2003

Getting One of the Basics Right for Distributed Simulations: A Mobility Service/Server for the Present and Future

Baylot, E. Alex; Goerger, Niki C.; Gates, Burhman

Monterey, California: Naval Postgraduate School.

Getting One of the Basics Right for Distributed Simulations: A Mobility Service/Server for the Present and Future

Mr. E. Alex Baylot
Dr. Niki C. Goerger
Mr. Burhman Gates
US Army Engineer Research and Development Center
3909 Halls Ferry Road, CEERD-GM
Vicksburg, Mississippi
601-634-3474, 831-656-3751, 601-634-3200
Alex.Baylot@us.army.mil, Niki.D.Goerger@erdc.usace.army.mil, Burhman.Gates@erdc.usace.army.mil

MAJ Simon R. Goerger
MOVES Institute
Naval Postgraduate School
Monterey, California
831-656-3733
srgoerge@nps.navy.mil

Key words:
Vehicle Mobility, Standards, Modeling and Simulation, NRMM, STNDMob API, XML, Java, HLA

ABSTRACT: *As computer hardware and models improve and the use of computer models and simulations (M&S) escalates, users subsequently demand more realism and, thus, fidelity requirements tend to increase. Many stand-alone, high fidelity, engineering level models have been developed, accepted, and repeatedly used in analysis and studies by the Department of Defense. For example, in the area of ground movement, the NATO Reference Mobility Model (NRMM) is the Army Model and Simulation Office (AMSO) standard for single vehicle ground movement representation. While representation of ground vehicle mobility in both entity- and aggregate-level M&S has typically been simplified, many of the speed limiters incorporated in NRMM are ignored. Developing M&S such as COMBAT^{XXI} and OneSAF Objective System (OOS) have functional and operational requirements to portray mobility at a higher fidelity. This paper describes the development of an NRMM-based Standard Mobility Application Programming Interface as a means of readily achieving higher-fidelity movement representation by incorporating terrain-limited speeds into M&S. The Standard Mobility API was written in Java and utilized Extensible Markup Language (XML) for database structures. The API design concept encompassed a range of M&S; this paper will focus on the tactical/entity-level implementation. The US Army Engineer Research and Development Center (ERDC) and the US Army TRADOC Analysis Center – White Sands Missile Range (TRAC) collaborated on the API development and integration into COMBAT^{XXI} as a testbed. By providing a standard interface for applications, this work helps reduce the proliferation of differing mobility models, provides access to standard speed prediction algorithms, and promotes reuse/consistency.*

1. Introduction

Mobility implementation in M&S is largely tailored for specific simulations, leading to inconsistencies in model calculations. The developing M&S systems will eventually be used for related studies and analysis and have identified NRMM-based mobility as a requirement (OneSAF Operational Requirements Document, COMBAT^{XXI} Functional Requirements Document). It is necessary to provide a means to promote consistency between M&S and to facilitate ready-integration of higher-fidelity standard algorithms into M&S, particularly in light of Army Transformation and Objective Forces analysis.

This paper describes the development of an NRMM-based Standard Mobility (STNDMob) Application Programming Interface (API) as a means of readily achieving higher-fidelity movement representation by incorporating terrain-limited speeds into M&S. The STNDMob API was written in Java and utilized Extensible Markup Language (XML) for database structures. The API design concept encompassed a range of M&S; tactical/entity-level implementation will be the focus of this paper. The ERDC and the TRAC collaborated on the API development and integration into COMBAT^{XXI} as a testbed to prove the usability of the API. By providing a standard interface for applications, this work helps reduce the proliferation of differing mobility models, provides access to standard speed prediction algorithms, and promotes reuse.

The STNDMob API development will facilitate integration of NRMM-based mobility into future simulations. Moreover, this approach will allow for model upgrades with minimal reengineering of simulation links. Furthermore, this approach should enable NRMM enhancements and upgrades to be readily implemented as the mobility model matures to accommodate advanced systems

2. Objectives & Scope

The goal of this research is to create an API to facilitate the implementation of NRMM-based mobility into modeling and simulation applications by providing a standard interface, to promote consistent representations of ground vehicle mobility across the hierarchy of M&S. As noted above, the API is referred to as the Standard Mobility API (STNDMob API). More specifically, the objectives of the research are to (a) identify the types and variations of API levels needed to meet the needs of the M&S community, (b) design the architecture for these APIs, (c) develop the associated algorithms, and (d) transition the products to the user community. The scope of this research is on ground vehicle mobility, or trafficability, in terms of achievable vehicle speed, and on M&S applications. The focus of this paper is on the accomplishments to date dealing with the design and development of the Level 1 Entity/Tactical hierarchy category

3. Background

3.1 M&S Taxonomy

M&S can be organized in a variety of ways, but Figure 3.1.1 provides the big picture in terms of next-generation M&S under development and a typical means of parsing the applications; namely by domain, type, and hierarchy. The STNDMob API suite is designed to support the various domains, types, and levels. The hierarchy of M&S can be categorized as engineering system or subsystem level through force-on-force engagement models that include one-on-one through many-on-many engagements aimed at the different echelons (brigade and below, theater, e.g.) [1]. Domain refers to the Army M&S domains, (a.) Advanced Concepts Requirements (ACR), (b.) Research Development and Acquisition (RDA), and (c.) Training Exercises and Military Operations (TEMO) [6]. The ACR domain is generally concerned with requirements and concept exploration. The activities within the RDA domain include basic and applied research and test and evaluation utilizing engineering and physics-based models. The TEMO domain is generally focused on individual and collective, including joint, training, operations planning, and mission rehearsal.

Simulations are also categorized as live, virtual, or constructive although there is no clear division between these categories [6]. Live simulations involve real persons operating in real systems. Virtual simulations involve real persons operating simulated equipment or systems, and constructive simulations involve simulated persons operating simulated systems.

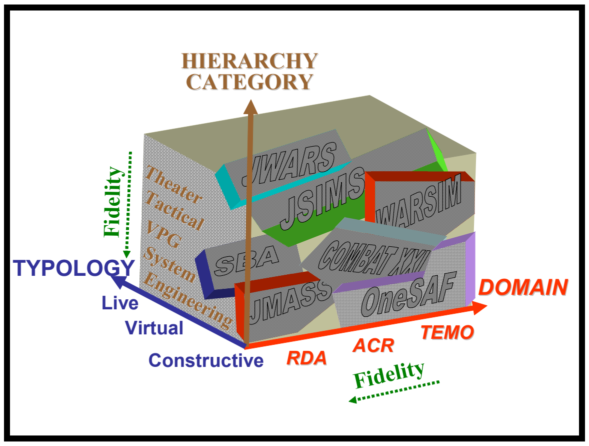

Figure 3.1.1. M&S Taxonomy with given future systems

3.2 Model Versions

To meet the mobility modeling needs of the M&S community, ERDC and TRAC proposed to develop three independent but related APIs to provide NRMM based terrain-limited speed predictions to Aggregate, Tactical/Entity, and Engineering Level models. Aggregate level models can be considered as those that model the movement of aggregate units as opposed to individual ground vehicles (Figure 3.2.1).

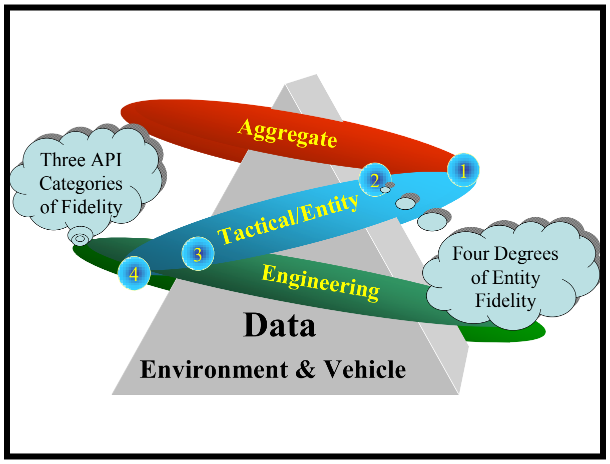

Figure 3.2.1. The suite of STNDMob APIs will span the hierarchy with expanded degrees of fidelity in the Tactical/Entity hierarchy level based on terrain and vehicle data resolution.

OOS development plans call for a phased implementation of mobility representation. The phases call for increasing the level of representation and fidelity of the predictions. COMBAT^{XXI} has similar development plans.

4. Description of Tactical/Entity Level API

Within the Entity/Tactical level API design, there are implementations that account for various degrees of fidelity associated with the resolution of the input and output parameters. For example, a study could be interested in representing vehicle performance and terrain factors at higher resolution or at lower resolution where levels are based on look-up-table computations. Compatibility and implementation of the API design with OOS, COMBAT^{XXI}, and WARSIM will be illustrated.

As stated previously, the Tactical/Entity Level API contains four levels of fidelity and will be hereafter referred to as Standard Mobility (STNDMob). This is needed to insure consistency between simulations and decision support systems that vary in entity definition. Some theater level simulations use an entity as large as a brigade with total vehicle count totaling in the hundreds. Furthermore, these entities are comprised of vehicles with differing performance potential and characteristics. Therefore, the least capable vehicle defines the performance of the group. Other simulations use a smaller group comprising of identical vehicles or they would use an individual vehicle. Thus, these levels of fidelity are meant to allow a user to find a common denominator and promote consistency between various simulations and decision support systems. Only the lowest levels of fidelity (1 & 2) will be discussed in the paper.

This level of representation is regarded as low resolution or Level 1. This method included accommodations necessary to insure compatibility with WARSIM 2000 and C4ISR systems [3]. Where possible, equations were reduced to look-up-tables to minimize runtime computational loads.

As shown in Figure 4..1, Level 1 mobility has two fidelity degree settings. Fidelity degree 1 refers to only using representative vehicles to model the performance of specific vehicles. Thus, specific vehicles are not explicitly modeled. Fidelity degree 2 is obtained by modifying the speed of Fidelity degree 1 by the speed reduction factor relationship of the representative vehicle to the specific vehicle [4][5].

Hierarchy

- Aggregate
- Tactical/Entity
    - Level 1
        - Fidelity Degree 1
        - Fidelity Degree 2
    - Level 2
        - Fidelity Degree 3
        - Fidelity Degree 4
- Engineering

Figure 4.1 Structure of model representation.

4.1. Inputs

The environmental input data parameters of the API were mapped to the OOS Environmental Data Model (EDM) version 1.0 [7]. The vehicle input data parameters were chosen from readily available sources [8].

4.1.1. Environmental

The parameters labeled in italics are native to the API and the parameters labeled within the parenthesis are from the OOS EDM.

- *Climate_Zone* (determines values for Soil_Wetness, Soil_Cone_Index_QB_Measurement, Terrain_roughness_root_mean_square, Mean_Stem_Diameter, Mean_Stem_Spacing_QB_Stem_Diameter for STGJ Code), index
- *Ground* (cross-country or road comes fromTerrain_Transportation_Route_Surface_Type, Road_Minimum_Traveled_Way_Width, Path_Count), index
- *Condition* (dry, wet, snow from Soil_Wetness and Frozen_Water_Type), index
- *STGJ* Codes (combination of Soil_Type, Vegetation_Type, and other factors, mapped to MLU Codes), index
- *Vis* (Maximum Visibility Range {four values, road only} derived from weather, sensor range, obscurants, illumination, etc.), index
- *VisObs* (Maximum Visibility Range {four values, same as Vis} and Obstacle Spacing {four values} combinations derived from Terrain_Obstacle_Type, Width, Overall_Vertical_Dimension, Row_Distance, Row_Spacing_Interval), index
- *Vehicle_Pitch* (use Surface_Slope or pitch along direction of vehicle travel {nine values}), percent

Note: See references for further clarification of units of measure or meaning of index values [5][7].

4.1.2 Vehicle

The twelve representative vehicles bins are given in Table 4.1.2.1.

Table 4.1.2.1. Bins and Representative Vehicles with Mappings

No.	Vehicle	OOS/WARSIM Name	CCTT-SAF
1	M1A1	High Mobility Tracked	High Mobility Tracked
2	M270 MLRS	Medium Mobility Tracked	Good Mobility Tracked
3	M60 AVLB	Low Mobility Tracked	Low Mobility Tracked
4	M1084 MTV	High Mobility Wheeled	High Mobility Wheeled
5	M985 HEMTT	Medium Mobility Wheeled	Low Mobility Wheeled
6	M917 Dump Truck	Low Mobility Wheeled	Not Applicable
7	M1084/M1094	High Mobility Wheeled w/Towed Trailer	Not Applicable
8	M985/M989	Medium Mobility Wheeled w/Towed Trailer	Not Applicable
9	M911/M747 HET	Low Mobility Wheeled w/Towed Trailer	Not Applicable
10*	M113A2	Tracked ACV	Moderate Mobility Tracked
11*	LAV25	Wheeled ACV	Not Applicable
12*	Kawasaki ATV (high shock)	Light ATV	Not Applicable
* Not yet approved by WARSIM, but implemented into JWARS			

Note: See references for justification of selected representative vehicles [8].

The vehicle data needed to determine what bin membership a specific vehicle belongs and it's relationship to a representative vehicle, is given below.

- *Type* (Traction Element: Track or Wheeled), number
- *Towing_Trailer* (Attached), number
- *Primary_Use* (Truck, Amphibious (or similar design) Combat Vehicle, Heavy Equipment Transporter, other), number
- *Gross_Weight* (Combat Vehicle Weight), kg
- *Engine_Power*, hp
- *Maximum_Gradient*, percent
- *Maximum_On_Road*, kph
- *Amphib_Design*, number

Additional vehicle is provided for characterizing the vehicle and establishing speed caps or boundaries. Representative_Bin, Speed_Factor, and Power_to_Weight_Ratio is computed using the above vehicle data.

- *Vehicle_Name*, text
- *Vehicle_ID*, number
- *Representative_Bin*, number
- *Fording* (speed), kph
- *Swimming* (speed), kph
- *Speed_Factor*, number
- *Power_to_Weight_Ratio*, number

4.2 Process

This method will insure consistent mobility representation with systems such as WARSIM 2000, C4ISR systems, theater level models, and other systems whereby the smallest entity is a unit or aggregation of individual entities. M&S that use smaller entities may use this level of fidelity, but the user must fully understand the limited relationship and representative vehicle-terrain interaction. This method will utilize representative vehicles to govern the maximum terrain limited speed of the least capable vehicle in the unit composition. Units will move at their commanded speed or the aforementioned maximum terrain limited speed, whichever is smaller. The terrain will utilize the WARSIM MLU (mobility look-up tables) or STGJ (Soil Trafficability Groups JSIMS) data model currently used by WARSIM 2000 or the revised environmental data model utilized by WARSIM 2000 at the release of OOS.

A routing service *(not part of STNDMob)* will query the maximum terrain limited speed via look-up-table indexed on the above described environmental inputs and vehicle inputs. The routing service will compute the position and heading information at each way-point as prescribed by WARSM. Actual speed will be the least of the commanded speed and maximum terrain limited speed. Commanded speed is computed as prescribed by WARSIM [9].

Exact terrain attributes are required except for slope along heading of vehicle (pitch). To obtain maximum terrain limited speed for values of slope that are not pre-processed or indexed, a linear interpolation of speed between given pre-processed slope values is performed. Guidance for translating the meaning of visibility, obstacle, and wetness index classes is provided in the references [5].

Fidelity degree 2 is a close match with current WARSIM 2000 implementation. The difference being, WARSIM 2000 uses data files containing the ratio of actual speed for each MLU to the maximum road speed, rather than the actual speed for each MLU. The inputs and outputs are the same as Fidelity degree 1, except a selected vehicle must be associated with a bin using the given attributes of vehicle data, and the maximum terrain limited speed is adjusted by a multiplicative factor.

Exact terrain attributes are required except for slope/pitch along heading of vehicle. For values of slope that are not pre-processed, a linear interpolation of vehicle speed between given slope/vehicle pitch values is performed to compute maximum terrain limited speed. Guidance for translating the meaning of visibility, obstacle, and wetness classes is provided in the references [5].

Using the given set of vehicle data, compute the bin membership or SAFOR class from the list of categories/bins given in Table 4.1.2.1[8]. Then, in the same manner as described for Fidelity degree 1, with the exception that once the maximum terrain limited speed for the representative vehicle is found, the maximum terrain limited speed for the given vehicle will be adjusted by a multiplicative factor computed as the ratio of the given vehicle maximum road speed to the representative vehicle maximum road speed of its bin membership. (Note: No known research has been conducted to quantify the accuracy of this multiplication factor. Accuracy is assumed to be sufficient for on-road and cross-country conditions when surfaces are hard and open).

It is conceivable that bin membership would be computed at simulation startup and not be recomputed during the course of the simulation. However, should the values of the factors that define bin membership change significantly, then a new bin computation might be warranted.

Once the specific (S) vehicle bin membership has been determined, apply the following equation to adjust the maximum terrain limited speed of the representative (R) vehicle found in the process as described for Fidelity degree 1. Default values for the bin membership value and factor on speed are given in the vehicle data files.

$$Speed_S = Speed_R \times \frac{MaximumRoadSpeed_S}{MaximumRoadSpeed_R}$$

Equation 4.2.1. Speed Formula

The procedure described above is to be used as a tool to consistently generically categorize vehicles and place them into bins. It is understood that for particular scenarios, some vehicles might be better represented in another bin. However, it is highly unlikely that it would be beyond the adjacent bin (i.e., low mobility versus high mobility).

Cross-validation with NRMM speed predictions was performed and there were no instances where the predicted speeds differed by more than one adjacent bin between NRMM and the binning procedure [8].

4.3 Example Input

An example of the data contained in the various input files is given in Table 4.3.1. The files are divided by climate zone and subdivided by condition. The reasoning is that it is unlikely that OOS will simulate a scenario involving more than one climate zone or soil condition. Thus, saving computer memory resources. However, this does not preclude additional climatic zone/soil conditions from being used. Locations of climate zones as 1-degree grids are given in the STNDMob. A utility function is provided for translation in STNDMob.

The range and meaning of index values are given in Table 4.3.2 and Table 4.3.3 provides the characterization data for the High Mobility Tracked representative vehicle and two other members of this bin. Vehicle ID values are arbitrary with values 1-99 reserved for representative bins.

Table 4.3.1. File Information

Title	NRMMII Predictions mapped to STGJ Codes									
Climate_Zone	2									
Bin	High Mobility Tracked									
Ground	cross country									
Condition	dry	***Speed (units) for the given slope/pitch in %***								
visobs	1	-40	-30	-20	-10	0	10	20	30	40
mlu	1	0.0	0.0	13.2	36.7	26.9	0.0	0.0	0.0	0.0
mlu	2	0.0	0.0	13.2	36.7	26.9	0.0	0.0	0.0	0.0
mlu	3	40.0	40.0	40.0	40.0	12.3	6.0	3.9	1.9	0.0
•••	•••	•••	•••	•••	•••	•••	•••	•••	•••	•••
mlu	256	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Condition	wet									
visobs	1	-40	-30	-20	-10	0	10	20	30	40
mlu	1	0.0	0.0	13.2	36.7	26.9	0.0	0.0	0.0	0.0
mlu	2	0.0	0.0	13.2	36.7	26.9	0.0	0.0	0.0	0.0
mlu	3	38.1	40.0	40.0	40.0	11.6	5.0	0.0	0.0	0.0
•••	•••	•••	•••	•••	•••	•••	•••	•••	•••	•••
mlu	256	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Condition	snow									
visobs	1	-40	-30	-20	-10	0	10	20	30	40
mlu	1	0.0	30.7	40.0	40.0	40.0	9.3	4.9	0.0	0.0
mlu	2	0.0	30.7	40.0	40.0	40.0	9.3	4.9	0.0	0.0
mlu	3	40.0	40.0	40.0	40.0	23.7	7.6	4.6	3.0	0.0
•••	•••	•••	•••	•••	•••	•••	•••	•••	•••	•••
mlu	256	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Ground	road									
Condition	dry	***Speed for the given slope/pitch in %***								
vis	1	-15	-12	-8	-4	0	4	8	12	15
mlu	726	27.7	30.0	30.0	30.0	26.6	14.6	0.0	0.0	0.0
mlu	727	27.7	30.0	30.0	30.0	26.6	14.6	0.0	0.0	0.0
mlu	728	30.0	30.0	30.0	23.6	12.3	8.7	6.8	5.5	4.8
•••	•••	•••	•••	•••	•••	•••	•••	•••	•••	•••
mlu	749	30.0	30.0	30.0	30.0	23.5	12.3	8.8	6.8	5.8

Table 4.3.2. Definition of Index Values

Index	Range of Values
Climate Zone	dry climates (2), humid mesothermal (3), humid microthermal (4), undifferentiated highland (6)
Condition	dry, wet, snow
Bin	1 – 12
Ground	cross-country, road
visobs	1 – 16
vis	1 – 4
mlu	1 – 256 (cross-country) 726 – 749 (road)
slope/pitch	-40, -30, -20, -10, 0, 10, 20, 30, 40 Cross-country) –15, -12, -8, -4, 0, 4, 8, 12, 15 Road

Table 4.3.3. Vehicle Information

Vehicle Name	M1A1	AMX 30 LeClerc	T-80
Vehicle ID	1	100	103
Representative Bin	1	1	1
Speed Factor	1.0	0.90	0.97
Gross Weight (kg)	54545	36000	42500
On Road Speed Max (kph)	72	65	70
Swimming Speed Max (kph)	0	0	0
Fording Speed Max (kph)	8	8	8
Amphibious Design (Y/N)	N	N	N
Maximum Gradient (%)	60	60	63
Engine Power (hp)	1500	793	1213
Type (tracked, wheeled)	Y	Y	Y
Primary Use	Tank	Tank	Tank
Towing Trailer (Y/N)	N	N	N

4.4 Example Output

The output from Maximum_Terrain_Limited_Speed will be used to govern whether a commanded speed is achievable or not. A routing service outside this model will determine the heading and position of the ground vehicle.

For the most part, the input data indexes to a given NRMM II representative vehicle speed prediction. The exception is that linear interpolation is performed on the speed prediction given two adjacent slopes on the index. Table 4.4.1 provides an example for a high mobility tracked vehicle.

Table 4.4.1. Predictions for High Mobility Tracked (Fidelity degree 1)

Input					Output
Climate Zone	Soil Wetness	Visibility/ Obstacles	STGJ Code (MLU)	Slope (%)	Speed (mph)
2	Dry	1	270 (19)	0	26.9
2	Dry	1	270 (19)	10	0
2	Dry	1	270 (19)	-10	30.0
2	Dry	16	270 (19)	0	0
2	Wet	1	313 (37)	0	11.6
2	Wet	1	313 (37)	10	5.0
2	Wet	1	313 (37)	5	8.3*
* Interpolated from 5 and 10% slopes for NRMM II predictions.					

Fidelity degree 2 uses the representative vehicle to serve effectively as the basis of the speed prediction of a specific vehicle. The ratio of the specific vehicles maximum road speed and the representative vehicle maximum road speed serves as a factor on the speed as given for Fidelity degree 1. Table 4.4.2 provides an example for a T-80 tank as represented by a High Mobility Tracked vehicle (M1A1).

Table 4.4.2. Predictions for a T-80 Tank (Fidelity 2)

Input					Output
Climate Zone	Soil Wetness	Visibility/ Obstacles	STGJ Code (MLU)	Slope (%)	Speed (mph)
2	Dry	1	270 (19)	0	26.2
2	Dry	1	270 (19)	10	0
2	Dry	1	270 (19)	-10	29.2
2	Dry	16	270 (19)	0	0
2	Wet	1	313 (37)	0	11.3
2	Wet	1	313 (37)	10	4.9
2	Wet	1	313 (37)	5	8.1*
* Interpolated from 5 and 10% slopes for NRMM II predictions. Note: Max speed for M1A1 = 72 kph, Max speed for T-80 = 70 kph					

5. Data and Software Standards

Initially, the native method for the API to retrieve the input data was through an Access database. The Access database remained in use until the development team formatted the data into compressed Extensible Markup Language (XML) files. The file format follows that of the OOS XML coding standard and COMBAT^{XXI} [10].

Vehicle specific data is not essential for running the API. If a lower level of resolution is applicable for the simulation, the API provides generic vehicle data, based on the twelve representative vehicles, four climate zones, cross-country, on-road, soils, and wet/dry/snow climate conditions. The API utilizes this data to generate maximum terrain limited speed predictions.

5.1 Extraction From NRMM

The data extracted from the NRMM consists of a series of tractive force vs speed curves. The API stores this static data along with the associated data used to reference the speed values in a separate database. During initialization of the API, this data is extracted form the database and placed into static java data structures for reference during runtime. The generic structures for the static data objects are found in the **standardmobility.interaction** package.

The initial version of the API stored static data in text files. Version two of the API used an Access database to provide an initial means of storing static data limiting use of the API to machines capable of utilizing an object database conductivity connection (ODBC) driver. The current implementation of the API uses compressed XML to store static data.

5.2 Open Standards (XML)

XML provides platform independent means of storing data. It is a text-based format with a series of markup tags used to categorize and store data. It is not as much a language as a set of standards for developing a language for the storage of data.

Version 3.0 of the STNDMob API stores the speed and the platform description data in XML files. Table 5.2.1 and Figure 5.2.1 show the tabular and the graphic lay down of the vehicle type to vehicle bin mapping XML file schema. Figure 5.2.2 is a graphical example of the XML file schema used to store climate zone and vehicle speed data.

These XML files provide the API with a platform independent data storage format. They also provide a means for users to request tractive force versus speed curve data for future systems from ERDC. The XML files provide a generalized template for engineers to collect data on existing systems not currently in NRMM and on future concepts. Additional data may be obtained through the US Army Material Systems Analysis Activity repository.

Table 5.2.1. Tabular definition of STNDMob API vehicle type mapping XML file

vehicle attributes	Name	Type	Use	Default	Fixed	Annotation	
	name	xs:string					
	vehicleTypeID	xs:integer	req'd				
	bin	xs:integer	req'd				
	binFactor	xs:double		1			
	loadedWeight	xs:integer		0		documentation	Measured in kg.
	maxRoadSpeed	xs:integer		0		documentation	Measured in kph.
	fordingSpeed	xs:double		0		documentation	Measured in kph.
	swimmingSpeed	xs:double		0		documentation	Measured in kph.
	amphibious	xs:integer		0			
	maxGradient	xs:integer		0		documentation	Measured in percent.
	power	xs:integer		0		documentation	Measured in hp.
	primaryUseCode	xs:integer		4		documentation	1 = Truck, 2 = ACV, 3 = Heavy Equipment Transport, 4 = Other
	trailer	xs:integer		0			

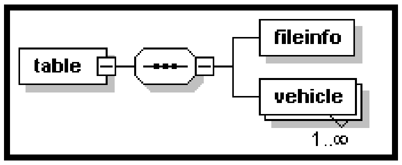

Figure 5.2.1. . Schema of STNDMob API vehicle type mapping XML file

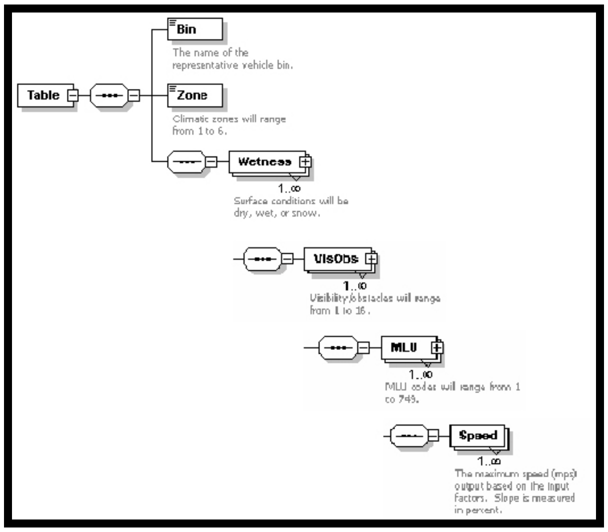
Table
Bin
The name of the representative vehicle bin.
Zone
Climatic zones will range from 1 to 6.
Wetness
1..∞
Surface conditions will be dry, wet, or snow.
VisObs
1..∞
Visibility/obstacles will range from 1 to 16.
MLU
1..∞
MLU codes will range from 1 to 749.
Speed
1..∞
The maximum speed (mps) output based on the input factors. Slope is measured in percent.

Figure 5.2.2. Schema of STNDMob API climate zone and vehicle speed XML file

5.3 Software Standards (Java)

The development team coded the Standard Mobility API utilizing an object orient paradigm to help segregate the interface from implementation specifics [11]. Any simulation wishing to use the STNDMob API, need only interact with the main class, **standardmobility.StandardMobility** [12]. From this Java class, a user has access to public methods that provide maximum terrain limited speed predictions for a platform in meters per second. A simulation gains access to a platform's terrain limited max speed using one of the overloaded methods (**calculateMaximumSpeed)**.

In Java, an overloaded method is a collection of methods with the same name but different input parameters. Using "**calculateMaximumSpeed"** allows users to access the functionality of the API using varied amounts and types of environmental information. This flexibility helps to ensure the API is non-simulation specific and valid for use in multiple scenarios and simulations.

Table 5.3.1. STNDMob API Package Structure

Package	Description
standardmobility	This is the top-level container package for all Standard Mobility API code generated directly by TRAC-WSMR developers. It holds the primary classes, interfaces, and supporting packages required for use of the API.
standardmobility.future	This package contains the classes and interfaces that maintain the data structures and algorithms required to run future versions of the STNDMob API.
standardmobility.mobility	This package contains the classes that maintain the data structures and algorithms required to run the STNDMob API.
standardmobility.util	This foundation package contains widely-used class definitions (Directions, Distance, etc.), utility algorithms, and enumerations of acceptable values ('types') for use by the STNDMob API.
standardmobility.util.geom	This package contains the classes that maintain the general utility data structures and algorithms required for the basic geometric functions (distance, location, etc) used to operate the STNDMob API.
standardmobility.util.types	This package contains the classes that define the general enumeration types (platform type - M1A2, T-72, BMP-3, APC, etc; NRMM vegetation type - bare, grass, forest, etc) required for the basic operation of the STNDMob API.

Internally, the API consists of six packages (Table 5.3.1). Figure 5.3.1 depicts the metaphorical relationship of these packages as a "Software Architecture". The API is composed of discrete packages of software, many of which are independent, reusable components that are not STNDMob API -proprietary. Like most software development efforts, the STNDMob API is composed of several abstraction tiers. These tiers are generic, but broadly-applicable. The services provide the foundation and core functionality that can be tailored for more specific purposes in representations that are meaningful from the context of the users needs.

This software architecture makes it easier to add features to the API, since many of the needed services will already exist and need not be re-coded and the users need only access the API through the integration package, **standardmobility**.

To date, a majority of the time developing the API was spent laying down the abstract services that we believe will be necessary no matter which STNDMob API or fidelity level is utilized. An Example would be to expand the Entity Level API to perform one of the Aggregate Level API functions by adding a method which takes an array of platform/bin types and a center of mass of the unit to determine the max speed of the unit based on the slowest mover in the vehicle array. This robust infrastructure is adopted from the one outlined in COMBAT^{XXI} but is generic in nature allowing for integration into most DoD models.

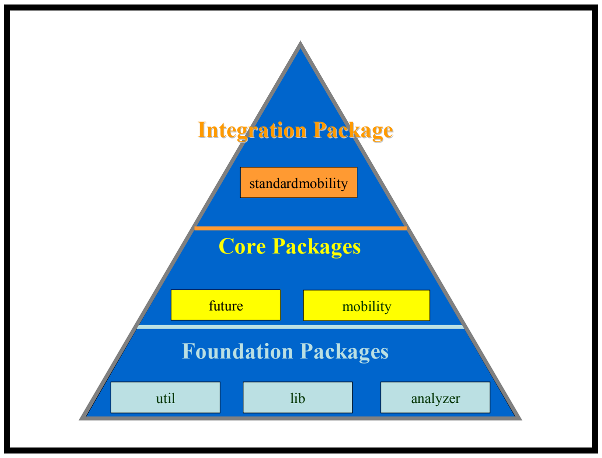

Figure 5.3.1. STNDMob API Software Architecture [4]

Again, the **standardmobility** package is the only package the programmer needs to interface with to gain access to NRMM functionality. This, as stated previously, is through the **standardmobility.StandardMobility** class. The **standardmobility.future** package is a placeholder for classes required for future expansion of the model. The **standardmobility.STNMobApp** class is the skeleton code for the future application implementation of elements of NRMM. The remaining four classes are the foundation of the API. They contain classes that access the functionality of NRMM and error detection capabilities of the API. The **standardmobility.StandardMobility** class calls the foundation classes and their associated methods. The foundation classes are contained in the final packages (**standardmobility.util.geom, standardmobility.util.types, standardmobility.util,** and **standardmobility.mobility**). The general relationship between the implementation classes and API packages is seen in Figure 5.3.2.

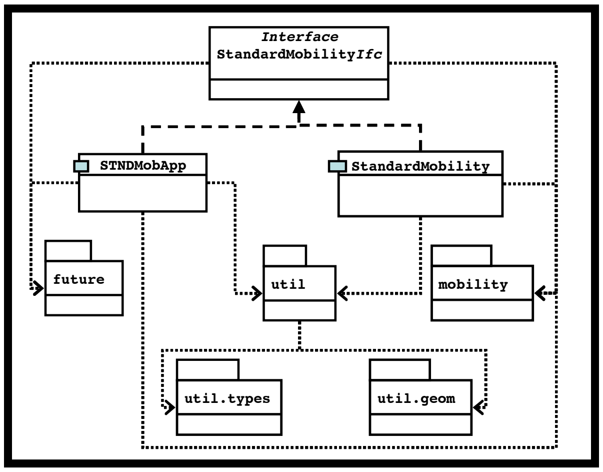

Figure 5.3.2. STNDMob API Package Diagram

To assist in maintenance and integration, the STNDMob API is coded in accordance with the Java Style Guide developed for use by COMBAT^{XXI} [11]. This includes the use of unit testing. The API uses these standards to ensure each class or interface is organized in a consistent manner. The standards also outline the requirements for in line documentation for each class and method. The in line documentation can be processed using Java tools to create JavaDocs. These documents assist project programmers, program mangers, and end users with understanding the code by providing a general layout of the program and definitions for the methods and variables used within the program.

6. Mobility Server

A software "wrapper" that takes advantage of IEEE specification 1516 (High Level Architecture / Run Time Interface (HLA/RTI)) was written to serve the mobility needs of the distributed simulation community. With STNDMob available as a server in distributed simulations, each participant that simulates ground vehicles has mobility representation readily available without the need to integrate with the API. This provides a consistent mobility representation across the distributed simulation.

Historically, mobility has been modeled within computer simulation using several approaches. The most common approach is a speed look up table based on a digital map. The digital speed map is referenced to a given vehicle. This method is employed for the Janus high-resolution combat model. However, a new digital speed map must be created for each new instance of a vehicle, scenario or terrain location. Another approach is the use of an API. The API is typically computer code introduced as a statically linked library. Statically linked libraries are compiled with the main program prior to run time. The program must be re-link to each participating program every time there is an update to the mobility model. This can be time consuming, where the participating programs in the distributed simulation are installed on heterogeneous platforms. i.e. Windows and Linux. Insuring the mobility libraries are the same across platforms is often difficult. In both of the previous cases there is also a problem with consistency across platforms. While the platforms may interoperate, the speed maps and libraries they use for mobility calculations can be different; this often results in an unfair fight. Two similar platforms operating on the same terrain and same scenario conditions may move at different rates because of inconsistencies in the movement algorithms. Also, if the terrain model increases in resolution, new weather conditions are introduced, or new vehicle systems are created, the mobility model should adapt to these changes.

To resolve many of the interoperability issues described above, an HLA/RTI interface (ambassador) was added to STNDMob so that it may participate in a federation of other applications. The objective of this add-on was to make terrain limited maximum speeds available to various simulators participating in a distributed simulation. Also, it was intended to establish a program that could be easily extended to serve other mobility parameters as well.

The HLA/RTI passes data between federates in an instantaneous form and a persistent form. Interactions are the instantaneous form. A federate is an executable which communicates/interacts to other federates within a larger federation. Interactions are one-shot and are not stored by the HLA/RTI. An example of an interaction would be one simulation requesting that another simulation create a certain type of vehicle. Since objects are of the persistent form, the vehicle is persistent. The federate that creates a vehicle announces the creation of the vehicle and which attributes it will update, or *publish* as the vehicle's state changes. If other federates have an interest in knowing when a vehicle typed object has been published or when a particular type of interaction has been published, then the federate *subscribes* to those communications. The HLA/RTI keeps up with the publications and subscriptions of the federate and passes data as needed.

Normally the federate that creates an object will control it. That federate will own the object's attributes -- that is, have permission and responsibility for updating the object's attributes. HLA/RTI has the ability to broker the ownership of object attributes between federates. This might happen if the federate requesting ownership has the ability to update the negotiated attribute better than the federate that created the object. In HLA/RTI, the notion of ownership is not associated with objects themselves. Instead ownership is connected with object attributes. Therefore, a simulated entity's attributes can be' split between federates. This is possible only if the federates have been programmed to take advantage of attribute brokering services offered by the HLA/RTI.

The HLA/RTI also regulates the elapse of simulation time through time management. Federates can either be time regulating or non-regulating and time constrained or unconstrained.

The availability of this mobility service will provide distributed simulations with maximum terrain limited speeds which are sensitive to soil type, soil strength, vehicle type, and slope. By using this service, distributed simulations will avoid integration of mobility models directly into each of the ground vehicle simulators. Also, as the mobility service is improved, the clients will benefit from the improvements without changing the client models themselves. This will reduce the maintenance and administrative costs of mobility representation in distributed simulations. And most importantly, client simulations can rely on consistent mobility representation across all the ground vehicle simulating federates.

7. Conclusion and Future Effort

To enhance the fidelity of M&S to support studies and analyses, developers are attempting to integrate engineering-level models into their products. It is imperative to facilitate standard implementations of these models to promote reuse and consistency,. This allows simulation developers to access the enhanced fidelity while maintaining the implementation and integrity of the engineering-level model.

The STNDMob API accomplished this by developing a standard, platform-independent implementation based on the AMSO standard for ground vehicle movement representation,NRMM, to facilitate the standardized integration of entity-level mobility constraints into DoD simulations. This was accomplished with the assistance of software enhancements such as Java, XML, and HLA. The implementation of the API in COMBAT^{XXI} demonstrated its viability as a DoD product.

The development of generic APIs or object models will facilitate integration of the model into future simulations. Moreover, this approach will allow for model upgrades with minimal reengineering of simulation links. By providing a standard interface for applications, this work helps reduce the proliferation of differing mobility models, provides access to standard speed prediction algorithms, promotes reuse/consistency, and saves developers and analysts time and money.

Future expansion will be accomplished through leveraging of the US Army Science & Technology Objectives, Battlespace Terrain Reasoning (BTRA) and High Fidelity Ground Platform & Terrain Mechanics Modeling (HGTM). BTRA is conducted by the ERDC and the HGTM is jointly conducted by the US Tank Automotive Command Research Development and Engineering Center (TARDEC) and ERDC.

8. References

[1] Army Modeling and Simulation Office (AMSO) (15 September 2000), “Planning Guidelines for Simulation and Modeling for Acquisition Requirements and Training” [WWW Document]. http://www.amso.army.mil/smart/documents/guidelines/guidelines.doc (viewed 25 July 2002).

[2] Ahlvin, R. B. and Haley, P. W. 1992 *NATO Reference Mobility Model User’s Guide*, edition II, ERDC TR GL-92-19, Vicksburg, MS.

[3] STRICOM, 2000 “One Semi-Automated Forces (OneSAF) Operational Requirements (ORD),” version 1.1, STRICOM, Orlando, FL

[4] Deliman, N. C., Baylot, E. A., and Goerger, S. R. 15 June 2001 (unpublished briefing), “*Standard Mobility Suite for Variable Resolution Analysis and Cross-Model Consistency* - IPR,” ERDC, Vicksburg, MS and TRAC, White Sands Missile Range, NM.

[5] Baylot, E.A., Goerger, N. C; March 2003 “Ground Vehicle Mobility Steady State: Low Resolution (Level 1),” KEMA070007, US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD.

[6] Department of the Army Regulation 10 July 1997, “Army Regulation (AR) 5-ll: Management of Army Models and Simulations”. Headquarters, Department of the Army, Washington, DC.

[7] Miller, D. D. 2002 “Environmental Data Model,” version 1.0, Lockheed-Martin, Orlando, FL.

[8] Baylot, E. A. and Gates, B. Q., 2002 *Procedure for Categorizing Ground Vehicles*, ERDC/GSL TR-02-21, Vicksburg, MS.

[9] STRICOM, 199? "Warfighter's Simulation 2000 Software Description Document," STRICOM Orlando, FL

[10] SAIC, Feb 2003, "OneSAF XML Coding Standard Guide," US Army STRICOM, Orlando, FL.

[11] Shattuck, M. and Goerger, S.R. 13 February 2001, "COMBAT^{XXI} Java Style Guide." TRAC, White Sands Missile Range, NM.

[12] Gates, Burhman. Mar 2003, "JavaDocs: Standard Mobility (STNDMob) API v3.0.2.0.," ERDC, Vicksburg, MS.

9. Author Biographies

E. ALEX BAYLOT is a Research Engineer with the Mobility Systems Branch, US Engineer Research and Development Center (ERDC), Vicksburg, Mississippi. Mr. Baylot has 15 years experience with the ERDC in the field of vehicle mobility, computer modeling, operations research and system simulation. Mr. Baylot has authored or co-authored 4 technical reports and numerous papers in technical conferences and proceedings. Mr. Baylot has a MS in Engineering Management and a BS in Industrial Engineering all from Mississippi State University. Mr. Baylot is a licensed Professional Engineer in the state of Mississippi.

NIKI C. GOERGER is a senior research engineer for the US Army Engineer Research and Development Center (ERDC), Vicksburg, MS, and is currently the ERDC liaison to the US Army TRADOC Analysis Center, Monterey. She earned her B.S in Biological Engineering and M.S. in Agricultural Engineering from Mississippi State University, and her Ph.D. in Industrial Engineering from Texas A&M University. She plans, directs, and coordinates the research and integration of models and functionality into Army and Joint models and simulations, including OneSAF, COMBAT^{XXI}, and JWARS. Dr. Goerger is the Army Model and Simulation Office Standards Category Coordinator for movement representation of platforms in Army M&S. She has made significant contributions to mobility as a principal investigator for several programs and has coauthored numerous articles and presentations.

BURHMAN GATES is a researcher for the US Army Engineer Research and Development Center (ERDC), Vicksburg, MS. He is currently pursuing a master's degree in Industrial Engineering with an emphasis in Operations Research. He received his Bachelor of Science degree in Electrical Engineering from Louisiana State University. He has been employed at ERDC since June 1990 in the Mobility Systems Branch of the Geotechnical & Structures Laboratory.

SIMON R. GOERGER is a Major in the United States Army and is currently assigned to the Naval Postgraduate School, conducting doctoral work in modeling and simulations. He has served the military as an infantry officer and successfully commanded light cavalry forces. He has been deployed and led soldiers in Egypt and Haiti as part of two separate peace keeping missions. He earned his masters in Computer Science from the Naval Postgraduate School prior to being assigned as a software engineer at the US Army's Training and Doctrine Command Analysis Center (TRADOC) at White Sands Missile Range, NM. During that assignment, MAJ Goerger worked on COMBAT^{XXI}, the US Army and Marine Corps next-generation stochastic, closed form brigade and below entity level combat simulation.