing two model elements, coming from different versions of the MIM, for semantic equivalence.[2]

[1] You can query information about `Line` by entering the URL in your favorite web browser.

[2] The Semantic IDs must not be confused with the GUIDs maintained by Sparx Enterprise Architect. Both Semantic IDs and EA’s GUIDs are expected to be unique within the scope of a particular version of the MIM. However, two model elements from different MIM versions

The key criterion for determining semantic equivalence is the definition of the model elements. Whenever a definition changes, the MIM team must decide whether it is an editorial change/a rewording or a fundamental change of the concept. In the latter case, a new Semantic ID must be assigned to the model element.

The concept of Semantic IDs is already in use in two ways:

- The OWL ontology of the MIM uses the Semantic IDs to establish sameAs relationships.
- The Linked Data Server, i.e., the application that publishes the MIM on the web, allows to query information for a specific Semantic ID. For instance, when sending an HTTP GET request to `https://www.mimworld.org/mim/semanticID/33ea6491-e65e-402b-8c8c-f3b4b469d33b`, all occurrences of this concept across all published MIM releases are listed.

3.3 Missing Information

During a multinational operation, there may be different reasons why specific information cannot be exchanged between military staff:

- The list of offered code values does not contain an appropriate entry (in which case the user may want to provide his/her own value).
- There is no useful value. The information does not make sense in the given context.
- The correct value is not readily available and may not even be available in the future.
- The value will be available later.
- The correct value is not known to, and not computable by, the sender. However, a correct value probably exists.
- The value is not divulged. Security restrictions prohibit its exchange.

The MIM defines the class NilReason to allow to specify why information is not provided. The concept is largely inspired by but is not identical to the NilReasonType in the *Geography Markup Language* (*GML*). The design of the MIP Information Model is based on the assumption that any property is *nillable*, i.e., a NilReason can be specified for any model element and, accordingly, a user may express the lack of information. Consequently, there is no (longer a) need to add special literals such as Other, Unknown or Not Otherwise Specified to individual enumerations – they are out of the scope of the MIM. The concept of *"nillability"* is not restricted to properties of type enumeration; it applies equally to properties of other types such as Boolean attributes. If the MIM defines a mandatory Boolean attribute (e.g., isActiveIndicator), with values true and false, it is assumed that an end user will still be able to express that the status cannot be delivered and is able to provide the reason why.

- may have the same Semantic ID but different GUIDs (the model element was replaced by another, semantically equivalent one)
- may have the same Semantic IDs and GUIDs,
- may have different Semantic IDs but identical GUIDs (the semantics of the model element has changed).

The *nillability* concept does not only apply to mandatory elements but also to optional ones. This allows the information provider to be more expressive as to why optional elements are not provided, if desired.

The concept of *nillable* properties must be taken into account when defining platform-specific schemas for an information exchange specification. The MIM itself does not mandate a specific technical solution. COIs may restrict the use of NilReason to selected properties in the context of specific business processes.

3.4 Miscellaneous Enhancements

The MIM 4.0 includes many additional changes that fall into one of the following categories:

Harmonization With Other Interoperability Standards Several areas of the information model have been updated according to new versions of authoritative sources. The Unit taxonomy has been (partially) aligned with the latest NATO standard for tactical symbols, APP-6(D). This simplifies the mapping of semantic concepts in the MIM onto a graphical depiction in line with APP-6(D). Several code lists that were adopted from former baselines of ADatP-3 have been aligned with the latest APP-11(D). The definitions of many facilities and features (primarily *geographic* features) have been aligned with the definitions of equivalent concepts used in the *NATO Geospatial Information Framework* (*NGIF*) and the *DGIWG Feature Data Dictionary* (*DFDD*). Since C2 information systems usually combine geospatial data with C2 data, harmonization of the MIM and the NGIF/DFDD helps to mitigate potential semantic conflicts.

Improved Modularity The MIM 4.0 defines individual packages for specific sub-trees of the Object hierarchy. These packages are nested according to the class hierarchy. For instance, class Material, which is an immediate subclass of Object, is placed in a package called Classifiers::Object::Material. All subclasses are placed in the package of their super class or in a sub-package thereof.

Moreover, data types (incl. enumerations) are defined in the package, in which they are used. If a data type is used by classes/data types in more than one package, the data type is placed in the "deepest" package that subsumes the packages. Data types used in multiple top-level packages are placed in a package called Generic. The same rules apply to association classes that link classes from two different packages.

The increased number of packages, in combination with the logical grouping of classes, data types, and enumerations, improves the modularity of the model.

Enhanced Operational Concepts Several operational concepts of the MIM have been improved. For instance, the specification of the *Military Load Classification* (*MLC*) of bridges on the one hand and vehicles on the other hand has been greatly simplified. In addition, the taxonomy of CBRN concepts (CBRN = Chemical, Biological, Radiological, Nuclear) has been expanded. For organization boundaries, it is easier to specify the names of the left and right organizations. In addition, it is possible to specify whether an action is hostile or not, something that is required when displaying events on a map according to APP-6.

General quality assurance The MIM has undergone a general revision. In many places, this has resulted in better (or new) definitions, more meaningful (self-explaining) names for model elements, and a more consistent modeling style.

Extended Documentation The MIM 4.0.1 includes many new class diagrams depicting specific parts of the model. It also contains additional examples in form of object diagrams.

4 Tailoring the MIM for COI-Specific Information Exchange

There are several Communities of Interest (COIs) that define interoperability standards for information exchange within the C2 domain. Those COIs address specific requirements, e.g., communication on the tactical level, intelligence, or C2-simulation information exchange. Yet, there is an overlap of concepts used by those communities.

The MIP Information Model is considered a semantic reference model that defines many concepts shared by various COIs. It can be used to overcome the n^2 problem, when n COIs need to implement $n - 1$ interfaces to exchange information with all other COIs.

The MIP Information Model defines operational concepts (in terms of UML classes) and their relationships. It purposely does not define platform-specific details, such as the format for serialization. Moreover, it is modeled independently from the *context* in which information is to be exchanged. The MIM is a model with a graph-like structure. UML classes are associated with each other in various ways, resulting in potential direct and indirect circular dependencies. In an exchange schema, such circular structures are unwanted. Instead, a tree structure is preferred for exchanging messages.

Figure 5 describes the conceptual process to derive a new exchange specification from the MIM.

- In the first step, the relevant parts of the MIM are identified and missing elements are added. The outcome is a tailored model that is still on the conceptual level. Section 4.1 describes on how this step should be performed.
- To derive a message model, the previous submodel has to be turned into a tree-like representation and message-specific structuring elements (such a Header and Body) have to be added. Sections 4.2 and 4.3 discuss several technical aspects of building a message model.
- Finally, an exchange schema can be derived from the message model in an automated manner.

Depending on the technical approach, steps 1 and 2 may be combined.

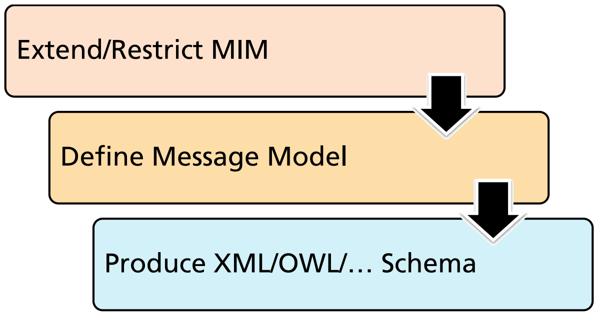

Figure 5: MIM Tailoring

4.1 Extending and Restricting the MIM

The MIM unifies information exchange requirements from many different sources and for many different purposes. Therefore, no single COI will adopt the MIM in its entirety. Moreover, there may be COI-specific requirements that are not reflected in the MIM (because they may not be relevant for cross-COI information exchange). They must be put in place when defining information exchange specifications.

Any COI may customize the MIM according to its needs and build its own COI-specific model (by reusing as much of the MIM as possible). However, this customization should not be done in an arbitrary manner. To support cross-COI information exchange, the following rule should apply:

> If a COI adopts a concept C from the MIM and tweaks it for its own purposes, resulting in C', then any instance of C' should also be a valid instance of C.

In terms of UML, the following restrictions may be made by COIs:

- **Restrict the multiplicity of properties (attributes and association ends)**

  The multiplicity in the COI submodel must be a subrange of the multiplicity in the MIM. Formally, let us assume that a property in the MIM has a multiplicity of $[x, y]$, where x is the lower bound, i.e., the minimum number of occurrences, and y is the upper bound. Then the multiplicity in the COI submodel must be $[x', y']$ with $x \leq x'$ and $y \geq y'$. Accordingly, optional attributes ($[0, 1]$) may be turned into mandatory ones ($[1, 1]$) or ignored completely ($[0, 0]$). However, a mandatory attribute ($[1, 1]$) must not be declared optional by a COI. The same applies to association ends, i.e., if a class C_1 has an association with a class C_2, where the multiplicity of the association end related to C_2 is $[0, *]$, then the association can be ignored completely from the perspective of C_1.

- **Restrict the set of code values**

  An enumeration in the MIM may define literals that do not make sense in a specific use case. A COI is allowed to restrict the set of literals. For example, VehicleCategoryCode defines many different types of vehicles but not all of them may be needed in a specific context.

- **Define default and fixed values**

  A COI might want to define default and fixed values for properties. For instance, the attribute isDecoyIndicator, which specifies whether an object has been designed expressly for purposes of acting as a decoy, is mandatory in the MIM. In some domains, we can assume that most objects are not decoys. In that case, the attribute should be set to the default value false.

  There are also cases where a fixed value should be assigned (in UML, an attribute has a fixed value if has a default value and is declared as *read-only*). For instance, when planning an own Order of Battle (ORBAT), we can assume that all involved units have its hostilityStatusCode set to Friendly.

- **Define domain-specific constraints**

  Using a general-purpose, formal language such as the Object Constraint Language (OCL; [5]), a COI may want to impose further constraints that cannot be expressed in UML. For instance, there may be domain-specific restrictions on the combination of valid values for different attributes. The types of constraints are not limited as long as they do not conflict with the underlying model (and, implicitly, with the rules stated above).

It is important to state that the rules above do not automatically enforce interoperability among different COIs in both directions. Any COI may share its information with other COIs based on the MIM. However, in order to support internal processes, additional constraints may have to be obeyed and the context of the information exchange needs to be considered. It is up to the individual COIs to decide on how to integrate (enrich/modify) MIM-compliant information coming from an external source.

4.2 Defining a Message Model

When building a message model, there are several aspects to consider:

- **Traceability to the MIM:** There should be a reference from the model elements in the COI message model to the corresponding concepts in the MIM. Ideally, the MIM elements should be reused in a way that standard tools will instantly recognize them as MIM concepts (by using the same URI). A weak solution is to introduce a link from the elements of the COI message model to their MIM counterparts (by using the Semantic ID; see section 3.2). The link could be specified in the message definition itself (e.g., as appinfo element in an XML schema that holds a MIM Semantic ID) or by means of an external mapping document.
- **Readability:** COI message models should be designed in a way that they are easy to understand by both subject matter experts and implementers.
- **Maintenance:** COI message models should be easy to maintain in response to changing requirements. In addition, it should be possible to make an impact analysis on COI models as the MIM evolves.

- **Reuse of Components:** There are components (also called *segments*) that occur in many different types of messages. A typical example is the combination of Unit and its Location. It should be possible to share such commonly used components across multiple COIs.
- **Support for Harmonization:** If several COIs introduce similar or identical concepts to their specific models, these concepts are candidates for unification within the MIM. The COI-specific extensions should be specified in a way that they can easily be fed back to the MIM.
- **Guided Tailoring Process:** Defining a COI-specific submodel should be supported by tools to ensure that only valid restrictions are applied.
- **Using Concepts in Different Contexts:** Experience has shown that the same operational concept may be used in different contexts within the same message. Depending on the context, different types of information may be relevant and, thus, different restrictions may have to be applied to the same concept. For instance, different meta data (reporting time, security classification. ...) may be needed at different places in a message model. In case of objects (in MIM terms), we talk about *roles* that they play. For example, a Unit in an ORBAT message model may play the role of the planning organization or of a subordinate unit. In a perfect model, all relevant roles are already defined; in practice, this is not the case.

4.3 Technical Approaches

There are several ways in which COIs may specify their logical message models. In the following, we discuss the pros and cons of some alternatives.

4.3.1 Copy and Adjust

The most trivial approach to build a COI-specific model is to

1. create a copy of the MIM,
2. remove all unwanted elements manually (e.g., by using the UML modeling tool), thereby following the constraints listed in section 4.1,
3. add missing elements for the top levels of the messages that glue the MIM concept together, and
4. create a new (XML) schema from the modified model with the XSD generator provided for the MIM.

Steps 1 and 2 can be combined and automated by selectively picking parts of the MIM when copying the model. This approach, as well as step 3, is already supported by the MIM tool suite. The needed elements as well as the new elements can be specified in XML format and created/manipulated by a tool, called CPProcessor.

The end result of the process above is a new exchange schema that follows the same naming and design rules of the MIM and keeps the model elements in their original (MIM) namespace, i.e., they are indistinguishable from the original MIM elements. (For the separation of concerns, it may be beneficial to place new COI-specific elements in a separate package.)

This approach is very flexible. However, it does not allow to use MIM concepts with different restrictions in different contexts – each MIM concept is only available with one specific way of restrictions. Moreover, the modeler defining the COI submodel still has to take care of associations resulting in non-tree structures.

4.3.2 Modeling Based on UML Class Diagrams

COI-specific submodels may also be specified in terms of UML diagrams. Formally, a UML diagram describes a subview of a larger model, not a submodel.

Modern modeling tools, such as the Sparx Enterprise Architect (Sparx EA), allow to specify what should be displayed in a UML diagram down to the level of individual attributes and literals. That means that optional attributes and associations may be hidden in a specific view and literals may not be listed.

In Sparx EA, UML class diagrams can be nested to indicate the hierarchical message structure. Figure 6 shows a PersonLocationReport that consists of two segments: ReportHeader and PersonLocation. Each segment is represented as another class diagram containing elements of the MIM.

When deriving an exchange schema, it is necessary to know the order of the nested elements. The relative position of their respective diagram frames may be used as an aid to determine the order (from top to bottom, left to right). Another option is to exploit the flexibility of the UML modeling tool and to specify "composition" relationships between the diagrams.[3] For instance, in the example of Figure 6, PersonLocationReport could be a diagram (or even a class) that has a composition relationship with ReportHeader and PersonLocation. By tagging the composition relationships with sequence numbers, it is possible to define the order, in which the elements will be specified in the exchange schema, unambiguously.

It is not possible to modify the multiplicity of a property in the scope of a specific diagram – such a modification must be done in the underlying model. Similarly, there is no formal means to express the multiplicity of nested diagrams. Therefore, class diagrams alone are not sufficient to fully specify a COI message model. One way to overcome this limitation is to specify all required restrictions in a formal manner next to the class diagrams. By means of an automated tool, a new submodel can be generated that (a) includes all elements listed in the class diagrams and (b) incorporates all restrictions.

Using UML class diagrams has the advantage that a strong coupling to the MIM is preserved. If the MIM changes, the class diagrams will change as well, thanks to the UML modeling tool. However, changes will propagate unnoticed by the modeler.

[3] Caution: This approach is not in line with UML semantics!

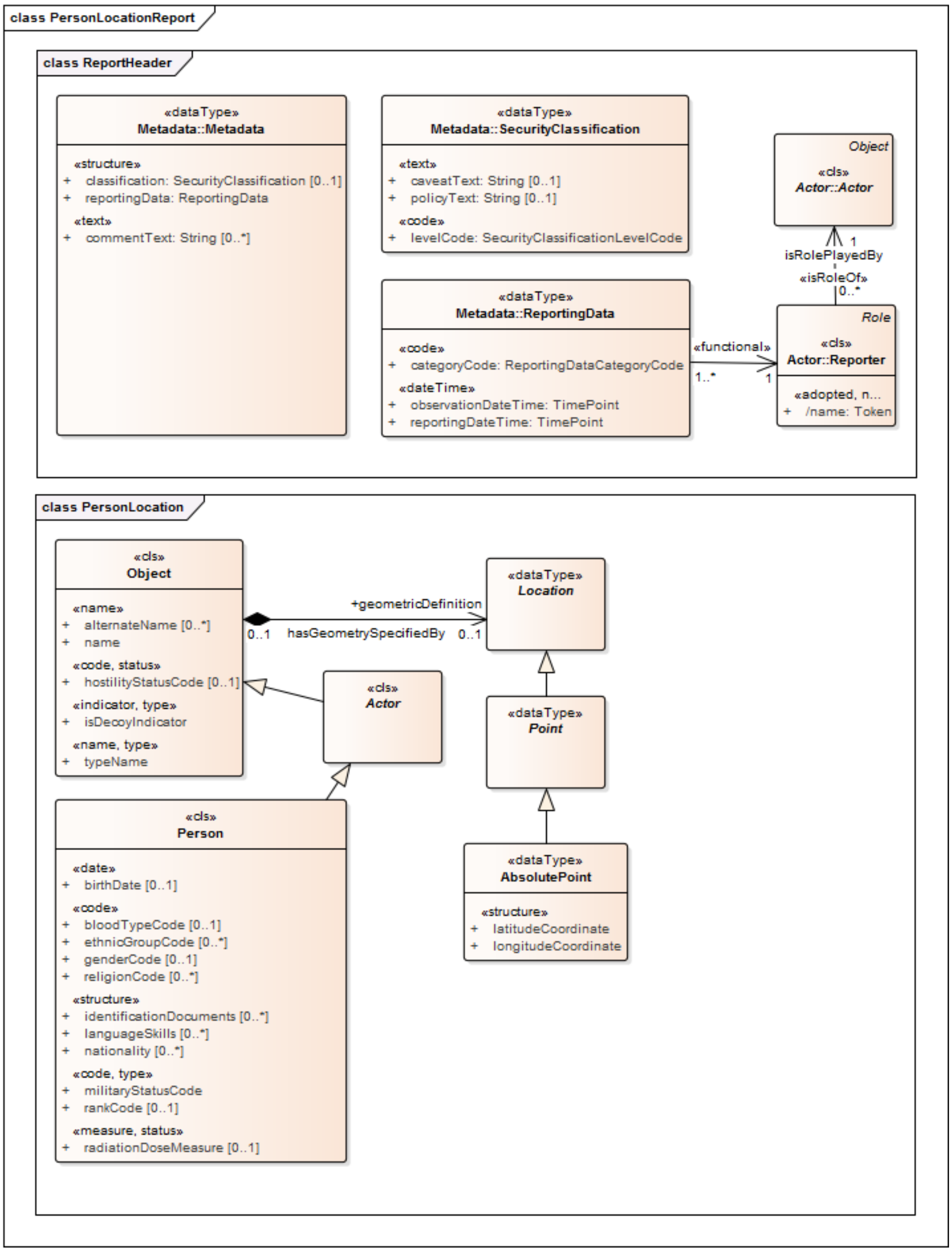

Figure 6: PersonLocationReport Defined by Nested Class Diagrams

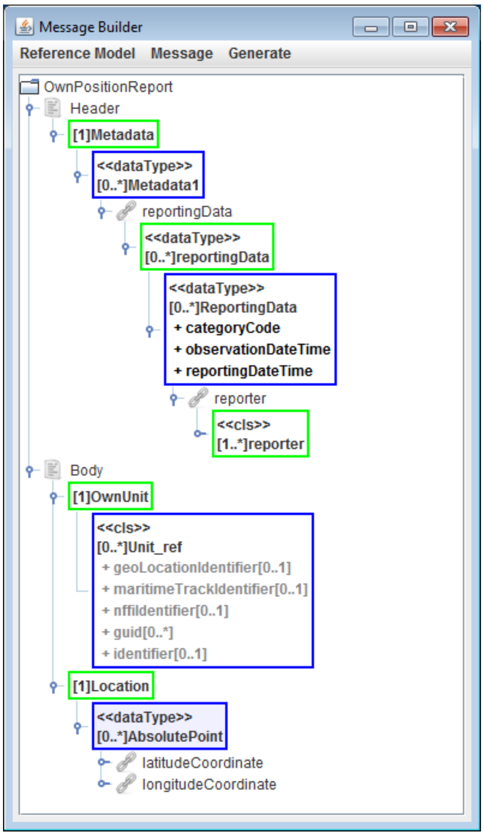

Figure 7: Screenshot of the Message Builder

The process of transforming diagrams and restrictions into a COI-specific submodel is repeatable and may unveil conflicts if the underlying MIM has changed in a way that is not compatible with the restrictions.

4.3.3 On-the-Fly Model Generation

At the Fraunhofer FKIE, we have realized another option to build message models based on tool support.

The idea is to build a tree-like message format in an interactive manner (see Figure 7). Starting with an empty message format, the user is asked to specify the top-level "segments" of a message, e.g., Header and Body. Each segment may contain other segments or elements referring the types in the MIM.

As the user defines its message format, a corresponding message model in UML is constructed on-the-fly. It copies relevant parts from the MIM, when needed. When the message format is complete, the UML model is finalized, too. Thereafter, an exchange schema in XML can be generated from the UML model.

By logging the user actions, the process of building the message model is repeatable. Any changes to the MIM that impact the COI model can be identified and presented to the user.

4.3.4 Extending MIM Concepts

The previous approaches modified the model elements (classes, attributes, ...) of the MIM and placed the modified versions either in the MIM namespace or in a specific COI namespace.

An alternative option is to keep the original MIM concepts untouched and derive subclasses from them that implement the restrictions. This approach is supported by UML but it is not commonly used[4]. Languages such as OWL and XML Schema support extensions and restrictions, too. For instance, it is possible to import MIM concepts into an OWL ontology ("coiA:MyUnit extends/restricts MIM:Unit"). Similarly, XML Schema supports two kinds of sub-typing: by extension and by restriction (the latter is basically in line with the rules given in section 4.1).

5 Summary

In this paper, we gave an overview of the concepts of the MIM. In particular, we highlighted some key improvements of the MIM 4.0. Among others, the latest version of the model provides a generic and consistent approach for systems and operators to deal with unknown information or information that is not covered by the MIM. By introducing Semantic IDs for all model elements, it has become easier to trace changes from one version of the model to another and to define mappings to other interoperability standards such as APP-6. In the second part, we described how Communities of Interest can take benefit of the concepts defined in the MIM. Several technical approaches have been presented and compared. Due to its "semantic richness", we think that the MIM is qualified as a semantic reference model that supports the standardization efforts not only within MIP but also across multiple COIs in or related to NATO.

References

[1] Gerz, M. & Meyer, O.: *Defining C2 Semantics by a Platform-Independent JC3IEDM.* International Journal for Intelligent Defence Support Systems. Special Issue on Command and Control Ontologies, Vol. 4, No. 3, pp. 263–285, 2011.

[2] Gerz, M., Mulikita, M., Bau, N. & Gökgöz, F.: *The MIP Information Model – A Semantic Reference for Command & Control.* International Conference on Military Communications and Information Systems (ICMCIS) / IST-119 Workshop on Semantic Interoperability, Krakow, Poland, 2015.

[4] UML provides many advanced features such as association subsetting and overriding of attributes but they are not much used in practice due to the complexity they introduce.

[3] Multilateral Interoperability Programme (MIP): *MIP Website*. `https://www.mip-interop.org`. (Last visited on 14 July 2016).

[4] Multilateral Interoperability Programme (MIP) / Fraunhofer FKIE: *Portal of the MIP Information Model*. `https://www.mim-world.org`. (Last visited on 14 July 2016).

[5] Object Management Group (OMG): *Object Constraint Language*. February 2014, `http://www.omg.org/spec/OCL/`. (Last visited on 14 July 2016).

[6] United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT): *Core Components Data Type Catalogue Version 3.1*. 17 October 2011, `http://www.unece.org/cefact/codesfortrade/ccts_datatypecatalogue.html`. (Last visited on 14 July 2016)