1. VV&A FOR DIME/PMESII MODELS

Dean S. Hartley III

Principal, Hartley Consulting

Abstract: Verification and validation (V&V) are difficult processes and the complexity of the DIME/PMESII (diplomatic, information, military, economic/political, military, economic, social, information, infrastructure) domain adds to the difficulty. However, V&V, under the proper approach is not impossible and supports the need for accreditation (the "A" in VV&A). This paper describes a risk-based, entrenched methodology that meets the needs of DIME/PMESII modeling. The paper also covers the special cases of agent-based models and compressed and hyper-compressed VV&A.

The following paper on Verification, Validation and Accreditation (***VV&A***) is adapted from the author's contribution to work performed on that subject for Defense Advanced Projects Agency (***DARPA***) [4]. The paper is divided into seven major sections. The first section reviews some standard definitions and concepts. The next two sections introduce and define the concepts of risk-based VV&A and entrenched VV&A. The fourth section describes the verification & validation (***V&V***) process and emphasizes certain critical elements of the process, including metrics. The fifth section covers the special case of agent-based models and the sixth section covers the special case of triggered VV&A. The final section contains conclusions.

1.1 GENERAL CONCEPTS

This section introduces the concepts and definitions needed for DIME/PMESII modeling and VV&A.

1.1.1 Definitions

DIME/PMESII or PMESII models are models that encompass the necessary variables to yield measures of merit such as stability, peacefulness, freedom, and economic security. These and other, similar, measures require information on the situation over time. This information consists of variables that describe the state of the situation, ***state variables***. For convenience, these state variables have been divided into six ***PMESII*** categories: Political, Military, Economic, Support, Information, and Infrastructure. Similarly, the national means to influence situations are divided into the four ***DIME*** categories: Diplomatic, Information, Military, and Economic.

The basic VV&A definitions are provided in Department of Defense (***DoD***) documents [5].

Validation is the process of determining the degree to which a model or simulation is an accurate representation of the real-world from the perspective of the intended uses of the model or simulation. ***Verification*** is the process of determining that a model or simulation implementation accurately represents the developer's conceptual description and specification. Verification also evaluates the extent to which the model or simulation has been developed using sound and established software engineering techniques. These actions emphasize the need for metamodels.

User Data Verification is the use of techniques and procedures to ensure that data meet user-specified constraints defined by data standards and business rules derived from process and data modeling and that data are transformed and formatted properly. ***User Data Validation*** is the documented assessment of data

by subject matter expert (***SME***) and its comparison to known or best-estimate values as appropriate for the use in the intended model or simulation. These actions emphasize the need for model required inputs metadata.

Producer Data Verification is the use of techniques and procedures to ensure that data meet constraints defined by data standards and business rules derived from process and data modeling. ***Producer Data Validation*** is the documented assessment of data by SME and its comparison to known or best-estimate values within stated criteria and assumptions. These actions emphasize the need for collected data metadata, some of which may rely on the delivered output metadata for other models.

Accreditation is an official determination that a model is acceptable for a specific purpose.

1.1.2 Model Creation and V&V

Figure 1 illustrates the processes involved in creating a model, in particular a PMESII model, and the associated risks. The most important problem actually occurs prior to any of the model creation steps: despite a desire to model the "real world," most of the needed values and relationships are not observable or are incompletely understood. Therefore, most of the resulting model will be created using some proxy for the real world or a perceived version of the real world, rather than having direct connection to the real world.

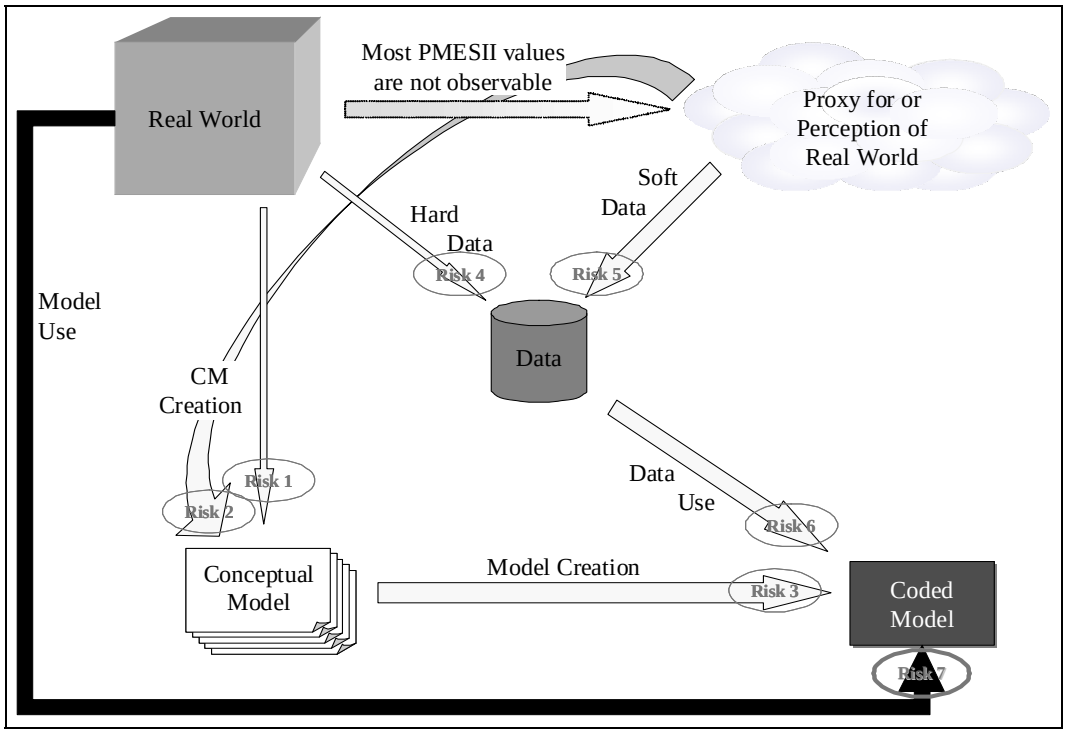

Figure 1. Construction of a DIME/PMESII Model

The canonical description of model creation specifies the creation of a conceptual model (***CM***) prior to creating the coded model. Most complex models actually are built incrementally, with coding of partial conceptual models, followed by additions to the conceptual model and more coding. However, for simplicity, this process is drawn as if it were a two step process. The conceptual model is created from observation, theory, and understanding of the real world (partially) and the proxy for the real world (more

substantially). The risks that something goes wrong in this step are labeled Risk 1 and Risk 2. The conversion of the CM to a coded model produces another opportunity for errors, labeled Risk 3.

Before the coded model can be used, data must be collected. Some of the data are available directly from the real world; however, in a PMESII model most of the data must be derived from a perception of the real world. (Note that the fact that a data value exists in the real world does not mean that it does not belong in the class of data from the proxy real world. If the data value represents an opinion or the result of a theoretical construct, it belongs in the proxy class.) The collection step introduces new opportunities for errors, labeled Risk 4 and Risk 5. The collected data must be prepared for use by the coded model. Each input must be in the proper format (syntactically correct) and must have the proper meaning (semantically correct). Semantic correctness is the more difficult problem, as it involves the matching of the models assumptions with the assumptions used in creating the data. This assumption matching can be as simple as unit matching (meters versus feet) or it can be very complex, involving issues of aggregation and population identification. The values must also be correct. The risk of errors in using data as model inputs is labeled Risk 6.

Once the model is created, the final risk lies in its use, which occurs in the real world. Depending on the nature of the use, the mismatches between the model and the real world may have large or small impacts. Note that, by definition, a model will have mismatches, as it is an abstraction – a purposeful omission of some detail – not an attempt at a full replication of the real world.

Verification and Validation are processes that have the purpose of identifying the problems that are the instantiations of the risks. Once problems have been identified, they can be either corrected or mitigated. Figure 2 illustrates the components of the V&V processes.

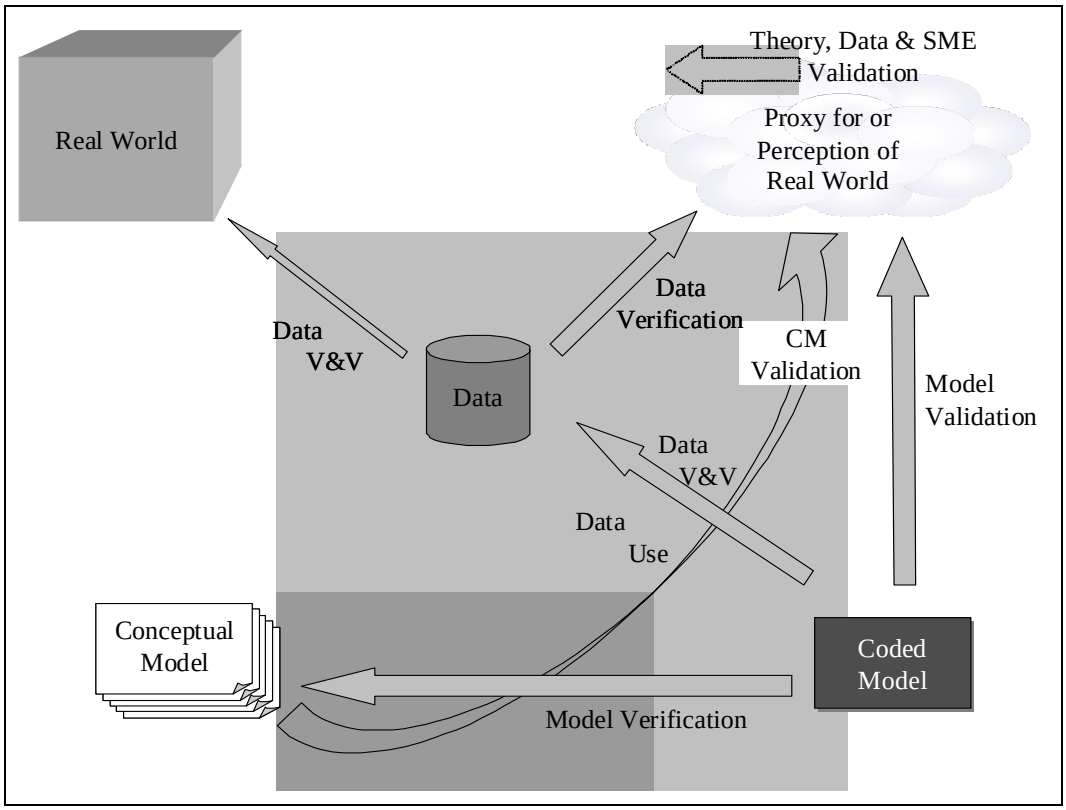

Figure 2. V&V of a Model

The first process consists of validation of the conceptual model. By the nature of the CM, it can only be directly checked against the proxy for the real world. Further, the appropriate criteria are determined by the intended uses of the model. The second step in this validation consists of checks on the theory and SME understanding that supports the CM, which, in the case of PMESII models, is likely to be limited (hence the arrow in the figure does not reach all the way back to the icon for the real world). The second process is the verification of the coded model. Model verification is a straightforward, if difficult, process.

User data V&V is shown by the V&V arrow between the coded model and the data icon in Figure 2. Producer data V&V is shown by the two arrows between the data icon and the real world and the proxy real world. The validation component is possible for that portion of the data drawn directly from the real world; however, data validation for the rest of the data is concerned with the connection of the data in the proxy real world and the real world. As indicated earlier, this arrow does not fully connect to the real world icon.

The final and most difficult process is model validation. Validating the coded model against the proxy reality is hard to do; however, that is not enough. The goal includes the second step of validation back to the real world (again using the short arrow at the top of the figure).

1.1.3 V&V Techniques

The bibliography in [4] contains a fairly comprehensive list of more than 70 citations, half on VV&A theory. However, there are two examples that are worthy of note. Osman Balci provides an excellent, brief description of the principles and techniques of verification and validation [2]. He classifies more than 75 techniques into four categories: informal, static, dynamic, and formal. These techniques range

from the informal *face validation*, in which domain experts judge whether the model seems to deliver believable results, to the formal *proof of correctness*, in which the model is proved to terminate and meet its specifications. In between, the techniques include the standard de-bugging and alpha testing and beta testing that are a necessary facet of rigorous modeling and programming. Knepell and Arangno produced a very readable and useful book on the subject [13]. They have extensive discussion of techniques, but also go beyond that to producing guides for the processes of using the techniques to assess models. They describe phases and introduce characterization of attributes, which can lead to verification and validation metrics.

1.1.4 The Reality of VV&A

There are few examples of rigorous applications of the VV&A process. Only 20 are listed in the bibliography in [4]. These range from the invalidation of a model [6] to the comparison of a combat model to an actual historical campaign [3] to a complete, multi-year VV&A effort on a PMESII modeling system [7, 8, 9, and 15]. The evidence is clear that VV&A is possible in practice, although it is not easy.

1.1.5 Special VV&A Problems for PMESII Models

The factors that generate the risks in Figure 1 are the same for general models and for DIME/PMESII models. The difference lies in the number of uncertainties and their range. In PMESII models almost nothing is certain and the range of possibilities seems unbounded. The point is that our understanding of reality in this area is poor and in those areas where we think we understand a little we know the inherent variability is large as compared to physical systems of inanimate objects.

The magnitude of the problem for verification and validation is not cause for despair; however, it does call for a careful evaluation of the standards that should be used in making an accreditation decision. When we speak of validation, we must emphasize coverage of the appropriate domains of events and variables and expect only general directional and magnitude correspondence for output values. When we address verification, we are challenged by the possible need to substitute a different social theory for the one used as a conceptual model, possibly within a single use of the model.

1.2 RISK-BASED VV&A

This VV&A methodology is an example of risk-based VV&A. Risk-based VV&A has four steps: identify and correct problems through V&V; identify residual risks; mitigate the risks; and decide on accreditation.

1.2.1 Identify and Correct Problems through V&V

Although for **very** simple models the V&V process might be sufficient to generate sufficient credibility for accreditation and use, in general the credibility generated by the V&V alone will not be sufficient to equal the validity needed. V&V should be an entrenched process throughout the modeling & simulation (***M&S***) lifecycle, identifying and correcting problems. Residual risks remain due to unknown differences between perceived/proxy world and real world.

1.2.2 Identify Residual Risks

The total risk is the difference between the credibility generated by the V&V and the validity needed for use. The first step is to categorize the types of risks. These types begin with the risk types identified in the discussion of V&V. These risk types are decomposed based on the modeling techniques used (e.g., agent-based modeling, systems dynamics, multi-attribute utility theory) and their particular vulnerabilities

to error. The risk types are also decomposed based on the subject matter of the model (e.g., economics, sociology, military activities) and its particular vulnerabilities to error. The second step is to determine the risk levels for each category for the M&S.

1.2.3 Mitigate the Residual Risks

The mitigation measures depend on the risk category and risk level. Each type of risk defines the types measures that are available to mitigate that risk. The risk level defines the urgency and/or quantity of mitigation that should be undertaken. Mitigation reduces, but doesn't eliminate risks. In general, not all possible mitigation measures that are available are selected for implementation. Some measures may be too expensive in time, effort, or funding to be performed. The remaining risk is accepted as necessary.

1.2.4 Decide on Accreditation

The nature and the level of the remaining risks may be acceptable for the intended M&S use. However, if the remaining risks are not generally acceptable, the accreditation may require caveats. The caveats may restrict the type of use or the problem domain or both. In any case, accreditation is a decision to accept risks, based on caveats. Obviously, if this is not possible, the decision must be made to not accredit the M&S.

1.3 ENTRENCHED VV&A

The V&V is conducted by asserting functionalities of the model and testing the assertions. In entrenched VV&A, there are three types of testing, developmental, triggered, and periodic. Additionally, there is an accreditation process. Figure 3 illustrates the elements of entrenched VV&A.

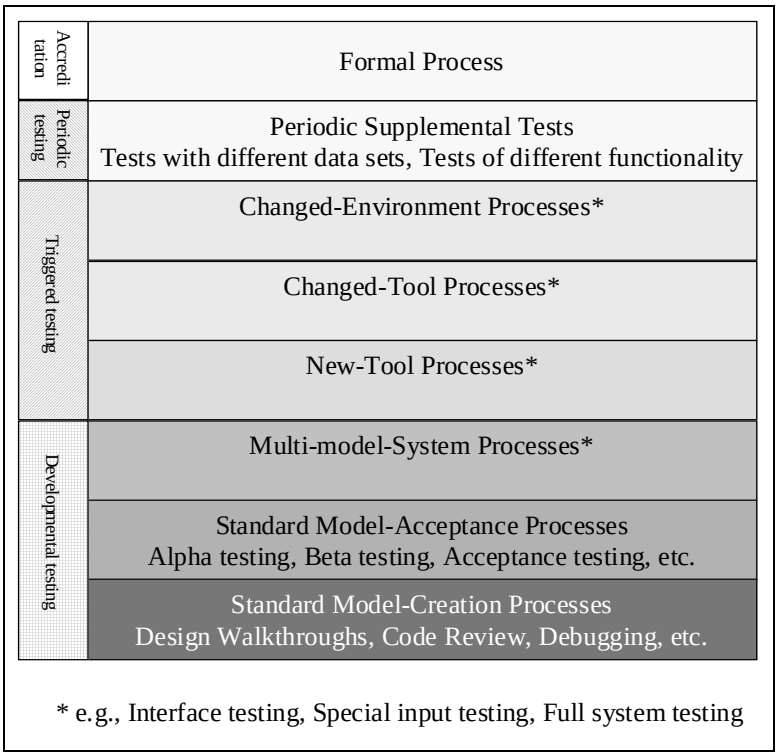

Figure 3. Entrenched VV&A

1.3.1 Developmental Testing

All tools go through developmental testing. Those created for the toolset undergo creation testing. New models, whether purpose created or brought in from outside, undergo acceptance testing. All tools undergo multi-tool system testing.

1.3.2 Triggered Testing

Certain events trigger additional testing. A new tool being brought into the toolset triggers the new-tool process. A proposed change to an existing tool (new sub-model, etc.) triggers the changed-tool process. Changes to the system being modeled require model changes.

The time available for VV&A in triggered testing is a major constraint. Two basic scenarios are envisioned, called compressed and hyper-compressed situations. In the compressed scenario, the events triggering the need for the changes allow for several weeks. In the hyper-compressed scenario, the time available is on the order of several hours. However, it is important that the available time comprises the entire time available and includes research, code changes, documentation of the changes, and the VV&A. Thus the time available for VV&A in the compressed situation is measured in days. In the hyper-compressed situation, the VV&A must be done in parallel with, or as part of the use of the modified simulation.

1.3.3 Periodic Testing

Periodically supplemental tests are performed to increase the understanding of the toolset. Tests are performed with new data sets. New tests of functionality that was previously rated as lower priority are

also introduced. In addition, a certain proportion of the tests include repeats of old tests to ensure the assumption that previously verified and validated functionality has not been lost.

1.3.4 V&V during Model Use

Each use of the model is an opportunity to observe it and either discover errors to be corrected or record successful runs. These discoveries should be used to increase the confidence in the model.

1.3.5 Accreditation

Accreditation follows the end of development, each triggered test, and each periodic test.

1.3.6 VV&A during the Model Lifecycle

The lifecycle of a model is divided into five phases (not including retirement), requirements, design, build, test and evaluate, and use. In general, the first four phases are illustrated as a spiral, repeating each phase until the model is ready for use. Figure 4 shows where the entrenched VV&A elements occur within this lifecycle.

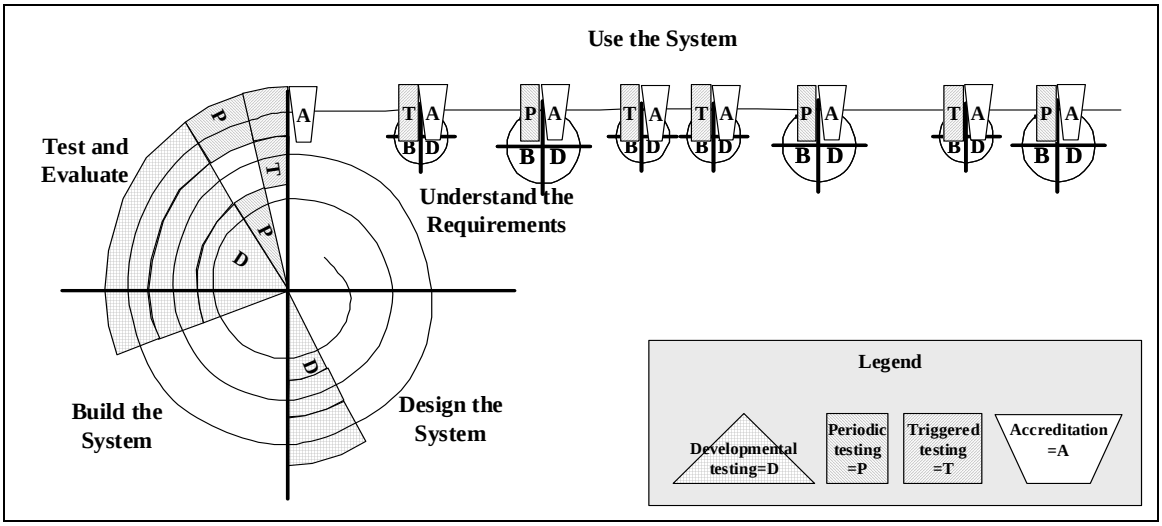

Figure 4. Lifecycle VV&A

Developmental testing includes V&V of conceptual models and debugging during the Build phase. The Periodic testing in this figure presumes a two cycle per year spiral process. The Triggered testing actually has no specific time relation – the events are placed to suggest this. Accreditation is similarly event-based, not time-based, following each periodic and triggered testing event.

1.4 VV&A METHODOLOGY

Figure 5 provides an overview of the VV&A process, which employs both risk-based VV&A and entrenched VV&A. As shown in the figure, once initiated the process is cyclical, with the details dependent on the non-V&V events that precede the V&V. The four steps of the process correspond fairly well to the phases defined in the book on simulation validation by Knepell and Arangno [12]. In that book, the initiate process is called the preparation phase; the define process is called the planning phase;

the execute process is called the application phase; and the evaluate process is divided into the evaluation and reporting phases.

Three of the branches in Figure 5 have already been discussed. The fourth branch, "Use the Model," emphasizes the point that at no time has the model been fully tested. As it is used, users will discover anomalies. These should be reported for correction or mitigation – or they may represent valuable insights into the workings of the real world. Additionally, reports of successful use should be recorded. These data will support future accreditation decisions.

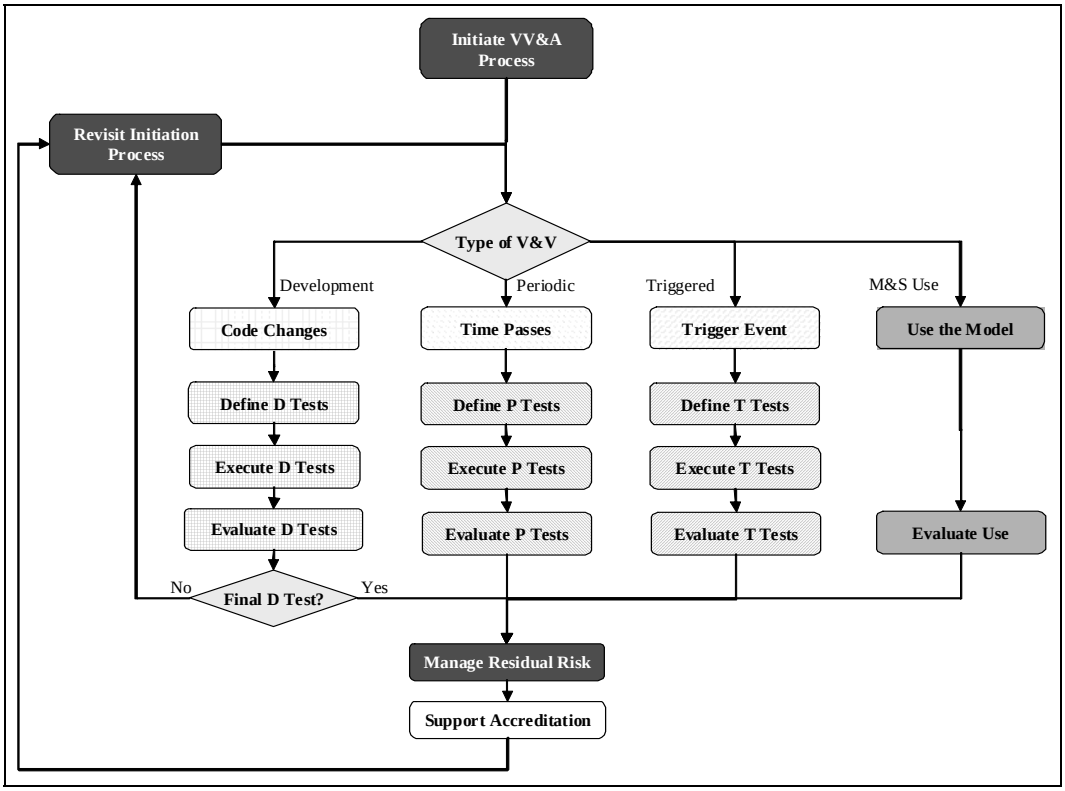

Figure 5. V&V Process Flow

There are a number of critical steps that must be performed at the beginning of the VV&A process. The first step is to define and assemble the VV&A team. Then the VV&A team must characterize the model and perform a stakeholder assessment. The documentation standards, V&V referents, validation criteria, and the general risks must all be defined. Finally, the team must perform some infrastructure creation to support the remainder of the VV&A process. The following subsections discuss several of the critical infrastructure elements.

1.4.1 Test Plans

Test plans are required to ensure that all tests are well thought-out, address the appropriate elements of V&V for the current stage of development and use, minimize the time and effort spent in their execution and evaluation, and are comprehensively recorded to prevent unwanted repetition and support desirable re-testing. Test plans will generally increase in comprehensiveness over time in Developmental Testing, will be abbreviated in Triggered Testing, and will have maximum comprehensiveness during Periodic Testing. Figure 6 shows a sample test plan outline.

- V&VPLAN EXECUTIVE SUMMARY
- PROBLEM STATEMENT
  - Intended Use
  - M&S Overview
  - M&S Application
  - Accreditation Scope
  - V&V Scope
- M&S REQUIREMENTS AND ACCEPTABILITY CRITERIA
- M&S ASSUMPTIONS, CAPABILITIES, LIMITATIONS, & RISKS/IMPACTS
  - M&S Assumptions
  - M&S Capabilities
  - M&S Limitations
  - M&S Risks/Impacts
- V&V METHODOLOGY
  - Planned Data V&V Tasks/Activities
  - Planned Conceptual Model Validation Tasks/Activities
  - Planned Design Verification Tasks/Activities
  - Planned Implementation Verification Tasks/Activities
  - Planned Results Validation Tasks/Activities
  - Planned V&V Reporting Tasks/Activities
- V&V ISSUES
- KEY PARTICIPANTS
  - Accreditation Participants
  - V&V Participants
  - Other Participants
- PLANNED V&V RESOURCES
  - Planned V&V Tasking and Funding
  - Planned V&V Timeline
- APPENDIX A: M&S DESCRIPTION
  - M&S Overview
  - M&S Development and Structure
  - M&S Capabilities and Limitations
  - M&S Use History
  - Data
  - Configuration Management
- APPENDIX B: M&S REQUIREMENTS TRACEABILITY MATRIX
- APPENDIX C: BASIS OF COMPARISON
- APPENDIX D: REFERENCES
- APPENDIX E: ACRONYMS
- APPENDIX F: GLOSSARY
- APPENDIX G: V&V PROGRAMMATICS
- APPENDIX H: DISTRIBUTION LIST
- APPENDIX I: ACCREDITATION PLAN

Figure 6. Test Plan Outline

1.4.2 Test Database

Testing requires a database (whether a formal, relational database or some other structure) to contain and retain the results of testing and to support the analysis and presentation of the results. Figure 7 displays the minimal set of elements required in a testing database. These elements have been identified through the actual exercise of the V&V process on a DIME/PMESII system of models [7, 8, 9, and 15]. Their purpose is to ensure both the improvement of the model and to support its accreditation.

- Contains a record for each test
- Pre-testing fields include, for example
  - Test ID #
  - Organization (group responsible for model or system)
  - Test type (Alpha, Beta, Periodic, Triggered, M&S Use)
  - Component (model name)
  - Subcomponent (module or function name, e.g., interface)
  - Function Type (group name, e.g., data, interface, software, human)
  - Function (particular function being tested within Function Type)
  - Test Sequence #
  - Date of test
  - Person performing test
  - Location of test
  - Hardware/Software Environment
  - Purpose of Test (description of test & its purpose)
  - Testing Conditions (prerequisites for test, such as data, state of the system)
  - Test Actions (specific instructions for performing test)
  - Expected Results (specific results that define successful completion of the test)
- Fields entered during testing
  - Actual Results (description of what actually happens)
  - Comments (free form)
  - Gravity of Problem (from enumerated list, e.g., minor, moderate, serious, catastrophic)
  - Recommended Fix (e.g., change code, change data, correct test procedure)
  - Timeline for Fix (estimated person hours required)
  - Fraction of Test Completed (1=100% of test was completed; 0=test was not performed)
  - Success Fraction (1=100% success; 0=complete failure)

Figure 7. Database Elements

1.4.3 Conceptual Model Metrics

Conceptual model validation result metrics are fairly simple. Because the validation problem is so difficult, the metrics must be defined by heuristics, which are easy to apply. Six grades from 5 descending to 0 are assigned to each component (model or module) and to the system as a whole, based on their conceptual models. The "5" label meets the most stringent standards, "4" next most stringent, and so forth. In general, the label for an ensemble will be lower than the labels for its components. Figure 8 contains the heuristics for each label, both for components and the system as a whole. Because of the nature of our knowledge in the DIME/PMESII domain, only levels 0 – 4 can be expected to be found in PMESII model components (highlighted). Our knowledge in the social sciences is so fragmented that connections among them are largely non-existent. This means that these connections, which must be created in the models, can be no greater than level 2. Therefore the expected ensemble levels can be no greater than level 2 (also highlighted).

In using data, distinctions are necessary to ensure that improper operations are not performed on numbers. ***Nominal data*** have labels associated which could be numbers or equally could be letters or words. No sequence is implied by the labels. The conceptual metrics are clearly not nominal data. ***Ordinal data*** also have labels which could be numbers or letters; however, a definite sequence, numerical or alphabetical is intended. An ***interval scale*** generally uses numbers and requires that the intervals have the same size; however, the zero point is arbitrary (as in the case for dates). Values on an interval scale can be subtracted from and added to by a scalar (unitless number), but not multiplied by a scalar. The conceptual metrics have a non-arbitrary zero point and it has been defined to have roughly equal intervals. That makes it ***continuous data***, which allows multiplication by a scalar. Such operations will be performed in the coded model validation, below.

Label	Component	Ensemble
5	Expresses fully validated theory, e.g., Newtonian physics with caveats on operations near light speed or in regimes subject to quantum effects	Expresses multiple Level 5 theories with fully engineered interfaces, e.g., fly-out model of ground-to-air rocket, involving, chemical reactions of propellants, ballistics, air flow, electronics, etc.
4	Expresses well researched theory involving considerable data checking and peer review, e.g., economic theory earning Nobel prize	Expresses multiple Level 4 or Level 5 theories with well researched interfaces, e.g., economic model ensemble used by Federal Reserve in setting U.S. interest and discount rates
3	Expresses theory supported by data and published in peer-reviewed literature	Expresses multiple Level 3, 4 or 5 theories with considerable peer-reviewed interfaces, e.g., some U.S. combat models
2	Expresses theory with rational basis, accepted by some experts as plausible (SWAG)	Expresses multiple Level 2 - 5 theories with plausible interfaces (SWAG)
1	Expresses a codified theory (WAG)	Expresses multiple Level 1 - 5 theories with codified interfaces (WAG)
0	Uncodified, mental model of uncertain consistency and completeness	Uncodified processes for connecting models of uncertain consistency and completeness

Figure 8. Conceptual Model Validation Metrics

1.4.4 Coded Model Verification Metrics

Because of the complexity of large PMESII models, complete verification is essentially impossible. Further, it is difficult to estimate the amount of progress toward complete verification that has been made. The basic measures are the number of successful verification tests and the number of unresolved problems. The results should be displayed over time to create a context for judging the results. The metrics should be collected for each model in the system and on system-wide tests and accumulated to produce overall metrics.

Rough metrics on attempted tests and successful tests should be prepared at the end of each testing effort. Figure 9 shows a report on the numbers and percentages of tests attempted and completed, divided by system module and test function. This report is taken from a Beta Test situation, in which 100% of all tests attempted are expected to be completed. In an Alpha Test situation, a 90% score would be considered a success, as the goal of an Alpha test is to identify problems that need correcting, rather than to ensure that problems have been fixed (as in a Beta test) or demonstrate the level of perfection of the system (as in a Final or Acceptance test).

Figure 10 shows a report of the accompanying report of numbers and percentages of successes vs. completed tests, with the same segmentation. These two sets of metrics emphasize current results.

Test Accomplishment by Objective & Tool

Company	(All)									
		Component								
Objective	Data	Controller	DIAMOND	ISSM	JCATS	Mine DB	Toolbox	UOB	XMT	Grand Total
1. Improve Function	Sum of Tests	17.00	22.00	9.00	17.00	7.00	7.00	27.00	8.00	114.00
	Sum of Desire	17.00	22.00	9.00	17.00	7.00	7.00	27.00	8.00	114.00
	Percent	100%	100%	100%	100%	100%	100%	100%	100%	100%
2. Integration w/XML	Sum of Tests	3.00	3.00		2.00		1.00	4.00	2.00	15.00
	Sum of Desire	3.00	3.00		2.00		1.00	4.00	2.00	15.00
	Percent	100%	100%		100%		100%	100%	100%	100%
3. Use C4I Data	Sum of Tests		7.00		6.00		1.00			14.00
	Sum of Desire		7.00		6.00		1.00			14.00
	Percent		100%		100%		100%			100%
4. OOTW Use Case	Sum of Tests		36.00	12.00	29.00		2.00	1.00		80.00
	Sum of Desire		36.00	12.00	29.00		2.00	1.00		80.00
	Percent		100%	100%	100%		100%	100%		100%
Total Sum of Tests		20.00	68.00	21.00	54.00	7.00	11.00	32.00	10.00	223.00
Total Sum of Desire		20.00	68.00	21.00	54.00	7.00	11.00	32.00	10.00	223.00
Total Percent		100%	100%	100%	100%	100%	100%	100%	100%	100%

Figure 9. Attempted vs. Completed Tests

Success by Objective & Tool

Company	(All)									
		Component								
Objective	Data	Controller	DIAMOND	ISSM	JCATS	Mine DB	Toolbox	UOB	XMT	Grand Total
1. Improve Function	Sum of Success	16.75	21.40	9.00	17.00	6.90	6.85	27.00	7.80	112.70
	Sum of Tests	17.00	22.00	9.00	17.00	7.00	7.00	27.00	8.00	114.00
	%Success	99%	97%	100%	100%	99%	98%	100%	98%	99%
2. Integration w/XML	Sum of Success	3.00	2.75		2.00		0.50	4.00	2.00	14.25
	Sum of Tests	3.00	3.00		2.00		1.00	4.00	2.00	15.00
	%Success	100%	92%		100%		50%	100%	100%	95%
3. Use C4I Data	Sum of Success		6.45		6.00		0.90			13.35
	Sum of Tests		7.00		6.00		1.00			14.00
	%Success		92%		100%		90%			95%
4. OOTW Use Case	Sum of Success		34.65	12.00	29.00		2.00	0.98		78.63
	Sum of Tests		36.00	12.00	29.00		2.00	1.00		80.00
	%Success		96%	100%	100%		100%	98%		98%
Total Sum of Success		19.75	65.25	21.00	54.00	6.90	10.25	31.98	9.80	218.93
Total Sum of Tests		20.00	68.00	21.00	54.00	7.00	11.00	32.00	10.00	223.00
Total %Success		99%	96%	100%	100%	99%	93%	100%	98%	98%

Figure 10. Completed vs. Successful Tests

Figure 11 gives an example of verification metrics that emphasize an historical perspective. Two years of tests are represented, with Alpha tests (whose purpose is to determine areas requiring work), Beta tests (whose purpose is to confirm corrections have been implemented), and Final tests (whose purpose is to demonstrate results to sponsor). “Name 1” and “Name 2” are the names of two models in the system. In the figure, “new tests” refer to tests that were not included in previous testing and “repeated tests” refer to tests that were included in previous testing. The repeated tests are especially significant when moving from one version to another.

Date	06/01/06	11/01/06	12/01/06	07/01/07	11/01/07	12/01/07
Event	Alpha Test	Beta Test	Final Test	Alpha Test	Beta Test	Final Test
Version Number	1	1	1	2	2	2
Successful Tests	81	106	109	99	116	119
Unresolved Problems	6	3	0	3	3	0
Total Tests	87	109	109	102	119	119
New Tests	87	68	0	65	79	0
Repeated Tests	0	41	109	37	40	119
System Tests						
Successful Tests	20	25	25	25	30	31
Unresolved Problems	1	0	0	2	1	0
Total Tests	21	25	25	27	31	31
New Tests	21	10	0	17	19	0
Repeated Tests		15	25	10	12	31
Name 1						
Version Number	2	2	2	3	3	3
Successful Tests	30	40	42	35	41	42
Unresolved Problems	3	2	0	1	1	0
Total Tests	33	42	42	36	42	42
New Tests	33	34	0	24	29	0
Repeated Tests		8	42	12	13	42
Name 2						
Version Number	3.1	3.1	3.1	3.7	3.7	3.7
Successful Tests	31	41	42	39	45	46
Unresolved Problems	2	1	0	0	1	0
Total Tests	33	42	42	39	46	46
New Tests	33	24		24	31	0
Repeated Tests		18	42	15	15	46

Figure 11. Historical Verification Metrics

1.4.5 Coded Model Validation Metrics

Figure 12 illustrates part of the process for developing useful metrics for validation of societal M&S. The central concept is coverage of the societal domain. At a gross level, an example would be that if a model did not cover any of the economic domain, its validity would be suspect. This figure shows that the political domain has been decomposed into a number of areas, some of which are PMESII only and some of which are also DIME (potential intervention type areas). Areas of no coverage at all are shown in this example with dark gray boxes. Because the system may have been decomposed by segregating some of the domains, you would not expect each model to cover all of the domains; however, the system as a whole should have good coverage, depending on the use to which it will be put.

In addition to coverage, the conceptual model validity for each component and its suitability for the given use are included in the considerations. The "scaled score" at the end of each section represents the average (scaled between zero and five) of the elements in the section. Later, these sections are averaged to produce values for each of the PMESII categories (not shown). The rows labeled as DIME factors are separately combined for each section and PMESII category (not shown). The DIME categories are combined because, as a practical matter, individual interventions might be performed by diplomats or military personnel, but have impact in one of the other areas, making disentanglement very difficult. In another part of the process (not shown), the tester considers the validity of the links among the areas.

Eventually, we arrive at the metrics. However, with hundreds of individual metrics, covering the many areas of the system, simple combinations of the results tend to obscure, rather than illuminate problems. To enhance the transparency of those tests, linked "spider" diagrams are useful (see Figure 13). The "spider" diagrams support visualization of multiple dimensions in a single chart and support an overview and segmentation by each individual model. Those diagrams provide the V&V Team with deep insight into the model's strengths and weaknesses that emerge from the VV&A process.

Name 2	Use A						
PMESII	DIME	PMESII Variables and DIME Functions	Relevant for this model? (0=no, 1=yes)	Theory (1=WAG, 2=SWAG, 3=Peer Rvw, 4=Accepted, 5=Proved)	Model Raw Score	Suitability for Use Fraction from Testing (granularity match, model match, right direction of change, right order of magnitude change)	Model Use Score
Political-Gov	DIME	Conducting nationwide elections	1	1	0.20	0.90	0.18
Political-Gov	DIME	Create local governments	1	1	0.20	1.00	0.20
Political-Gov	DIME	Educate local governments	1	1	0.20	0.90	0.18
Political-Gov	DIME	Establishing a mechanism for constitutional reform	1	1	0.20	0.90	0.18
Political-Gov	DIME	Establishing, staffing & funding effective transition national govt	1	1	0.20	0.90	0.18
Political-Gov	DIME	Providing advisors to national govt officials	1	1	0.20	1.00	0.20
Political-Gov	DIME	Supply local governments	1	1	0.20	0.90	0.18
Political-Gov	DIME	Training newly elected national political leaders	1	1	0.20	0.90	0.18
Political-Gov		Central authority is effective	1	1	0.20	0.90	0.18
Political-Gov		Central government exists	1	1	0.20	0.95	0.19
Political-Gov		Social services are adequate	1	1	0.20	0.90	0.18
		Sums	11		2.20		2.03
		Scaled Scores			1.00		0.92
Political-Pol	DIME	Maintaining compliance with peace accord milestones & conditions	1	1	0.20	0.90	0.18
Political-Pol	DIME	Mediating & negotiating w/ conflicting parties	1	1	0.20	0.90	0.18
Political-Pol	DIME	Monitoring government powersharing arrangements	1	1	0.20	0.90	0.18
Political-Pol	DIME	Transferring control of government functions to host nation officials	1	1	0.20	1.00	0.20
Political-Pol		Opposition party does not espouse force	1	1	0.20	0.90	0.18
Political-Pol		Opposition party doesn't attempt to dominate by force	1	1	0.20	1.00	0.20
Political-Pol		There are charismatic leaders advocating peace and stability	1	1	0.20	0.90	0.18
Political-Pol		There are no charismatic leaders advocating dissension	1	1	0.20	0.90	0.18
Political-Pol		There are no external forces advocating conflict	1	1	0.20	0.90	0.18
		Sums	9		1.80		1.66
		Scaled Scores			1.00		0.92
Political-ROL	DIME	(Re)building & monitoring new police force	0	3	0.00	0.00	0.00
Political-ROL	DIME	Assisting in establishing humane penal systems	0	1	0.00	0.00	0.00
Political-ROL	DIME	Assisting in establishing/reforming legitimate legal system	0	1	0.00	0.00	0.00
Political-ROL	DIME	Conducting constabulary operations	0	1	0.00	0.00	0.00
Political-ROL	DIME	Conducting war crimes investigations, tribunals, etc.	0	1	0.00	0.00	0.00
Political-ROL	DIME	Monitoring and reporting on corruption by govt officials	1	1	0.20	1.00	0.20
Political-ROL	DIME	Monitoring human rights practices	1	1	0.20	0.90	0.18

Figure 12. Validation Metrics Data Entry

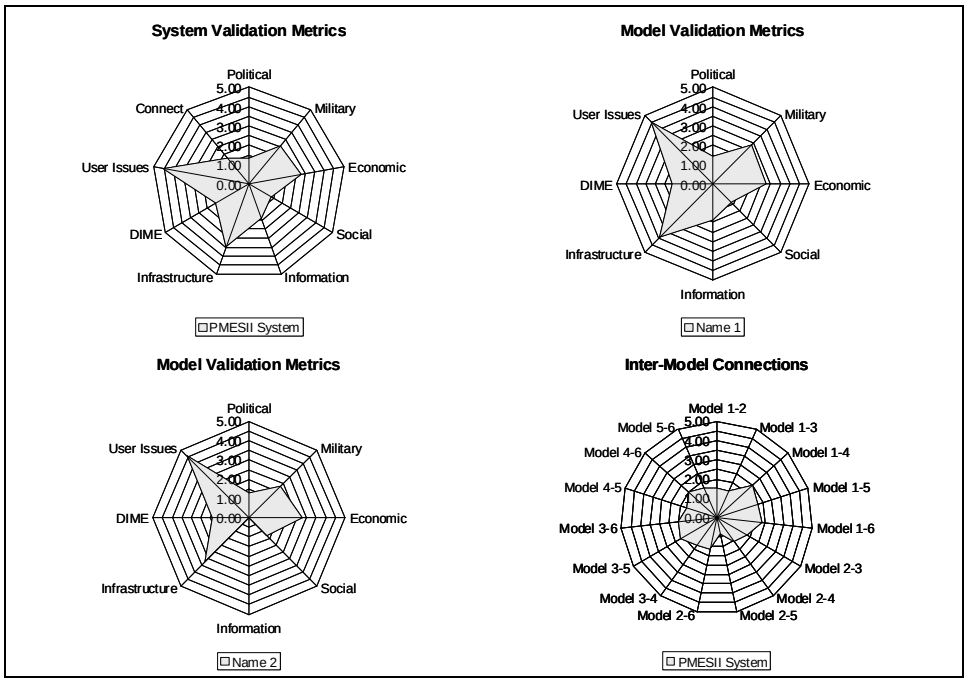

Figure 13. Coded Model Validation Display

1.5 V&V FOR AGENT-BASED MODELS

Dr. Michael Bailey characterized the problem of V&V for agent-based models (***ABM***s) in his keynote address to a workshop on the topic [1].

Agent-based simulations use modular rules and local reasoning to produce realistic and/or interesting emergent aggregate behavior.

Surprise is good**

Successful simulation testing (core to face/results validation) based on demonstrating credibility across the range of potential input.

Surprise not good**

** Refined later in this talk

It is not clear that the workshop resolved the problem.

1.5.1 Types of Agent-Based Models

Agent-based models come in two varieties, thin-agent and thick-agent models.

1.5.1.1 Thin-Agent Models

Thin-agent models are exemplified by the earliest agent-based models, such as the Irreducible Semi-Autonomous Adaptive Combat (***ISAAC***) model [12]. These models generate a set of symbols (often dots) within a (usually) rectangular grid. The dots may only reside at the discrete grid locations and move from one location to another and perform other actions based on pre-determined rules. The user defines a particular run by defining the values for the rule parameter set and setting up the grid space. In thin-agent models, the rule and rule parameter sets are small and the rules are simple.

For example, movement in ISAAC is determined by parameters that value the proximity to other agents of the various types, the goals (special grid points), and any barriers (forbidden grid points). The rule that governs movement is a combination of rules that calculate vectors toward (or away from) the various agent types and goals. ISAAC also includes parameters for capability to sense other agents (obviously impacting the movement rules) and the range attribute of the agent's weapon. The rules for using the weapon depend on sensing another agent, identifying it as a target, being in range, and so forth. ISAAC also provides parameters that define situational adjustments to the other parameters. The outcomes of the rules are modified through the use of random numbers generated by the model system.

Some thin-agent models, such as ISAAC, have an additional self-modification capability. This self-modification is called adaptation and involves sequential multiple runs of a scenario in which the parameters are varied with the aim of improving the score of each side. A typical methodology for this adaptation is the use of a genetic algorithm.

There are more recent thin-agent models that have more complex parameter sets and more complex rule sets; however, the general aim of thin-agent models is to have extremely fast models that can be run hundreds of thousands to millions of times (on super-computers) to explore the very large data space that even these simple models create.

These models might be more properly described as modeling environments than as simple models because the connection to the real world is defined by the user in setting up the scenario. That is, the user might decide that the dots represent individual soldiers, combat, units, mines in a mine field, or some other entity.

1.5.1.2 Thick-Agent Models

Thick-agent models are similar to thin-agent models in representing entities as having autonomy, attributes, methods, and goals and in allowing the simulation flow to be directed by the interplay of the actions of these autonomous agents, rather than by a script. However, in thick-agent models, the agents have a much richer, but more circumscribed identity.

For example, the Diplomatic and Military Operaions in a Non-warfighting Domain (***DIAMOND***) model is built with a specific connection to the real world in mind [14]. The entities include a special "civilian population" entity and numerous user-defined entities that are meant to represent organizational agents, such as military units, non-governmental organizations (***NGO***s), and terrorist organizations. These agents may have organizational command relationships and do have "party" or side affiliations, with specified relationships among the parties. There are complex relationships among the agent attributes and the terrain attributes that relate the size of the agent with its speed of passage. There are complex attrition methods that are associated with the agents. And there are complex communications and mission definition methods that are associated with the agents. These and other relationships all assume the identification of the agents that are created by the user with a particular type of real world entities. A particular DIAMOND entity may be created to be a model for a different real world entity, such as an Improvised Explosive Device (***IED***); however, the user must exercise great care in its attribute assignment and use to avoid violating the intrinsic DIAMOND model assumptions.

The general aim of thick-agent models is to model a particular subset of the real world in a particular manner. It is desirable for a thick-agent model to run as fast as possible, to support rapid analysis; however the goal is to enable 10 – 50 runs, rather than thousands or millions of runs as is the case for thin-agent models.

1.5.2 Named Input Variables

Named input variables are inputs with names that carry semantic content, such as "weapon type," as opposed to content free names, such as "x1."

1.5.2.1 Bare Model V&V

An ideal thin-agent model has no content and is perfectly protean. That is it can represent anything, given the proper assignment of meaning to its variables at run-time. Thus, there is nothing to verify or validate until a scenario is instantiated. There is no referent against which the basic model can or should be compared, no conceptual model for verification and no real world for validation.

As a practical matter, there are no ideal thin-agent models. Even ISAAC uses English terms, such as "time," "fire," and "proximity." It is not perfectly protean. These terms imply a connection to reality which requires V&V. Are the "distance" calculations valid for some notion of distance? Are there any problems with travel times or firing trajectories as far as violating fundamental laws of nature?

Some thin-agent models and all thick agent models employ other "named" variables. Each instance of a named variable implies a conceptual model or when defined explicitly specifies a conceptual model and thus presents the opportunity for and requires V&V. For example, if a model were to identify a parameter as a quantity of "love" between two agents, then the conceptual model of "love" must be identified and the implementation of this conceptual model must be verified. This conceptual model must be validated against the real world. Further, the coded model must be validated with respect to its implementation and its interaction with the rest of the model to ascertain that these two agents in fact act as they should with this quantity of love between them. However, for thin-agent models, their protean nature means that the bulk of the model is yet to be defined, and thus not available for V&V prior to the model use.

For the thick-agent model, this is the realm of traditional V&V. Almost all of the model is defined in the bare model and is subject to V&V.

1.5.2.2 Instantiated Model V&V

When a user defines a scenario for an agent-based model, he adds new content to the model. The scenario represents a particular instantiation of the bare model and is endowed with additional connections to the real world.

For the thick-agent model, we are still in the traditional world of V&V. The implementation of the IED mentioned above provides a simple example. Was it coded (including parameter selection) correctly and does it work in the model in the way a real IED would work (as well as needed for the particular use of the model). Because such additions to the model, often called "work-arounds," are relatively rare and constitute a small portion of the entire model content, the additional V&V required for the use of the model is clearly recognizable and not too onerous.

For the thin-agent model, most of the V&V must be delayed until the model is ready to use. This can place severe strains on the model use because time is a significant constraining factor. It is at this point that most variables are actually named. For example, the user may decide that the "fire" event in ISAAC could be used to represent entering into a "love" relationship, with a first "hit" by the "love" weapon generating "liking" and a second "hit" generating "loving." Validation of this added content could prove interesting. One strategy for avoiding this V&V process is to declare that a million runs will be made and that from these million it will be determined within which subset of the entire parameter space this meaning is valid. This pushes the V&V process from the time prior to running the model until after running the model.

1.5.3 Pattern Discovery

Pattern discovery pertains to searching for patterns within multitudes of thin-agent model outputs. Observing that one run out of thousands looks "like a classic pincer movement" does not validate the model. When a piece of toast is discovered with the "image of Elvis Presley" on it, that image may be attributed to a miracle or coincidence, but not to some property of the bread. Given a sufficiently large sample, the chances approach unity that a pattern of some sort will be discovered. This is the reason that, for a given significance level, repeated analyses of a set of data must be evaluated using a tighter significance level, whereas a single analysis is evaluated using the given significance level.

The implication is that the discovery of a "significant" pattern of behavior in a group of model outputs or even a region of "significant" patterns must be judged to be a weak validation event. Pushing V&V to an after the event process reduces its value.

1.5.4 ABM Conclusions

The emergence of complex behaviors from simple sets of rules is the principal reason for using ABMs. The claim is that just as the complexity of chemistry emerges unpredictably from the simple rules of atomic electron interactions, the best way to understand the complexity of human interactions is to investigate the emergence of complex interactions from simple simulation rules in an ABM. The more complex ABMs incorporate variable behaviors. These ABMs support connected sequences of runs in which the results of previous runs are used to modify the behaviors in subsequent runs. These ABMs are adaptive and can generate the co-evolution of the behaviors of one or more groups of agents (sides).

The ABMs of interest here are the models that include PMESII-type attributes, such as emotions, opinions, and social grouping valences. These attributes present problems because they are labeled as

PMESII-type attributes; however, it is not clear that they actually represent what their labels claim. Because our understanding of the true relationships among such variables is poor, validating the code is difficult. Further, the claim is made by some ABM proponents that the primitives that make up the relationships are what should be modeled and that by observing the emergent behaviors, users can make correlations between these and real-world behaviors with these labels, producing the desired validation. This claim is also problematic.

Therefore, this author recommends pursuing all V&V activities for ABMs just as described in the rest of this paper, with the exception that a very large portion of the work must occur after the scenario has been specified and before the model has been run.

1.6 TRIGGERED VV&A AS A SPECIAL CASE

As described above, triggered VV&A events are not planned. And in the normal course of events they will occur at the worst of times, usually when the pressure of a deadline makes time a scarce commodity. The DARPA project [4] considered these circumstances explicitly and developed three methodologies that would support VV&A under compressed or hyper-compressed situations. Triggers for reassessment include introduction of a new tool, modifications to an old model, and a changed environment. In the ***compressed*** scenario, the events triggering the need for the changes allow for several weeks for the model changes. In the ***hyper-compressed*** scenario, the time available is on the order of several hours. Note, however, that the available time will be employed to perform research, implement code changes, document the changes, and perform VV&A. Thus the time available for VV&A is measured in days for the compressed situation and in tens of minutes for the hyper-compressed situation.

1.6.1 Experimental Design Methodology

Prior to the need for compressed or hyper-compressed VV&A (in the “past” of Figure 14), the V&V team must characterize the societal model by creating a database of inputs with related outputs. This database represents the results of exploring the problem space, using experimental design techniques to reduce the data points needed. The team then must analyze the results to find a useful subset of tests that represent key elements of the useful input space of the model (filtering process in the figure).

At the time of need (“now” in the figure), the V&V team runs the subset against the modified model and compares the results against the stored results. The subject matter experts judge the results to determine whether the modified model should be considered “no worse than” the model prior to modification. If the SMEs are ambivalent or judge that something is wrong, further examination of the model is called for.

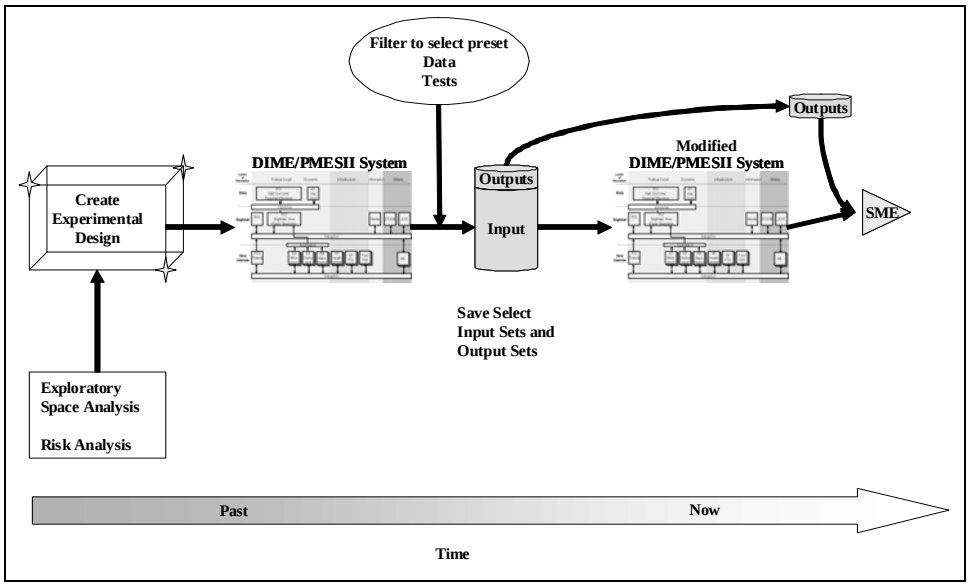

Figure 14. Experimental Design Methodology

1.6.2 Embedded Results-Traceback Methodology

In this methodology, the model must be instrumented to output sets of time-coded internal data that represent the system state over a model run. The system must also include storage of these data and a results-traceback system. Further, the results-traceback system must have knowledge of which variables feed into other variables. Naturally, this system must be constructed prior to the need for compressed or hyper-compressed VV&A.

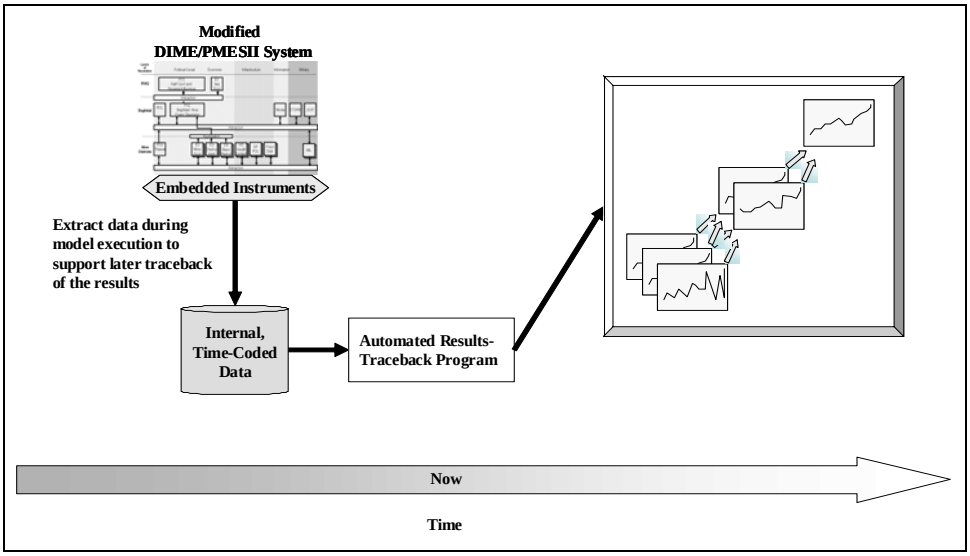

Figure 15. Embedded Results-Traceback Methodology

However, during a model run (shown as "now" in Figure 15) the user simply invokes the traceback system if any output variables seem anomalous. The system displays the results over time for a particular

variable, the results over time for predecessor variables, and the results over time for predecessors of those predecessors. The simplest analysis uses the pattern matching abilities of the human operator to identify the origin of unexpected results. More complex systems could invoke correlations of lagged variables to determine likely candidates for further inspection. In this methodology the user decides whether the predecessor variables adequately justify the questioned results. Continued uncertainty should result in further examination of the model.

1.6.3 Model Comparison Methodology

This concept relies on the use of an external, very high-level DIME/PMESII model which has entirely independent logic and on pre-processing of historical data. In this case, the Interim Semi-static Stability Model (ISSM) [11] is used as the high-level model. The V&V team calibrates the target model prior to using it on a new scenario by finding a convenient historical starting point and beginning the model then ("past" in Figure 16) and proceeding to the desired starting point of the simulation ("now" in the figure). The target model and the high-level model are both fed the starting data and any exogenous data modifications (for example, DIME interventions) made in the target model are also fed to the high-level model. At the end of the simulation period, the V&V team uses SMEs to evaluate the results of the target model, the high-level model, and the current situation. The concept is based on the idea that SME evaluations of the target model results, the high-level model results, and the historical results will yield a fair idea of the validity of the target model. If the results are judged to be insufficiently close, attempts are made to calibrate the target model and re-test. Once the calibration is successful, the users can run the target model with some degree of confidence that its results will be a fair extrapolation of the future state of the real situation.

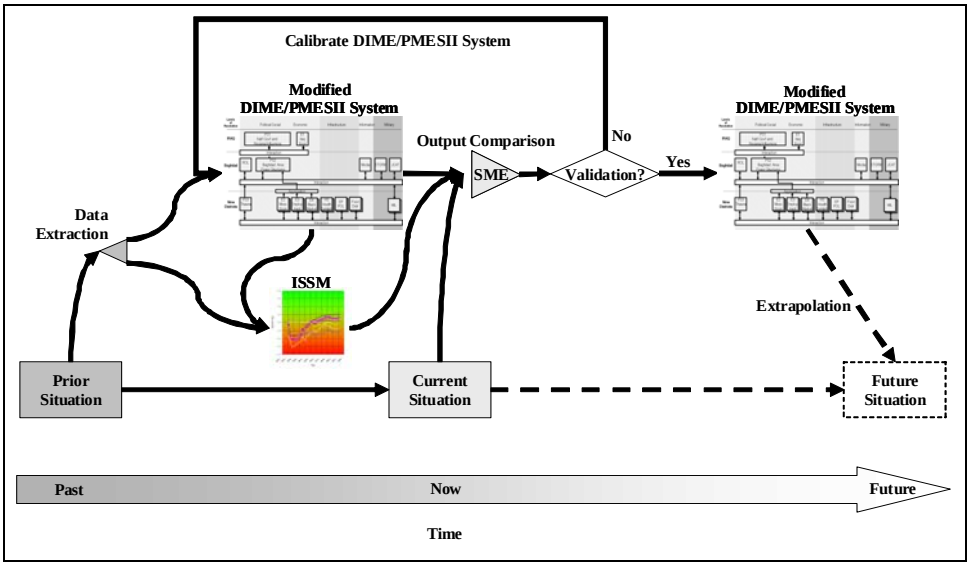

Figure 16. Model Comparison Methodology

1.6.4 Triggered VV&A Conclusions

The exploratory design methodology may require extremes in computing power in situations where data farming methods are required. It requires a large amount of preparatory work that must be done each time a new model version is released. It should be adequate in compressed situations; however, it may be too slow for use in hyper-compressed situations.

The embedded results-traceback methodology should be usable in compressed and hyper-compressed situations. However, it requires the creation of a data capture subsystem which must be updated each time a new model version is created. Further, the embedded nature of this approach means that it can be used at any time during the use of the PMESII model to investigate any questionable results.

The model comparison methodology should be usable in compressed and hyper-compressed situations. Its use with projected SME results during hyper-compressed situations may not be possible because of time constraints. It requires only minimal changes due to model version changes to ensure that any new exogenous events are automatically passed to the high-level model. Further, the embedded nature of this approach means that it can be used at all times during the use of the societal M&S to produce calibrated results. This approach is known to be feasible using the ISSM as the simpler model. Its use with other high-level models remains to be explored.

The use of any of these methodologies during baseline (or non-time compressed) VV&A will provide useful information, both for the baseline VV&A and for later compressed and (possibly) hyper-compressed situations. Using combinations of these methodologies would be beneficial in all situations: baseline, compressed, and hyper-compressed. These approaches to the problem of validating a modified model are orthogonal and thus likely to yield the most information with the least effort. However, these methodologies must be exercised during baseline VV&A and periodically during model use or the users will be too unfamiliar with their operation to use them effectively in time compressed situations.

1.7 OVERALL CONCLUSIONS

VV&A is a means to an end, not an end in itself. Verifying that a complex model has been implemented correctly is always difficult. Of course, DIME/PMESII models can be *very* complex. Validating one of these models is extremely daunting – if you think of the goal as being complete validation. However, if you think of it as a process of improving your confidence and understanding of the model – it is merely daunting. Further, the accreditation decision requires information that can only gained by a good verification and validation process. The reason anyone does VV&A is that he wants to *use* the model for an intended purpose. The specific use impacts the details of the verification and validation processes and the accreditation decision. In particular, V&V based on the specific use yields mitigation strategies that will improve the value of the use of the model.

BIBLIOGRAPHY

1. Bailey, Michael. "Validation Methodology for Agent-Based Simulations" in *Validation Methodology for Agent-Based Simulations Workshop, 1 May 2007*. Marine Corps Combat Development Command, 2007.
2. Balci, Osman. "Verification, validation and testing of models," in *Encyclopedia of Operations Research and Management Science*. Saul I. Gass and Carl M. Harris, Editors. Kluwer Academic Publishers, Boston, MA, 2001.
3. Bauman, Walter. *Ardennes Campaign Simulation (ARCAS)*, CAA-SR-95-8. US Army Concepts Analysis Agency, Bethesda, MD, 1995.
4. Clemence, Robert C., et al. *Verification, Validation, And Accreditation (VV&A)*. Evidence Based Research, Vienna, VA, 2007.
5. DoDD 5000.59.
6. Hartley, Dean S., III. *An Examination of a Distribution of TAC CONTENDER Solutions*, NMCSSC, Washington, DC, TM 101-75, 1975.
7. Hartley, Dean S., III. *FAST for the Warfighter: Test Strategy & Plan (Revision 3)*. DRC, Vienna, VA, 2004.

8. Hartley, Dean S., III. *MOOTW FAST Prototype Toolbox: FY04 Validation Strategy & Plan*. DRC, Vienna, VA, 2005.
9. Hartley, Dean S., III. *MOOTW FAST Prototype Toolbox: FY05 Validation Strategy & Plan*. DRC, Vienna, VA, 2005.
10. Hartley, Dean S., III. *MOOTW FAST Prototype Toolbox: FY05 Validation Strategy & Plan*. DRC, Orlando, FL, 2005.
11. Hartley, Dean S., III. *Operations Other Than War (OOTW) Flexible Asymmetric Simulation Technologies (FAST) Prototype Toolbox: ISSM v4.00 Analysts' Guide*. DRC, Orlando, FL, 2006.
12. Ilachinski, Andrew. "Irreducible Semi-Autonomous Adaptive Combat," in *Maneuver Warfare Science 1998*, Eds. F. G. Hoffman and Gary E. Horne. Marine Corps Combat Development Command, Quantico, VA, 1998.
13. Knepell, Peter L. and Deborah C. Arangno. *Simulation Validation: A Confidence Assessment Methodology*. IEEE Computer Society Press, Los Alamitos, CA, 1993.
14. Montera, Rick and Dean Hartley. *Military Operations Other Than War (MOOTW) Flexible Asymmetric Simulation Technologies (FAST) Prototype Toolbox*: DIAMOND Users' Guide. DRC, Orlando, FL, 2006.
15. Senko, Robert M. *Flexible Asymmetric Simulation Technologies (FAST) for the Warfighter: FY05 Final Verification Test Report*. DRC, Vienna, VA, 2005.

ABOUT THE AUTHOR

DEAN S. HARTLEY III is the Principal of Hartley Consulting. Previously he was a Senior Member of the Research Staff at the Department of Energy Oak Ridge Facilities (Oak Ridge National Laboratory, Y12 Site and East Tennessee Technology Park).

Hartley graduated *Summa Cum Laude*, Phi Beta Kappa from Wofford College in 1968, majoring in mathematics and foreign languages. He received his Ph.D. in piecewise linear topology from the University of Georgia in 1973.

Hartley is a past Vice President of the Institute for Operations Research and Management Science (INFORMS), a past Director of the Military Operations Research Society (MORS), past President of the Military Applications Society (MAS), and a member of the College on Simulation of INFORMS. He also serves as the Technical Advisor for Operations Research and Modeling to the International Psychopharmacology Algorithm Project (IPAP). Hartley is a Senior Fellow with the George Mason University School of Public Policy, a consulting resource for the Naval Postgraduate School (NPS), Modeling, Virtual Environments & Simulation (MOVES) Institute, and a Research Fellow with the University of Alabama in Huntsville, Center for the Management of Science and Technology (CMOST).

Hartley has published *Predicting Combat Effects*, contributed chapters to three other books, and written more than 150 articles and technical documents. His interests include modeling of combat, operations other than war (OOTW), stability and support operations (SASO), and stability, security, transition and reconstruction (SSTR) operations, verification and validation of models, psychopharmacology modeling, and simulation. His website is http://dshartley3.home.comcast.net.