Intelligent Integration of Medical Models: A Rigorous Approach to Resolving the Dichotomy in Medical Models

John James
Intermetrics, Inc.
7918 Jones Branch Dr. Suite 710
McLean, VA 22102 USA
james@potomac.wash.inmet.com

Nicholas DeClaris
Departments of Pathology, Epidemiology and Preventive Medicine
of the School of Medicine and the
Electrical Engineering Department of the School of Engineering
University of Maryland, College Park, MD 20742 USA
declaris@medinfo.eng.umd.edu

Wolf Kohn
Intermetrics, Inc.
1750 112th Ave NE, Suite D-151
Bellevue, WA 98004 USA
wfk@minnie.bell.inmet.com

Anil Nerode
Director, Mathematical Sciences Institute
White Hall, Cornell University
Ithaca, NY 14853-7901 USA
nerode@mssun7.msi.cornell.edu

Abstract

Today, the dominant modeling concepts for medical information systems are based on a) "homeostasis" and b) "physical dynamics" – such as those encountered in pharmacology, quantitative pathophysiology, and more recently in immunology. We introduce the novel idea of the *hybrid system state* which is used to provide a basis for modeling of medical systems and the *carrier manifold* which is used for analysis, design, and integration of medical models. We discuss the intelligent integration of what is now a dichotomy in medical models. The dichotomy we refer to occurs between the phenomenological models of high-level decisions and the evolution models of low-level dynamics. By intelligent we mean an explicit declaration of the rules for composition of heterogeneous models. By rigorous we mean that, to the extent that the phenomenological models are correct, the evolution models are correct, and the composition rules are correct, our approach for integration of medical models will extract a correct, integrated program.

1 Introduction

A long-standing issue in building intelligent systems has been the integration of symbolic and numeric processing. The steady march toward ever more capable computer and communication systems has resulted in being able to build ever more complex software systems. In the past, an *ad hoc* approach was acceptable to validate that system requirements met user needs and to verify that integration of heterogeneous models produced software which met those requirements. However, given the increasing complexity of large-scale systems, an *ad hoc* approach, ever when enganced with current concepts for software architectures and reusable components, is no longer acceptable from a cost perspective or from a safety and reliability perspective.

1.1 Fundamental constraint of the current approach to integration of medical models

Integration of heterogeneous system models is currently driven by the need to *exhaustively experiment* with failure modes of computer-controlled information systems. The engineering approach for construction of these systems is to build a periodic process which is used to control the switching between modes of control. These modes of control are experimentally determined to be "good enough" around "equilibrium conditions." Precision control in each of the modes is executed by algorithms constructed from differential equation models of system dynamics. The switching between control modes is determined by high-level logic which also coordinates interface to the user, unit test, and data base access functions.

Engineering experience and heuristics are used to select conditions for the switching modes and *exhaustive experimentation* is required to perform verification and validation of the composite of logical and evolution equations, such as the interface between a message monitor for handling asynchronous (logical) inputs (e.g. change operating modes) to a periodic process (evolution equations) in one mode to a different set of evolution equations in another mode. Commercially available software is available to aid in conducting the experiments, however, the current technology *does not support* attainment of exhaustive testing of alternatives.

The current technical approach for integration of dissimilar medical information models is the "experiment, observe, analyze, and correct" sequence used for test and verification of other information systems. Once a level of integration is achieved and an incremental addition of a new system is desired (or a new feature is added to an existing subsystem), the effect of the change on the existing system must be determined through *repetition of the experimentation process*.

Medical information systems share scale-up and integration challenges with other complex information systems. But, unlike those systems, medical systems are faced with even more stringent safety and security requirements, and they must meet a need to drive down the costs of health care delivery while maintaining high quality service. Given the trend to integrate ever larger medical information systems, there is a pressing need to improve this technology. In the remainder of this section we develop a more precise statement of the integration problem.

1.2 Statement of the medical system integration problem

Consider the challenge of intelligent integration of medical information systems posed by a medical emergency for even a single patient. Even at this level, the setting for medical decision-making is very complex (see Table 1). Clinicians consider data from patient interviews, previous interventions, physical examination, laboratory tests, and various models to perform assessment diagnosis, evaluate patient prognosis, and prepare therapeutic plans. Information system support for clinical activities requires acquisition and efficient processing of data from many sources and depends heavily on the use of models.

The qualitative reasoning of homeostasis accommodates clinical intuition and empirical associations, which is of great importance among humans, but very difficult to systematize or automate for representation as computable models. Modeling of homeostasis information has often been captured as discrete changes or decisions (events) described by rules. Selection, verification and validation of sets of rules and their use in medical decision making has been a long-standing issue [1, 5, 10].

On the other hand, the quantitative reasoning of

Class	Qualitative Characteristics	Math. Basis	Unified Model-Based Software		Areas of Applicability
			Old	New	
Dynamic Systems Models	• Quantitative reasoning • Continuous change • Clinical analysis of dynamical change (e.g. pharmacology, immunology, and quantitative pathophysiology) • Analytical associations	Differential Operators	Procedural Digital Machines	Distributed/ unified medical model	• Prediction of: disease growth, immune system reactions, pharmacological variations, chemical reactions, . . . • Continuous monitoring • Analysis of radiological morphological variables
Phenom-enological Models	• Qualitative reasoning • Homeostasis • Clinical intuition • Empirical associations	Algebraic Topologies (T-Zero)	AI/predicate calculus		• Diagnosis • Patient management • Therapeutic interventions • Epidemiology • Safety • Starting and stopping • Data exchange protocols

Table 1: Dichotomy of Models Used in Medical Information Systems Development

analytical models of physical dynamics accommodates well-defined conceptual representations of continuous change in spatial and temporal attributes of: patient variables (e.g. response of the immune system to an infectious organism, amount of oxygen in the blood, or concentration of a particular chemical), medical equipment variables (e.g. flow rate of oxygen in a respirator, rate of administration of a particular medicine, data transfer rate for a temporary communications link), and support structure variables (e.g. light intensity, room temperature, background noise, cost of treatment, profit, loss). Also analytical representations of information can be captured as differential equations, stochastic processes and numerical input-output tables which can be used for efficient processing of computable models. However presentation of results from analytical models creates substantial practical difficulties compared to the more intuitive qualitative reasoning of homeostasis.

A major difficulty of current systems is that there is a dichotomy in mathematical methodology when dealing with these two representation classes of medical models [1]. The coupling of qualitative abstractions with quantitative dynamics has been previously recognized as a crucial challenge for construction of intelligent systems. This challenge has been variously described as the "pixel-to-predicate" or "signal-to-symbol" transformation problem [15, 16].

Both classes of models (logical and evolution) are subject to incremental review and change over time. Treatment protocols are updated, new chemical process models are discovered, new forms are required for patient monitoring and reporting, analytical accuracy and detail of radiographic results change with improvement in equipment. Various research thrusts revolve around the determination of how to represent and incrementally adapt the degree of confidence we have in the accuracy of the models being examined [4].

1.3 Overcoming a fundamental constraint in computable models

A key feature of our approach for integration of medical models is the application of a new technology for coupling of numeric and symbolic computation which is based on *unification* of trusted models instead of *experimentation* with trusted models [2]. This new approach eliminates the dichotomy in models by representing the state of the system as a point in a manifold (explained below) which admits both logical and evolution models. Table 1 notionally summarizes the idea that hybrid system models developed using the multiple-agent declarative control architecture approach can achieve the desired integration. We provide an overview of the hybrid systems model in the next section. Space limitations preclude a more complete mathematical justification of our notional assertions (see [2]).

Recent medical informatics work by Dr. Nicholas DeClaris has addressed unifying dynamical system models and phenomenological algorithms for medical decision making. He has introduced a novel conceptual scheme based on "Disease-Therapeutic Dynamics" which involves computer-aided reasining from the outset. There are five essential medical elements in this conceptualization: pathophysiological dynamics, immunological and allergic dynamics, pharmacokinetics, observations and measurements, and iatrogenics. The medical elements are conceptualized by means of analytical (differential equations, random processes, etc.) and algorithmic techniques (data structures, etc.) as well as phenomenological associations involving skill and specialized knowledge based on training and experience.

2 Hybrid System State

For purposes of our exposition, there are three cases which need to be considered in the characterization of unified-medical-information represented in the computer. These cases collectively contain the kinds of information that describe the *state of the system.* The cases are:

- information derived from monitoring continuous variables,
- information derived from monitoring variables which normally evolve continuously but which may exhibit logical changes (jumps), and

- information which is derived from logical variables.

To represent the state of the system in a computable model, the values of continuous variables must be approximated, as must the values of variables which are normally continuous but may occasionally exhibit jumps. This is achieved by A-D and D-A transducers. Exact representation of logical variables can be achieved using transducers. The data structure of the hybrid system state in the computational environment is a composition of logical and evolution variables. The state of the system in the physical environment is approximated by the state of the system in the computational environment (the hybrid system state). The information system of the user includes direct observation of the physical system as well as information available from the user interface of the computation environment. The hybrid system state evolves over time as the physical environment is altered by the user(s) and by the actuators of the system (see Figure 1).

2.1 Hybrid Systems Language Requirements

A formal language that encodes hybrid systems must be able to express: evolution models, logic models, interfaces of evolution and logic models, behavior requirements, and real time constraints.

Evolution Models: An evolution model is a composite of 3 items: an Evolution Domain, a Set of Transformations on that domain and Boundary Conditions.

- *Evolution Domain:* The domain of a Hybrid system is always a Manifold, called the Carrier Manifold or domains derived from it such as vector bundles, jet spaces, congrence manifolds, etc. (Vector spaces are the simplest examples of manifolds.) A manifold is a set of points S together with a countable set of maps called Coordinate Maps. The domain of each coordinate map is an open subset of S, the range is an open subset in a Euclidean space. The coordinate maps are specified by a set of Generic and Particular properties. The generic properties are common to all manifolds. The particular properties describe characteristics of each application. Given these properties and the set S the manifold is completely determined. Thus, a language expressing hybrid systems must provide primitives for encoding sets and coordinate transformation properties. Moreover it must provide means to express domain items that depend on the manifold (ex: tangent spaces, vector bundles, submanifolds, immersions, jet spaces, products of manifolds, etc.).

- *Transformation Set:* A transformation is an item that transforms a point or points in the manifold to another point or points. Examples include differential (ordinary or partial) equations, difference equations, a stochastic process, integral equations, algebraic transformations, a variational expression, an algorithm, a set of rules and conjunctions or disjunctions of the above. Without loss of generality, a set of transformations generate Trajectories in the carrier manifold or in its associated domains. Thus, the language of hybrid systems must be able to express transformations and the outcome of transformations (namely trajectories). Note that in general trajectories are infinite objects so the language must exhibit the capabilities to express infinite objects with finitary representations (example: flows).

- *Boundary Conditions:* These are standard initial or constraint items for the transformations defined above. They define subdomains within the manifold or its associated domain items.

Logic Models: A logic model is an object composed of three items: a Logic Domain, an Axiom Base and a Proof or Discrimination System.

- *Logic Domain:* The domain of the logic model are certain classes of trajectories of evolution models on the carrying manifold called *relaxed curves*. The state of a hybrid system and its behavior are relaxed curves. These curves must

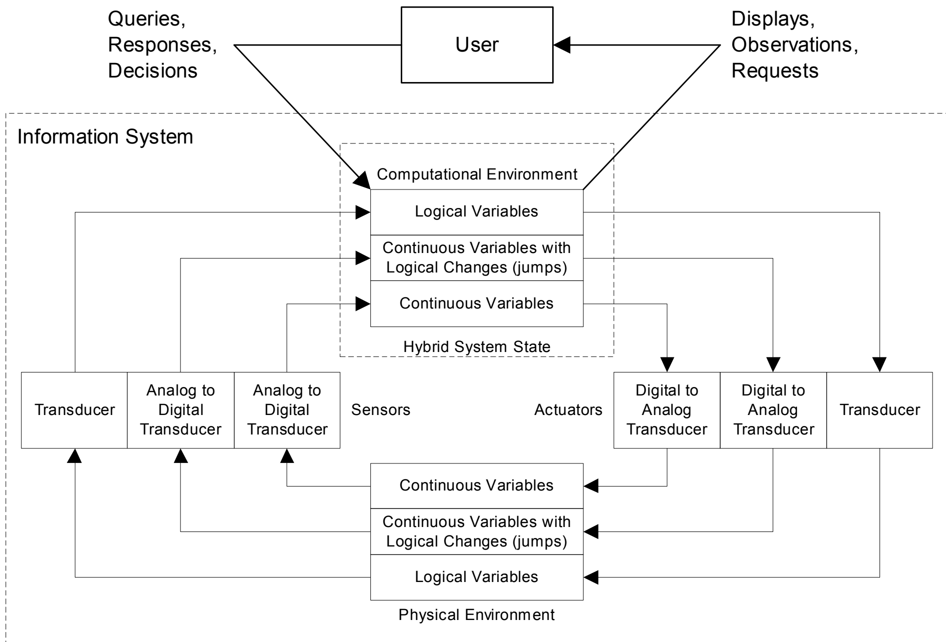

Figure 1: Hybrid System State Representation

possess certain porperties. (If the hybrid system is a model of a control problem these properties are requirements.) A language for hybrid systems must have primitives for encoding, manipulating, connecting and transforming Relaxed Curves. In the Kohn-Nerode theory a central result states that relaxed curves can be expressed by one or more strings of the form $q(X,\ldots,Y) rel p(X,\ldots,Y)$ with p and q algebraic expressions in a certain algebra and *rel* one of three relational connectives: equal ($=$), not equal ($\neq$), and partial order ($\leq$).

- *Axiom Base:* The Axiom Base is an encoding of the properties listed above, together with the axioms of algebraic logic with equality, and the axioms of properties of the logic operators (and, or, $\leq$, not) and the properties of certain class of operators, called Inference operators that transform elements of the domain into other elements of the domain. For example unify, which implements Robinson unification, transforms a set of relaxed curves into a relaxed curve whose properties are common to the set. The Axiom base also encodes properties of the hybrid system as a whole (global properties). The central ones are Stability, Controllability, Goal Reachability, Observability, and Self-Awareness.

- *Proof System:* The proof System is a formal representation of the "Intelligence" of the hybrid model. It uses the axiom base to generate Relaxed curves that correspond to the behavior of the system. It encodes a *strategy* (Monotone Dynamic Programming in the Kohn-Nerode Theory) for the effective construction of relaxed curves that characterize the behavior of the hybrid system if they exist or Modification instructions (to the model) if they do not exist.

Interface: The interface between Evolution Models and Logic models characterizes the central property for well-posedness of hybrid models: *continuity* of behavior trajectories (relaxed curves) in the carrier manifold. By continuity we mean that the Transformation that maps the requirements encoded in the axiom base to the space of behavior trajectories in the manifold or its derivatives is continuous. When one talks about continuity, one must specify a topology; in hybrid systems the topology is the topology of the carrier manifold which is fully determined once the coordinate maps are defined. In our approach to hybrid systems the axiom base must define the coordinate maps at each instant. A formal language for hybrid systems must contain primitives and a composition strategy for expressing desired continuity definition. For example, in the commercial airplane problem, desired continuity is that the behavior trajectories be such that "a coffee cup 3/4 full in any location of the airplane should not spill (relative topology)."

Behavior Requirements: Language requirements for a scenario-based approach for determining system requirements and expressing these results in a formal language have been studied. Descriptions about primitives for expressing scenario-based behavior requirements are summarized in [17]. Also, these primitives are described in the Hybrid Systems book recently published. However we emphasize here two primitives that the behavior requirements must specify; these are the Generalized Lyapunov stability check and the goal reachability check. Both of these can be written as equations with variables in the space of relaxed trajectories. Each of these equations have non-empty solution sets only if the corresponding properties are satisfied.

real Time Constraints: These are equational forms that characterize the computer environment. A language for hybrid systems must have primitives that express real-time frames, infinitesimal concurrency, relaxation level, "sufficiently good paradigms," any time computing, etc. These real-time constraint primitives must be expressible in terms of *continuity.*

2.2 Use of hybrid system state for integrating medical models

We have developed preliminary ideas concerning use of Multiple Agent Declarative Control Architecture (MA-DCA) with results from medical informatics. The MA-DCA architecture is based on a theory developed by Wolf Kohn and Anil Nerode over the past 3 years. This theory extends the concepts, principles and algorithms of Single Agent Declarative control theory, and merges them with the principles of concurrent computing and dynamical hybrid systems. The multiple agent architecture is a collection of these agents and an inter-agent connectivity network. The central concept for MA-DCA is that an on-line restrictive mechanical theorem prover will exist within each agent. The theorem prover of each agent consists of five elements: a Knowledge Base, a Planner, an Inferencer, an Adapter, and a Knowledge Decoder. The Knowledge Base stores the goal for the agent, system constraints, inputs and inference operations. The Planner generates the theorem which represents the goal. For some agents, this goal will govern the behavior local to that agent. For other agets, the goal will also include behaviors global to the system. The Inferencer proves the theorem. If the theorem is true, control actions, computed during inferencing, are issued to the plant. If the theorem is false, the Adapter processes the failed terms in the theorem for replacement or modification. Data from other agents is provided to the Planner for incorporation as constraints into the theorem and passes through the Knowledge Decoder for entry into the Knowledge Base.

The MA-DCA has several key capabilities:

- *Reactive*: The theorem proving function of each agent on the architecture operates according to a first principles feedback paradigm.

- *Adaptive:* The knowledge base of each agent is open and modifiable by sensory data. Theorem failure triggers tuning and corrective action.

- *Distributive with Coordination:* The theorem proving is carried out distributively over the agents. The coordination scheme is implicit without umpire.

- *Dynamic Hierarchization:* The architecture can operate simultaneously at different levels of abstraction.

- *Figure of Merit:* The behavior of the closed loop distributive system is determined by proving that there exists a command trajectory that minimizes a goal functional.

- *Real-Time:* Constraints for real-time performance are explicit and part of the knowledge base. This is important because real time constraints cannot be fully instantiated at design time.

3 Carrier Manifold

While the *hybrid system state* is used to provide a basis for building unified information models of medical systems, the *carrier manifold* is necessary to provide the mathematical basis for analysis, design, and synthesis of programs which apply unified medical models. The Kohn-Nerode approach for unification of logical and evolution models is based on introducing the idea of continuity of the hybrid state representation. The continuity argument and the constructive extraction of automata which comply with the continuity constraint is accomplished by using the mathematics of manifolds (see Figure 2). At this point it is appropriate to reemphasize our comment in the introduction that engineers and scientists have built *ad hoc* integrated information models of hybrid systems for decades. Anyone who has built an implementation of a computer-controlled dynamical system will recognize the necessity to explicitly accommodate both logical and (approximations of) continuous variables in the information model of the system being controlled. What is new in our approach is the explicit interpretation of these existing medical models as representing the hybrid system state and the application of the new concept of a carrier manifold as the mathematical tool for analysis and manipulation of the system state. Furthermore, the current technical approach for construction of computational models for sending signals to actuators is based on experimentally integrating logical and evolution models of the physical environment. The *inefficiency* in requiring experimental verification and validation that the safety, security, and functional system requirements are satisfied is a *fundamental barrier* to lowering the costs of integrating existing medical information systems. Previous efforts to improve the integration process have been hampered by the fact that there has not been a mathematical framework for simultaneously analyzing safety and security requirements while also considering temporal and spatial constraints. Our creation of the hybrid system state provides the technical foundation to achieve the *unification* of logical and evolution models. Our multiple-agent declarative control architecture provides the analytical framework to simultaneously comply with evolution and logical constraints. Our analysis of the hybrid system state as a point in a carrier manifold provides the mathematical basis for extraction of programs which simultaneously satisfy continuity constraints imposed logical and evolution behavior requirements.

More explicitly, a point in a manifold supports unification of logic and evolution models. T-zero topologies have a one-to-one correspondence with Horn clauses of logical representations. This enables us to model the Discrete-Event Dynamic System (DEDS) sampling rule models. Lie algebra infinitesimals allow us to consider all the standard evolution models of differential operators and DEDS evolution models. We embed logical models in continuous models in order to construct automata which comply with logical and continuum constraints. We assert and emphasize here that for systems which meet the conditions for creation of a hybrid system state, the revolutionary nature of our approach has two benefits: (1) creation of a unified mathematical foundation for analysis and synthesis of models which for decades have been treated separately, and (2) creation of a rigorous process for incremental expansion of trusted systems which must comply with stringent safety and performance constraints.

References

[1] DeClaris, N., "Unifying Dynamical System Models and Phenomenological Algorithms for Medical Decision Making," USSR Academy of Science Workshop on Mathematical Models in

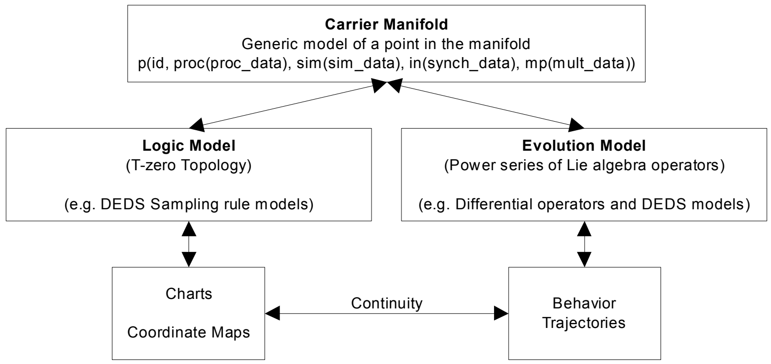

Figure 2: Continuity in the Topology of the Representation

Immunology and Medicine, Kiev USSR, September 1989. Presented by invitation and published the following year in its proceedings by IASA. A summary of this paper also appears in: DeClaris, N., "A Systems Approach to Medical Decision Aiding" (invited), Proceedings of the 1991 International Conference on Systems, Man and Cybernetics, University of Virginia, IEEE Service Center, pp. 2103–2108.

[2] Kohn, W., J. James, and A. Nerode, "Multiple Agent Reactive Control of Medical Information Processes through a Heterogeneous Network," Intermetrics White Paper, December 1993.

[3] Boehm, B. and W. Scherlis, "Megaprogramming," Defense Advanced Research Projects Agency, Software and Intelligent Systems Technology Office (DARPA-SISTO) Software Technology Conference, Los Angeles, CA, April 1992.

[4] Bonissone, P. P., "Approximate Reasoning Systems: A Personal Perspective," Plenary Presentation at the Ninth National Conference on Artificial Intelligence (AAAI '91), Los Angeles, CA, 1991.

[5] Cannata, P., "Carnot Mediator Frameworks," Intelligent Information Integration Systems (I^3S) Workshop, Reston, VA, March 1993.

[6] Chakravarthy, S., "A Federated Multi-media DBMS for Medical Research," Intelligent Information Integration Systems (I^3S) Workshop, Reston, VA, March 1993.

[7] Chu, W. W., A. F. Cardenas, and R. K. Taira, "A Knowledge-Based Multimedia Medical Distributed Database System – KMeD," Intelligent Information Integration Systems (I^3S) Workshop, Reston, VA, March 1993.

[8] DeClaris, N., R. A. Cowley, and B. F. Trump, "Systems Approach to Intensive Care Medicine," American Surgeon, 1979.

[9] DeClaris, N., "An Adaptive Medical Information System," Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics, Atlanta, GA, 1986, pp. 286–292.

[10] Clancey, W. J., "From Guidon to Neomycin and Heracles in Twenty Short Lessons: ONR Final Report 1979–1985," AI Magazine, Vol. vii, No. 3, pp. 40–60, August 1986.

[11] Boult, T., "Image Understanding Environment," ARPA DSSA VII Workshop, Key West, FL, July 1993.

[12] DeClaris, N. and McCarter, R., "Coorperative Information Systems for Epidemiological Research," in 1992 Proceedings of the 1992 IEEE International Conference on Systems, Man and Cybernetics, Emergent Innovations in Information Transfer, Processing and Decesion Making, Chicago, 1992.

[13] DeClaris, N., Havas, S., and McCarter, R., "A Systems Approach to Medical Informatics Academic Research and Training," in MEDINFO 92, Vol. 2, KC Lun (ed.), North Holland, Elsevier Science Publishers, 1992.

[14] Kohn, W. and Nerode, A., "An Autonomous Systems Control Theory: An Overview," Proceedings of IEEE CACSD '92, March 17–19, Napa, CA, pp. 200–220.

[15] Proceedings of the AAAI Workshop on Coupling of Symbolic and Numeric Processing, Seattle, WA, 1988.

[16] James, J., P. P. Bonissone, D. K. Frederick, and J. H. Taylor, "A Retrospective View of CACE-III: Considerations in Coordinating Symbolic and Numeric Computation in a Rule-Based Expert System," Proceedings of the Second Conference on Artificial Intelligence Applications, Miami Beach, FL, December 1985.

[17] Hufnagel, S., K. Harbison, and D. Goldstein, "Scenario Driven Requirements Analysis Method," Proceedings of the Joint Services Guidance and Control Workshop, December 1992.