

Keeping up with the Military Scenario Definition Language (MSDL)

Robert L. Wittman Jr., PhD.
The MITRE CORPORATION
3504 Lake Lynda Drive, Orlando, FL 32817
321-235-7601
rwittman@mitre.org

Jeff Abbott
Acusoft
3045 Technology Parkway
Orlando, FL 32826
321-235-7716
jabbott@ideorlando.org

Keywords: OneSAF, Computer Generated Forces (CGF), Semi Automated Forces (SAF), Command, Control, Communication, and Computers, and Intelligence (C4I), Software Reuse

ABSTRACT: *The Military Scenario Definition Language multi-national standardization effort is poised to enter the Product Development phase of the Simulation Interoperability Standards Organization (SISO) standardization process. During calendar year 2005 a SISO MSDL study group was formed, met several times, and collaboratively developed a number products to capture and focus interest, define the product set for standardization, produce draft standards, and generate interest for a follow on product development group. Concurrently, the MSDL product itself, an XML-based schema representation of a military scenario has continued to evolve with inputs coming from several programs: the Modeling Architecture for Technology, Research, and Experimentation (MATREX), the Army Command Control, Communications, Computers and Intelligence (C4I) and Simulation Initialization System (ACSIS), and the OneSAF Objective System (OOS). This paper quickly summarizes the calendar year 2005 efforts, the planned standardization efforts for CY06, and describes the major enhancements to MSDL driven by the multi-program inputs.*

1 Introduction

MSDL intends to unambiguously capture a military scenario in an open, well-defined format for population by compliant applications and consumption by computer-based simulations. MSDL originated as an effort by the Program Manager One Semi-Automated Forces (PM OneSAF) Military Scenario Development Environment (MSDE) task order to redesign and enhance the Close Combat Tactical Training (CCTT) System's Commander's Exercise Initialization Toolkit (CEIT) to comply with the OneSAF Objective System (OOS) Product Line Architecture Framework (PLAF). The resulting separation of the military scenario data from CCTT specific initialization code provided the greatest opportunity for reuse by other simulations.

Within the OOS architecture, the redesigned CEIT is called the Military Scenario Development Environment and the resulting XML-based Data Interchange Format of MSDE's output is called the Military Scenario Definition Language (MSDL). Currently MSDL is being studied by the Simulation Interoperability Standards Organization (SISO) as a possible international military scenario definition standard.

Within the OOS architecture MSDL offers a hallmark example of several important architectural characteristics:

- **Separation of code from data:** An MSDL formatted document includes all of the data necessary to define the initial conditions of a military scenario and the initially-planned missions and tasks associated with the organizations defined

within the MSDL document. Furthermore, an MSDL document is defined using XML and is independent of any other programming language, database, or application.

- **Use of industry standards for data format and content definition:** MSDL is defined using the World-Wide Web Consortium (WC3) eXtensible Markup Language (XML) definition [1]. MSDL is defined using an XML schema that allows for format and content validation prior to document use.
- **Application independence:** MSDL documents are fully explained by their accompanying standardized XML schema and accompanying MSDL information products. MSDL documents can be, and have been, populated manually, via database applications, via C4I device output, or via specially designed tools.
- **Separation of concerns:** MSDL's exclusive focus is restricted to information relevant to a military scenario and does not include application specific, training specific, exercise control specific, or other types of simulation initialization type information. This type of information might include the speed of simulation execution, specifying the forces that will be represented either in a live, virtual, or constructive manner, etc.

Based on the results of the SISO MSDL Study Group these same characteristics make the MSDL valuable to the SISO community [2].

The following sections briefly explain MSDL efforts and technical enhancements in Calendar Year (CY) 2005 and the expected MSDL standards related efforts in CY 2006.

2 MSDL Efforts in Calendar Year 2005

Throughout 2005 MSDL continued to evolve through a number of formalized program-level efforts and through international collaboration associated with the SISO MSDL Study Group effort. The following sections discuss the primary MSDL-related activities during 2005, starting with the SISO MSDL Study Group, continuing with the MATREX integration

effort, and concluding with OOS MSDL development highlights.

2.1 SISO MSDL Study Group

During 2005 a SISO-accredited MSDL study group was formed and met 5 times, with 4 meetings in the US and 1 in Toulouse, France, in conjunction with Euro-SIW [2]. There were over 100 participants at the face-to-face meetings, and 98 study group members participating via the MSDL Study Group reflector to meet the assigned activities within the MSDL Study Group Terms of Reference (ToR). The entire membership brought a broad knowledge base and a substantial international perspective in successfully addressing the study group's tasks and developing the required products stated within the ToR. The requirements identified within the TOR include:[3]

Tasks accomplished

1. Reach out to the community to gather further requirements and input to MSDL.
2. Evaluate Battle Management Language (BML) and Coalition BML (CBML) work to ensure MSDL meets the needs of all three communities.
3. Prepare progress report to be presented at the Euro 05 SIW
4. Perform analysis and prepare a set of recommendations for steps required to define the specifications of the eventual product.
5. Define the final set of MSDL products that should be pursued by a follow-on SISO MSDL product development group (PDG).
6. Prepare the final report to be presented at the Fall 2005 SIW.

Products developed

1. Reports to Euro SIW 2005, Fall SIW 2005
2. Table of Contents for Final Report (Euro SIW)
3. Final Report of Study Group (Due Fall 2005), including:
4. Description of the MSDL standards.
5. Description and analysis of issues and proposed alternative solutions.

6. Recommendation to SISO on the topics and issues described in this TOR, and (if appropriate) an accompanying standards Product Nomination.

In addition to completing these assigned activities the study group collaboratively developed an MSDL domain spreadsheet identifying the organizations with a direct interest in populating or leveraging MSDL data. The group developed a review and feedback process and associated comment form for the initial and subsequent MSDL specification drafts, and the Study Group vetted version .01 of the MSDL specification. Finally, at the time of this writing, the study group has resubmitted the product nomination and final report in response to Standards Activity Committee's (SAC) comments and is awaiting additional comments or approval on the MSDL product nomination. All of the Study Group products and correspondence are available on the MSDL Study Group Reflector at <http://discussions.sisostds.org/default.asp?action=10&fid=84>.

2.2 MATREX Integration

Throughout 2005 MSDL developers have been working with the Modeling Architecture for Technology, Research, and Experimentation (MATREX) team to create a comprehensive system for federation initialization. MATREX is a toolset within the Army's Research, Development, and Acquisition (RDA) modeling and simulation domain and is sponsored by the US Army Research, Development, and Engineering Command (RDECOM). MATREX offers a set of model and simulation federates, supporting tools for initialization, data collection, after action review, and federation management, and a configuration managed FOM to support system acquisition and operational concept analysis [4]. As shown in Figure 1, MSDL is central to the design principle of providing a single centralized data source for the interacting federates.

Figure 1: MATREX Simulation Initialization [4]

As described by Prochnow et al., [4] the MATREX initialization scheme has the potential to reduce federation scenario development time by providing appropriate tools for scenario construction and distribution to the required federates and a process to support such distribution. This process is expected to produce a number of positive results. First it is expected that there will be reduced federate miscommunication due to enumeration-type mapping errors by providing a common data set for each federate to base its common names and enumerations on. Secondly, it is expected there will be reduced configuration management overhead by since a single centralized data set will be used. Finally it is expected there will be a scenario development cost reduction due to the process and tools that allow reuse and sharing of scenario data by all participating federates. [4]

To support MSDL's role in the MATREX environment data element extensions to MSDL's XML schema were necessary. MATREX engineers worked to define required elements and propose these elements for inclusion into the MSDL schema. The Attack Guidance Matrix (AGM) domain area was used as the first addition based on the MATREX analysis. At the time of this writing the MSDL developers are finalizing the implementation support for the new AGM data elements [5]. We expect that additional collaboration and MSDL extensions will continue in support of MATREX through CY 2006.

The AGM represents a battalion-level and above planning artifact extension to the MSDL. The AGM identifies the important targets and available munitions and then makes specific weapon to target pairings.

Although MSDL development has focused on brigade-level and below, there does not appear to be any substantial technical issues for including the higher level AGM information.

It should be noted the AGM data structures are still being integrated into the MSDL baseline and may change between the publication of this paper and the ultimate AGM representation within the MSDL. Because the AGM information is associated with target-based planning it logically fits as a subordinate XML structure under the Coarse of Action (COA) element and at a top level will be structured as depicted in Figure 2.

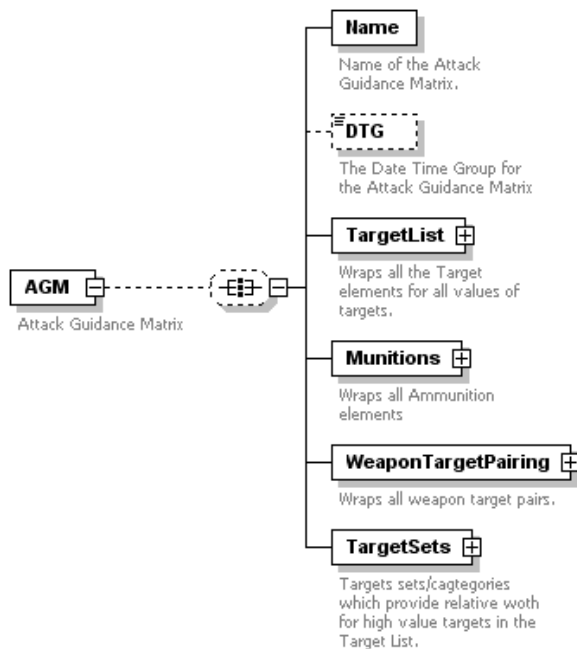


Figure 2: The AGM Data Structure within the MSDL

As denoted by the XMLSPY graphic depiction of the top-level elements the AGM structure includes the following XML elements:

- **Name:** Holds the name of the AGM.
- **Date Time Group (DTG):** Holds the Data Time Group associated with the AGM.
- **TargetList:** Defines the structure of a list of all the identified targets. The targets themselves are XML elements and contain information describing the target type, the priority of the target, and the

expected effect of the attack on the target, and optionally the targeting system.

- **Munitions:** Defines the structure of a list of all munitions associated with the AGM. Each ammunition item within the list uniquely identifies the munition type, a description of the ammunition, number of rounds, munition status, and the type of platform holding the munition.
- **WeaponTargetPairing:** Defines the structure of a list of all weapon (ammunition) target pairs. Each of these items is an element containing a reference to the target, reference to the ammunition, the size and area of the target location, the effect to be accomplished, the timing associated with executing the weapon on target, a Land Air Mobility modifier, the priority, the number of rounds to fire, the fuse type, and any remarks.
- **TargetSets:** Defines the structure of a list of target sets that identify the relative worth of sets of targets.

This effort has provided many benefits, in addition to the AGM data extensions, to the MSDL effort in terms of creating the appropriate information to guide extensions to MSDL. The AGM data model was developed from Army doctrine as specified in FM 6-20-10 “The Targeting Process.”

2.3 OOS Development

The One Semi-Automated Forces (OneSAF) Objective System (OOS) is the U.S. Army’s next generation, composable, entity based simulation system. It is being developed to provide an integral simulation service to the Advanced Concepts and Requirements (ACR); Training, Exercises, and Military Operations (TEMO); and Research, Development, and Acquisition (RDA) domains. With requirements ranging from closed-form analytical support to command-level human-in-the-loop training, OneSAF will be a High Level Architecture (HLA)/Distributed Interaction Simulation (DIS) compliant, entity-level simulation, providing a common simulation system for a broad range of user requirements.

MSDL exists as a Data Interchange Format (DIF) within the OOS Architecture providing the open, well-defined

interface to the transfer of military scenario information between the Military Scenario Development Environment and the OOS simulation core components. Figure 3, shows the OOS Product Line Architecture Framework and highlights the component (MSDE) that creates the MSDL and the component where the MSDL is reposed. During calendar year (CY) 2005 MSDL developers were integral to the OOS iterative and incremental development process. Driving this process was a 10-12 week development and integration (build) cycle where MSDL additions implemented in a previous build were tested during the next integration cycle.

In addition to PM OneSAF's support of SISO's standardization process several MSDL enhancements were prototyped or implemented as part of OOS's 2005 delivery schedule. Three of the major enhancements are described below and include ACSIS team generated modifications to support more robust C4I device connectivity, initial support for unit tasking within the MSDL schema, and a restructuring of the MSDL-related schema files.

2.3.1 Army Command and Control, Communications, and Intelligence and Simulation Initialization System (ACSIS) Coordination

In early 2005, as part of a combined ACSIS and OOS prototype effort, ACSIS engineers enhanced the MSDL schema to allow the OOS C4I adapter and models to initialize from a single MSDL-based data source. The updated schema and ACSIS generated MSDL compliant file were provided to OOS for review. In general the review was favorable; although, the development schedule for OOS did not allow for the prototype MSDL schema modifications to be refined and integrated into a baselined MSDL product. It is expected that the modifications will be integrated in the 2006 timeframe. Figures 4 and 5 show the structures for the newly created PlatformRoleInstance element and the modified CommunicationNetReference elements. Both of these elements are located under the Equipment element within the MSDL schema.

As shown in the XMLSPY-generated figures the new structures provide necessary information to support models initialization consistent with real-world C4I assets. An additional benefit is that the OOS C4I adapter can also use this same information to initialize connectivity services it provides between the models and the real-world C4I devices. It is expected in CY 2006 that collaboration between the ACSIS and MSDL team's will continue and that these elements will be refined and integrated into the base MSDL schema.

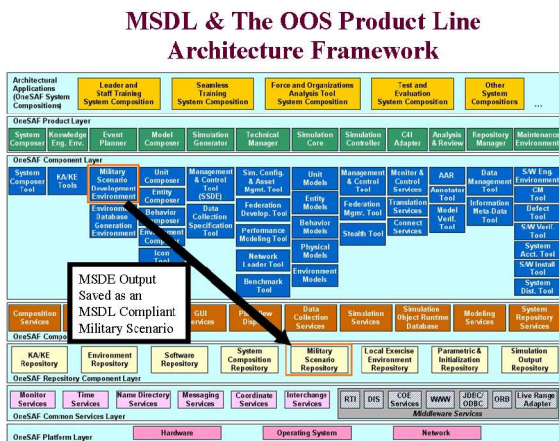


Figure 3: Where MSDL Fits into the OOS Architecture

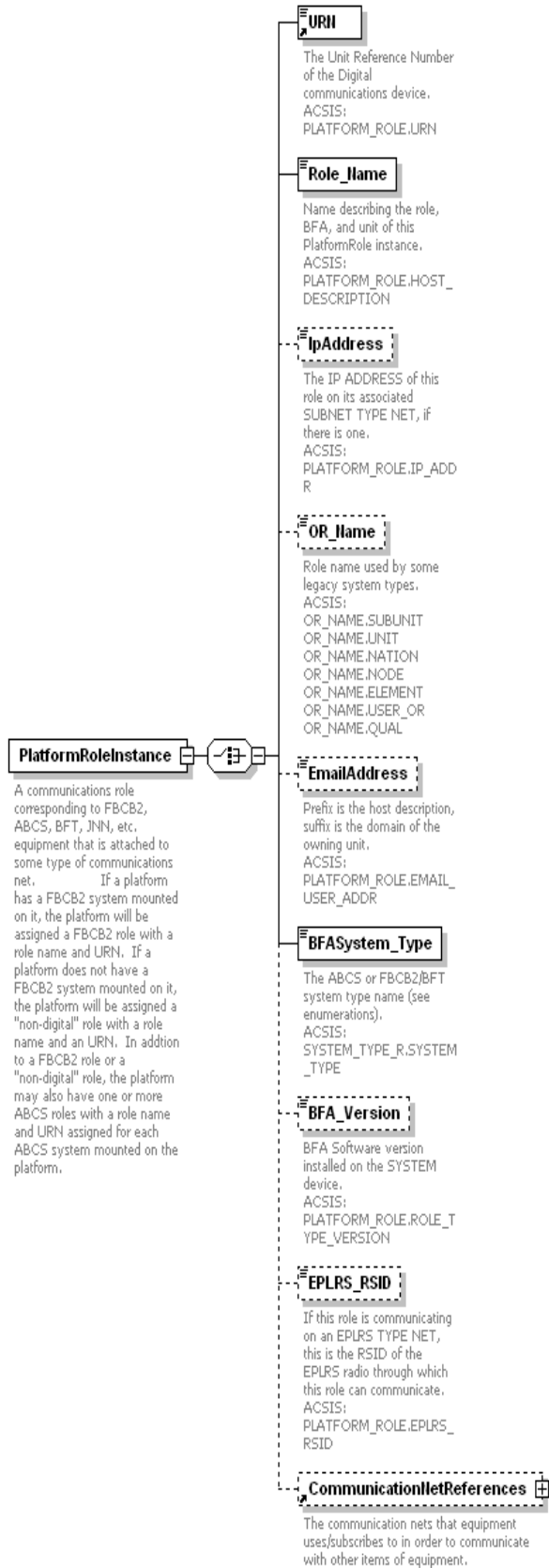


Figure 4: New PlatformRoleInstance Element

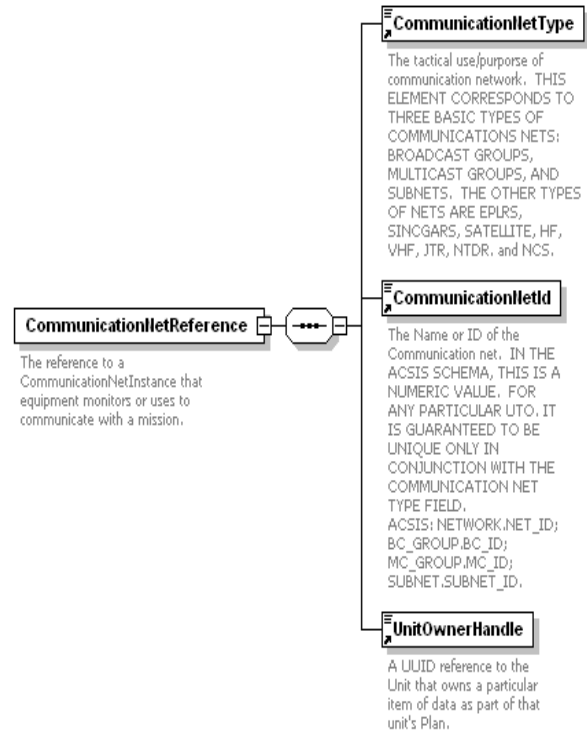


Figure 5: Modified CommunicationNetReference Element

Although the benefits of this effort are not yet realized, it is expected that this centralized consistent approach to initialize the C4I devices, simulation devices, and the necessary adapters will strengthen the analytical and training results when using the resulting system.

2.3.2 Initial Tasking Language Representation:

Also in 2005 an initial tasking language vocabulary was created largely based on the Army Universal Task List (AUTL) and the Command and Control Information Exchange Data Model (C2IEDM). This effort was initiated as a proof of principal as part of an OOS and Army-BML prototyping and demonstration effort that was cancelled in the second quarter of FY05 due to lack of funding. A cornerstone of the MSDL tasking language comes from the the BML "5Ws" – "Who", "What", "When", "Where", and "Why". The units represented within MSDL file provide the Who. The MSDL Task Element discussed below provides the "What". The Trigger data provides the "When" and "Where" the task will be triggered. The Why is provided as part of the Task Element. The Task

Element is central to the tasking language within MSDL. It is composed of three supporting elements:

- The **ActivityType** element is restricted to C2IEDM actions type enumerations and holds a general enumerate explaining the type of activity such as OPERATION_MAIN_EFFORT the tasked organization is performing.
- The **WhyEnumeration** element provides an enumerated value explaining why the task is to be performed and is largely based on values from FM 101-5-1 and includes values such as BLOCK, BREACH, CONTACT_POTENTIAL_THREAT, etc.
- The **Enumeration** element is of type TaskEnumeration and contains an enumerated value from the list of Army tasks within the Army Universal Task List.

The tasks are linked to the remaining “5W” information set by the overarching COAPhase element that describes the unit activities associated with a specific phase of a specific COA. It should be noted this initial tasking language structure provides a starting point for the emphasis of the MSDL phase II standardization effort.

2.3.3 Schema Component Restructure:

A final highlight of the OOS MSDL development activity was an effort to reduce the complexity and number of separate schema files associated with MSDL. Originally the MSDL schema files were segmented largely according to the main elements within the Military Scenario structure and other common or Military standards areas which left 18 separate schema files to manage and track. The original 18 files included:

MilitaryScenario.xsd: Defines the overall military scenario structure and references to the other subschema definitions.

CommandRelations.xsd: Defines the structure to hold the command relations of an MSDL compliance military scenario.

CommonElements.xsd: Defines the common elements that are used by the other schemas defining the MSDL.

CoursesOfAction.xsd: Defines the structure to hold the Coarse of Action structure that provides task and plan information to entities within the scenario.

Plan.xsd: Defines the structure in addition to the COA structure to hold planning data associated with an MSDL compliant military scenario.

Environment.xsd: Defines the structure to hold environment data of an MSDL compliant military scenario.

Equipment.xsd : Defines the structure to hold equipment data of an MSDL compliant military scenario.

ForceSide.xsd : Defines the structure to hold force and side data of an MSDL compliant military scenario.

Installations.xsd : Defines the structure to hold military installation data of an MSDL compliant military scenario.

METOC.xsd : Defines the structure to hold weather-related data of an MSDL compliant military scenario.

MILSTD2525B.xsd : Defines the structure to hold 2525B-related data of an MSDL compliant military scenario.

MOOTW.xsd : Defines the structure to hold tactical graphics associated with Military Operations Other Than War of an MSDL compliant Military Scenario.

Options.xsd : Defines the structure of resolution, scenario standards, enumeration sets, and coordinate systems to be used within as part of an MSDL compliant military scenario.

Overlays.xsd : Defines the structure of overlay data of an MSDL complaint military scenario.

TacticalGraphics.xsd: Defines the structure for tactical graphics that link overlays to specific units of an MSDL compliant military scenario.

Threats.xsd: Defines the structure to provide the relationship of threats to forces and sides, MOOTW activities, and weather data of an MSDL compliant military scenario.

Units.xsd: Defines the structure of unit data of an MSDL compliant military scenario.

dcr.xsd: Defines the structure of the Data Interchange Format Control Record (DCR) that must accompany all OOS DIFs.

As part of the redesign all of the schema elements except the MilitaryScenario element were moved into the msdlElements.xsd file. Three new enumeration files were developed to restrict the values associated with the military units and equipment within the scenario. The final structure segments the overall schema into 5 files:

MilitaryScenario.xsd Defines the overall military scenario structure and references to the other subschema definitions.

UnitEnumerations.xsd: Provides the valid unit enumerations, based on the 2525B hierarchical representation of Army units, to be used when defining the task organizations within an MSDL compliant military scenario.

EquipmentEnumerations.xsd: Provides the valid equipment enumerations, based on the 2525B hierarchical representation of equipment, to be used when defining the equipment within an MSDL compliant military scenario.

TaskEnumerations.xsd, Provides the valid task enumerations, based on the AUTL, to be used when defining COA data within an MSDL compliant military scenario.

msdlElements.xsd Defines all of the elements referenced by the MilitaryScenario element.

The resulting schema has been posted to the SISO and onesaf.org websites and is the basis for the version .01 MSDL specification vetted through the MSDL Study Group.

3 Planned Standardization Efforts in 2006

The MSDL Study Group plans on transitioning to a Product Development Group in 2006. Initial plans, based upon receiving the Standards Activity Group (SAC) and Executive Committee (EXCOM) approval, are to hold a joint MSDL and C-BML Product Development Group kickoff meeting at the end of February to establish officers and operating procedures.

Shortly thereafter, a second meeting is planned during the Spring SIW in Huntsville, AL, to continue the technical work initiated by the MSDL Study Group to create a ballotable version of the MSDL specification.

The goal for the first year of the PDG is to successfully ballot an initial version of the MSDL standard specification focusing on ground combat and to be consistent with NATO standardization efforts. The initial product will include a content schema along and the associated implementation rules and guidelines. In following years, the MSDL standardization product will be refined on a 2 year schedule consistent with the C-BML standardization schedule.

It is expected that this initial PDG effort will collaborate and align with respect to the following: [2]

- The 5 Ws concept with potential extensions to support Which and How attribution.
- The naming conventions and types of units, activities, interactions, and relationships.
- The initial rules and guidelines for structure, content, and format.
- The initial context for structure, content, and format (NATO, Land, etc.).
- Continued collaborative development of the formal grammar.

Phase II of the MSDL & C-BML Product Development Group is anticipated to start in CY07. During this phase both groups refine and extend their initial PDG processes and products. The focus of the collaboration shifts to extending the context of entities, activities, relationship, and interactions and grammar development. It is expected that this second PDG efforts will collaborate and align with respect to the following:

- Refinement and extension of the naming conventions for units, activities, interactions, and relationships.
- Refinement and extension of the standardized rules and guidelines for structure, content, and format.
- Refinement and extension of the standardized context for structure, content, and format.
- Initial standard-based definition of the formal grammar.

1 Summary

2005 was a busy year for MSDL. It included SISO standardization related activities as well as PEO-STRI (OOS) and RDECOM (MATREX) related development activities. 2006 promises to be even busier with an aggressive lean toward SISO standardization and balloting while at the same time managing additional functional related enhancements to support a more robust distributed and standalone simulation initialization systems and more robust C4I connectivity.

2 Acknowledgements

The authors would like to thank the specific MATREX and ASCIS team members for their collaborative efforts.

3 References

- [1] <http://www.w3.org/XML>.
- [2] Surdu, John; Wittman, Robert; Abbot, Jeff. (2005) "Military Scenario Definition Language Study Group Final Report", 05F-SIW-017, 2005.
- [3] MSDL Study Group Terms of Reference, <http://www.sisostds.org/index.php?tg=fileman&idx=list&id=31&gr=Y&path=>, 2005-004-06.
- [4] Prochnow, David; Fogus, Michael; Vintilescu, John; Benton, Borum. (2005) "Initialization of Distributed Simulations: A Better Way?", 05F-SIW-073, 2005.
- [5] Thakkar, Naimish; Lindy, Erica. (2005) "Initialization of the MATREX Environment using MSDL", 05F-SIW-016, 2005.

Washington State University, a M.S. in Software Engineering from the University of West Florida, and a Ph.D. in Industrial Engineering from the University of Central Florida.

JEFF ABBOTT is a Senior Software Engineer at AcuSoft. He has been involved in the OneSAF Objective System program in the capacity of developing the Military Scenario Development Environment, a military scenario planning capability. Mr. Abbott was a lead designer of the initial OneSAF MSDL specification.

Author Biographies

ROBERT L. WITTMAN JR. currently works for the MITRE Corporation supporting the OneSAF program. He has been part of the U.S. DoD M&S community since 1990. He holds a B.S. in Computer Science from