Primitives of Meaning in Support of Interoperability

Charles Turnitsa  
(cturnits@odu.edu)
Andreas Tolk  
(atolk@odu.edu)

Virginia Modeling Analysis and Simulation Center  
Old Dominion University  
Norfolk, VA

Robert Kewley  
(Robert.Kewley@usma.edu)

Operations Research Center  
US Military Academy  
West Point, NY
Presentation Roadmap

- Introduction
- Problem Addressed
  - Semantic Misalignment
- Primitives of Meaning
  - What are they, and how do they relate to the Problem
- Experimental Application
  - Representing doctrine for training in the MATREX FOM
- Results of Analysis
  - How will Primitives of Meaning, if applied, affect Military M&S
- Summary/Results
  - Next steps
Introduction

- This presentation is possible because of merging two efforts

- The first is a theoretical approach to decomposing the meaning of objects and processes in a model to something called “primitives of meaning” (more later)
  - Based on the primary author’s (Turnitsa) master’s thesis in M&S

- The second is a practical approach to decomposing a model into a lowest granularity, so that all actions and elements can be represented
  - Based on a PEO Soldier project to model small unit operations in federates employing the MATREX FOM
PROBLEM BEING ADDRESSED
Problem Being Addressed

- Interoperability between two models is enabled by representing information from one into the other.

- It is often convenient (or “natural”) to think of information in terms of a number of ideas grouped together.

- When two different models group those same basic ideas together in different ways, we get semantic misalignment.

- Important to realize that a model here is not ONLY a simulation system – but could also be a model for communication (like doctrinal C2 commands).
Semantic Misalignment

PRIMITIVES OF MEANING

COMPOSITIONS OF "LETTER" SYSTEM

COMPOSITIONS OF "NUMBER" SYSTEM

Entity 1

Entity 2
When two or more systems must share information, but there are differences to

- Structure (which entity has which primitive)
- Scope (some primitives in one model are not in the other)
- Resolution (modeling efforts are based on different levels of detail focus in the real world)

When one or more of these features are out of alignment – meaning a difference between the structure, scope or resolution of one model and another – the models are said to have Semantic Misalignment.
PRIMITIVES OF MEANING
Primitives of Meaning

- Primitives of Meaning are a name for the simplest, atomic (indivisible) concepts, that when placed together make up an entity of the model.

- Objects (entities, things) of a model can be based on Primitives of Meaning (in this case, different contributing concepts each are responsible for one or more attributes of the Primitive).

- Processes (activity, tasks) of a model can be based on Primitives of Meaning (in this case they represent the simplest atomic action – several of which are strung together in order to show a “task”)

---

Primitives of Meaning are a name for the simplest, atomic (indivisible) concepts, that when placed together make up an entity of the model.

Objects (entities, things) of a model can be based on Primitives of Meaning (in this case, different contributing concepts each are responsible for one or more attributes of the Primitive).

Processes (activity, tasks) of a model can be based on Primitives of Meaning (in this case they represent the simplest atomic action – several of which are strung together in order to show a “task”).
Why work with these Primitives

There are at least two possible reasons why identifying the primitives is a worthwhile pursuit:

1. Primitives are identified and relied on for building a transport/translation mechanism between systems.
2. Primitives are identified within a requirement for representation in a system, which is independent from the entities that originally included them in aggregation.

The first reason is core (for example) to the approach of interoperability that uses a Common Reference Model, to decompose and recompose information using high granularity concepts (such as C-BML).

The second reason is reasonable when a new representation requirement comes around to show (for example) the radios on tanks, when they were previously assumed to be present whenever a tank entity was.
Primitives of Meaning and MBDE

Model Based Data Engineering (MBDE) is a technique (refined and published by VMASC) that tries to solve the first problem.
- This is the approach that C-BML is taking
- While perhaps not at the highest level of granularity, nevertheless the atomic elements of the JC3IEDM (that are composed for interoperability exchanges of C2 information through a C-BML modeled CRM) are high enough granularity to support the unambiguous transmittal of data

The MBDE approach to data modeling (based on ISO 11179 standard for information metadata) allows for identifying entity attributes, and the values of those attributes
- These attributes would be the Primitives of Meaning of the entities
MBDE/ISO 11179
Primitives of Meaning and LCIM

The Levels of Conceptual Interoperability Model (LCIM) is a method for evaluating the conceptual richness of information exchange between two interoperating systems.

By exposing the *structure, scope, and resolution* of the information exchanged (what VMASC often refers to as the “meaning” of the data), vs. just a *name* for the data, levels at and above the Semantic Level (level 3) of the LCIM can be supported by using Primitives of Meaning to express the “meaning” of the information exchanged.
LCIM

Level 6
Conceptual Interoperability

Level 5
Dynamic Interoperability

Level 4
Pragmatic Interoperability

Level 3
Semantic Interoperability

Level 2
Syntactic Interoperability

Level 1
Technical Interoperability

Level 0
No Interoperability

Increasing Capability for Interoperation

Modeling / Abstraction
Simulation / Implementation
Network / Connectivity

Composability
Interoperability
Integrability
EXPERIMENTAL APPLICATION
 Doctrine to FOM

- Within the PEO Soldier project, there is a goal to represent Doctrinal commands within the MATREX FOM (and finally, within the federates using the FOM)

- There are far too many Doctrinal commands (or tasks) that can be given to small infantry teams for it to be practical to model each individually

- This application is a combination of both case 1 and case 2
  - Case 1 is the interoperability between systems of information, based on a method that decomposes it into Primitives of Meaning
  - Case 2 is the representation of some of the Primitives of Meaning separate from the original information aggregates that contained them
The models involved are (1) US doctrinal commands for Small Infantry Units, and (2) the MATREX Federation Object Model (FOM).

It is desired to show (for training events) the individual elements of the doctrinal commands, so that a training audience can interact with any piece of that command, and also so that each can be represented in a Simulation Federate that interacts via the FOM.

- ONESAF
- Combat XXII
- IWARS
- Etc

First test case was decided to look at the “Support by Fire” command.
Establish a Support by Fire (SBF):

Again, we turn to the PLT / SQD level doctrine, this time on page 4-42 of FM 3-21.9. The PL’s order might sound something like this. “WPNs, establish an SBF on the berm to the south of Bldg 100 and orient your fires to the North. On the codeword “ANVIL 1,” engage the first and second floor with 10 secs of cyclical M240B fire, subsequently talking your guns at a rapid rate. Additionally, maintain a watch and shoot to the East of Bldg 100 to prevent the enemy’s exfil. On “HAMMER,” shift your fires to the east of PL RED onto Bldg 101, and standby for “ANVIL 2.”

From the example by MAJ Dabkowski, Feb 17, 2009.


4-38. SUPPORT BY FIRE

The platoon maneuvers to a position on the battlefield from which it can observe the enemy and engage him with direct and indirect fires. The purpose of support by fire is to prevent the enemy from engaging friendly forces.

a. To accomplish this task, the platoon must maintain orientation both on the enemy force and on the friendly maneuver force it is supporting. The platoon leader should plan and prepare by:

• Conducting line-of-sight analysis to identify the most advantageous support-by-fire positions.
• Conducting planning and integration for direct and indirect fires.
• Determining triggers for lifting, shifting, or ceasing direct and indirect fires.
• Planning and rehearsing actions on contact.
• Planning for large Class V expenditures, especially for the weapons squad and support elements since they must calculate rounds per minute. (The platoon leader and weapons squad leader must consider a number of factors in assessing Class V requirements, to include the desired effects of platoon fires; the time required for suppressing the enemy; and the composition, disposition, and strength of the enemy force.)

b. A comprehensive understanding of the battlefield and enemy and friendly disposition is a crucial factor in all support-by-fire operations. The platoon leader uses all available intelligence and information resources to stay abreast of events on the battlefield. Additional considerations may apply. The platoon may have to execute an attack to secure the terrain from which it will conduct the support by fire. The initial support-by-fire position may not afford adequate security or may not allow the platoon to achieve its intended purpose. This could force the platoon to reposition to maintain the desired weapons effects on the enemy. The platoon leader must ensure the platoon adheres to these guidelines:

• Maintain communications with the moving force.
• Be prepared to support the moving force with both direct and indirect fires.
• Be ready to lift, shift, or cease fires when masked by the moving force.
• Scan the area of operations and prepare to acquire and destroy any enemy element that threatens the moving force.
• Maintain 360-degree security.
• Use ICVs and Javelins to destroy any exposed enemy vehicles.
• Employ squads to lay a base of sustained fire to keep the enemy fixed or suppressed in his fighting positions.
• Prevent the enemy from employing accurate direct fires against the protected force.
• Move to given location of SBF (Support by Fire), and take up position, using given Orientation

• On “engage” code word, fire at stated target

• On “shift” code word, cease firing, re-orient, and begin firing at new stated target

• On “lift” code word, cease firing
Identifying Objects & Processes

**OBJECTS**

UNIT – the military unit being ordered to assume the SBF. UNIT is an explicit object.

LOCATION – the spot on the map/chart where the SBF is to take place. Location is an explicit object.

TARGET – the location where the fire is to be directed at. Target is an explicit object.

MAP/CHART – an implicit object, represented by a reference system that provides a domain and range for the locating objects and describing actions.

CLASS V expendables – if these are modeled by the simulation, then there will be an expenditure rate that depletes these during the existence of a FIRE process. If the expendables are depleted, then the FIRE process implicitly stops.

**PROCESSES**

MOVE – the UNIT changes its location on the map/chart, to match the LOCATION given in the order. MOVE is a process that will end as soon as the UNIT’s LOCATION is the same as given with the MOVE process.

ORIENT – the UNIT changes it’s “facing” to match the one given with the ORIENT process command. ORIENT is a process that will end as soon as the UNIT’s orientation matches that given with the ORIENT process.

FIRE – the UNIT begins to discharge its weapons at the stated TARGET. The FIRE process is continuous, so a start and stop for the period of fire is indicated by the corresponding code words.

**UNIT will MOVE to LOCATION**

**UNIT will ORIENT in a stated direction, once at LOCATION**

**UNIT will await “fire code word”**

**UNIT will await additional “code words”**

**On “fire code word”, UNIT will begin to FIRE at stated TARGET**

**On “lift code word”, UNIT will cease the FIRE activity**

**On “shift code word”, UNIT will cease the FIRE activity at the current TARGET, re-ORIENT, and begin to FIRE at a new stated TARGET**
## Objects

<table>
<thead>
<tr>
<th>Doctrine Object</th>
<th>FOM Representation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit</td>
<td>AggregateUnit</td>
<td>Gives parameters for all of the information concerning a unit’s current status (ID, type, side, country, damage, ammo, fuel, appearance, location, orientation, etc)</td>
</tr>
<tr>
<td>Location</td>
<td>LatLongAltPositionCDT</td>
<td>This is a complex data type, that is part of the StateVectorCDT, which is the UnitLocation parameter’s data type. This CDT gives the x,y,z location of the object it is with. As mentioned, this is part of the StateVectorCDT which also gives velocity, position, orientation, orientation rate, and orientation acceleration, as well as a timestamp. Changes to the StateVectorCDT represent the results of a MOVE process.</td>
</tr>
<tr>
<td>Target</td>
<td>LatLongAltPositionCDT</td>
<td>Given as the parameter “EngagementArea” as part of the “AttackByFire” interaction. This has a cardinality of 2+, as it is intended to be a bounded geographic area, and the different parameters represent the vertices of the area.</td>
</tr>
<tr>
<td>Map/Chart</td>
<td>Implied as the boundaries to the values permitted in the LatLongAltPositionCDT</td>
<td>The playbox of the federation will have some limit to what values can be used for locations, or more precisely, as parameters to the LatLongAltPositionCDT complex data type.</td>
</tr>
<tr>
<td>Class V Expendables (ammo)</td>
<td>AmmoStateCDT</td>
<td>Part of the AggregateUnit object – lists all the ammo types and quantities carried by a unit.</td>
</tr>
<tr>
<td>Doctrinal Process</td>
<td>FOM Representation</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------</td>
<td>-------</td>
</tr>
<tr>
<td>MOVE</td>
<td>MOVE</td>
<td>The MOVE interaction has, as parameters, the Destination of the move (which is a LatLongAltPositionCDT), the Route followed (which uses the SimpleWayPointCDT), and a rate of speed, which is given in meters per second.</td>
</tr>
<tr>
<td>ORIENT</td>
<td>LatLongAltPositionCDT</td>
<td>This is given as part of MOVE, and is a parameter of that interaction. In order to change the current orientation, issue a MOVE interaction, with the current location as the destination, but with a new orientation.</td>
</tr>
<tr>
<td>FIRE</td>
<td>UnitAttack</td>
<td>This interaction gives the unit information on where to move to, and where to attack to. If the unit has already moved to its position before the “fire” codeword is given, then use the parameters for this interaction that show the current location/orientation/etc. If the halt or change codewords are given, weapons free status is turned off (still researching how this is done).</td>
</tr>
</tbody>
</table>
Representation in the FOM

- The representation of the Objects in the FOM seems to be pretty well supported.

- The representation of the Processes in the FOM seems to be lacking somewhat:
  - In order to fully support doctrine, some of the identified processes of a command that are represented by the current interactions of the Matrex FOM need to be identified and isolated.
  - Most of these have to do with separating the C2 functionality from the command.
  - Example used – Support by Fire, doctrinally the command is given, then elements of the command are performed following command “trigger” words:
    - While this can be done using mechanisms like a military event task list, or through operator interaction, allowing the constructive SIM to do it, can help train command units.
Extending the MATREX FOM

- Analysis was performed of what new interactions would be needed in order to separate the Primitives of C2 triggers, etc from the Interactions that represent the doctrinal command

- Following this, a new Base interaction was suggested, that other interactions in the FOM can be extended from

- The steps, including the newly suggested interactions, are presented here
New Base Interaction

Proposed that a new Base interaction be added to the FOM

C2Command - (similar to Command) Extends Communication

- ContributingIDs (0+ x FederateIDCDT) from the SituationReports, as with the interaction "Command"
- StartSignalID (1 x long) Correlates to a CommandID to start the C2Command
- StopSignalID (0-1 x long) Correlates to a CommandID to stop the C2Command. Only needed for interactions that extend this command who require a "stop" command - like C2Fire

Each new interaction that extends C2Command, will therefore be associated with a StartSignalID, and if necessary, a StopSignalID
New Entity Interactions

- **C2Move** - (similar to Move, could also be based on MoveAlongRoute)
  Extends C2Command
  - Destination (1 x LatLongAltPositionCDT)
  - Route (1+ x SimpleWayPointCDT)
  - Rate of Speed (1 x long)

- **C2Orient** - (new interaction) Extends C2Command
  - Orientation (1 x double) this is the new orientation of the entity, given in degrees from due north, increasing clockwise.

- **C2Fire** - (similar to Assault) Extends C2Command
  - EngagementTRP (2+ x LatLongAltPositionCDT) Gives the engagement area.

- **C2CommandSignal** - (new interaction) Extends Communication
  - CommandID (1 x long) Supplies the CommandID
New Unit Interactions

C2UnitMove - (similar to UnitMarch) extends C2Command
- Route (1+ x SimpleWayPointCDT)
- OccupyPosition (1+ x LatLongAltPositionCDT) Gives the location (either point or area) of unit once it completes the route move
- Rate of Speed (1 x long)
- Formation (1 x MoveFormationEDT) Gives the formation of the unit during the move

C2UnitOrient (new interaction) Extends C2Command
- Orientation (1 x double) this is the new orientation of the entity, given in degrees from due north, increasing clockwise
- OccupyPosition (1+ x LatLongAltPositionCDT) Gives location of unit once it re-orientes

C2UnitFire (similar to UnitAttack)
- EngagementTRP (2+ x LatLongAltPositionCDT) Gives the engagement area.

C2UnitCommandSignal
- CommandID (1 x long) Supplies the CommandID
RESULTS OF ANALYSIS
Design Philosophy

- Models can be representative of a referent in two ways
  - It can behave in the most efficient manner, in order to exhibit an “appearance” that is as close to that of the referent as possible
  - It can mimic the behaviors that the referent goes through

- The approach here takes the second of these, in order to be as close to doctrinal behavior as possible (i.e. – requiring the passing of a command word to initiate the fire, hold, re-orient)

- There are two ways to model any process – the ‘activity’ of the process, or the ‘results’ of the process. This method chose the former, as it was necessary to isolate the primitives representing the C2 exchanges
In an IABG report (Germany, 1999), a number of different combat primitives (under a different name) were identified, that could be combined to show doctrinal activities:

- Attrition
- Movement
- Observation
- C2 (Decision)
- Communication
- Maintenance (repair)
- Supply/Resupply
In a recent MORS Presentation (P. Works, 2007) the decomposition of Joint Mission Threads into different Role-Based Behaviors was identified to include the following:

- Update Situational Awareness (Observation)
- Informational Message (Communications)
- Call for fire Message (Communications)
- Other Message (Communications)
- Command Self (C2)
- Command Subordinate (C2)
- Maneuver (Movement)
- Fire/Engage (Attrition)

The similarity of these two lists (The equivalents of the first list are included in parentheses on the second list) seems to validate our idea of combat modeling primitives of meaning (at least for processes
SUMMARY/RESULTS
In order to evaluate the idea of Primitives of Meaning, we worked with two Models (Doctrinal Command, and the MATREX FOM) each of which were representations of a referent (i.e. – Small Unit Infantry Actions), but which represented the “meaning” of objects and processes differently.

In order to bring to the surface some of the primitives of the processes (i.e. – the C2 exchanges) for the training audience, decomposition of the FOM into higher granularity primitives was done.

This resulted in some recommended additions being suggested for the MATREX FOM in order to accommodate these new primitives.

The idea of the combat modeling primitives was compared to a literature review to see if the idea holds with what other researchers have done – it does.
Results

- The new additions/extensions to the MATREX FOM have been suggested.
- These may result in some programming changes to the participating federates so that the primitives can be shown.
- Further exploration of the primitives idea will require some way to “script” the primitives, so that more complex ideas (doctrinal commands representing tasks) will be needed.
- If the identified requisite combat modeling primitives are enabled, then a wide variety of new (but doctrinally sound) tasks should be possibly merely by transmitting a script from a C2 federate (representing a task) to the enabling federate (who will interpret the primitives) without requiring new programming.
Next Steps

Several things are going on that will impact further iterations of this work:

1. The development of a Modeled Process Specification method to describe the defining elements of a process (starting req’ts, ending req’ts, timing details, etc) – this should assist with the definition of these elements for process primitives.

2. The evaluation of the behavior composers for several small unit federates, to see if they support a similar list of process primitives as has been identified here.

3. The further analysis of more doctrinal commands, to evaluate if the list of process primitives is sufficient to represent newly required tasks.
QUESTIONS?

THANK YOU!

Charles Turnitsa  
(cturnits@odu.edu)

Andreas Tolk  
(atolk@odu.edu)

Virginia Modeling Analysis and Simulation Center  
Old Dominion University  
Norfolk, VA

Robert Kewley  
(Robert.Kewley@usma.edu)

Operations Research Center  
US Military Academy  
West Point, NY