Conceptual Model Development for C4ISR Simulations

Dale K. Pace*
The Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, Maryland 20723-6099
USA
(240) 228-5650; (240) 228-5910 (FAX)
dale.pace@jhuapl.edu

Abstract

A simulation conceptual model is the simulation developer's way of translating modeling requirements (i. e., what is to be represented by the simulation) into a detailed design framework (i. e., how it is to be done), from which the software, hardware, networks (in the case of distributed simulation), and systems/equipment that will make up the simulation can be built. Standards for describing systems that exchange information and for interoperability among simulations are enabling enhanced capabilities; but, unfortunately, similar standards do not exist for simulation decomposition into entities and processes, for representation abstraction of the subject simulated, and for how to describe and document the simulation conceptual model. This situation is already a problem for command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) simulations and will become a greater problem as expectations about C4ISR simulation capabilities increase, especially relative to information operations. This paper discusses fundamental issues in simulation conceptual model development with particular attention to guidelines for conceptual model decomposition and for abstraction in describing simulation elements. The paper also suggests how to document a simulation conceptual model. (Keywords: conceptual model, validation, fidelity, simulation).

1. Introduction

Increased interoperability among military systems both enhances their capabilities and creates problems in their use, especially problems related to transformation of data into knowledge and information. For example, if sensor tracking information can now be shared so that data from more than one sensors may be presented at a single location, how are differences about target location, vector, identification, characterization, etc. to be addressed? Will the first report dominate? Will data from the "best" sensor dominate? Will a composite be developed from the collection of data? Will all receiving the data process it in the same way? This is not a new kind of problem. Making sense of variant data from different sources and fusion of disparate data have long challenged military personnel. The new dimension of this challenge is the vast increased in the volume of data which now must be addressed by computational machinery without extensive human involvement, yet the resulting processed data ("knowledge" and "information") must be reliable enough across the entire spectrum of possible situations that operators and commanders may act upon it appropriately.

* Material in this paper drew heavily upon simulation conceptual model research sponsored by the Defense Modeling and Simulation Office (DMSO).

Military simulation faces a similar challenge. Interoperability advances make it possible to connect simulations together, even to interact with live forces[1]. This places new emphasis on reliable and compatible results from the simulations (their "data") so that their interaction can be meaningful and correct. This kind of capability has driven a vision for interaction among simulations from different parts of the defense community, so that data from simulations used for system design in acquisition can interact with data in a campaign simulation analyzing force structure as well as with a decision support system that is involved in training or even in support of an operation. This vision is called Simulation Based Acquisition (or some similar rubric).

Interoperability has given much greater importance to compatibility and reliability of both operational data and simulation data. This is true for military systems with the wide variety of information from sensors and about force readiness/capability, for simulations which are embedded in military systems and used in their operation and control, and for simulations which represent military systems in analysis, planning, and assessment. A key to achieving such compatibility and reliability in simulation data is the simulation conceptual model because the simulation conceptual model is the basis for judgment about the appropriateness (validity[2] evaluation) of simulation data for all conditions not specifically tested.

Economic considerations emphasize the importance of reuse of simulation components. Likewise, the movement toward simulation based acquisition emphasizes better understanding of simulation quality (especially articulation of simulation fidelity) and compatibility among simulations, reusable simulation components, real systems, and standard data. These emphases increase the importance of conceptual model development methods since the conceptual model is the basis for judgment about reuse appropriateness and simulation fidelity. This gives conceptual model development increasing importance in C4ISR simulations because of C4ISR simulations' important role in military assessments in the information-intensive future that lies before us.

In recognition of the importance of simulation conceptual model methods for simulation fidelity and validity as well as for reuse of simulation components, the Defense Modeling and Simulation Office (DMSO) has sponsored conceptual model research as part of its simulation verification, validation, and accreditation (VV&A) endeavors. This paper draws heavily upon material published under DMSO sponsorship about simulation conceptual model development at the Simulation Interoperability Standards Organization (SISO) Simulation Interoperability Workshops (SIWs) 1998-2000, Society for Computer Simulation (SCS) conferences and Military Operations Research Society (MORS) symposia in 1999 and 2000, and elsewhere, including the updated version of the DoD *Recommended Practices Guide for VV&A* of simulations [accessible

---

[1] Current distributed simulation terminology uses "live forces" for actual military systems that interact with the simulation, "virtual forces" for manned simulators involved in the simulation, and "constructive forces" for military forces that are represented only by computer code within the simulation.

[2] Simulation "validity" and "fidelity" are often confused. Fidelity is an absolute measure of the closeness of the simulation's representation to the real world. On the other hand, validity is a relative measure. It addresses the appropriateness of the simulation for an intended or specified application. Thus, a simulation, whose fidelity is unchanged, could be valid for one application but not for a different application.

via the DMSO website: http://www.dmso.mil].

Version 3.0 (November 15, 1999) of the DoD Joint Technical Architecture (JTA) prescribes Integrated Computer Aided Manufacturing Definition (IDEF) models as standards for data and process descriptions of systems that exchange information [http://www-jta.itsi.disa.mil/]. This methodology is also being applied specifically to object oriented systems in a new set of IEEE standards [IEEE, 1998]. This has increased understanding of architecture issues for C4ISR and other systems. The High Level Architecture (HLA) [accessible via the DMSO website: http://www.dmso.mil] is likewise imposing a coherent discipline for information exchanges and interoperability among simulations in distributed simulation applications within the U.S. Defense community, and has also influenced distributed simulation activities in Europe and elsewhere. Unfortunately at present, similar widely accepted approaches do not exist for developing and documenting a simulation conceptual model. This situation is already a problem for command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) simulations and will become more of a problem as expectations about C4ISR simulation capabilities increase, especially relative to information operations[3]. Only as the entire C4ISR community comes to appreciate issues related to simulation conceptual model development will needed improvements occur. This community includes those who set requirements for C4ISR systems, those who use them, those who develop simulations of C4ISR systems, and those who analyze systems and capabilities using C4ISR simulations. Only when all of these understand why improvements in methods for developing conceptual models of C4ISR simulations are essential will those improvements occur as needed throughout the entire C4ISR domain.

This paper starts with a description of the simulation conceptual model, its components, its interaction with simulation requirements, and an overview of the conceptual model development process. Then it addresses decomposition of the mission space (the domain represented in the simulation) into entities and processes for the simulation. Next the paper discusses representation abstraction for those entities and processes. This is followed by a discussion of conceptual model documentation. In each sections, the paper focuses on C4ISR simulations.

2. The Simulation Conceptual Model

2.1 Terminology

Terminology is always a problem as a discipline evolves, and this is certainly true in regard to the connotation for "simulation conceptual model." The current version of the DoD *Glossary of Modeling and Simulation Terms* [accessible via the DMSO website: http://www.dmso.mil] does not define "simulation conceptual model." It follows the approach used by the Distributed Interactive Simulation (DIS) community in the early and mid-1990s by defining "conceptual model" as the agreement between the simulation developer and user about what the simulation

[3] Human behavior representation in simulations and use of computer generated forces/semi-automated forces in simulations that interact with people in war games, in simulators, or in live forces is receiving significant attention -- as illustrated by the number of participants and papers in the annual conferences on Computer Generated Forces (CGF) and Behavior Representation (BR) [accessible at http://www.sisostds.org/], and has major consequences for C4ISR simulations. Unfortunately, representation of human behavior in military simulations is still in its infancy (Pew and Mavor 1998).

will do. The HLA Federation Development and Execution Process (FEDEP) gives the Federation Conceptual Model a slightly different flavor from that presented in this paper because the FEDEP uses the term Federation Objectives for what most simulation developments call "requirements" and the term Federation Requirements for what most simulation developments call "specifications."

Others also have different connotations for conceptual models. The *Journal of Conceptual Modeling* [http://www.inconcept.com/JCM] is dedicated to data modeling, design, and implementation issues for those concerned with databases. The terms "conceptual", "logical", and "physical" are frequently used in data modeling to differentiate levels of abstraction versus detail in the model although there is no general agreement about precise definitions of these terms. Some approach conceptual modeling from knowledge engineering and cognitive science perspectives. For them, conceptual modeling involves constructing representations of human knowledge. Dalugach and Skipper [2000] suggests dynamic conceptual graphs and Tracked Repertory Grids for large scale knowledge engineering problems characteristic of many contemporary simulations. One of the primary uses of such representations is to verify analysts' understanding of users' knowledge of an application domain before proceeding with database/simulation design and implementation. Others use system engineering tools and principles to link simulation requirements and behavior representation in conjunction with DMSO's Conceptual Model of the Mission Space's technical framework (standards for knowledge creation and integration), a common repository of mission space models, and toolset [Dubois and Might, 2000]. Research in conceptual modeling for this community often focuses on developing formalisms (often graphical) to represent knowledge.

Some take a restrictive view of the conceptual model, restricting it to simulation representational aspects and not addressing simulation control features, as in early verification and validation oriented paradigms developed in the late-1970s by the Society for Computer Simulation's Technical Committee on Model Credibility [Schlesinger, 1979] and elaborated upon by Sargent [1985] and later by Balci [1991]. This restrictive view of conceptual modeling activities is also used by Oberkampf *et al.* [2000] in their work on estimation of total uncertainty in computational simulations that deal with numerical solution of systems of partial differential equations.

The connotation for simulation conceptual model used in this paper is that in the current (revised) version of the DoD *Recommended Practices Guide for Verification, Validation, and Accreditation (VV&A)* [accessible via the DMSO website: http://www.dmso.mil]. This connotation is particularly useful for simulation development and to support validation activities.

Definition: A simulation conceptual model is the simulation developer's way of translating modeling requirements (i. e., what is to be represented by the simulation) into a detailed design framework (i. e., how it is to be done), from which the software, hardware, networks (in the case of distributed simulation), and systems/equipment that will make up the simulation can be built.

A conceptual model is the collection of information that describes a simulation developer's concept about the simulation and its pieces. That information consists of assumptions,

algorithms, characteristics, relationships, and data, which describe how the simulation developer understands what is to be represented by the simulation (entities, actions, tasks, processes, interactions, etc.) and how that representation will satisfy simulation requirements. Because some simulation requirements have implementation implications (such as the simulation must interact in real-time with hardware, software, or systems), the simulation conceptual model must be responsive to these requirements as well as to representation factors. This is one reason a restrictive view of the simulation conceptual model can not be fully responsive to simulation requirements. The more perspicuous and precise the conceptual model, the more likely the simulation development can fully satisfy requirements and can demonstrate that the requirements are satisfied (i.e., validation). As Teeuw and van den Berg [1997] put it, “A model in the ‘conceptual world’ captures all essential aspects of the "real world". The word essential refers to a specific modeling objective.”

The term “simulation developer” is used broadly in this paper. It includes those who create new individual simulations, “federates” in HLA parlance, those who create distributed simulations, HLA “federations,” and those who modify existing (i.e., legacy) simulations. High quality simulation conceptual models are needed for new individual simulations (whether used as imbedded support for decision support systems or used in other ways), for distributed simulations, and for modifications of existing simulations.

A simulation conceptual model *should be* a primary mechanism for clear and comprehensive communication among simulation developer design and implementation personnel (systems analysts, system engineers, software designers, code developers, testers, etc.), simulation users, subject matter experts (SMEs) involved in simulation reviews, and evaluation personnel, such as those involved in verification, validation, and accreditation (VV&A).

2.2 Simulation Conceptual Model Components

A simulation’s conceptual model consists of the simulation context, the simulation concept (with its mission space and simulation space aspects), and simulation elements. Each of these is discussed briefly below. The relationships among these are illustrated by Figure 1.

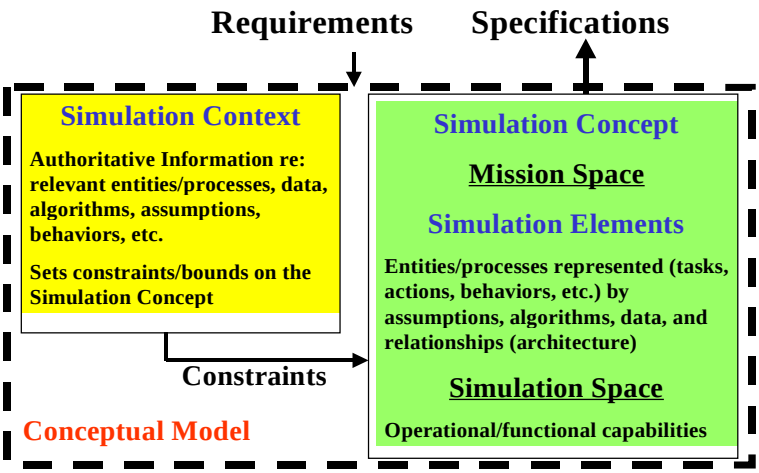

Figure 1. Conceptual Model Components

1) The *simulation context* provides "authoritative" information about the domain which the simulation is to address. In simulations that provide realistic representation of physical processes, the laws of physics and principles of engineering are part of the simulation context. For many military-related simulations, the simulation context includes standard organizational structures and general doctrine, strategy, and tactics. This is especially characteristic of C4ISR simulations. Often the simulation context is merely a collection of pointers and references to sources that define behaviors and processes for things that will be represented within the simulation. Special care, especially for distributed simulations, must be used when algorithms are taken from more than one source to ensure that sources do not employ contradictory assumptions or factors (such as different models for the shape of the Earth, differences in characterizing the environment, etc.). The information contained in the simulation context establishes boundaries on how the simulation developer can properly build the simulation.

2) The *simulation concept* describes the simulation developer's concept for the entire simulation application (all the federates and other pieces in a distributed simulation, i.e., everything that comprises the simulation) and explains how the simulation developer expects to build a simulation that can fully satisfy user-defined requirements. The *simulation context* establishes constraints and boundary conditions for the *simulation concept*. If the simulation is concerned with realistic representation of missile flight, then laws of physics and principles of aerodynamics are part of the *simulation context*, making the *simulation concept* accommodate conservation of momentum, etc. Unrealistic, cartoon representation of missile flight would not necessarily be so constrained. The *simulation concept* includes *simulation elements*, i.e., the things represented in the simulation. The *simulation concept* includes all *simulation elements* and specifies how they interact with one another: the simulation's *mission space*.

The *simulation space* part of the *simulation concept* includes *all additional information needed* to explain how the simulation will satisfy its objectives. Such *additional information* often addresses control capabilities intended for the simulation, such as pause and restart capabilities, data collection and display capabilities, and how data and simulation control factors can be

entered into the simulation (by keyboard, by voice, by gesture or touch, or by feedback from parts of the simulation). *Simulation space* characteristics range from identification of specific kinds of computing systems (hardware and operating systems) and timing constraints so that real systems can be part of the simulation (such as hardware in the loop unitary simulations or involvement of live forces in distributed simulations) to the kinds of simulation control capabilities described above. Some simulation space considerations are closely related to implementation issues for the simulation. For example, selection of a parallel computing architecture has implications for algorithms used to describe simulation elements.

3) A *simulation element* consists of the information describing concepts for an entity, a composite or collection of entities, or process which is represented within a simulation. It includes assumptions, algorithms, characteristics, relationships (especially interactions with other things within the simulation), data, etc. that identify and describe that item's possible states, tasks, events, behavior and performance, parameters and attributes, etc. A simulation element can address a complete system (such as a missile or radar), a subsystem (such as the antenna of a radar), an element within a subsystem (such as a circuit within a radar transmitter), or even a fundamental item (such as an atom). It can also address composites of systems (such as a ship with its collection of sensors, weapons, etc.). It should be noted that a person, part of a person (such as a hand), or a group of people can likewise be addressed by a simulation element. It can also address a process such as environmental effects on sensor performance.

A primary function of the *simulation conceptual model* is to serve as the mechanism by which simulation requirements are transformed into detailed simulation specifications (and associated simulation design) which fully satisfy the requirements. This transformation is easiest and most reliable if both the requirements and the specifications can be expressed in the same descriptive formalism because every translation from one descriptive formalism to another introduces an additional source of potential error, even for mundane transformations. Errors may result from something as simple as failure to translate units, which caused failure of NASA's Mars Climate Orbiter in September 1999 [http://lunar.ksc.nasa.gov/mars/msp98/orbiter/]. Misinterpretation of symbols and factors is also possible, especially for terms with multiple definitions. For example, the statistical term "mean" can be confusing because the arithmetic mean is quite different from the geometric mean. Similar problems result from absence of a convenient construct or concept in the subsequent descriptive format to address all aspects of information contained in the format from which the information is being translated. This is a well-known problem in natural language translation, and is important for C4ISR simulations of Joint and Combined operations.

2.3 Evaluation Criteria for a Simulation Conceptual Model

There are four primary evaluation criteria for a simulation conceptual model:

1) **Completeness:** the simulation conceptual model identifies *all* representational entities and processes of the problem domain, the "mission space" in DoD parlance, and *all* control and operating characteristics of the simulation, "simulation space," needed to ensure that specifications for the simulation fully satisfy simulation requirements.

2) **Consistency:** representational entities and processes within the conceptual model are addressed from compatible perspectives in regard to such features as coordinate systems and units, levels of aggregation/deaggregation (which is of particular importance in C4ISR simulation), precision, accuracy, and descriptive paradigms.
3) **Coherence:** the conceptual model is organized so that all elements of both mission space and simulation space have function (i.e., there are not extraneous items) and potential (i.e., there are no parts of the conceptual model which are impossible to activate).
4) **Correctness:** the simulation conceptual model is appropriate for the intended application and has potential to perform in such a way as to fully satisfy simulation requirements.

Lindland et al [1994], Teeuw and van den Berg [1997], and others address quality of conceptual models from various perspectives. C*ompleteness*, *propriety* (pertinence), *clarity*, *consistency*, *orthogonality* (*modularity*, the independence of aspects of the subject represented), and *generality* (implementation independence).

2.4 Implementation Dependence

Implementation independence is sometimes presented as an essential aspect of a simulation conceptual model. The HLA FEDEP [accessible via the DMSO website: http://www.dmso.mil] says, "The federation conceptual model provides an implementation-independent representation that serves as a vehicle for transforming objectives into functional and behavioral capabilities, and provides a crucial traceability link between the federation objectives and the design implementation. This model can be used as the structural basis for many federation design and development activities (including scenario development), and can highlight correctable problems early in the federation development process when properly validated." This use of "implementation independent" is not absolute. If the objectives for the federation indicate that a particular simulation (federate) is to be involved or that particular kinds of live forces are to be involved, then the federation conceptual model can not be totally implementation independent. Such objectives are normally stated for a HLA federation.

It is very desirable for a simulation conceptual model to have "reasonable" implementation independence. This allows it to be most useful (not uniquely tied to a particular implementation) and makes evaluation of it easier (since some implementation aspects need not be involved in the evaluation). However, most simulation requirements have a variety of implementation aspects. They may specify that the simulation must be capable of running on a certain kind of computer platform, use a particular operating system, be capable of real-time operation, etc. Even the HLA compliance requirement for contemporary DoD simulations keeps a simulation conceptual model from implementation independence in an absolute sense. A conceptual model that is fully responsive to the simulation requirements must accommodate such implementation aspects of the requirements. Therefore, a simulation conceptual model will typically have some level of implementation dependence.

3. Development of a Simulation Conceptual Model

3.1 Interaction with Simulation Requirements

Simulation requirements and conceptual model development are a classic "chicken-egg" pair. They each stimulate and derive from the other. Conceptual model development may even begin prior to completion of simulation requirements. Conceptual model development may reveal problems with simulation requirements, especially if there has not been a rigorous validation of simulation requirements prior to initiation of conceptual model development or if the best of contemporary requirements engineering practices have not been employed comprehensively. As the simulation conceptual model is developed to fully satisfy simulation requirements, inconsistencies among requirements and lack of balance among the requirements (some very lax and others very stringent in the same general area) may become apparent. Development of the conceptual model may also reveal serious holes in the requirements. "Holes" are areas where the simulation developer is left to his own initiative about what the simulation should be able to do. A good simulation development program will insist upon use of the best requirements engineering practices, will encourage early formal and rigorous validation of simulation requirements, and will ensure that requirement deficiencies uncovered during conceptual model development are corrected with appropriate modifications to simulation requirements.

Development of an explicit conceptual model that is complete, consistent, clear, and correct will cause most deficiencies in simulation requirements to become evident. Such requirement deficiencies are inconsistencies and contradictions, holes, inclusion of extraneous demands, lack of balance, and lack of clarity. Such deficiencies are not a critique about the appropriateness of what the simulation is to do. Detection and correction of such requirements deficiencies early is very important for affordable, successful simulation development since the majority of software faults are attributed to requirements deficiencies and it can cost up to 100 times as much to correct an error late in development as it costs to correct it early [Nelson *et al.*, 1999].

Finalization of requirements and conceptual model development often occur iteratively; each process stimulates the other. "Finalization of requirements" means verification and validation of requirements so that they are complete, consistent, and clear. This should not be confused with "requirements creep," which is the expansion of requirements for additional functionality that normally occurs over the life of a simulation project (in software projects, a requirements creep of 20% a year of project duration is common and is planned for by wise program managers). Many believe that requirements creep is due in part to increased understanding about the problem, a result of information growth [Nelson *et al.*, 1999]. "Finalization of requirements" has to occur iteratively as "requirements creep" add additional requirements. Since the simulation conceptual model is responsive to simulation requirements, the conceptual model must also evolve with changes in simulation requirements. This is one reason that effective configuration management is needed for both requirements and the simulation conceptual model, at least to the extent of tracking unique identification for the evolving versions of requirements and conceptual models with clear indications of which is related to the other.

3.2 Steps in Conceptual Model Development

There are four basic steps in simulation conceptual model development: 1) collect authoritative information about the simulation context, 2) identify entities and processes to be represented in the simulation (application domain decomposition), 3) develop simulation elements (representational abstraction), and 4) define interactions and relations among simulation elements to ensure that a) constraints and boundary conditions imposed by the simulation context are accommodated, and b) simulation space issues (such as control capabilities that allow the simulation to be stopped, backed up in time, restarted, etc. or that identify data to be collected during the simulation) are addressed appropriately. These steps are iterated until the conceptual model is completed. Unfortunately, widely accepted approaches do not exist for how to perform these steps. The first three of these steps are discussed in subsequent sections of this paper.

3.2.1 Authoritative Information about the Simulation Context

The first step in conceptual model development is to collect authoritative information about the intended application domain that will comprise the simulation context. Development of the simulation concept and collection of authoritative information for the simulation context is likely to occur iteratively as the entities and processes to be represented in the simulation are defined. Authoritative descriptions of military activities such as contained in Conceptual Models of the Mission Space (CMMSs) in the Defense Modeling and Simulation Office (DMSO)'s repository [Sheehan *et al.*, 1998] can be used in the simulation context when appropriate for a simulation's intended application, as can the laws of physics and similar principles.

It is unlikely that the formal, documented simulation context, should one exist, will address everything needed to fully describe the domain of a simulation. CMMS endeavors emphasize a disciplined procedure by which the simulation developer is systematically informed about the real world and a set of information standards that simulation SMEs employ to communicate with and obtain feedback from military operations SMEs. Common semantics and syntax, a common format database management system (DBMS), and data interchange formats (DIF) are keys to removing potential ambiguity between the ideas of the warfighting SMEs and the simulation development SMEs. Significant progress has been made in developing a CMMS toolset to provide the keys noted above – and as noted earlier, these have been found helpful in developing models of human behavior [Dalugach and Skipper, 2000], but information beyond that likely to be obtained in the first level abstraction (i.e., the CMMS itself) may be required for a simulation conceptual model. SMEs may be "called upon to fill in details needed by [simulation] developers" that are "not provided in doctrinal and/or authoritative sources" [Johnson, 1998].

Unfortunately, caution is also required. It has also long been known that inserting the knowledge engineer (or other agent to transform information) in the role of an intermediary between the expert and the knowledge-based systems (or simulation developer) may create as many problems as it solves [Gaines, 1987]. Many approaches have been developed to address this. For example, Sharp [1998] developed a process to ensure that the expert and the knowledge engineer or program developer have the same understanding of the information being acquired and transferred to a knowledge-based system or to some other form of expression.

The more complete and clearly stated a simulation context is, the easier it will be to understand how one simulation entity (or simulation in a federation) may differ from another in its assumptions about the domain which is addressed. This becomes very important when questions of compatibility among simulations (federates) considered for a distributed simulation implementation (federation) are addressed as well as in assessment of coherence and compatibility of simulation parts.

3.2.2 Simulation Decomposition

The second step in conceptual model development is to identify entities and processes that must be represented for the simulation to accomplish its objectives. This enumeration process is fundamental in conceptual model development. Here basic decisions are made about the level of detail and aggregation that is appropriate to support simulation requirements. These decisions determine whether a system (such as a radar) will be represented as a single entity, as a composite of subsystem entities (such as an antenna or receiver), or as a composite of composites of ever smaller entities (to whatever level of detail is needed for the purpose of the simulation). Decisions are made here about the level of representation of human decisions and actions. For example, in the movement of a platform (tank, aircraft, ship, etc.), are the decisions and responses of all the people involved (the crew) represented implicitly as a single aspect of the movement control process or does each individual involved get represented explicitly (as in a tank simulator with a position for every member of the tank crew)?

Booch, Rumbaugh, and Jacobson [Booch *et al.*, 1999] identify four basic principles of modeling: 1) The choice of what models to create has a profound influence on how a problem is attacked and how a solution is shaped, 2) Every model may be expressed at different levels of precision, 3) The best models are connected to reality, and 4) No single model is sufficient. These principles suggest that modeling is essentially an art that has not yet matured to a scientific method. This is especially true for simulation conceptual model development. However, this does not prevent application of rational processes to conceptual model development.

Conceptual model decomposition determines simulation elements of the conceptual model. This determines the scope of representation in the simulation and the discernible levels of the simulation. The following six-item rationale can guide determination of what simulation elements should be in the conceptual model, forming a checklist for conceptual model decomposition. Using this rationale will help a conceptual model to be complete and coherent.

1) *There should be a specific simulation element for every item (parameter, attribute, entity, process, etc.) specified for representation by simulation requirements.* This re-emphasizes the importance of appropriate requirements for the simulation. Since a primary function of the conceptual model is to facilitate transformation of requirements into specifications, there must be a total tracking of items in the requirements to the conceptual model. This establishes the minimum level of detail in simulation decomposition.

2) *There should be a specific simulation element for every item (parameter, attribute, entity, task, state, etc.) of potential assessment interest related to the purpose of the simulation.* This stresses the importance of understanding potential use of the simulation so that all measures of performance (MOPs), measures of effectiveness (MOEs), and measures of merit (MOMs) that might be associated with use of the simulation are fully understood so that the conceptual model may fully accommodate them. This has implications for simulation space aspects of the simulation (especially in regard to data collection and display capabilities) as well as for representational aspects of mission space.

3) *As far as possible, there should be "real world" counterparts (objects, parameters for which data exist or could exist, etc.) for every simulation element.* The potential impact of data and metadata structures on simulation elements and on the overall simulation conceptual model should not be underestimated. Credibility for simulations with realistic representations is always enhanced when there is easy correspondence between its elements and the real world so that direct comparisons can be made between the real world and the simulation more readily.

4) *Wherever possible, the simulation elements should correspond to standard and widely accepted decomposition paradigms to facilitate acceptance of the conceptual model and effective interaction (including reuse of algorithms and other simulation components) with other simulation endeavors.* In most disciplines, there are standard paradigms for how an entity or process is described, measured, and evaluated. The clarity and credibility of a simulation conceptual model are enhanced when such standard paradigms are employed.

5) *Simulation elements required for computational considerations* (such as an approximation used as a surrogate for a more desirable parameter which is not computationally viable) *that fail to meet any of the previously stated rationale should be used only when absolutely essential.* Many computationally intensive problems require use of approximations or less than the most rigorous mathematical expressions. Such should be used only when necessary, and a clear statement of the reason for use should be included as part of the conceptual model.

6) *There should not be extraneous simulation elements.* Elements not directly related to specific items in the simulation requirements, not implied directly by potential assessment issues, and without a specific counterpart in the real world or in standard decomposition paradigms should not be included in the simulation conceptual model. Every extraneous simulation element is an unnecessary source of potential simulation problems. Eliminating extraneous items from a simulation conceptual model removes unnecessary potential sources of errors, and follows the principle of parsimony encapsulated in Ockham's Razor.

General guidance about construction of an HLA federation is available [accessible from the DMSO website: http://www.dmso.mil]; but rationale for how to decompose a HLA federation (or other distributed simulation) into federates (component simulations) is embryonic. Pollack and Baker [1999] developed HLA-specific software metrics for use in determining the appropriate level of decomposition in a specific HLA application. The metrics provided a quantitative

measure for achieving a balanced federation in the effort to optimize the sometimes competing goals of utility and efficiency. As others perform similar assessments, it is likely that a set of metrics having general utility will emerge and become accepted by the community. Until these are developed, it will be necessary to consider the availability of compatible simulation components as well as computational and bandwidth factors in determining how to best decompose the distributed simulation (HLA federation) for the intended application.

An important aspect of C4ISR simulation is decomposition of command processes and human elements. At what level of granularity will command processes and human activities be represented? For example, will the command processes for firing a weapon reflected in the observe, orient, decide, act (OODA) loop be represented by a single aggregated algorithm that contains a probability of firing and associated time delay – or will the command process for firing that weapon be represented explicitly addressing every sensor and information source, every operator and staff member, every communication channel and information flow path, all timing factors, etc.? How such questions are answered will impact how the simulation can be used and its compatibility (or incompatibility) with other simulations and systems. Application of the six factors mentioned above will help the simulation developer make a conceptual model that will fully satisfy simulation requirements in this regard.

3.2.3 Representational Abstraction

The third step in conceptual model development is development of simulation elements. Identification of simulation elements in conceptual model decomposition determines the scope of subject representation and the discernible levels possible in the mission space. How the characteristics of the simulation elements are abstracted determines the accuracy and precision of the representation. Because no single model is sufficient, the unified modeling language (UML) that Booch *et al.* [1999] developed has nine kinds of diagrams (class, object, use case, sequence, collaboration, state chart, activity, component, and deployment) to fully express the five most useful views (use case, design, process, implementation, and deployment) that comprise the architecture of a software-intensive system. Similarly, representational abstraction for simulation elements is likely to need a multifaceted descriptive approach.

Limitations of a particular single model are patently clear in human behavior representation, a key aspect of many C4ISR simulations. Even in a realm as narrowly defined as that of "user models" (a person's interaction with a computer system – reflects user intent, information available to the user, and response opportunities), each of the user models is probably useful in applications for which it was designed; but a particular user model may be useless for a somewhat different application since "no model represents everything" [Banks and Strytz, 2000].

Simulation fidelity is a complex function of the scope and discernible levels of the simulation as well as accuracy, precision, and other parameter quality characteristics. During recent years, substantial attention has been paid to describing simulation fidelity, with progress being made toward standardizing connotations for fidelity-related terms and awareness of issues associated with simulation fidelity [Gross, 1999]. Widely-accepted principles for determining required levels of fidelity and abstraction approaches that will produce them have not yet evolved.

Knowledge engineering provides abstraction principles that can be helpful in developing a simulation conceptual model. Theoretical approaches to knowledge engineering typically break it into three phases: knowledge acquisition, knowledge elicitation, and knowledge representation. Such theoretical approaches usually identify three knowledge structures: declarative knowledge (why things work the way they do), procedural knowledge (how to perform a task), and strategic knowledge (the basis for problem solving). Typically different acquisition, elicitation, and representation techniques are used for each kind of knowledge. Unfortunately, although there are substantial efforts to establish common approaches [Schreiber *et al.*, 2000] these theoretical approaches do not yet allow abstraction to be performed as a scientific method; thus, it remains an art. Even a casual review of articles in *Journal of Data and Knowledge Engineering* [http://www.elsevier.com/inca/publications/store/5/0/6/0/8/] and similar publications reveals that contemporary researchers in this arena often have to develop a "new" descriptive language (or dialect of a language) or formalism for the problem at hand because current techniques do not yet have broad, general application capabilities.

Banks and Strytz [2000] make this same point specifically for human behavior representation: "the usability of modeling techniques is very low; requiring experienced modelers to set-up and use the models and model programming experts to develop any new modeling situations and explain the results." The pervasiveness of this problem is not always appreciated. It extends to the most fundamental simulations in computational science, which are mainly focused on numerical solution of partial differential equations describing fundamental physical phenomena – such as the computer codes used in Computational Fluid Dynamics (CFD). Porter [1996] noted that "the utility of CFD" in a particular project for the US Air Force "will be as dependent on the skill and experience of the analysts as on the codes themselves." This is basically because no model is complete and perfect in its representation. It's proper use (so that it is not relied upon outside its realm of appropriate application) is a primary responsibility of the analyst in this case.

Quality conceptual models help to identify potential inappropriate representation. This is especially important when simulation components are mixed together. For example, planning to support a Joint Force Commander is uses a variety of tools available through the Global Command and Control System (GCCS) and its Service variations. These tools combine data and simulations from a variety of sources. Examination of conceptual models in these tools may be required to determine limits on their appropriate application, especially for new and delicate situations in which past experience with the tools for other situations may not be a totally reliable guide. Generally confidence in tools such as these is gained by using them, by training related to them (usually in programs for those headed for staff positions in joint commmands), or by playing games related to them.[4]

[4] The Joint Chiefs of Staff (JCS) Joint Force Employment (JFE) CD-ROM (2000 Version) uses the latest video game technology to enhance the user's awareness of joint doctrine employment concepts with scenarios set within the framework of the crisis action planning (CAP) system – but does not faithfully replicate all tactical aspects of the battlespace. While the simulation plays at both the tactical and operational levels of war, descriptions and functioning of weapons systems and formations are notional, with focus on key joint doctrine employment concepts.

Failure to fully account for uncertainties and errors that exist in data and information used as the basis for models, algorithms, entity characteristics and behaviors, processes, and other aspects of a simulation is a common problem in simulation verification and validation [Roache, 1998]. This issue, like fidelity, is an important consideration in representational abstraction. Likewise, sensitivity of the conceptual model to implementation characteristics is often overlooked. Very large, even drastic consequences can result from very small changes when a simulation manifests mathematical instability or chaotic behavior. Dewar *et al.* [1995] cite a war game simulation example of a well-respective land combat model in which results varied radically simply as a consequence of changing the computer hardware (from a VAX computer to a CRAY-2 computer) and software operating system so that different computational round-off processes were used. Without careful analysis of a simulation's algorithms to ensure that it is not vulnerable to such problems, it is hard to generate high levels of confidence in simulation results. Similar concerns exist for analyses which employ multiple simulations with different levels of fidelity [Pace and Shea, 1992] and simulations that employ multiple levels of resolution. Many simulations, such as the Joint Conflict and Tactical Simulation (JCATS, a multi-sided interactive entity level conflict simulation to be utilized by military and site security organizations as an exercise driver and a tool for training, analysis, and mission planning) and the Defense Information Systems Agency (DISA)'s C4ISR model and Joint Theater Level Simulation (JTLS), have to face these issues and address concerns common to all multi-resolution simulations [Ball *et al.*, 2000].

As a simulation conceptual model is developed, it should be evaluated by the completeness, consistency, coherence, and correctness criteria. It is important to record model assessments and note why the model is changed in response to the evaluation, and how criteria for a quality conceptual model are met more fully. Otherwise, rationale for some changes (and their benefits) may be lost as time passes, and lessons learned from the conceptual model development will not be so readily available for use in subsequent developments.

4. Conceptual Model Documentation

A major problem in most simulation developments in the past has been failure to produce distinct documentation of the simulation conceptual model. It is unfortunate that many simulation contracts fail to require distinct documentation of the simulation conceptual model. Lack of adequate documentation of the simulation conceptual model makes it much more difficult and costly to validate simulations and reduces their potential reuse. This true for parts of the simulation as well as for the simulation as a whole. It has been recommended that simulation conceptual model documentation employ the scientific paper approach, even if also employing the design accommodation approach by using an implementation-oriented descriptive format such as UML [Pace, 1999]. Nine items (listed below) are suggested for the description of a portion of the conceptual model (such as a simulation element) or of the entire conceptual model in the scientific paper approach to documenting a simulation conceptual model:

1) Conceptual model portion identification;
2) Principal simulation developer point(s) of contact (POCs) for the conceptual model (or part of it);

3) Requirements and purpose;
4) Overview;
5) General assumptions;
6) Identification of possible states, tasks, actions, and behaviors, relationships and interactions, events, and parameters and factors for entities and processes being described;
7) Identification of algorithms;
8) Simulation development plans; and
9) Summary and synopsis.

Discussion of these items may be found at the DMSO website [http://www.dmso.mil]for *the VV&A Recommended Practices Guide* and elsewhere [Pace, 1999]. It should be noted that this list of items is functionally equivalent to the 10 items in the generic content guidelines from IEEE/EIA Industry Implementation of International Standard ISO/IEC: ISO/IEC12207 *Standard for Information Technology Software Life Cycle Processes* for describing a planned or actual function, design, performance or process: date of issue and status, scope, issuing organization, references, context, notation for description, body, summary, glossary, and change history [Moore, 1999]. As more simulation developers use this standard approach to documentation for the simulation conceptual model (and its parts), reliable comparison of simulations (and simulation parts) will become easier and more affordable.

The importance of good conceptual model documentation for C4ISR simulations can not be over stressed. As contemporary military operations are dominated by contingency operations, operations other than war (OOTW), etc., C4ISR simulations used to analyze them or to their support planning and execution may include elements that are derived from very varied perspectives. For example, the Baseline Collective Assessment Report for Adaptive Joint Command and Control (C2) identified the Defense Information Systems Agency (DISA) C4ISR Federation Model, the Federal Emergency Management Agency (FEMA) Consequences Assessment Tool Set (CATS), JWARS approaches to representation and analysis of C4ISR, and other simulations/tools as potentially useful for Adaptive Joint C2. It is very likely that these contain inconsistent and perhaps incompatible algorithms and data since they were developed by different agencies with varied perspectives. Quality conceptual model documentation will help those modifying, combining, or using these simulations and tools not to blunder.

5. Conclusion

This paper has focuses on basic methodology, not just on C4ISR. However, it's pertinence and importance for C4ISR should be evident. Growing reliance upon simulations for analysis, planning, assessment and evaluation, and even support in executing operations makes it imperative that appropriate credibility for these simulations must be apparent and limitations on their applicability must be well understood. Quality, explicit conceptual models are an important aspect of obtaining this position. This paper has described what conceptual models are, explained how to decompose a simulation, discusses issues of representational abstraction, and provided suggestions about documentation. All elements of the C4ISR community are

encouraged to insist upon development and documentation of quality conceptual model for any simulation for which they have responsibility or influence.

6. References

[Balci, 1990] Osman Balci, "Guidelines for Successful Simulation Studies," *Proceedings of the 1990 Winter Simulation Conference*, New orleans, LA, 1990.

[Ball *et al.*, 2000] George L. Ball, Bernard P. Zeigler, Richard Schlichting, Michael Marefat, and D. Phillip Guertin, *Problems of Multi-resolution Integration in Dynamic Simulation* [available at: http://www.sbg.ac.at/geo/idrisi/gis_environmental_modeling/sf_papers/ball_george/santafe.html].

[Banks and Strytz, 2000] Sheila B. Banks and martin R. Strytz, "Advancing the State of Human Behavior Representation for Modeling and Simulation: Technologies and Techniques," Paper 023 in *Proceedings of the 9th Annual Conference on Computer Generated Forces (CGF) and Behavior Representation (BR)*, May 16-18, 2000, Orlando, FL. Accessible at the SISO website: http://www.sisostds.org/ **.**

[Booch *et al*., 1999] Grady Booch, James Rumbaugh, and Ivar Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, 1999.

[Dalugach and Skipper, 2000] Harry S. Delugach and David J, Skipper, "Knowledge Techniques for Advanced Conceptual Modeling," Paper 004 in *Proceedings of the 9th Annual Conference on Computer Generated Forces (CGF) and Behavior Representation (BR)*, May 16-18, 2000, Orlando, FL. Accessible at the SISO website: http://www.sisostds.org/ .

[Dewar et al, 1995] J. Dewar, Bankes, S., Hodges, J., Lucas, T., Saunders-Newton, D. And Vye, P., *Credible Uses of the Direstributed Interactive Simulation (DIS) System*. RAND Corporation Report Mr-607-A.

[Dubois and Might, 2000] Richard D. Dubois, Jr., and Robert J. Might, "Conceptual Modeling of Human Behavior (CMHB)," Paper 087 in *Proceedings of the 9th Annual Conference on Computer Generated Forces (CGF) and Behavior Representation (BR)*, May 16-18, 2000, Orlando, FL. Accessible at the SISO website: http://www.sisostds.org/ .

[IEEE, 1998] IEEE Standard 1320.2-1998, *Conceptual Modeling Language - Syntax and Semantics for IDEF1X97 (IDEFobject)*.

[Gaines, 1987] B. R. Gaines, "An Overview of Knowledge Acquisition and Transfer," *International Journal of Man-Machine Studies*, Vol. 26, No. 4 (April 1987), pp. 453-472

[Gross, 1999] Gross, David C (redactor), "Report from the Fidelity Implementation Study Group", 99S-SIW-167, *1999 Fall Simulation Interoperability Workshop Papers*, March 1999.

[Johnson, 1998] Thomas H. Johnson, "Mission Space Model Development, Reuse and the Conceptual Models of the Mission Space Toolset" *98 Spring Simulation Interoperability Workshop Papers*, March 1998, Vol. 2, pp. 893-900.

[Lindland *et al.*, 1994] O. I. Lindland, G. Sindre, and A. Solvberg, "Understanding Quality in Conceptual Modeling," *IEEE Software*, Vol. 11, No. 2, (March 1994), pp. 42-49.

[Moore, 1999] James W. Moore, "An Integrated Collection of Software Engineering Standards," *IEEE Software*, November/December 1999, pp. 51-57.

[Nelson *et al.*, 1999] Mike Nelson, James Clark, and Martha Ann Spurlock, "Curing the Software Requirements and Cost Estimating Blues: The Fix is Easier Than You Might Think," *Program Manager*, Vol. XXVIII, No. 6, Defense Systems Management College (DSMC) 153, November-December 1999, pp. 54-60.

[Oberkampf *et al.*, 2000] William L. Oberkampf, Sharon M. DeLand, Brian M. Rutherford, Kathleen V. Diegert, and Kenneth F. Alvin, SANDIA Report Sand2000-0824, *Estimation of Total Uncertainty in Modeling and Simulation*, April 2000.

[Pace, 1999] Dale K. Pace, "Development and Documentation of a Simulation Conceptual Model," *99 Fall Simulation Interoperability Workshop Papers*, September 1999.

[Pace and Shey, 1992] D. K. Pace and Shea, D.P., "Validation of Analysis Which Employs Multiple Computer Simulations" *Proceedings of the 1992 Summer Computer Simulation Conference, Reno, NV, July 27-30*, Society for Computer Simulation, pp. 144-149.

[Pew and Mavor, 1998] Richard W. Pew and Anne S. Mavor (eds.), *Modeling Human and Organizational Behavior: Application to Military Simulations*, National Academy Press, Washington, DC, 1998.

[Pollack and Baker, 1999] Anne F. Pollack and John P. Baker, "Federation Decomposition: HLA Metrics," *99 Fall Simulation Interoperability Workshop Papers*, September 1999.

[Porter, 1996] J. L. Porter, "A Summary/Overview of Selected Computational Fluid Dynamics (CFD) Code Validation/Calibration Activities," AIAA Paper 95-2053, *27th AIAA Fluid Dynamics Conference*, New Orleans, LA, June 17-20, 1996.

[Roache, 1998] Patrick J. Roache, *Verification and Validation in Computational Science and Engineering*, Albuquerque, NM: Hermosa Publishers, 1998.

[Sargent, 1985] Robert G. Sargent, "An Exposition on Verification and Validation of Simulation Models," *1985 Winter Simulation Conference*, Sacremento, CA, 1985.

[Schlesinger, 1979] Stuart Schlesinger, "Terminology for Model Credibility," *Simulation*, Vol. 32, No. 3, 1979, pp. 103-104.

[Schreiber *et al.*, 2000] Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Robert de Hoog, Nigel Shadbolt, Walter Van de Velde, and Bob Wielinga, *Knowledge Engineering and Management: The CommonKADS Methodology*, Cambridge, MA: The MIT Press, 2000.

[Sharp, 1998] John K. Sharp, "Validating an Object-Oriented Model," *Journal of Conceptual Modeling*, Issue No. 6, December 1998 [http://www.inconcept.com/JCM].

[Sheehan *et al.*, 1998] Jack Sheehan, Terry Prosser, Harry Conley, George Stone, Kevin Yentz, and Janet Morrow, "Conceptual Models of the Mission Space (CMMS): Basic Concepts, Advanced Techniques, and Pragmatic Examples," *98 Spring Simulation Interoperability Workshop Papers*, March 1998, Vol. 2, pp. 744-751.

[Teeuw and van den Berg, 1997] W. B. Teeuw and H. van den Berg, "On the Quality of Conceptual Models," *16th International Conference on Conceptual Modeling, 3-6 November 1997 (ER'97) in Los Angeles, CA* [http://osm7.cs.byu.edu/ER97/workshop4/tvdb.html].