2009

Applying Model-Based Data Engineering to Evaluate the Alignment of Information Modeled Within JC3IEDM, MSDL, and MATREX-FOM

Anil Ustun
Old Dominion University, austu001@odu.edu

Andreas Tolk
Old Dominion University, atolk@odu.edu

Saikou Diallo
Old Dominion University, sdial002@odu.edu

Original Publication Citation

Ustun, A., Tolk, A., & Diallo, S. (2009). *Applying model-based data engineering to evaluate the alignment of information modeled within JC3IEDM, MSDL, and MATREX-FOM.* Paper presented at the 2009 Fall Simulation Interoperability Workshop, Orlando, FL.

Applying Model-based Data Engineering to Evaluate the Alignment of Information modeled within JC3IEDM, MSDL, and MATREX-FOM

Anil Ustun & Andreas Tolk
Department of Engineering Management and Systems Engineering
Old Dominion University
Norfolk, VA 23529
United States

austu001@odu.edu, atolk@odu.edu

Saikou Diallo
Virginia Modeling Analysis & Simulation Center
Old Dominion University
Norfolk, VA 23529
United States

sdiallo@odu.edu

Keywords:
Joint Consultation, Command, and Control Information Exchange Data Model (JC3IEDM), The Military Scenario Description Language (MSDL), The Modeling Architecture for Technology, Research, and Experimentation (MATREX) Federation Object Model (FOM), Model-based Data Engineering (MBDE)

ABSTRACT: *The need for a common representation of entities and their relations to support the easier composition and federation of independently developed solutions in support of the user has been identified and addressed in several papers presented during recent simulation interoperability workshop. One of the underlying assumptions is that standards derived from the same conceptual domain can easily be converted into each others, as they deal with the same concepts. In a project conducted for the U.S. Army's Program Executive Office (PEO) Soldier, three of such solutions for military operations (with focus on the land forces) were utilized to capture the underlying concepts of land warfare: the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM), the Military Scenario Description Language (MSDL), and the Modeling Architecture for Technology, Research, and Experimentation (MATREX) Federation Object Model (FOM).*
When we applied the methods of Model-based Data Engineering (MBDE) we observed, that these three standards are not conceptually as well aligned as we assumed. We identified several significant gaps. The findings of this paper will contribute to support designers, engineers and project managers in a better way to understand, (1) which data are needed operationally, (2) how gaps can be identified regarding supporting standards, (3) how the gaps can be closed, and (4) what data transformation must be conducted when dealing with different standards in data-rich integration projects to ensure cost-efficient and operationally effective solutions.

1. Introduction

The PEO Soldier program was instituted to evaluate the use of M&S to support the procurement process and to provide a more cost efficient way to solve complex R&D problems. A review of existing simulations showed that there is no single model that fully represents the acquisition process, thus the need for a federated approach. In the initial phase of the program an integrated solution using different standards and existing simulations showed how M&S can be used in the acquisition of BA.

It is often assumed that standards derived from the same conceptual domain can easily be converted into each others, as they deal with the same concepts. Applying the methods of Model-based Data Engineering (MBDE) shows that these three views are not conceptually aligned and where the gaps are. PEO Soldier project as well as each of these standards will be introduced in more detail in the sections below.

1.1 U.S. Army's PEO Soldier's Project

The primary task of the PEO Soldier federation model [1] is to produce appropriate data for in-depth analysis to make an intelligent decision on which type of body armor (BA) to procure. In order for the correct data to be generated several battle situations and phases have to be predetermined to cover all aspects of battle for any given soldier.

In the first phase of the project, the modeling teams worked together to develop a federated real-time capability between the heterogeneous simulation models using the Research, Development and Engineering Command's (RDECOM) Modeling Architecture for Technology, Research, and Experimentation (MATREX) simulation architecture.

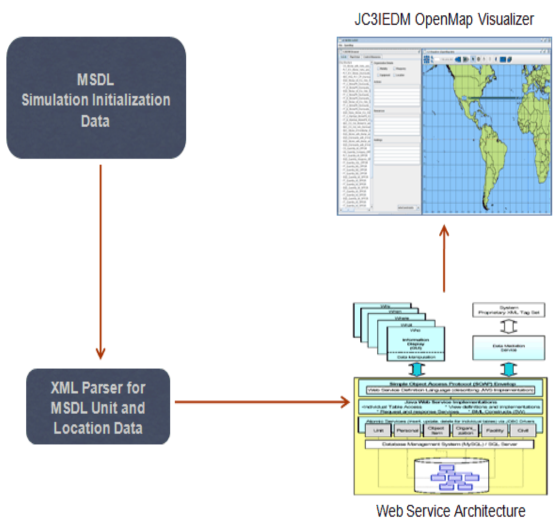

Figure 1: Components of the PEO Soldier Project

Within this project, our focus was on displaying the MSDL (Unit and Location) and MATREX FOM (Situational Awareness) data on the JC3IEDM OpenMap visualizer. The main challenge was how to consistently integrate the data from these three heterogeneous sources: JC3IEDM, MSDL and FOM.

Figure 1 shows the starting point and steps of the project. The reference model utilized in the development of the prototype is the JC3IEDM, which is maintained and distributed by the Multilateral Interoperability Program (MIP) as its reference model. The web service architecture is built to facilitate access to the five W components of the JC3IEDM. The overall JC3IEDM data model is implemented as a MySQL database. The JC3IEDM Openmap tool is very easy to use and facilitates many methods of entity visualization. Icon sets can be fed to the tool via external services/servers. The default icon set distributed with the tool is the military standard 2525 symbol set. These icons are stored locally and are easily buried under a service which returns the icon file given a string identifier. XML Parser software basically reads the necessary data from the target MSDL file and pushes the data to the JC3IEDM viewer via web servers to be rendered on a world map. More details can be found in this paper [1]. This paper focuses on the mapping of three heterogeneous sources in a consistent way. The following sections will present a short description of these sources.

1.2 Standard for Information Exchange Data Model for the Sharing of C2 Information: JC3IEDM

The JC3IEDM is developed by North Atlantic Treaty Organization's (NATO) Multinational Interoperability Programme (MIP) to represent military situations in order to support communication and interoperability among NATO forces. JC3IEDM is a very detailed and comprehensive data model as it consists of 289 entities, 396 relationships between entities, 1729 entity attributes and nearly 7000 value codes. MIP Data Model impressively captures the elements of battlespace objects and features, their properties, and situations made up of facts about objects and activities involving collections of objects [2]. In addition, JC3IEDM allows the representation of the land, sea, and air as well as certain aspects of the communications infrastructure. It represents nearly all objects of interest including organizations, persons, equipment, facilities, geographic features, weather, capabilities, and military control measure such as boundaries.

1.3 Standard for Scenario Initialization: MSDL

This is being accomplished by utilizing scenario development tools such as the Military Scenario Development Environment (MSDE) [3] tool, which allows a subject matter expert SME to construct a scenario, complete with task organization, order of battle, and force lay down on a terrain. MSDE exports all of this data into an XML schema language called the Military Scenario Definition Language (MSDL) [3] which is a simulation environment independent Extended Markup Language (XML) format thus enabling simulation systems to share their scenarios with other simulation and Control, Communications, and Computers and Intelligence (C4I) systems. MSDL includes the information which is current situation (t_n) data and course of action (COA) (t_{n+1}) of a military scenario. For instance, the military scenario data comprise of unit task organization data, force and side data, unit and entity locations, tactical graphics and overlays, environment and geographic location data, initial weather conditions, and planned tasking data.

1.4 Environment for Information Exchange at Runtime: MATREX

In 2003 the U.S. Army started a program called MATREX and developed a standard Federation Object Model (FOM) to be used in all future and current standalone and federation simulations [4]. 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]. PEO Soldier is one of the few that have utilized the standard MATREX FOM into their models among many of the Army's older existing simulation programs.

In order to support MSDL's role in the MATREX environment data element extensions to MSDL's XML schema were necessary. MATREX engineers worked to identify necessary elements and propose these elements for inclusion into the MSDL schema. It is expected that additional collaboration and MSDL extensions to be continued in support of MATREX throughout the following years. Next section briefly describes the engineering approach we used for the mapping process of the three sources.

1.5 Data Management and Alignment

In model-based data engineering (MBDE) [5], the reference model essentially serves as the common language. The data structures describing military operations are too complicated to be handled, managed, and mapped automatically. Therefore, when information is exchanged, we have to ensure semantic consistency with knowing the services at definition and implementation time. Previous efforts have been limited to individually designed point-to-point interfaces. Literature surveys of interoperability have indicated that the mapping problem is an n2 problem: Whenever a new system is introduced, it must be mapped to every potential partner. If all participating systems use a common reference model, this effort is theoretically reduced to an n problem: The new system must align only with the reference model, rather than each participating partner. Figure 2 shows the five-step MBDE integration process [5]:

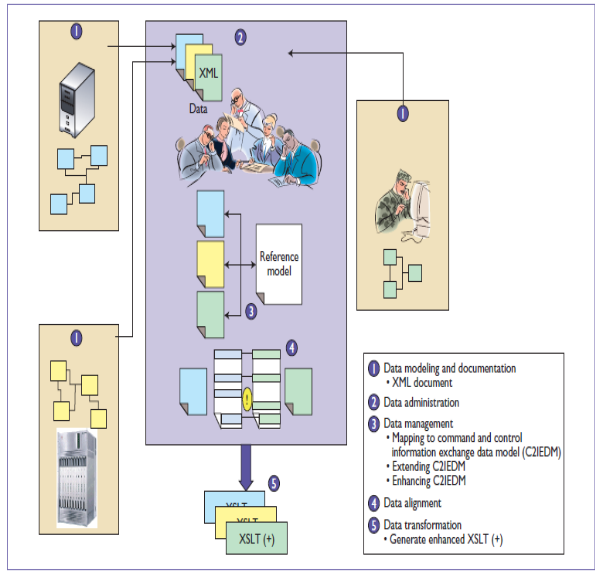

Figure 2: Steps of the MBDE integration

1. Data modeling and documentation: Service developers or providers from participating systems use XML to model and document data and document their interfaces.
2. Data administration: Data administrators in participating organizations collect and store all XML documents using UDDI or alternatives.
3. Data management: The organization's data management agency uses the common reference-information-exchange data model to unambiguously define all data elements' meaning, resulting in a mapping of the target XML tag set to the standardized XML tag set.
4. Data alignment: The organization's data management agency compares the data deliverers' supported tag sets with the requested data consumer's tag sets. If all requested tag sets
can be delivered, there's no problem; otherwise, the data can't be obtained from that source.
5. Data transformation: Based on the results, service providers can automatically document the mapping as enhanced XSLT documents. This results in configuration files for software layers, and hence eliminates the source of ambiguous interpretation of documents by developers.

The following sections briefly explain mapping efforts and identify the type of issues faced in the process.

2. Methodology

To perform any sort of data modeling a methodical approach is required. The proposed methodology is described below. Details regarding the other suggested methodologies can be found in this paper [6].

2.1 Top-Down versus Bottom-Up Data Mapping

To deal with the assignments required for data mapping, the general idea of properties, propertied concepts, and associated concepts can be applied [6].

- *Property values* are the allowed values for a specifying characteristic. Of particular interest are enumerations. Within relational databases, these are enumeration values for attributes.
- *Propertied concepts* are a collection of specifying characteristics for an entity in the domain of knowledge. In ontologies using data models to structure its information, this can be mapped to tables and their attributes. Within relational databases, these are tables.
- *Associated concepts* are semantic entities in which data is given in a broader context. Within data models, these are the views or replication domain sets.

Both MSDL and JC3IEDM have the ability to define objects that may appear within a battle space, as well as describing its relationships to other objects. This project describes an initial effort to implement the method of data engineering to a real world problem. Generally, two approaches are available for mapping tasks [6]:

- The *Top-Down* approach starts with the associated concepts (meaningful subsets) and maps them to each other. Next, the tables are mapped and finally missing properties identified. The advantage is that common knowledge and structure of the mapping domain is used to reduce the complexity by dividing the mapping problem into subsequent classes. The disadvantage is that the associated concepts and propertied concepts can differ significantly.

- The Bottom-Up approach starts with the mapping of the properties. This means starting with the smallest common denominator. How these properties are structured into propertied concepts and associated concepts doesn't matter in the first step. Only in the second step, when the properties have to be tied together for obtainability in the applications, the propertied concept descriptions may have to be completed and associated respectively.

In the end, the team agreed to use the top-down approach for the mapping process.

3. Implementation

3.1 Applying Model-Based Data Engineering

For the mapping process, data must be captured and its structure must be mapped to the reference model data sets that have the same definition. This implementation will require an analysis on the correlation between data fields in the JC3IEDM, MSDL and FOM message formats. Steps of proposed mapping process using MBDE is shown in Table 1. According to the steps proposed, primarily we have to look for the key similarities and data sets between the standards. After identifying the objects, equivalent information expressions should be mapped to each other. After the mapping process, we should be able to spot the gaps and fill the missing values. In some cases, equivalent data sets would not have the same data structure; therefore transformation of the data structure is also required.

Mapping Steps	Detail
Data Administration in support of requirements	Identification of the objects, missing tag sets, values.
Data Management in support of semantic integration	Identification and description of the data elements, and mapping equivalent information expressions to each other.
Data Alignment in support of semantic integrity	Comparison of the mapped model's tag sets with every tag set in the target data model. Result is awareness of the gaps and taking actions to fill these gaps.
Data Transformation in support of implementation	Technical process of transforming the systems' information to match information exchange requirements.

Table 1: Steps for Data Mapping

3.2 Conceptual Mapping: Identifying Key Similarities

3.2.1 Conceptual Definitions: JC3IEDM

This paper focuses on the Unit and Location information. At the center of the mapping is the OBJECT-ITEM. This class is used to implement specific instances of objects described in the messages. There are five subclasses of OBJECT-ITEM, although for our mapping it would be sufficient to just have the subclasses ORGANISATION (used to define military organisations and reporting units). The OBJECT-ITEM class is paralleled by the OBJECT-TYPE class, which also has five subclasses as shown in the figure; again only the ORGANISATION-TYPE (along with its subclasses GOVERNMENT-ORGANISATION-TYPE, MILITARY-ORGANISATION-TYPE and UNIT-TYPE) subclasses are needed for our mapping process.

- OBJECT-TYPE is used for more static information associated with an entire class of objects (i.e., the values of the attributes are not likely to change very often over time)
- OBJECT-ITEM is used to capture information specific to individuals (e.g., dynamic, the speed of a tank, the fact it has 5 gallons of gas).

The OBJECT-TYPE of an OBJECT-ITEM describes the object's inherent characteristics. The association between an OBJECT-ITEM instance and its OBJECT-TYPE is achieved through the use of an OBJECT-ITEM-TYPE instance. Data characteristics are entered either on the item side or the type side as appropriate. Any characteristic described on the type side also applies to the item when the item is assigned a type classification [2, 7, 10].

The key elements of an instance of an OBJECT-ITEM referring to a Unit are the Unit's affiliation (e.g., Turkey), its type (e.g., Artillery), its status (e.g. friendly), and its position information (e.g., location, heading, speed, etc). This information is captured in associated instances of the UNIT-TYPE, OBJECT-ITEM-AFFILIATION, OBJECT-ITEM-TYPE, OBJECT-ITEM-STATUS, and OBJECT-ITEM-LOCATION classes, respectively [7, 10].

LOCATION class is required to represent Unit's position. In JC3IEDM, every object can be assigned a location via an OBJECT-ITEM-LOCATION. We suspect that the only other subclasses that might be needed to represent the Unit's position data are POINT, ABSOLUTE-POINT and GEOGRAPHIC-POINT, which would be needed, for example, to represent Unit's geodetic (latitude/longitude). Table 2 shows some of the necessary JC3IEDM tables to describe the Units and their Location..

Unit	Location	Linkage
Object-Type Object-Item	Location Point Absolute Point Geographic Point	Object-Item-Type Object-Item-Location

Table 2: JC3IEDM Tables for Unit and Location

The affiliation of a specific Unit is defined using an instance of the OBJECT-ITEM-AFFILIATION class that references an instance of AFFILIATION. For the sample data all AFFILIATIONS are from the subclass AFFILIATION-GEOPOLITICAL that includes instances for the nationalities of Belgium (BEL), Germany (DEU), Ghana (GHA), Spain (SP), Turkey (TUR), United States (USA) and an unspecified nation symbolized as "NOS" (Not otherwise specified).

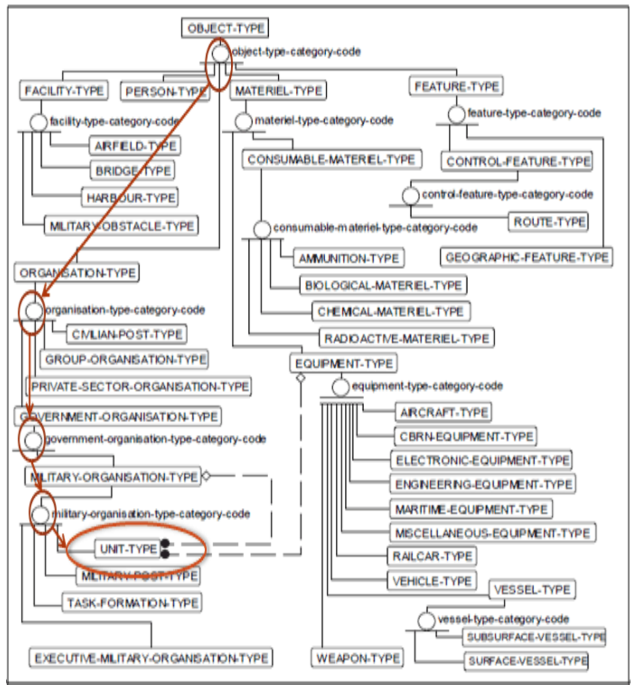

Figure 3: Data Management Process

JC3IEDM is intended to allow a common representational structure for enemy and friendly troops. The OBJECT-ITEM-HOSTILITY-STATUS for an OBJECT-ITEM specifies its hostility code status. Object-Item-Hostility-Status-Code attribute, which is HO for Hostiles, FR for Friendly, and so on for other statuses of forces (e.g. suspect, neutral, unknown, etc.). Enemy actions are expressed in the same way as friendly actions.

Figure 3 shows how we are arrive to the Unit-Type table. Object-type-category-code is the main attribute that separates OBJECT-TYPE into sub-categories. Filling the correct category-codes will lead you to the desired tables.

3.2.2 Conceptual Definitions: MSDL

An MSDL formatted document includes all of the data necessary to define the initial conditions of a military scenario and is defined using XML, thus independent of any other programming language, database, or application. The MSDL schema files are segmented into five files [3]:

- 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 military scenario element.

In our case, we are focused in Unit.xsd data which is under the MilitaryScenario.xsd file. This structure must be mapped to the JC3IEDM data sets that have the same definition. The results of this analysis will be a series of mapping processes that each connecting the MSDL's Unit.xsd dataset to the corresponding JC3IEDM dataset [3]. Some of the important attributes of each Unit are:

- The *ObjectHandle* uniquely identifies a unit.
- The *Name* provides readable name for each unit.
- The *UnitSymbolModifiers* represents Unit Symbol.
- The *SymbolId* provides the MIL STD 2525B Symbol Identifier for the Unit from the War fighting track.
- The *CommunicationNetInstance* provides the communication nets that the unit uses to communicate with other units during a mission.
- The *Location* element represents the data type for MSDL Coordinate.

3.3 Aligning MSDL with the JC3IEDM

The starting point for mapping is going to be the *SymbolId*. This is a symbol identification coding (SIDC) scheme which is a 15-character alphanumeric identifier that provides the information necessary to display or transmit a tactical symbol between MIL-STD-2525B [8] compliant systems. Positions of this scheme will be used to fill the tables of JC3IEDM.

The positions of the *SymbolId* code are described below [8]:

- A dash (-) is used to fill each unused position.
- An asterisk (*) indicates positions that are user defined based on specific symbol circumstances, such as affiliation or echelon/mobility.

For more in-depth understanding of this mapping process, let's take a look at one of the Symbol ID that is identified in the Unit.xsd file. The definition of each position is described below:

S F G P U C I - - - - E U S -

- Position 1: S indicates that this symbol belongs to warfighting.
- Position 2: F indicates that the symbol is friendly. Position 2 is also going to identify the category-code of JC3IEDM's AFFILIATION-GEOPOLITICAL table.
- Position 3: G indicates that this is a ground symbol. Position 3 is also going to identify the unit-type-general-mobility-code of JC3IEDM's UNIT-TYPE table.
- Position 4: Status indicates the symbol's planned or present status. P indicates that this symbol's status is present.
- Positions 5 through 10: Function ID, identifies a symbol's function. Each position indicates an increasing level of detail and specialization.
  - U indicates that this symbol is a unit. Therefore, Position 5 is also going to identify the category-code of JC3IEDM's MILITARY-ORGANIZATION-TYPE table and category-code of the JC3IEDM's ORGANIZATION table.
  - C indicates that this symbol is a combat unit. This also identifies the unit type category code of JC3IEDM's UNIT-TYPE table. A UNIT-TYPE who closes with and destroys enemy forces or provides firepower and destructive capabilities in the battlespace.
  - I indicates this symbol is a infantry. This is also the specific value that represents the designation of a military branch for a particular UNIT-TYPE. A UNIT-TYPE whose principal designation is the employment of non-mechanized or lorry-(truck-) borne infantry.
  - Positions 8,9,10 are unused position.

- Positions 11 and 12: Symbol modifier indicator, identifies indicators present on the symbol such as echelon, feint/dummy, installation, task force, headquarters staff, and equipment mobility.
  - Position 11 is unused.
  - E indicates that symbol's echelon is a company. This also identifies the unit-type-size-code of UNIT-TYPE. The specific value that represents the relative size of the commonly accepted configuration of military formations.
- Positions 13 and 14: Country code identifies the country with which a symbol is associated.
  - US indicate that symbol is associated to United States. This also attribute (affiliation-category-code) of the JC3IEDM AFFILIATION table.
- Position 15: Order of battle, provides additional information about the role of a symbol in the battlespace.
  - Position 15 is unused.

Figure 3 shows the Data Alignment process conducted to some of the JC3IEDM tables. First column shows the name of the entities. Second column lists the attributes associated with each table. Third column defines the data type of each attribute. Fourth column aligns each Unit's SIDC code to corresponding JC3IEDM physical value. As you can see from the figure, some values are generated in the code. The reason for this is that MSDL Unit does not have a long type unit-type-id. Instead it has the ObjectHandle string type data that identifies each Unit.

Therefore the code should generate those ids's (Primary Key/Foreign Key) that is necessary for the database. Last column shows that if that attribute is mandatory or optional optional. If the attribute is optional, it is up to you to fill that value or not. As you can see, we are now able to locate the missing data values. Category-codes are the main attributes of each entity and defined by SIDC positions of MSDL.

3.4 Aligning MATREX FOM with the JC3IEDM

The primary principle of FOM is to develop a reusable software component – a model that describes: object classes, attributes, and interaction classes – that reduces development time and allows software engineers and programmers to easily understand another's object model without having an in-depth knowledge of the inner workings of the other simulation model. The FOM also provides information on the capabilities of a federate to exchange information and communicate inside a federation [9].

A class diagram for the MATREX FOM Structure has been manually generated as illustrated in Figure 4. Exemplified diagram is conceptual and some details are not necessary for our example. Once a FOM is in the class diagram format it can be modified using the graphical editor. For this example, class diagram was produced with the MagicDraw™ UML modeling and CASE tool on a Windows XP platform. Class structure reflects the MATREX FOM version 3.0 hierarchy. For the purpose of denoting attribute and parameter usage within PEO Soldier attributes are placed under child objects.

Entity	*Attributes*	*Data Type*	*Mapping Result (JC3IEDM↔MSDL)*	*Opt*
OBJECT-TYPE	object-type-id (PK)	long	Generated in the code	MA
	object-type-category-code	String	OR ↔ SIDC Position 5	MA
	object-type-decoy-indicator-code	String	Generated in the code	MA
	object-type-name-text	String	Name (MSDL)	MA
ORGANISATION-TYPE	organisation-type-id (PK) (FK)	long	Generated in the code	MA
	organisation-type-category-code	String	GVTORGSIDC ↔ Position 5	MA
	organisation-type-command-and-control-category-code	String	null	OP
	organisation-type-command-function-indicator-code	String	YES	MA
	organisation-type-description-text	String	null	OP
GOVERNMENT-ORGANISATION-TYPE	government-organisation-type-id (PK) (FK)	long	Generated in the code	MA
	government-organisation-type-category-code	String	MILORG ↔ SIDC Position 5	MA
	government-organisation-type-main-activity-code	String	null	OP
MILITARY-ORGANISATION-TYPE	military-organisation-type-id (PK) (FK)	long	Generated in the code	MA
	military-organisation-type-category-code	String	UNIT ↔ SIDC Position 5	MA
	military-organisation-type-service-code	String	ARMY ↔ SIDC Position 5	MA
UNIT-TYPE	unit-type-id (PK) (FK)	long	Generated in the code	MA
	unit-type-arm-category-code	String	INFANTRY ↔ SIDC Position 7	MA
	unit-type-arm-specialisation-code	String	null	OP
	unit-type-category-code	String	COMBAT ↔ SIDC Position 6	MA
	unit-type-general-mobility-code	String	LAND ↔ SIDC Position 3	OP
	unit-type-principal-equipment-type-id (FK)	String	null	OP
	unit-type-qualifier-code	String	null	OP
	unit-type-size-code	String	COMPANY ↔ SIDC Position 12	MA
	unit-type-supplementary-specialisation-code	String	null	OP
	unit-type-supported-military-organisation-type-id (FK)	long	null	OP

Figure 4: Data Alignment Process

In the MATREX FOM, SAObject is responsible for the perceived truth entities maintained on the situational awareness network; SpottedObject is the generic attributes for spotted objects, both friendly and not; Target represents a target in the SA picture; Friend represents a friendly unit in the SA picture. In addition, the SAobject has the attributes; CellID represent the ID of the sender/updater of this object, ForceID represents the Friendly and foe type, Location is represented in the format of latitude, longitude and altitude, Timestamp is the simulation time, in seconds, at which the updated attributes for this object are valid. Information such as force size, entity or aggregate side (friend, foe), status, location, speed and bearing should be the sorts of events that we want to subscribe and capture and mapped to the correct data field.

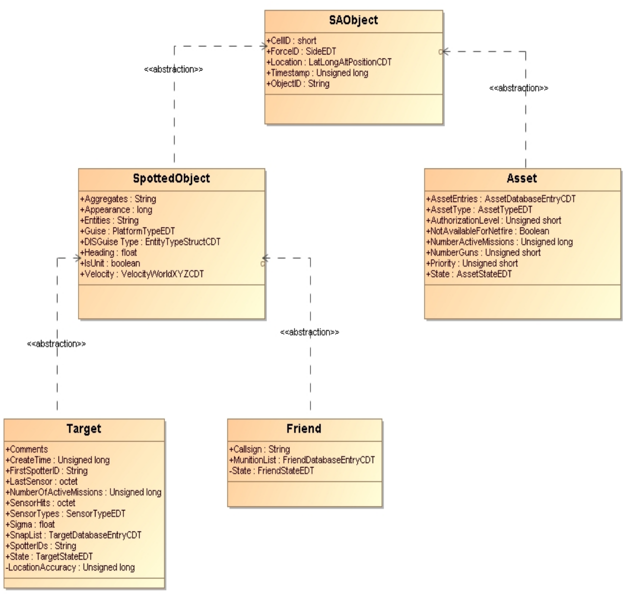

Figure 5: Data Management Process

4. Results

Within the mapping process, data instances passed back and forth between the systems were not matched in the sense of a data structure and representation/syntax. In this case the data mismatch has to be addressed and our reference model should be extended accordingly. The common mismatch problems and other important results can be seen as below:

- *Structure Mismatch:* Diverse relationship between the FOM Class diagram and JC3IEDM schema is causing problems. Therefore sometimes it might not be possible to capture every possible object, attribute, and interaction from the FOM.
- *Syntax Mismatch:* Correlation between data fields in FOM and JCEIDM does not match. For instance, FOM and JC3IEDM represent the Location information in different formats of latitude, longitude and altitude data fields.
- JC3IEDM does not allow the representation of the current formation of a Unit, unlike in MSDL.
- MSDL is using the Military Grid Reference System (MGRS) which is a geocoordinate standard used by NATO militaries for locating Unit points on the earth. However, JC3IEDM only supports the representation of geodetic (latitude/longitude) values. In order to map the Unit's locations from MSDL to JC3IEDM, primarily the values should be converted to geodetic data. In this case, data transformation process is recommended.
- JC3IEDM requires long type unique ID's for each Unit in order to distinguish each table. MSDL does not support this.

5. Conclusion

This paper documents the successful application of MBDE in support of data gap analysis. These results are currently used to identify solutions to close these gaps and recommend respective changes to participating organizations as well.

6. References

[1] A. Ustun, S. Diallo "Modeling and Simulation in the Procurement Process: An Update on PEO Soldier," Proceedings of the M&S Capstone Conference, Old Dominion University, April, 2009.

[2] B. Ulicny, C. J. Matheus, G. Powell, R. Dionne & M. M. K,okar "The Representability of METT-TC Factors in JC3IEDM" Twelfth International Command and Control Research and Technology Symposium, June, 2007.

[3] R. L. Wittman & J. Abbott "Keeping up with the Military Scenario Definition Language (MSDL)", Spring 2006 Simulation Interoperability Workship, 06S-SIW-018

[4] E. Lindy & N. Thakkar "Initialization of the MATREX Environment using MSDL", Spring 2005, 05F-SIW-016

[5] A. Tolk, & S. Y. Diallo, "Model-based data engineering for Web services" Internet Computing, IEEE Volume 9, Issue 4, July-Aug. 2005 Page(s): 65 – 70

[6] C. Turnitsa, S. Kovurri, A. Tolk, L. DeMasi, V. Dobbs, W. Sudnikovich "Lessons Learned from C2IEDM Mappings within XBML," Paper 04F-SIW-111, 2004 Fall Simulation Interoperability Workshop, Orlando, Florida, September 2004

[7] The Joint C3 Information Exchange Data Model Overview, MIP-NATO Management Board (MNMB).

[8] DOD MIL-STD-2525 Common Warfighting Symbology, Defense Information Systems Agency (DEPSO)

[9] R. Dumond and R. Little, "A Federation Object Model (FOM) Flexible Federate Framework," Technical Note, Carnegie Mellon Univ., Pittsburg, PA, 2003.

[10] R. Wittman & S. Lopez-Couto, "Exploring Simulation and JC3IEDM alignment: A OneSAF Example", SimTecT 2007 Simulation Conference: Simulation - Improving Capability and Competitiveness (SimTecT 2007)

Author Biographies

ANIL USTUN is pursuing a Ph.D. in Engineering Management in the Department of Engineering Management and Systems Engineering at Old Dominion University. He received a M.S. degree in Information Systems from the Informatics Institute at the Middle East Technical University, in Ankara, Turkey. His research interest is in the domain of combat modeling and simulation (M&S) and the relation to command and control systems (C2S). He conducted the mapping and research.

SAIKOU Y. DIALLO is a Ph.D. candidate and Senior Project Scientist at the Virginia Modeling Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU). He received his B.S. in Computer Engineering (2003) and his M.S in Modeling & Simulation (2006) from ODU. His Ph.D. research focuses on the domain of Model Based Data Engineering and Web Services for M&S applications. He was responsible for the technical implementation.

ANDREAS TOLK is Associate Professor in the Faculty for Modeling, Simulation, and Visualization at the Engineering Management Department of the College of Engineering and Technology at Old Dominion University (ODU) of Norfolk, Virginia. He is affiliated with the Virginia Modeling Analysis & Simulation Center (VMASC). His domain of expertise is the integration of M&S functionality into real world applications based on open standards. He received a Ph.D. and an M.S in Computer Science from the University of the Federal Armed Forces in Munich, Germany.