ARMY RESEARCH LABORATORY

A Time-Discrete Vulnerability/Lethality (V/L) Process Structure

Brian G. Ruth
Phillip J. Hanes

ARL-TR-1222 November 1996

NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1996	Final, Nov 94 - Oct 95

4. TITLE AND SUBTITLE
A Time-Discrete Vulnerability/Lethality (V/L) Process Structure

5. FUNDING NUMBERS
PR: 622120AH25

6. AUTHOR(S)
Brian G. Ruth and Phillip J. Hanes

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-SL-CM
Aberdeen Proving Ground, MD 21010-5423

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1222

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

All real-world military systems active within a battlefield can be considered dynamical systems, which are constantly changing or varying in terms of their descriptive parameters, such as position, velocity, field strength, and so on. In this report, a variation on the established Vulnerability/Lethality (V/L) Taxonomy is introduced, which could be used as an architecture for the integrated survivability, lethality, and vulnerability (SLV) analysis of a dynamic military system exposed to the full spectrum of battlefield threats, where the threats can be applied stochastically as a function of time. This methodology basically involves the introduction of a time axis orthogonal to the process flow within the current V/L process structure. The overall architecture of the new structure is first presented; then, each element within the architecture is examined in greater detail. Finally, an example application using the new structure to analyze a battlefield system is described.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Vulnerability/Lethality Analysis, V/L Taxonomy, Integrated Analysis	49
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

ACKNOWLEDGMENTS

The authors would like to thank those who participated in the numerous discussions that led to this report. In particular, they would like to thank Drs. Paul H. Deitz and James N. Walbert for recognizing the need to include time within a vulnerability/lethality (V/L) process structure. The authors would also like to thank Messrs. William J. Hughes, Drew B. Farenwald, Richard L. zum Brunnen, and Lynn H. Davis for their collective insights into the dynamical nature of V/L processes. Finally, the authors would like to thank Mr. Robert W. Kunkel, Jr., for his technical review of this report.

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

INTENTIONALLY LEFT BLANK.

LIST OF FIGURES

LIST OF TABLES

INTENTIONALLY LEFT BLANK.

1. INTRODUCTION

A physical *dynamical system* is a real-world construct that is constantly changing or varying in terms of its descriptive parameters, such as position, velocity, field strength, and so on. There are three principal paradigms associated with a dynamical system: (1) a space, known as the *manifold*, in which the motion or activity of the system takes place, (2) a rule (or set of rules), known as the *vector field* (or vector fields), to determine the progression of the system in space and time, and (3) the *state* of the system, which is the set of numerical values of all the system's descriptive parameters sampled at an instant of time *t*. What distinguishes a dynamical system from a *static* system is the former's evolution as a function of time. If α is a vector whose elements describe the state of a system as a function of time, then the time derivative of the vector $d\alpha/dt = 0$ for a static system and $d\alpha/dt > 0$ for a dynamical system. The vector field for a system may also be time-dependent, so that a system may jump from a static to a dynamic condition (or vise-versa) as a function of time.

All real-world military systems active within a battlefield can be considered dynamical systems. The battlefield, which is the system's manifold, is, by its very nature, an area of constant change and activity. The vector fields, which determine the progressive states of the military system as a function of time, are themselves functions not only of system parameters but also of external battlefield-environment parameters, such as terrain, atmospheric conditions, and potential threats to the system. To address the last area mentioned in the previous statement, a process structure, or Vulnerability/Lethality (V/L) Taxonomy, has been developed for the V/L analysis of military systems exposed to battlefield threats (Deitz 1986; Deitz and Ozolins 1989; Deitz et al. 1990; Klopcic, Starks, and Walbert 1992). This V/L Taxonomy, which is really a mathematical framework for V/L analysis developed by the Ballistic Vulnerability Lethality Division (BVLD)* of the Survivability Lethality Analysis Directorate (SLAD), U.S. Army Research Laboratory (ARL), clearly defines the elements of the V/L analysis process as: (1) generation/formation of the threat event, (2) interaction between the threat and the target system, (3) component response within the system, and (4) final remaining system capability levels. The V/L Taxonomy will be addressed in further detail in the next section.

The SLAD of ARL has the responsibility of evaluating the survivability, lethality, and vulnerability (SLV) of U.S. Army systems against the full spectrum of battlefield threats, including conventional

* BVLD was formerly known as the Vulnerability Lethality Division of the U.S. Army Ballistic Research Laboratory (BRL).

ballistic, electronic warfare (EW), nuclear, chemical, biological, smoke/obscurants, and atmospheric. The three divisions within SLAD (BVLD, Electronic Warfare Division [EWD], and Chemical/Biological, Nuclear, and Environmental Effects Division [CBNED]) are tasked with applying their capabilities to perform SLV simulations, investigations, and analyses to support program managers (PMs), independent evaluators, and decision makers. In order to better implement these capabilities, the V/L Taxonomy has been applied to all battlefield threats within the spectrum of SLAD's responsibilities to generate threat-specific V/L process structures (Deitz 1986; Deitz and Ozolins 1989; Deitz et al. 1990; Klopcic, Starks, and Walbert 1992; Ruth 1994; Walbert 1994; Mar 1995; zum Brunnen 1995). Although a recent effort has applied the V/L Taxonomy to address the battlefield combined arms threat (Hughes 1995), the challenge still remains of integrating these process structures into a composite architecture with the capability to *treat time as a stochastic variable.*

The objective of this report is to present a variation on the established V/L Taxonomy that could be used as an architecture for the integrated SLV analysis of a military system exposed to the full spectrum of battlefield threats, where the threats can be applied stochastically as a function of time. This methodology basically involves the introduction of a time axis orthogonal to the process flow within the current V/L process structure. The overall architecture of the new structure is first presented; then, each element within the architecture is examined in greater detail. Finally, an example application using the new structure to analyze a battlefield system is described.

2. BACKGROUND

The V/L Taxonomy is a mathematical framework developed by BVLD, which clearly defines the elements of the V/L analysis process (Deitz 1986; Deitz and Ozolins 1989; Deitz et al. 1990; Klopcic, Starks, and Walbert 1992). Within this framework, two critical concepts are defined:

- *Vulnerability Space or Level.* A vulnerability space or vulnerability level (VL) is defined as a set of points, where each point is a vector whose elements each define the status of a particular aspect of the system under analysis (SUA) or subsystem under analysis (SSUA). The number of points in a particular level is a function of the analytical granularity imposed on the SUA. There are five separate levels in the Taxonomy: (1) VL0, the set of all possible vectors describing threat configurations for threat definition; (2) VL1, the set of all possible vectors defining initial conditions, consisting of the vector elements *threat definition* and *target definition* for each point in the space;

(3) VL2, the set of all possible vectors defining damaged components; (4) VL3, the set of all possible vectors defining a *new* system (a degradation of the original SUA); and (5) VL4, which is the set of all possible vectors defining the overall post-threat battlefield utility of the SUA.

- *Mapping.* A mapping is a function that operates on a point (state vector) in one level to generate a time-evolved image point in the next level. The mapping function itself is an algorithm (or set of algorithms) that incorporates the physics or engineering of a real-time and real-space process (such as electromagnetic pulse [EMP] coupling into a cable or chemical agent penetration into an enclosure). The mapping operator $O_{n,n+1}$ is defined as the noninvertible function that maps a point in VL_n to an image point or locus of points in VL_{n+1}.

Figure 1 illustrates the generic V/L Taxonomy.

The V/L Taxonomy is a framework for understanding the steps involved in evaluating the effect of a threat on a target asset. The primary purpose of the Taxonomy is not so much to impose a methodology for developing V/L simulations, but to provide a common language in which to discuss the processes involved in performing these simulations. In so doing, it has also helped to clarify the significance of various issues in vulnerability.

The concept of a vulnerability level, as described previously, is such that at each level, the state of the threat/target combination can be described by a set of values that are measurable and *meaningful.* It has been shown that mathematical spaces can be defined at each level (Walbert 1994). The structure requires mapping functions to determine the values in one level from the values in the previous level. These mapping functions are the heart of the vulnerability analysis. They determine the outcome of the analysis and must be continually refined to provide the highest quality models possible.

Now, we will describe the individual elements of the V/L Taxonomy in greater detail. VL0 describes the conditions of initial threat generation. This includes information such as the initial launch conditions of a tactical missile, the detonation of a thermonuclear device, or the initial generation of a radio frequency (RF) jamming signal. The initial mapping process within the V/L Taxonomy is the $O_{0,1}$ mapping, which describes the evolution of the threat from its initial generation in VL0 to the point of its first moment of interaction with a target system.

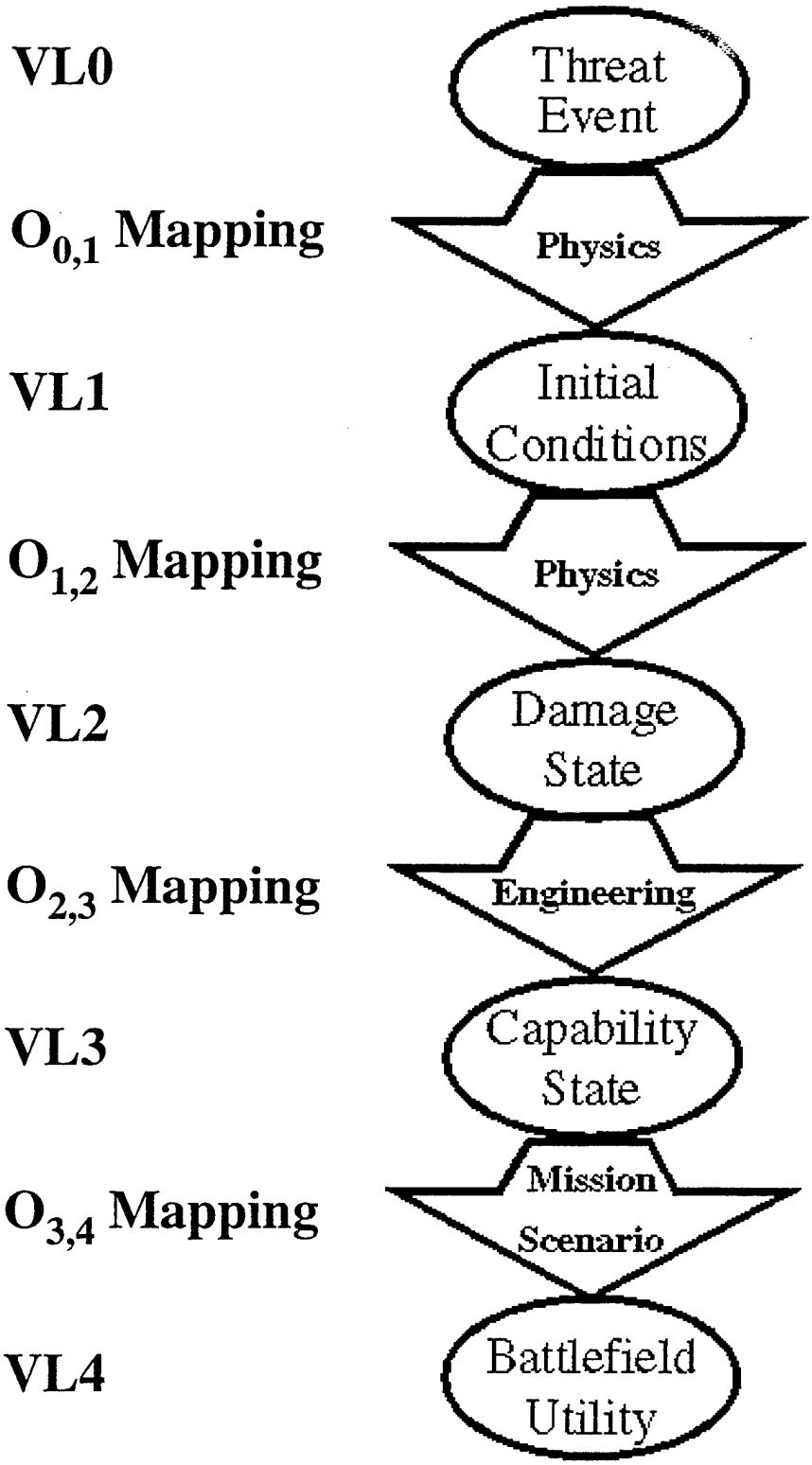

Figure 1. The V/L Taxonomy.

VL1 describes the initial threat/target system encounter conditions. The location and motion of both target and threat, plus other relevant physical parameters are important constituents of VL1 metrics. For an air defense problem, this would be the motion of the aircraft, the trajectory of the munition, as well as the payload on the munition (high explosive, fragments, incendiary, etc.). For an EW threat, the target may be in the same location performing the same maneuvers, but the information required (material properties, electronic equipment on board) will be different, and the nature of the threat and the models required to compute VL2 metrics will be completely different.

The collection of equations/decisions required to determine the VL2 metrics from the encounter conditions form the $O_{1,2}$ mapping. As previously mentioned, VL2 refers to the damage to the individual components within the target. A component is an item which performs a particular function within the overall system and which may be considered to fail or degrade as a unit. The numerical value associated with a component for VL2 measures is usually a number between 0.0 and 1.0 inclusive (these numbers are currently restricted to 0 or 1, indicating complete nonfunctionality or complete functionality, respectively). This number is either a value indicating partial functionality of the component, or a complementary value indicating loss of functionality within the same component. This flexibility of definitions requires clear communication from the model developer to the analyst to explain precisely what the values mean and how they are to be used.

It must be understood that the definition of "component" is rather flexible. It simply refers to the lowest level of detail; in other words, a component is the smallest thing that the analyst will consider as a single unit. Thus, to one analyst, an engine might be a component, while to another, the carburetor, distributor, cylinders, and water pump might be separate components. The level of detail in the physical definition of a component is referred to as the "granularity." The granularity required for an analysis is dictated by the type of questions being asked and the time available to prepare the inputs.

VL3 refers to the remaining capabilities of the target system as a whole. That is, what is the system capable of doing after the encounter with the threat? This is the level with which the soldier in the field usually interacts. When his tank, helicopter, or unit is hit, the first thing the soldier knows (if he survives the initial impact) is that it will no longer do certain things that it used to. VL3 values are usually determined from VL2 functionality values by means of capability or fault trees, which will be explained later in this report. This is referred to as the $O_{2,3}$ mapping.

VL4 is the battlefield utility of the target after it has been impacted (and possibly modified) by the threat. Traditionally, this was expressed as a list of "kill values" for various types or levels of damage to the target. For aircraft, this meant values for attrition (loss of the aircraft and crew), forced landing, and mission abort. For ground vehicles, this usually meant values for firepower, mobility, and catastrophic kill. However, since the advent of the capability metrics, it has become clear that these values are inadequate for expressing the response of the damaged target to the total battlefield environment. A damaged target that is considered useless in one scenario may be quite useful in a different scenario, depending on the capabilities required for that scenario.

One limitation with the V/L Taxonomy is the representation of time within the process structure. A specific time is assigned to all points within a vulnerability level; this representation limits the ability to overlap or embed dynamical multithreat/target interactions. The current V/L structure requires that each threat-specific V/L analytical process (which, when properly coupled together, would form an integrated analysis) involving a target system be executed in series or in sequence. This sequential structure limits the flexibility of dynamically combining threats; in a real battlefield, many threat-specific V/L processes run parallel (rather than serial) as a function of increasing time, which is difficult to convey within the existing process structure. What is required is an architecture that clearly delineates the flow of forward time (either linear or nonlinear) within a V/L process structure and also provides flexibility for dynamically combining multithreat effects on a target system.

3. TIME-DISCRETE V/L PROCESS STRUCTURE

3.1 Model Architecture. In the real battlefield, dynamical V/L processes are continuous. For analytical purposes, the continuum that contains all V/L process vectors can be discretized into a set of time-sampled states, where the total number of time samplings is finite. In this approach, the discretization of continuous time into a set of intervals (which may be either homogeneous or variable) is driven by two factors: (1) the relative time scales of the system dynamics, including both the threat/target system interaction physics and the post-interaction subsystem component response, and (2) the analytical granularity that a threat-specific model imposes upon the system dynamics. The number of possible states that might occur at any sample time is countably infinite and is a function of both the "V/L process time" and the computational complexity of the dynamical V/L processes.

As previously described in this report, the state of a dynamical system at a given instant can be envisioned as a "snapshot" in time, fully describing the system dynamics (in terms of descriptive parameters) at the sample time. Within a dynamic V/L process, the state of the system, sampled at time t_n, can be decomposed into four *substates*; within each of these substates is a *supervector*, or set of state vectors. Each substate is defined to contain exactly one supervector, where

- the *Threat* substate contains the supervector $[\mathbf{T}(t_n)]$,
- the *Target System* substate contains the supervector $[\mathbf{S}(t_n)]$,
- the *Component Functionality* substate contains the supervector $[\mathbf{F}(t_n)]$, and
- the *System Capability* substate contains the supervector $[\mathbf{C}(t_n)]$.

The notation $[\mathbf{V}(t_n)]$ implies that V is a supervector within a substate sampled at time t_n. Each of these supervectors contain information concerning the state of the threat, target system, component functionality levels, and system capability levels required by a V/L analyst for a system analysis. The threat, target system, component functionality, and system capability substates roughly correspond to VL0, VL1, VL2, and VL3 within the V/L Taxonomy, respectively (the correspondence is only approximate, since VL0 and VL1 are each constrained to one instant in time, namely the threat event initialization and the threat/target interaction initialization, respectively). Figure 2 shows an example of a V/L process supervector ($[\mathbf{T}(t_n)]$ for a nuclear EMP event).

Within the state sampled at time t_n, one can define mappings between substates that operate in a manner similar to the inter-VL mappings within the V/L Taxonomy. From the threat event initial time t_0 to the first instant of threat/target system interaction t_i, only the threat supervector $[\mathbf{T}(t_n)]$ dynamically evolves; however, a complete description of the target system exists within the supervector $[\mathbf{S}(t_n)]$ from time t_0 onward. Commencing at time t_i, $[\mathbf{T}(t_n)]$ and $[\mathbf{S}(t_n)]$ begin to interleaf, which is actually a coupling of state vectors in $[\mathbf{T}(t_n)]$ to other state vectors in $[\mathbf{S}(t_n)]$ and is represented by the coupling mapping $\mathbf{O}_{\mathbf{T+S}}$. This operation is a static information mapping rather than a dynamic coupling, where threat state vectors are added to specific portions of the target description. The information in $[\mathbf{S}(t_n)]$ is updated at every sample time (which is the time-dependent state of the target system) and then evaluated, resulting in the component functionality supervector $[\mathbf{F}(t_n)]$, which includes functionality state vectors, each of which contains both a metric for determining the function of a specific component and a functionality level chosen from the inclusive interval [0, 1] pertaining to that component. This evaluation process is essentially a mapping of physical system information to functional component information, and is

$$[T(t_n)] = \begin{bmatrix} T_1(t_n)\ \text{(Nuclear event parameters such as detonation energy and height of burst)} \\ T_2(t_n)\ \text{(EM field parameters such as electric field and magnetic field)} \\ T_3(t_n)\ \text{(Waveform parameters such as pulse width, frequency spectrum, rise time, and fall–off time)} \\ T_4(t_n)\ \text{(Environmental parameters such as atmospheric conditions and terrain topographical and electrical characteristics)} \end{bmatrix}$$

Figure 2. Example of the threat supervector $[T(t_n)]$ for a nuclear EMP event.

represented by the functional mapping $O_{S,F}$. This information can further be combined via a capability tree/network mapping to produce system capability levels associated with each sample time (which are represented by capability state vectors that populate $[C(t_n)]$); this mapping of component functional information to system-level capability information is represented by the capability tree/network mapping $O_{F,C}$.

Once the structure of operational mappings within a V/L process state sampled at time t_n is established, an operator must be defined, which maps this state to a proceeding state sampled at time t_{n+1}. This operator executes a mapping of the form

$$\left[M\left(t_{n+1}\right)\right] = F\left(\Delta t_{n,n+1}\right)\ \left[M\left(t_n\right)\right] + \left[\zeta\left(\Delta t_{n,n+1}\right)\right], \tag{1}$$

where

$[M(t_{n+1})]$ = meta-vector at time t_{n+1}

$F(\Delta t_{n,n+1})$ = transition matrix operating on $[M(t_n)]$

$[M(t_n)]$ = meta-vector at time t_n, composed of the supervector set $\{[T(t_n)], [S(t_n)], [F(t_n)]$, and $[C(t_n)]\}$.

$[\zeta(\Delta t_{n,n+1})]$ = stochastic "noise" vector representing possible random processes within the time interval $[t_n, t_{n+1}]$.

The dynamic meta-vector [M], which is the set of all meta-vectors $[M(t_n)]$ sampled over the entire "V/L process" time window, is actually all of the information which the V/L analyst requires to carry out a complete system analysis. The transition matrix $F(\Delta t_{n,n+1})$ operates on the existing state at t_n to determine a portion of $[M(t_{n+1})]$ (the remainder of $[M(t_{n+1})]$ being determined by stochastic processes which may be embedded within the interval $[t_n, t_{n+1}]$); if $[\zeta(\Delta t_{n,n+1})]$ is of zero amplitude, then the above time-mapping can be considered a Markovian process, in the sense that the state of the system at t_{n+1} is dependent only on its state at t_n (this assumes that threat events occur randomly with no a priori knowledge of their definite occurrence in space and time).

Now that both the operational mappings within a V/L process state and an inter-state time-mapping operator have been defined, the two can be combined in a V/L process architecture. Figure 3 illustrates this architecture. Note that the operational mappings between substates occur along an axis orthogonal to the horizontal time axis. The dynamic continuum representing the "V/L process" time is discretized into m intervals, where the nth interval is of length $t_{n,n+1} = t_{n+1} - t_n$. At the left of the figure, the threat event initial time is characterized by t_0; at time t_i, $[\mathbf{T}(t_n)]$ and the target system supervector $[\mathbf{S}(t_n)]$ begin to interact. At time $\geq t_i$, the information within $[\mathbf{T}(t_n)]$ and $[\mathbf{S}(t_n)]$ becomes "inter-leafed," so that, as time evolves, the state vectors within $[\mathbf{T}(t_n)]$ perturb the state vectors within $[\mathbf{S}(t_n)]$ and vice-versa. These perturbations may result in component damage or malfunction, which would, in turn, modify $[\mathbf{F}(t_n)]$ and $[\mathbf{C}(t_n)]$ as a function of time.

3.2 The Threat and Target System Supervectors $[\mathbf{T}(t_n)]$ and $[\mathbf{S}(t_n)]$ and the Threat/Target System Coupling Mapping O_{T+S}.

3.2.1 Threats of Concern. There are a multiplicity of battlefield threats of concern to SLAD. These threats can be generalized into generic classes of threats, as listed below (for more detailed information on these threats, see zum Brunnen, Kunkel, and Reza 1995).

- Ballistic threats, which include:
  - Armor piercing and armor-piercing incendiary
  - High-explosive and high-explosive incendiary
  - Threat platforms
  - Propagators
- Chemical and biological agents
- Obscurants
- Atmospheric and environmental effects

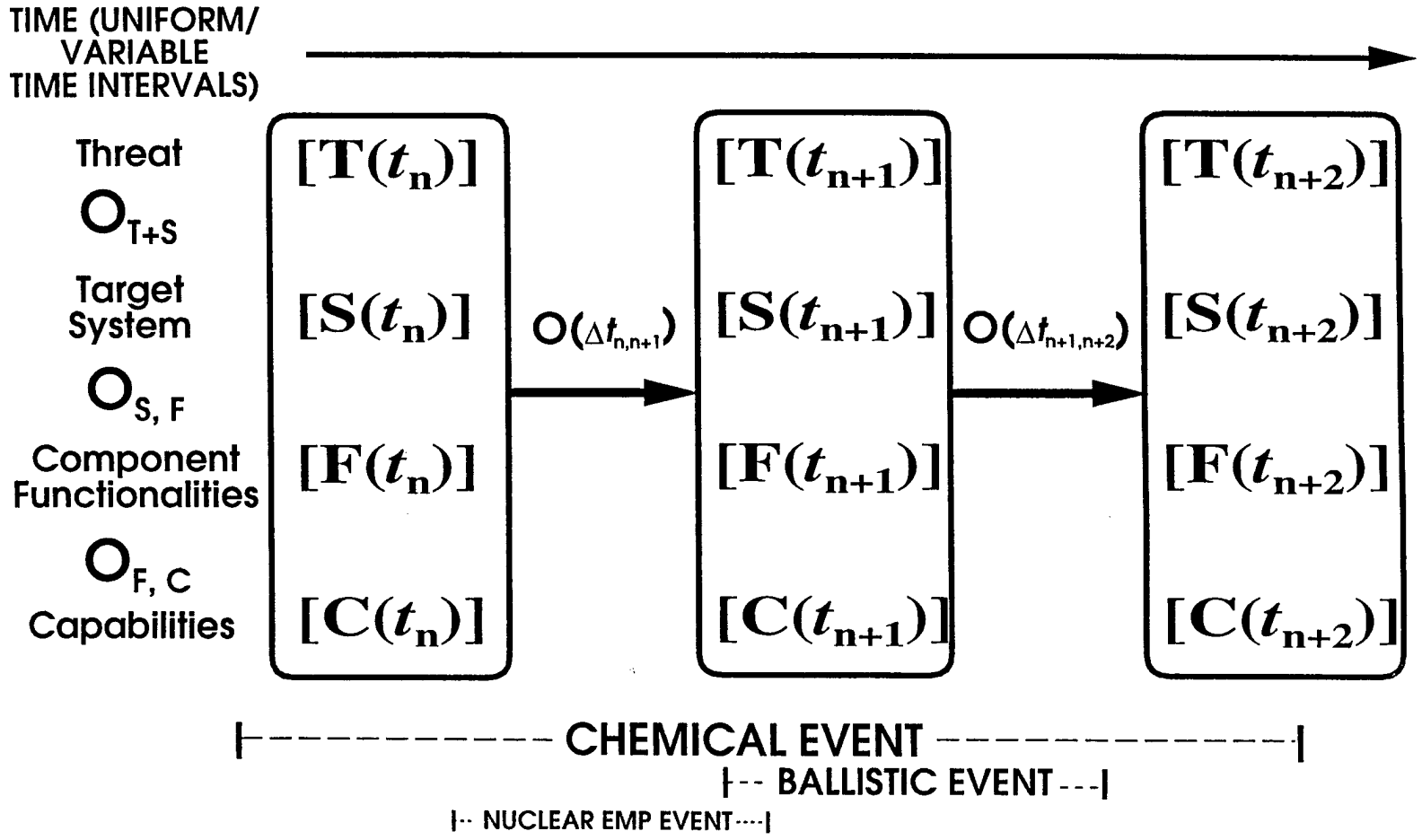

Figure 3. The time-discrete V/L process structure.

- Nuclear weapons effects, which include:
  - Thermal radiation
  - Blast and shock
  - Initial nuclear radiation (INR)
  - EMP
- EW threats, which include:
  - RF jamming
  - High-power microwaves (HPM)
  - Electromagnetic compatibility (EMC)
  - Electromagnetic interference (EMI)
- Information warfare/information viruses

Each of these threats, being real-world events, has a specific behavior with respect to time. In the time-discrete V/L process structure, this time behavior (for all threats) is described by the supervector time sequence $[T(t_0)], [T(t_1)], [T(t_2)], [T(t_3)], \ldots, [T(t_i)], \ldots, [T(t_{final})]$, where t_{final} = final time sample within a particular analysis. There is a possibly of coupling between threats prior to initial interaction with the target system ($t_0 \leq t_n < t_i$); in this situation, new coupled-threat state vectors would be introduced into $[T(t_n)]$.

3.2.2 V/L Kolmogorov Complexity. In this subsection, a metric is introduced for the purpose of determining the amount of information required in a target description in order to perform a threat-specific V/L analysis. In the 1960s, the Russian mathematician, A. N. Kolmogorov, considered the problem of the intrinsic descriptive complexity of a binary symbol string. This led to the formulation of the *Kolmogorov complexity* $K_U(x)$ of a symbol string x with respect to a universal computer U, defined as

$$K_U(x) = \min_{p:U(p)=x} l(p), \tag{2}$$

where min $l(p)$ = the minimum length of the symbol string p, and $U(p) = x$ defines p as the symbol string (or computer program) which, when fed into U, produces x. Thus p can be viewed as the minimum amount of information required by the universal computer U (which is approximately equivalent to any digital computer) in order to calculate x.

As an example, consider a binary string of 1,000 bits. If the sequence of bits in the binary string is generated in a completely random fashion, such as in, for example, the string 1001000011100110110...1001011, then the shortest program p_1 to generate this string is "x = 1001000011100110110...1001011." Assuming that the information content of an alpha-numeric symbol (other than 0 or 1) is about 5 bits, then the length of p_1 in terms of information is 1,010 bits. Thus, in this instance, K_U(1001000011100110110...1001011) = 1,010 bits.

Now suppose that we generate another binary string of 1,000 bits, namely 1010101010101010...10101010. The shortest program p_2 to generate this second binary string might be "write '10' 500 times." Using the above criteria for alpha-numeric characters, the length of p_2 = K_U(1010101010101010...10101010) = 67 bits, so that

$$.K_U(1010101010101010...10101010) << K_U(1001000011100110 11...10010110). \tag{3}$$

This notion of algorithmic complexity can be extended for application within the V/L process structure. The *V/L Kolmogorov complexity* (or just V/L complexity) $K_{V/L}(x_{inter.anal.})$ is thus defined as

$$K_{V/L}\left(x_{inter.anal.}\right) = \min_{[S(t_i)]:\, Process_{V/L}([S(t_i)]) \,=\, x_{inter.anal.}} l([S(t_i)]), \tag{4}$$

where min $l([S(t_i)])$ = the minimum size of the target system supervector as sampled at the initial threat/target interaction time t_i, and the expression $[S(t_i)]$: $Process_{V/L}([S(t_i)]) = x_{inter.anal.}$ can be interpreted as "the target system supervector sampled at time t_i, which, when input into the V/L process

structure, will yield a *complete* threat/target interaction analysis specific to threat x." In other words, $K_{V/L}(x_{inter.anal.})$ is the minimum amount of system information required by the V/L analyst to perform a threat/target interaction analysis, specific to threat x, from times t_i to t_p, where t_p is the time when all primary (if not secondary) influence of the threat on the system is complete. The expression "*complete* threat/target interaction analysis" refers to the condition that the target description contains enough detail so that all damage vectors attributable to threat x (as described within $[\mathbf{T}(t_i)]$) are calculated.

Zum Brunnen, Kunkel, and Reza (1995) address an aspect of V/L complexity by observing that the complexity of a target description is directly proportional to the granularity of the target description. This observation supports the notion of $K_{V/L}(x_{inter.anal.})$, in that the amount of information required for the analysis of the interaction between a threat x and a subsystem s is less than (or equal to) the amount required for an analysis of interaction between x and system S, where s is a subsystem of S, and the state vector $[s(t_i)]$ describing s is an element of the supervector $[\mathbf{S}(t_i)]$ describing S. Perhaps the most interesting use of $K_{V/L}(x_{inter.anal.})$ is in the relative ranking of V/L complexity as a function of threat. As an example, consider a communications shelter containing numerous computers, required for routing and switching electronic communications. For a ballistic threat/target interaction analysis, the computers within the shelter are modeled as critical components with appropriate material properties, and thus the internal circuit boards need only be modeled as low-resolution plates (and, in some instances, may not need to be modeled at all). For a nuclear EMP or an HPM interaction analysis, the computer model would require a granularity up to at least the integrated circuit (IC) chip level, so that, in this instance, $K_{V/L}(\text{ballistic}_{inter.anal.}) < K_{V/L}(\text{nuclear EMP}_{inter.anal.}) \cong K_{V/L}(\text{HPM}_{inter.anal.})$. In theory, the V/L complexity of the computer specific to all threats of concern to SLAD can be calculated and then used to rank the various threat-specific target information requirements within an integrated target description.

3.2.3 Structure of the O_{T+S} Mapping. The purpose of the O_{T+S} coupling mapping is to couple state vectors in $[\mathbf{T}(t_n)]$ to other state vectors in $[\mathbf{S}(t_n)]$ (as introduced in section 3.1). As previously mentioned, this operation is a static mapping rather than a dynamic coupling, and perhaps is best described as a process which "overlays" variable-granularity threat state vectors at spatial points around and within a three-dimensional solid geometric (BRL-CAD) model of the target system. This "overlay" operation is iterative with respect to time, commencing with the meta-vector sampled at t_i and continuing throughout succeeding time samples until the termination of all threat/target system interaction. At each time sample t_k, $[\mathbf{T}(t_k)]$ is updated, and the state vectors within $[\mathbf{T}(t_k)]$ interacting with the target system under analysis are remapped to current interaction sites.

3.3 The Functional Mapping $O_{S,F}$ and the Component Functionality Supervector $[F(t_n)]$.

3.3.1 Modular Unix-Based Vulnerability Estimation Suite (MUVES) Evaluation Module Concept. Before we address the $O_{S,F}$ functional mapping and the component functionality supervector $[F(t_n)]$ within the time-discrete V/L process structure, a similar process within the V/L Taxonomy is introduced as it is implemented within MUVES, a comprehensive software package designed for ballistic V/L analysis developed by BVLD of ARL (Murray, Moss, and Coates, in publication). The process for determining the VL2 component damage metrics within the V/L Taxonomy from VL1 information is implemented in the MUVES software as a two-step procedure. The first step involves the physics of the interaction. In this step, parameters such as mass, velocity, and material properties are used to calculate damage mechanisms, such as depth of penetration and impulse loading. This is where the model will track secondary threats (those created by the interaction of the primary threat with the target), such as spall or fire. One indicator of the fidelity of a particular model is if and how it tracks secondary threats, and the detail with which it does so.

The next step of the MUVES implementation of the $O_{1,2}$ mapping is called the *evaluation module.* This combines the total physical damage to a component into a single level of *conceptual* damage, where the measure of damage is chosen as part of the analysis methodology. Physical damage may be a hole in a component, temperature elevation (by exposure to fire), or myosis in a soldier among a seemingly infinite variety of possibilities. All known damage is rolled into a single value for the component, which is the measure of its damage. This could be a probability of dysfunction, or a degraded level of functionality, or some other metric required by the analyst and useful as an input to the $O_{2,3}$ mapping. Within the context of current ballistic analyses, this mapping of physical to conceptual damage is accomplished through the use of a probability of component dysfunction given a hit ($P_{cd/h}$) curve, where the $P_{cd/h}$ for a particular component under analysis is a function of both the mass and velocity of the impacting ballistic threat.

3.3.2 Models of Component Functionality. Once physical damage has been mapped into residual component-level functionality, the component functionality supervector $[F(t_n)]$ is established. As introduced in section 3.1, $[F(t_n)]$ is the collection of all available information addressing the functional states of all components within the target system. Each component is represented by a two-element functionality state vector, the second element being a numerical functional level selected from the interval

[0, 1]. One method of modeling continuous functionality levels is to discretize the continuum [0, 1] into a finite set of functionality values relevant to a component's functional metric.

As an example, consider a halogen flashlight which a soldier might use in the field. The critical components within the flashlight required to produce illumination include a dry-cell DC source, a halogen lightbulb, conducting cables which electrically connect the battery to the headlight, and an off/on switch. The DC source contains six dry-cell batteries connected in series, each of which can produce a potential difference of 2 V; thus, the total battery voltage will vary between 0–12 V, depending on the residual charge within each of the batteries. The battery functionality state vector $[F_{source}(t_n)]$ can be expressed as [*function* = "provide a direct current potential difference of 12 V between the positive and negative terminals," F_{source}] where F_{source} is the DC source functionality level. The functionality of the DC source, which is a continuum of possible values, could then be discretized into seven different values contained in the set $\{F_{source}\} = \{0, 1/6, 1/3, 1/2, 2/3, 5/6, 1\}$ which represent available source voltages of 0, 2, 4, 6, 8, 10, and 12 V, respectively. In this model, we assume that each of the six batteries either works or doesn't work, resulting in the six different levels of positive battery functionality (plus the completely nonfunctional state). Thus, within the context of a ballistic threat such as a bullet, any one of the six batteries (as well as combinations of the six) could be shorted out, reducing the total source voltage by 2 V per each damaged battery.

At this point, it is useful to introduce a concept from information theory in order to characterize the degree of outcome uncertainty associated with a stochastic process. The *entropy* $H(X)$ of a *discrete* random variable X is defined by Shannon (1948) as

$$H(K) = -\sum_{i=1}^{n} p_i \log_2 (p_i), \tag{5}$$

where the sample space of possible values of the random variable is formed by the discretization of the real number line $\mathbf{R}^1$ into the value set $\{X_1, X_2, X_3, \ldots, X_n\}$ with respective outcome probabilities $\{p_1, p_2, p_3, \ldots, p_n\}$ (Shannon 1948). In this context, the entropy defines the level of average uncertainty associated with each X_k in $\{X_1, X_2, X_3, \ldots, X_n\}$. For a discrete uniform random binary variable, the sample space consists of $\{X_1 = 0, X_2 = 1\}$ with respective outcome probabilities $\{p_1 = 0.5, p_2 = 0.5\}$ so that $H(X) = -((0.5*\log_2(0.5)) + (0.5*\log_2(0.5))) = 1$ bit of information. This quantity, which describes

the uncertainty in outcome of the classic random coin toss problem, defines one unit of entropy, and thus serves as a metric by which the relative average uncertainty of more complex (as well as simpler) events can be evaluated. If the above functionality variable F_{source} follows a uniform random distribution, then the associated entropy $H(F_{source}) = -(7*(1/7)*\log_2(1/7)) = 2.81$ bits. Roughly, this means that *almost three times as much information is required to determine the value of* F_{source} *compared to the case where* F_{source} *is defined as a binary variable (voltage/no voltage).* Another way to look at the Shannon entropy is that $H(F_{source})$ is the average amount of information required to uniquely select any one element of the set $\{F_{source}\}$ as the outcome state of a stochastic threat/target interaction event. If, on the other hand, the seven values in $\{F_{source}\}$ respectively follow the discrete outcome probability distribution {0, 1/7, 2/7, 3/7, 1/14, 1/14, 0} (shown in Figure 4), then the associated entropy $H(F_{source}) = -((1/7)*\log_2(1/7) + (2/7)*\log_2(2/7) + (3/7)*\log_2(3/7) + 2*(1/14)*\log_2(1/14)) = 1.99$ bits. As one would expect, the uniform random distribution yields the greater entropy; in fact, this distribution produces the maximal entropy per a given sample space cardinality.

Next, consider the halogen lightbulb within the flashlight. The lightbulb functionality state vector $[F_{lightbulb}(t_n)]$ can be expressed [*function* = "provide an optimal illumination rated at 200 W," $F_{lightbulb}$], where $F_{lightbulb}$ is characterized by the binary states "illumination" and "no illumination." Given a uniform random distribution of associated outcome probabilities, the entropy of the lightbulb functionality sample space $H(F_{lightbulb}) = 1$ bit (since $F_{lightbulb}$ is a binary variable). Similarly, conducting cable (or simply *cable*) and on/off switch (or simply *switch*) functionality state vectors are defined as [*function* = "provide a continuous electrical path," F_{cable}] and [*function* = "provide a continuous electrical path when in the *on* position and an electrical disconnect when in the *off* position," F_{switch}], respectively. If F_{cable} and F_{switch} are also defined as binary variables following a uniform random distribution, then $H(F_{cable}) = H(F_{switch}) = 1$ bit. Given that the states F_{source}, $F_{lightbulb}$, F_{cable}, and F_{switch} are *independent statistical events* (in that the damage state of one component does not contribute to the damage state of another component), then $H(F_{source}) + H(F_{lightbulb}) + H(F_{cable}) + H(F_{switch}) = 5.81$ bits. In other words, the information entropies of a set of discrete random variables are additive if and only if all variables in the set are statistically independent of each other. It is interesting to compare this value of total "system damage" entropy with that value given that all four component functionalities are binary, i.e., $H(F_{source}) + H(F_{lightbulb}) + H(F_{cable}) + H(F_{switch}) = 4$ bits. This difference indicates the amount of additional information required when modeling F_{source} with fractional values.

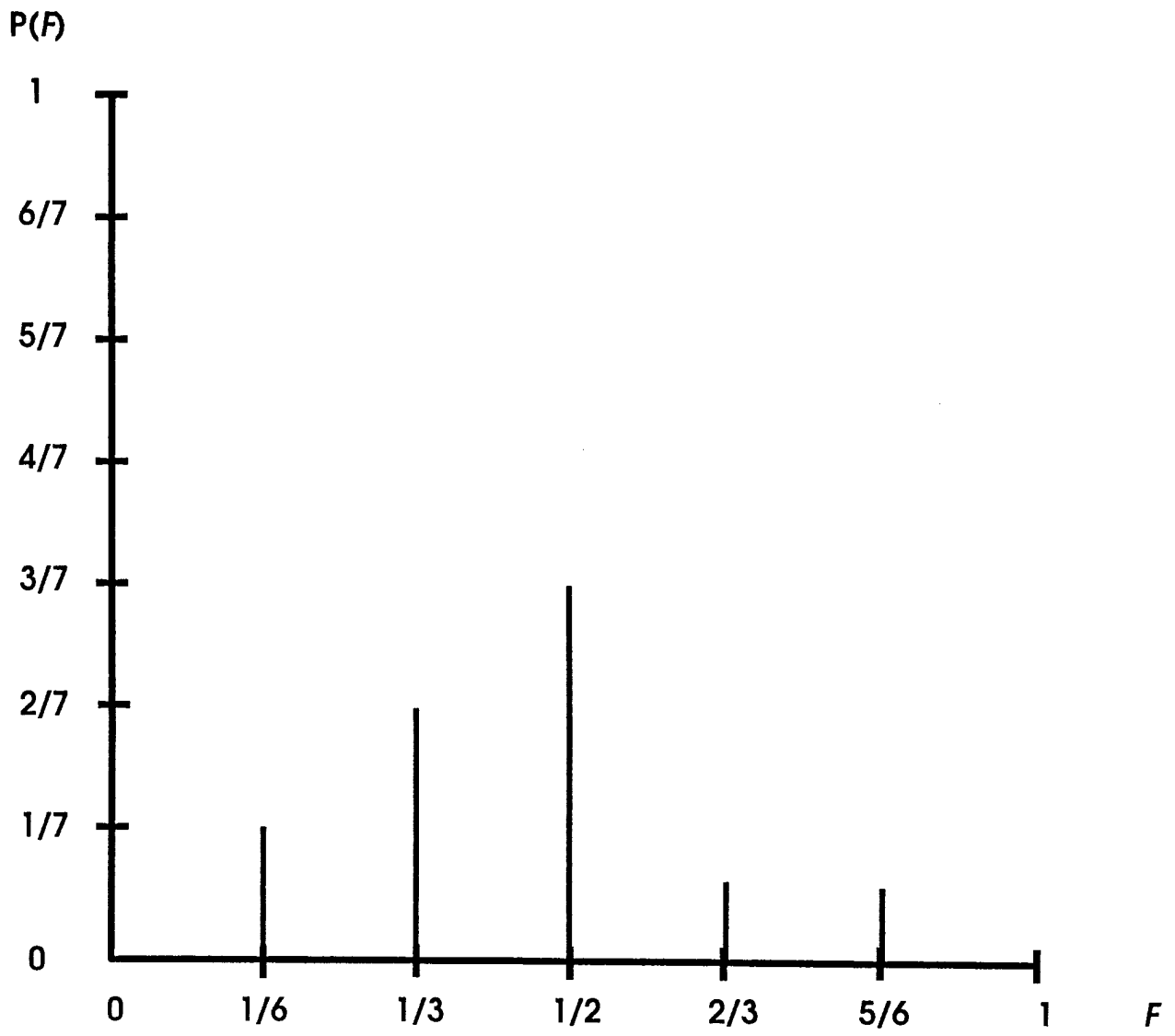

Figure 4. Discrete probability distribution of F_{source}.

3.3.3 Effects of Extensive Component Damage on Functional Models. One important point which we have not yet addressed is the mutable effect of component damage on functionality models. The examples given so far address only changes in component functionality levels; what if the effects of the threat-induced damage actually alter the physical aspects of component configuration to a point where the nature of the associated functionality sample space is changed?

Consider the DC source described in section 3.3.2. If one of the batteries is shorted by a bullet, then, assuming that each cell produces 2 V, the maximum voltage output from the DC source is reduced from 12 to 10 V. This fits within the model which we have thus far developed. But what if a chemical agent dissolves the insulator jacket on a battery cable, connected to the positive battery terminal, to a point where the cable intermittently and randomly arcs to ground? This new component configuration would modify the $O_{S,F}$ mapping, resulting in a new distribution of component functionality levels.

In theory, there are an infinite number of damage configurations for a particular component that would allow for some residual post-damage functionality within the semi-inclusive interval (0, 1]. Of course, this number would be greatly reduced by the practical requirements of the system-level capability, which depends on the component in question. The functional performance metric of the component would not be altered, only the distribution of functionality levels. One could discretize the continuum of possible damage configurations into several classes and then assign a specific component functionality model to each class (this subject will be further addressed in a followup technical report).

3.4 The Capability Tree/Network Mapping $O_{F,C}$ and the System Capability Supervector $[C(t_n)]$.

3.4.1 Boolean Capability/Fault Trees. One possible candidate for use in implementing the $O_{F,C}$ mapping within the time-discrete V/L process structure is the capability/fault tree methodology currently utilized within the MUVES analysis environment. This methodology, which essentially executes the $O_{2,3}$ mapping within the V/L Taxonomy, is encapsulated, for a particular target system, within what is called a *system definition file*, which is a description of target subsystem operation in terms of critical components. Critical components are components required for the continued performance of specific subsystem-level capabilities, such as mobility, speed, firepower, communication, and so on. In MUVES terminology, target subsystems are defined by relationships between groups of target components; thus, VL3 capability states are the output product of target subsystems. The operational states of individual critical components, which are currently limited to binary values (0/1), can be expressed in terms of (1) *lof*

(loss of function), (2) *frf* (fractional remaining functionality), where frf = 1 –lof, (3) probability of kill (p_k), (4) *hit* (hit flag, 0 or 1), and (5) *killed* (killed flag, 0 or 1). The system definition file then defines how these component states are combined to result in capability metrics.

System definition expressions are created by combining component states with either unary or binary operators or specific functions. A unary operator operates on a single component state, while a binary operator evaluates two component states which it joins. Functions are either user-defined (IF and IFELSE) or predefined (random uniform, random normal, and equal) and can operate on one or more component states. Table 1 gives a summary of all available operators and functions within MUVES.

Using a system definition file, either a capability or a fault tree operating under the rules of Boolean logic can be constructed for each system-level capability associated with the system under analysis. The difference between a capability tree and a fault tree is that the former evaluates component states as *frf*'s, linking these states through positive logic, while a fault tree evaluates component states as *lof*'s, linking these states through negative logic (Kunkel, in preparation). In this report, we will focus only on capability trees. These trees are series/parallel constructs, a series construct requires all component states in the construct to be nonzero for a resultant nonzero capability level (logical AND), while a parallel construct requires only one nonzero component state for a nonzero capability (logical OR). Typically, most systems will require a complex network of connected series and parallel constructs for capability-state evaluation. Also, all component states are assumed to be pairwise independent. Currently, component states are modeled simply as binary functions, where "1" indicates full functionality and "0" indicates full nonfunctionality (Roach 1993).

As an example of a Boolean capability tree, consider the halogen flashlight system discussed in section 3.3.2. If **source**, **cables**, **switch**, and **lightbulb** represent the binary *frf*'s of the DC source, cables, on/off switch, and lightbulb, respectively, then the *frf* of the illumination capability is expressed through the system definition

$$\textbf{illumination\_frf = source \& cables \& switch \& lightbulb,} \qquad (6)$$

where "&" is the Backus-Naur notation for the logical AND function. Figure 5 illustrates the graphical structure of this capability tree. In this case, all components within the tree either work (functionality

Table 1. Operators and Functions Within MUVES

Unary Operators:

Absolute Value
Boolean Value
(if operand > 0 => 1.0; else => 0.0)
NOT (1.0 – operand)

Binary Operators:

AND (a AND b = a x b)
OR
(a OR b = 1.0 – ((1.0 – a) x (1.0 – b)))
XOR
(a XOR b = (a + b – (a x b)) x (1.0 – (a x b))
Sum (a + b)
Difference (a – b)
Product (a x b)
MAX (maximum of a and b)
MIN (minimum of a and b)

Functions

IF (condition, exp)
(conditional function:
if condition is true (>0),
then exp; else 0.0)
IFELSE (condition, true–exp, false–exp)
(two–way conditional func:
if condition is true (>0),
then true–exp; else false–exp)
RANDOM UNIFORM (min, max)
(provides a random number r such that min < r < max with uniform distribution)
RANDOM NORMAL (mu, sigma)
(provides a random number from a Gaussian distribution with mean mu and standard deviation sigma)
EQUAL (a, b)
(returns 1.0 if both arguments are equal in value; else 0.0)

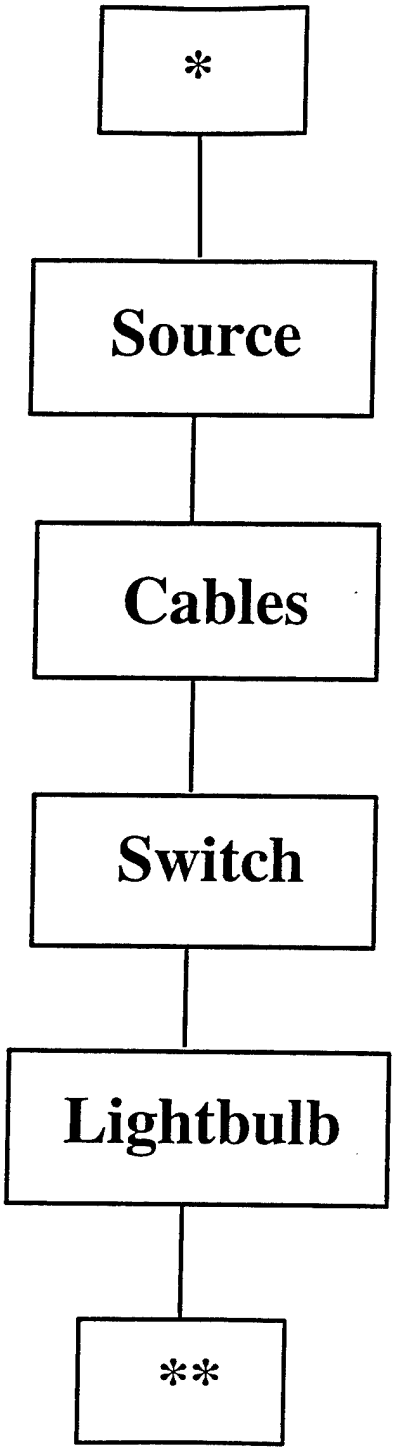

Figure 5. Capability tree for the "illumination" capability of the halogen flashlight system.

state = 1) or do not work (functionality state = 0). Also, the illumination capability of the flashlight is a binary metric, either producing illumination (**illumination_frf** = 1) or no illumination (**illumination_frf** = 0). Since all components are connected in series, there is only one tree configuration that will produce an illumination capability, namely **source** = **cables** = **switch** = **lightbulb** = 1. In other words, all components must function in order for the capability to operate.

3.4.2 Transfer Functions. The biggest limitation of the Boolean capability/fault tree methodology (as is currently used) is its inability to account for pairwise dependence between component states. As previously defined in section 3.3.2, component functionalities are mutually independent of one another as far as they are affected by damage vectors. But when components are linked together to produce a capability, there is usually some level of mutual dependence between the components involved. We choose to call the mutually independent states *independent functionalities* and the mutually dependent states *dependent functionalities*; in general, the total operational function of a component is itself a function of both its independent functionality (based on component damage) and its dependent functionality (due to connections with other components). Again, let us consider the halogen flashlight system as an example. In a real system, the ability of the lightbulb to produce light *is* dependent on the functionality of the DC source in a manner which is described by the relation

$$Power_{\text{headlight}} = \frac{\left(Voltage_{\text{source}}\right)^2}{Resistance_{\text{headlight filament}}}, \tag{7}$$

where $Resistance_{\text{lightbulb filament}} = (12\ \text{V})^2/200\ \text{W} = 0.72\ \Omega$ (here we make the assumption that filament resistance is not dependent on filament temperature). We call the above equation a *transfer function*, in that it transfers a quantity (voltage) produced in one component (the DC source) into another quantity (illuminative power) produced in a second component (lightbulb). The cables and switch serve as the *transfer channel*. We can convert the above equation to a normalized transfer function involving F_{source} and $C_{\text{illumination}}$ by defining

$$F_{\text{source}} = \frac{\text{DC source voltage}}{12\ \text{V (maximum voltage)}} \tag{8}$$

and

$$C_{\text{illumination}} = \frac{\text{lightbulb power}}{200\ \text{W (maximum rated power)}}, \tag{9}$$

and then replacing equations (8) and (9) into equation (7):

$$200\ \text{W} * C_{\text{illumination}} = \frac{\left(12\ \text{V} * F_{\text{source}}\right)^2}{0.72\ \Omega}, \tag{10}$$

which reduces to

$$C_{\text{illumination}} = \frac{144\ \text{V}^2 * F_{\text{source}}^2}{200\ \text{W} * 0.72\ \Omega} = F_{\text{source}}^2. \tag{11}$$

Note the difference between equation (11) (where $C_{\text{illumination}}$ is a function of F^2_{source}) and the Boolean product expressed in equation (6) (where $C_{\text{illumination}}$ is a function of F_{source}).

Within a complex subsystem containing numerous components, one could define transfer functions that move quantities (such as voltage, current, electrical or mechanical power, and information) from component to component. Given that the engineering process equations are known, these transfer functions could be implemented within a capability tree, allowing for continuous rather than binary values of component functionality. However, since transfer functions assume an inherent intercomponent causality (i.e., the state of component 2 is caused by the state of component 1), the capability tree must be modified in order to incorporate flow vectors between components. This directional information within the tree results in a *capability network*. Given a system consisting of n critical components operationally connected in serial, the network transfer function would be of the form

$$C = C\left(F_{\text{COMP}n}\left(F_{\text{COMP}n-1}\left(F_{\text{COMP}n-2}\left(\ldots F_{\text{COMP2}}\left(F_{\text{COMP1}}\right)\ldots\right)\right)\right)\right), \tag{12}$$

which is read as "the capability C is a function of F_{COMPn}, which is a function of $F_{COMPn-1}$, which is a function of $F_{COMPn-2}$, which is a function of . . . F_{COMP2}, which is a function of F_{COMP1}," where F_{COMPk} = functionality of the kth component. If all transfer functions within the capability network were linear (i.e., $F_{COMPk+1} = \alpha_k * F_{COMPk}$, where α_k is a constant), then the Boolean capability/fault tree methodology would suffice. In the case of nonlinear transfer functions (as in equation (7)), a network approach is the more desirable. This concept will be further investigated in a followup technical report.

3.4.3 Multibranch Capability Trees. As a final candidate for use in implementing the $O_{F,C}$ mapping, we now consider an extension of the Boolean capability tree. In section 3.4.2, the notion of inter-dependent components was introduced, where the functional state of component y is dependent on the functional state of component x. This dependence can be characterized by a conditional rule of the type "IF $F_x = A_j$, THEN $F_y = \{B_i, B_{i+1}, B_{i+2},..., B_{i+l}\}$," where A_j and the elements of $\{B_i, B_{i+1}, B_{i+2},..., B_{i+l}\}$ are discrete intervals within the inclusive range [0, 1]. In the generic case, A_j is a random discrete functionality variable sampled from the domain set $\{A_1, A_2, A_3,..., A_m\}$, and $\{B_i, B_{i+1}, B_{i+2},..., B_{i+l}\}$ is a discrete random scheme subset to the discrete random scheme $\{B_1, B_2, B_3,..., B_n\}$, where $H(\{B_i, B_{i+1}, B_{i+2},..., B_{i+l}\}) \leq H(\{B_1, B_2, B_3,..., B_n\})$. If the range set $\{B_i, B_{i+1}, B_{i+2},..., B_{i+l}\}$ contains only one element, then F_y is completely deterministic. Also, A_j can be a logical product (using the AND operator), a logical sum (using the OR operator), or a combination of products and sums among a collection of discrete random functionality variables $A_j^1, A_j^2, A_j^3,..., A_j^q$.

The IF/THEN type of logical production rules described previously can be used to form a *multibranch capability tree*. This is a structure which links all critical component functionality levels to corresponding system capability levels via logical production rules, and is similar to the notion of a Boolean capability tree except that pair-wise dependence between components is considered. The generic multibranch capability tree is graphically represented as a sequence of data nodes linked by vertical "branches." A data node is created by discretizing the continuous functionality range [0, 1] associated with a component into the n+1 intervals [0], (0, X_1], (X_1, X_2], (X_2, X_3],..., (X_{n-2}, X_{n-1}], (X_{n-1}, 1], which are then mapped into the data node values 0, X_1, X_2, X_3,..., X_{n-1}, 1, respectively.

To illustrate the concept of a multibranch capability tree, we again consider the halogen flashlight system. Let us further assume that we lack equation (7) (relating DC source voltage to lightbulb power), so that our understanding of the operation of the flashlight is based solely on empirical data. Table 2 presents data gathered from a series of hypothetical experiments on the flashlight, where seven permissible

values of F_{source} were combined with binary values of $F_{lightbulb}$, F_{cable}, and F_{switch}, and the resulting illumination capability was evaluated by a group of soldiers with 20/20 vision. The illumination capability metrics used were perceptual in nature: no illumination ($C_{illumination} = 0$), dim ($C_{illumination} = 1/3$), moderate ($C_{illumination} = 2/3$), and bright ($C_{illumination} = 1$). Although $C_{illumination}$ is really a continuous function, the soldiers reported perceived illumination at the granularity of the above four metrics. The production rules in Table 2 are thus generated by statistically fitting the responses of the sample group of soldiers to the four above capability metrics. Figure 6 shows how this empirical data can be graphically represented as a multibranch capability tree. In this example, the value of F_{source} determines which of four possible branches is followed within the capability tree, where each branch maps to a unique state within the output data node (representing the capability to provide illumination in terms of lightbulb rated power).

Table 2. Production Rules (Based on Empirical Data) for the Evaluation of $C_{illumination}$

IF F_{source} = [0, 0.4) AND $F_{cables} = F_{switch} = F_{lightbulb} = 1$, THEN $C_{illumination} = 0$ (no illumination),
IF F_{source} = [0.4, 0.6) AND $F_{cables} = F_{switch} = F_{lightbulb} = 1$, THEN $C_{illumination} = 1/3$ (dim),
IF F_{source} = [0.6, 0.9) AND $F_{cables} = F_{switch} = F_{lightbulb} = 1$, THEN $C_{illumination} = 2/3$ (moderate),
IF F_{source} = [0.9, 1.0] AND $F_{cables} = F_{switch} = F_{lightbulb} = 1$, THEN $C_{illumination} = 1$ (bright),

3.4.4 Comparison of Methodologies. Figure 7 displays a comparison of the three $O_{F,C}$ mapping methodologies presented in the previous sections: a graph which plots discrete values of $C_{illumination}$ as a function of F_{source}, where $F_{cables} = F_{switch} = F_{lightbulb} = 1$. Given that F_{source} may assume one of the seven values in the set {0, 1/6, 1/3, 1/2, 2/3, 5/6, 1}, $C_{illumination}$ is calculated three different ways: (1) $C_{illumination} = F_{source} * F_{cables} * F_{switch} * F_{lightbulb}$ (Boolean capability tree), (2) by using equation (11) (normalized transfer function), and (3) by using Table 2 (multibranch capability tree). Although there is complete agreement between the three methodologies at $C_{illumination} = 0$ and at $C_{illumination} = 1$, there is considerable variation in the interval $C_{illumination} = (0, 1)$. Which of these methodologies is best to use or is at least sufficient for a particular analysis is a judgement call left to the analyst.

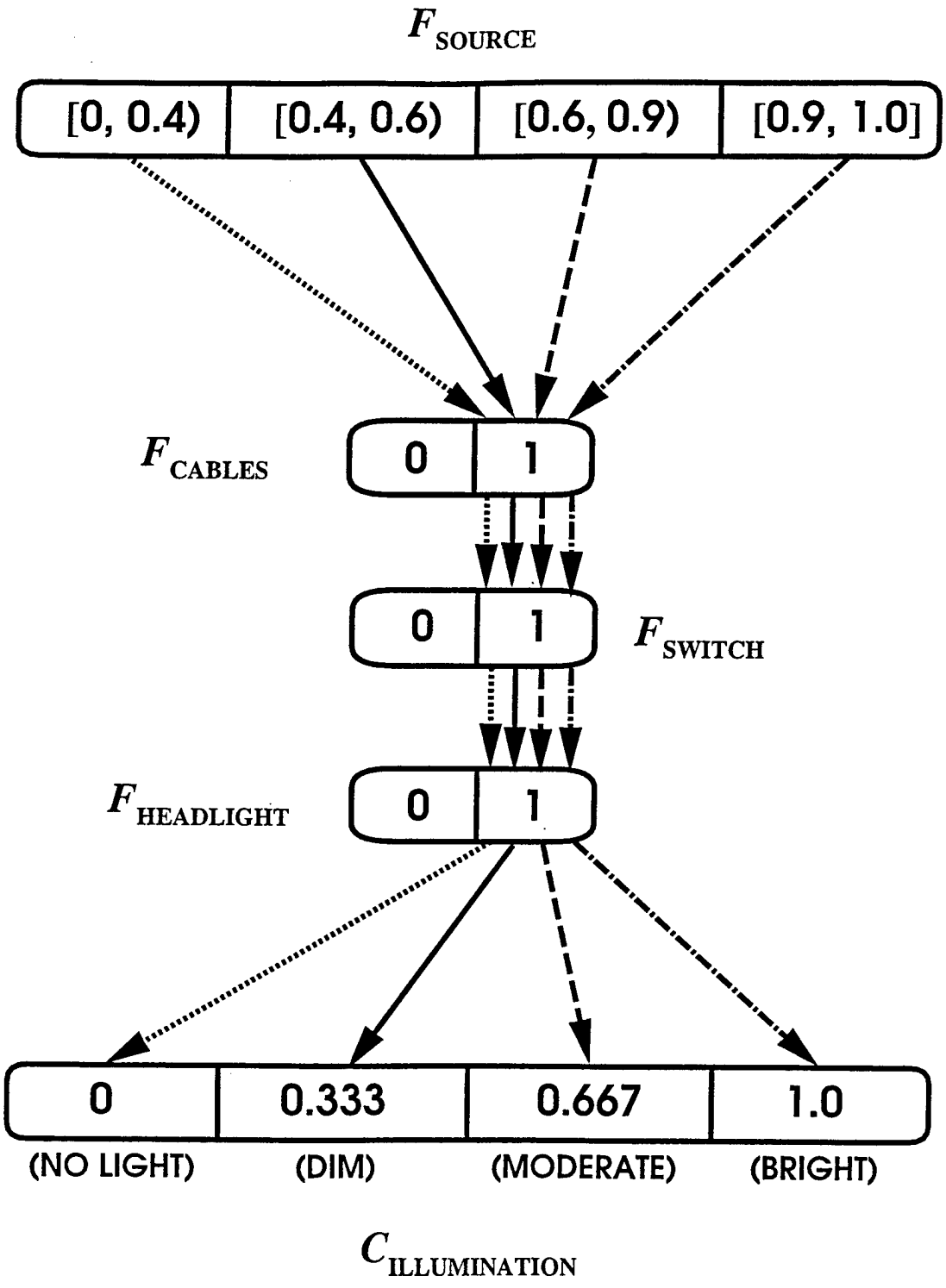

Figure 6. Multibranch capability tree for the "illumination" capability of the halogen flashlight system.

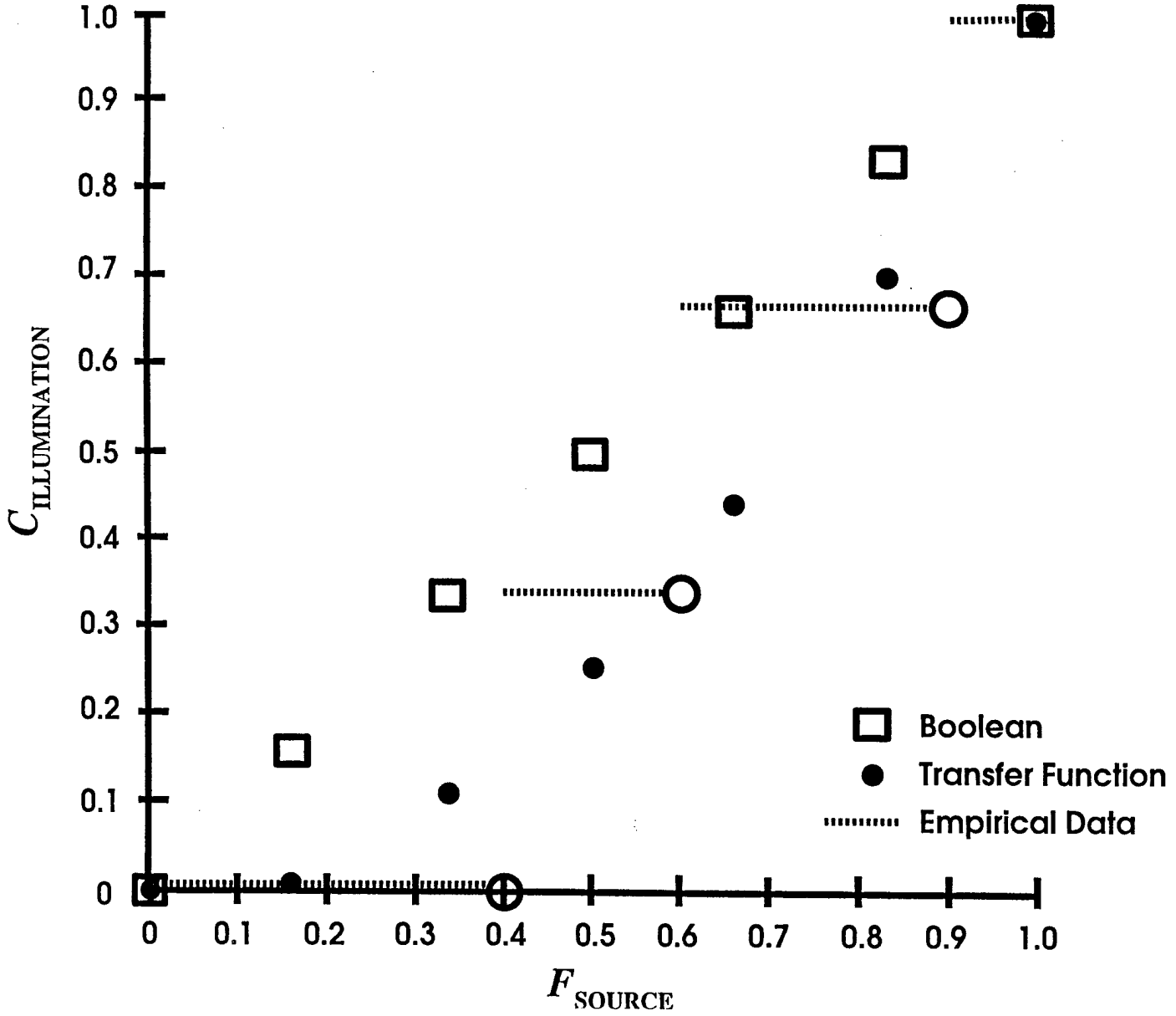

Figure 7. Comparison of $O_{F,C}$ mapping methodologies.

3.5 The Discrete Time Mapping $O(\Delta t_{n,\,n+1})$.

3.5.1 Deterministic vs. Stochastic Processes. As discussed in section 3.1, the discrete time mapping operator $O(\Delta t_{n,\,n+1})$ considers both deterministic and stochastic outcome events occurring at time t_{n+1} based on the state of the meta-vector $[M(t_n)]$. Deterministic events are representable by systems of differential equations, where the time differential dt is approximated by the mapping interval $\Delta t_{n,\,n+1}$. Stochastic events, on the other hand, are functions of random variables, which collapse to specific values at a sampling time t_n. Most events are actually combinations of both deterministic and stochastic events, where a random variable characterizing a system state might collapse to a particular value at t_n, which is then mapped to a state at t_{n+1} via differential equations. Thus, the transition matrix in Equation (1) can be decomposed into

$$F\left(\Delta t_{n,n+1}\right) = \left\{F_{sto}\left(t_n\right) + F_{det}\left(\Delta t_{n,n+1}\right)\right\}, \tag{13}$$

where

$F_{sto}(t_n)$ = stochastic transition matrix operating on $[M(t_n)]$.

$F_{det}(\Delta t_{n,n+1})$ = deterministic transition matrix operating on $[M(t_n)]$.

The nature of stochastic events requires that multiple trials are run, where each trial collapses one or more random variables associated with a stochastic process into measurable states via a random selection process (Monte Carlo routine). Thus, in general, the meta-vector $[M(t_n)]$ is decomposable into a set of m meta-vectors $\{[M(t_n)^1], [M(t_n)^2], [M(t_n)^3],..., [M(t_n)^m]\}$ associated with m trials, the resulting statistics of which are then used to make inferences within a V/L analysis. The Shannon information entropy $H(X)$ (introduced in section 3.3.2) can be utilized to study the relative outcome uncertainty of all random events within $[M(t_n)]$ (since random component functionality levels are resultant from component damage caused by both randomly generated threat events and secondary-effects damage mechanisms). In this application, one would expect entropy levels, for a particular analysis, to be a function of stochastic trial frequency.

When considering time-embedded events, the relative timescales become a problem when $\Delta t_{n,n+1}$ corresponding to one type of threat/target interaction is several orders of magnitude smaller than a $\Delta t_{n,n+1}$

corresponding to another type of threat/target interaction, as in the case of an electromagnetic threat time-embedded within a chemical threat (where the entire lifetime of the particular threat in question is considered). This phenomena is accounted for within the $O(\Delta t_{n,n+1})$ operator in the form of the stochastic "noise" vector $[\zeta(\Delta t_{n,n+1})]$, which represents possible random processes within the time interval $[t_n, t_{n+1}]$ (as introduced in section 3.1). In this sense, all processes, either deterministic, stochastic, or a mixture of the two, which evolve according to a timescale defined by $\Delta t_{n,n+1}{}^{\alpha}$ appear as "system noise" when embedded within another process which evolves on a timescale defined by $\Delta t_{n,n+1}{}^{\beta}$, where $\Delta t_{n,n+1}{}^{\alpha} << \alpha t_{n,n+1}{}^{\beta}$ (a difference of approximately two orders of magnitude or greater). This system noise is analogous to random noise that might arise within an electronic finite-state machine, such as a digital computer; the noise is actually generated by the operation of the system (or process) itself. In the case of V/L analyses, this is not really true, but, when examining two threats that evolve according to greatly disparate timescales (as in the instance of an EM threat and a chemical threat), the "system noise" model should prove to be a good approximation.

3.5.2 V/L Heuristics.

Combined Effects Heuristics. One of the primary advantages of an integrated system analysis is the opportunity to determine the net effects of dynamically combined threats. However, these combined threats are very complex in nature due to (1) greatly disparate time scales among the threats of consideration and (2) complex coupling between threat/target interaction processes. It is reasonably straightforward, however, to develop a set of rules of thumb in order to determine first-order combined-threat effects on systems. These generic (nonsystem-specific) rules will be referred to as *combined-effects heuristics* and will serve to supplement the $O(\Delta t_{n,n+1})$ mapping.

Since many threat-specific analyses utilize computer codes and experimental results that ignore the possible influence of other battlefield threats, the combined-effects heuristics are based on the collective expertise of experienced analysts in all threat areas of concern. This multiple expertise is integrated into a set of production rules operating within an IF/THEN structure. Only a very small percentage of threat events would actually combine synchronously; most combined-threat effects are created by sequential threat events occurring within a "small" interval of time, or embedded events (where a 10-ms ballistic event might occur within the 3-hr life span of a chemical event, as discussed in section 3.5.1). In the case of sequential threat events, the analyst must be careful not to assume commutivity between event times. As an example, consider a communication shelter; a ballistic event followed by a nuclear EMP event

might result in a compromise of the shelter's EM shielding and thus higher coupling of the EMP into the system, while the reverse situation would offer little or no interthreat enhancement.

The IF/THEN-type of logical production rules that form the structure of the combined-effects heuristics can be graphically represented within a multibranch tree structure similar to that of the capability trees discussed in section 3.4.3. Figure 8 presents an example of such a combined-effects heuristics tree, with multistate data nodes, which describes two different combined ballistic/pulsed EM-event (where the ballistic event precedes the EM event) conditional production rules: (1) "IF the average weight of a ballistic fragment = {less than 500 grains} AND the average number of apertures in the metallic portion of a system's exterior body (as a resultant from penetrating fragments) per square meter = {9 to 15} AND $E_{>0.5}$ (the frequency range within which greater than 50% of the total EM spectral energy is located) for a pulsed EM event = {f < 100 MHz}, THEN the additional EM coupling (E-field) due to a combined ballistic/EM effect = {0 to 10 dB}," and (2) "IF the average weight of a ballistic fragment = {500 to 1,500 gr} AND the average number of apertures in the metallic portion of a system's exterior body per square meter = {1 to 3} AND $E_{>0.5}$ for a pulsed EM event = {1 GHz ≤ f < 10 GHz}, THEN the additional EM coupling due to a combined ballistic/EM effect = {51 to 100 dB}." In theory, one could construct 60 different production rules using the four input data nodes in combination with AND conjunctive operators, given that each production rule is defendable in terms of either prior analysis, experimentation, or viable engineering judgement.

Secondary Effects Heuristics. In addition to combined effects, an equally-important aspect of V/L analyses are secondary threat effects, which arise within a system when damaged components interact in a systemic fashion. These secondary effects are connected to a particular threat event in the sense that the threat event is an indirect (rather than a direct) cause of the effect (as opposed to a primary threat effect, which has as a direct cause a particular threat/target interaction resulting in component damage or malfunction). Secondary-effects models are as complex to formulate as are combined-effects models, and thus, in many cases, the development of *secondary-effects heuristics* would benefit the V/L analyst in a similar manner as would combined-effects heuristics. These secondary-effects heuristics also serve to supplement the $O(\Delta t_{n,n+1})$ mapping.

Secondary-effects heuristics may also be graphically represented in a tree structure, as shown in Figure 9. In this scenario, a missile is intercepted by an RF jamming signal, which upsets the missile's telemetry computer. A possible resulting secondary effect that would degrade the missile's lethality is

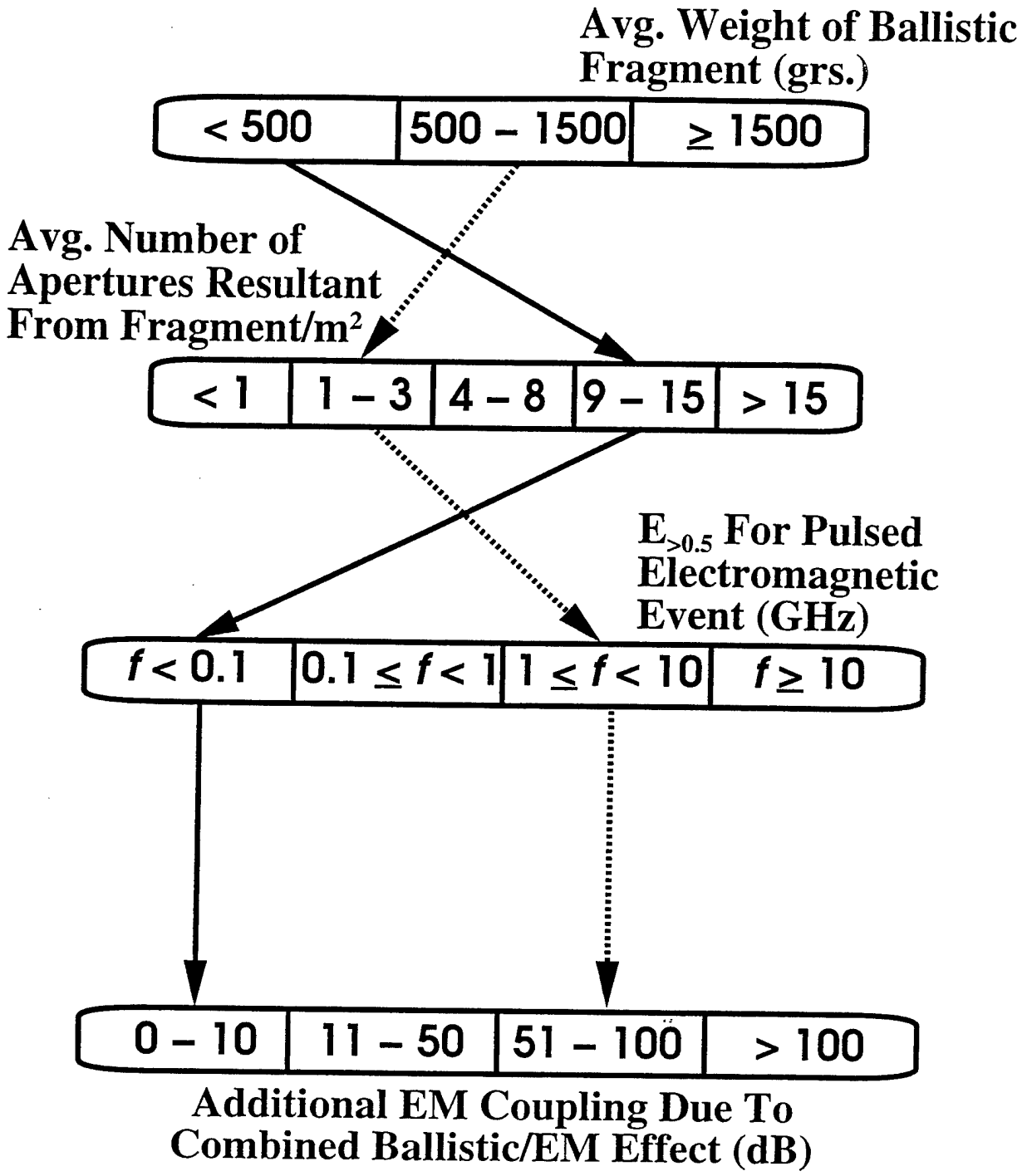

Figure 8. Ballistic/nuclear EMP combined-effects heuristics.

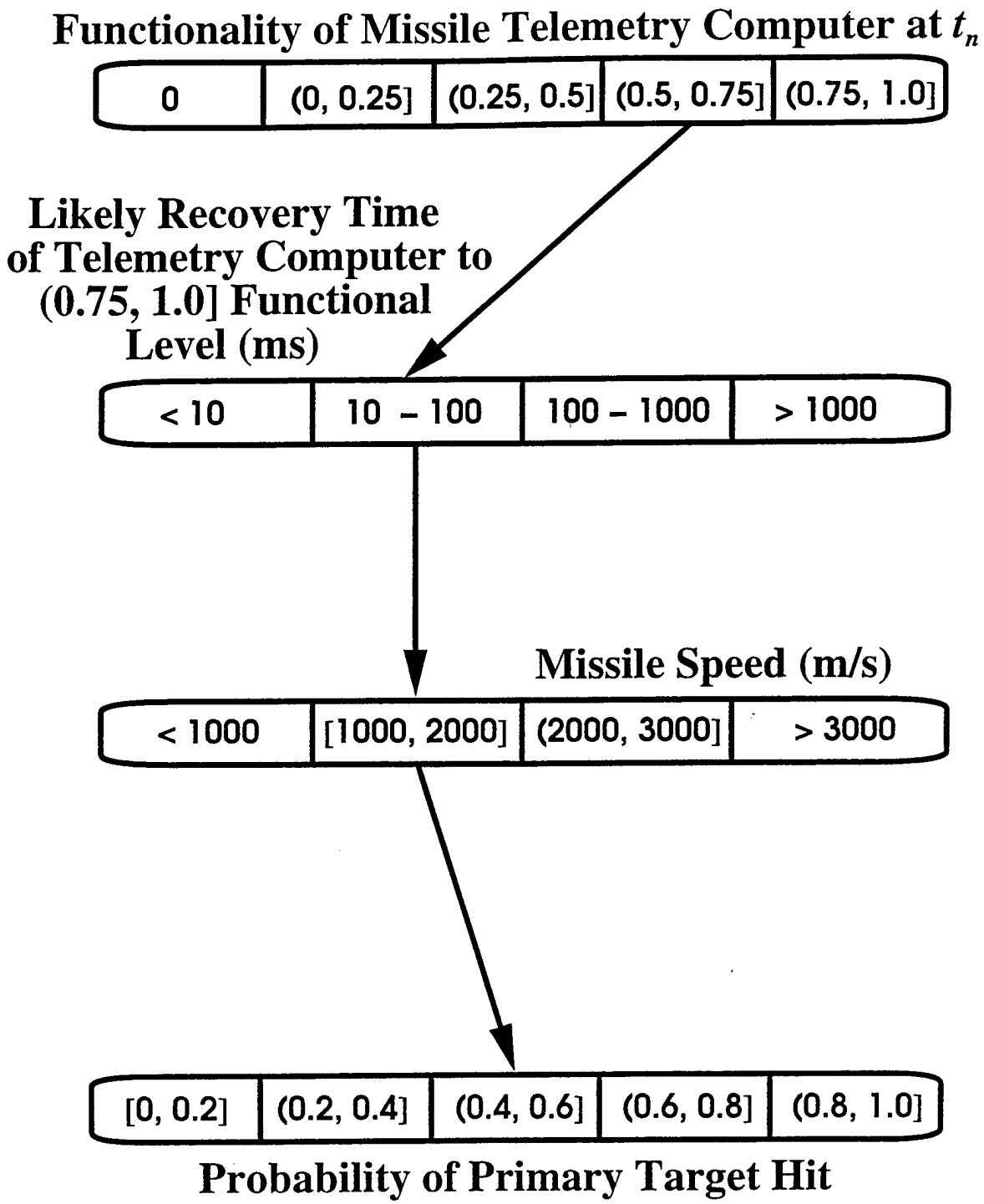

Figure 9. Missile/RF jamming signal secondary-effects heuristics.

described in the production rule "IF the functionality level of the missile telemetry computer at time t_n = (0.5, 0.75] AND the likely recovery time of the telemetry computer to a functional level within the interval (0.75, 1.0] = {10–100 ms} AND the missile speed = [1,000, 2,000] miles/hour, THEN the probability of a primary target hit = (0.4, 0.6]." Since the states represented within a supervector are discrete in nature, the above probability of a primary target hit must be "collapsed" to a distinct hit/no hit outcome via a random decision process such as a Monte Carlo algorithm (where the "hit" threshold could be set anywhere within the semibounded interval (0.4, 0.6]). Within this secondary-effects tree, there are 80 different possible production rules; in this case, statistical data based on simulations or experimentation is required for rule verification.

4. APPLICATION OF THE TIME DISCRETE V/L PROCESS STRUCTURE

Many of the concepts introduced in the previous sections can be applied to the V/L simulation of a military system, such as an attack helicopter interacting with both a long-term chemical threat and several near-instantaneous ballistic threats. The first step in the application is to formulate the parameters of both chemical and ballistic threats for inclusion within the threat supervector $[\mathbf{T}(t_n)]$. For a kinetic energy (KE) penetrator, threat parameters would include projectile dimension, material properties, fragment characteristics, functional capacity, and state of motion (Burdeshaw 1995). For a chemical cloud, threat parameters would include such information as height of the chemical threat event (initial release of the chemical agent), type of chemical agent, agent propagation medium (dry powder, liquid, slurry), properties of the agent propagation medium (particle size distribution, density of liquid), and meteorological conditions.

After $[\mathbf{T}(t_n)]$ is determined, the next step in setting up the SLV "architecture" is to construct the system-level capability supervector $[\mathbf{C}(t_n)]$. It is necessary to set up $[\mathbf{C}(t_n)]$ next, since elements of the critical component functionality supervector $[\mathbf{F}(t_n)]$ are determined from the contents of $[\mathbf{C}(t_n)]$. The capability supervector for an attack helicopter would contain a series of capability state vectors divided between the three capability classes MOVE, OPERATE, and COMMUNICATE (Kunkel 1995). MOVE capabilities would include subsystem functions such as engine fuel delivery, mechanical engine processes such as combustion and compression, and flight controls. OPERATE capabilities would include subsystem operations such as navigation, target detection, acquisition, and identification, and munitions control. Finally, COMMUNICATE capabilities would include subsystem applications for both voice and data communications.

Next, the critical component functionality supervector $[F(t_n)]$ is formulated. This process involves identifying sets of components contained within the attack helicopter system that are critical for the operational implementation of the capabilities contained in $[C(t_n)]$. At this point, we can also formulate the $O_{F,C}$ mapping. The logic used within this mapping operation is dependent on whether the analyst is interested in calculating residual capability (thus using positive logic) or loss of capability (thus using negative logic) as metrics within $[C(t_n)]$. It is recommended that the analyst utilize networks, as opposed to trees, when mapping component functionalities to nonbinary capability metrics.

Finally, the target system supervector $[S(t_n)]$ is formulated. At the very least, $[S(t_n)]$ must contain information on the physical configuration of all critical components listed within $[F(t_n)]$. In addition, $[S(t_n)]$ must also contain information on the physical configuration of noncritical components which, upon interacting with a threat, will affect the physical state of one or more critical components (i.e., a rubber seal surrounding the cockpit door on the attack helicopter must be modeled in order to determine chemical agent infiltration into the cockpit, which could degrade the functionality of the pilot, a critical component within the flight control capability). The contents of the target system supervector $[S(t_n)]$ would consist of a BRL-CAD target description of the attack helicopter, including information necessary for both ballistic and chemical analyses of all critical components identified in $[F(t_n)]$, such as material type and density.

Once all elements of the meta-vector $[M(t_n)] = \{[T(t_n)], [S(t_n)], [F(t_n)], [C(t_n)]\}$ and the $O_{F,C}$ mapping have been formulated, the time mapping $O(\Delta t_{n,n+1})$ is constructed. This mapping would consist of algorithms that model the dynamical behavior of the threat, the threat/target system interaction, and the post-threat-interaction response of components within a subsystem (engineering performance models). In the case of the present example, computer codes such as the Chemical/Biological Agent Vapor, Liquid, and Solid Tracking model VLSTRACK (chemical cloud formation and dynamics), and the Chemical Defense Materials Database (chemical agent effects on materials) would be used for the chemical threat dynamic analysis, while codes which use the JTCG/ME penetration modules in MUVES (penetrator dynamics and resulting target system damage) would be used for the ballistic threat dynamic analysis (Burdeshaw 1995). We will assume that engineering performance models (EPMs) also are available to model the dynamical post-threat/target-interaction subsystem response.

Once the V/L process architecture for this example is completed, the simulation is ready to run. The global time scale of the simulation is set to one time sample per second (t_{n+1}(global) – t_n(global) = 1.0 s)

to track the infiltration of the chemical agent into the helicopter cockpit. Within several of these intervals, KE penetrators impact the body of the helicopter, resulting in several holes in the cockpit wall. The local time scale for these ballistic interactions is set to one time sample per 1 ms (t_{n+1}(local) – t_n(local) = 10^{-3} s). Figure 10 illustrates the time-discrete V/L process structure applied to the global time interval between t_n(global) = 6.00×10^2 s (10 min) and t_{n+1}(global) = 6.01×10^2 s (10 min and 1 s). Within this interval, a KE penetrator produces a hole in the cockpit wall. As a result, the dynamical history of the chemical cloud will indicate a discontinuity within this interval as the cloud begins to seep through the hole. This discontinuity is attributable to the cockpit hole that seems to "instantaneously" appear with respect to the global time scale. The total duration of the ballistic "V/L process" is 100 ms.

5. CONCLUSIONS

In this report, we have introduced the time-discrete V/L process structure, a mathematical architecture, based on the V/L Taxonomy, for the conduct of dynamic V/L analyses of battlefield threat/target system interactions and subsequent degradation of the mission capabilities of the target system. By decoupling the mappings within a V/L process structure into those which statically process all information on the state of a system at a given time and one which dynamically evolves the state of the system, the time-discrete V/L process structure allows the analyst to determine the time-history of the capability states of a system. Also, by discretizing the "V/L process window" into user-determined time steps, the new process structure allows for the overlap or time-embedding of different threat/target system interaction processes that may be greatly disparate in time scale. This last feature suggests the time-discrete V/L process structure as an ideal platform for an integrated V/L analysis.

Since the ultimate products of a V/L process structure are the (possibly-degraded) capability states of a target system, the process architecture must be formulated in a reverse order. First, all mission-related capabilities related to a military system must be determined. Then, the components and subsystems (within the system) that are critically linked to the operation of a system-level capability must be identified. Finally, all physical and structural information describing these critical components/subsystems and how they fit together into the system must be collated into a target description. The complete target system description is actually a combination of a geometrical description and an operational description (both of which are hierarchical). The geometrical description (typically a BRL-CAD file) is contained within [$S(t_n)$], while the operational description (the operational couplings between components which, when integrated together, result in a specific capability) is within [$F(t_n)$] and $O_{F,C}$. Both geometrical

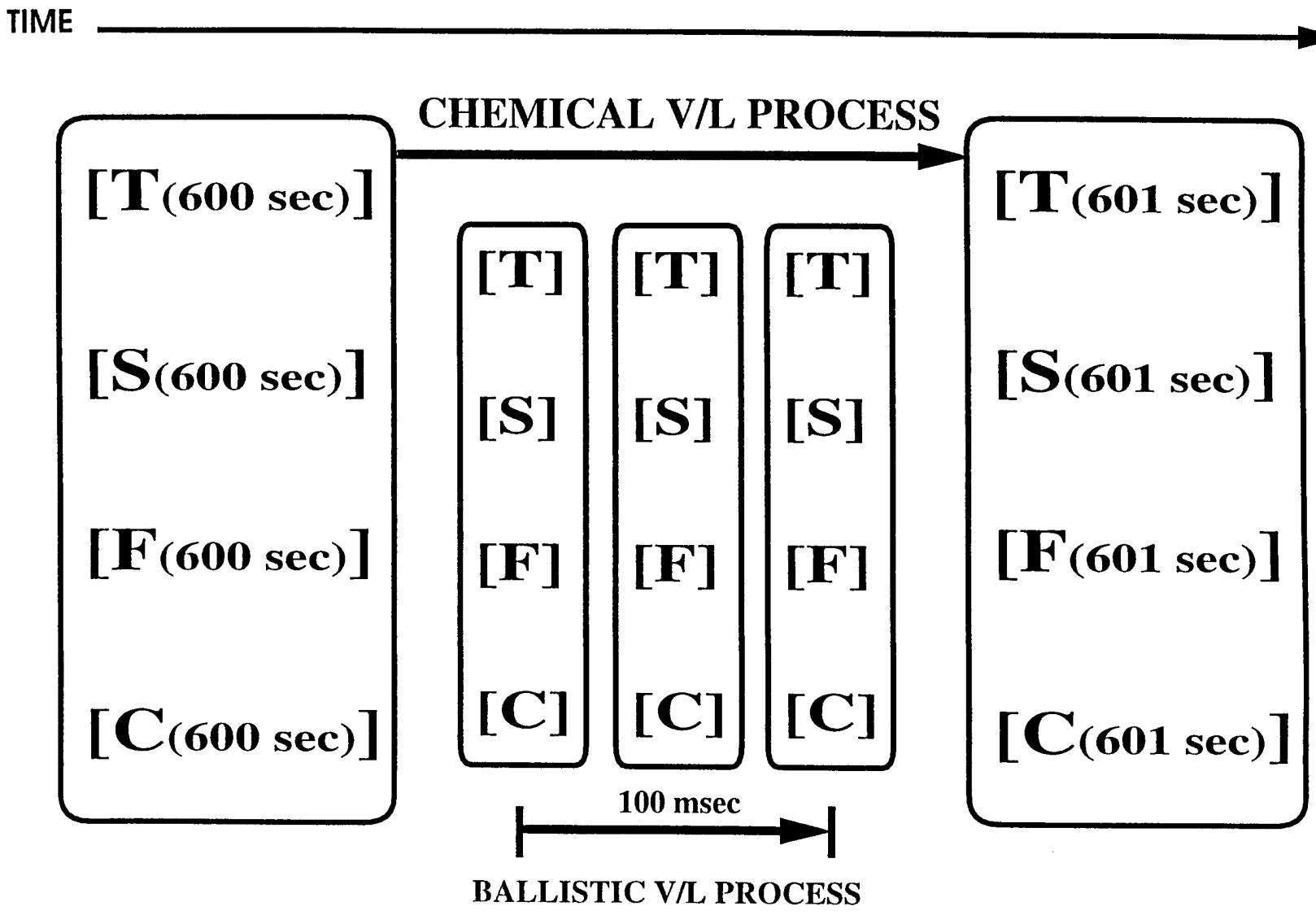

Figure 10. Example of a ballistic V/L process "time-embedded" within a chemical V/L process.

and operational descriptions are required to define a dynamic system; the descriptions must be confluent with one another.

In a followup technical report, the concepts introduced in sections 3.3 and 3.4 (and included subsections) will be further explored, including an in-depth tradeoff study of the relative applicability of all capability tree/network methodologies addressed within the current report.

6. REFERENCES

Burdeshaw, M. D. "The MAVEN Approximation Method Within the MUVES Environment." ARL-TR-787, U.S. Army Research Laboratory, July 1995.

Deitz, P. H. "Comments by the BRL in Response to the Report of the Board on Army Science and Technology's Committee on Vulnerability." U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, July 1986 (unpublished).

Deitz, P. H., and A. Ozolins. "Computer Simulation of the Abrams Live-Fire Field Testing." BRL-MR-3755, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, May 1989.

Deitz, P. H., M. W. Starks, J. H. Smith, and A. Ozolins. "Current Simulation Methods in Military Systems Vulnerability Assessment." BRL-MR-3880, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, November 1990.

Hughes, W. J. "A Taxonomy for the Combined Arms Threat." *Chemical Biological/Smoke Modeling and Simulation Newsletter*, vol. 1, no. 3, published by the Chemical Biological Information Analysis Center, Fall 1995.

Klopcic, J. T., M. W. Starks, and J. N. Walbert. "A Taxonomy for the Vulnerability/Lethality Analysis Process." BRL-MR-3972, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, May 1992.

Kunkel, Jr., R. W. "Degraded States and Fault Tree Analysis of LONGBOW APACHE." ARL-TR-801, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, July 1995.

Kunkel, Jr., R. W. "Degraded States Vulnerability Methodology and Fault Tree Analysis of the B-52H and Air Launched Cruise Missile (ALCM)." U.S. Army Research Laboratory, Aberdeen Proving Ground, MD (in preparation).

Mar, M. H. "A Nuclear Electromagnetic Pulse (EMP) Vulnerability/Lethality (V/L) Taxonomy With Focus on EMP Coupling." ARL-TR-786, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, July 1995.

Murray, K. R., G. S. Moss, and S. A. Coates. "Modular Unix-Based Vulnerability Estimation Suite (MUVES) Analyst's Guide (Release 2.0)." U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, 1994.

Roach, L. K. "Fault Tree Analysis and Extensions of the V/L Process Structure." ARL-TR-149, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, June 1993.

Ruth, B. G. "A Nuclear EMP Vulnerability/Lethality Taxonomy With Focus on Component Assessment." ARL-TR-205, U. S. Army Research Laboratory, Aberdeen Proving Ground, MD, November 1994.

Shannon, C. E. "A Mathematical Theory of Communication." *Bell System Tech. J.*, vol. 27, 1948.

Walbert, J. N. "The Mathematical Structure of the Vulnerability Spaces." ARL-TR-634, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, November 1994.

zum Brunnen, R. L. "Introducing Chemical/Biological Effects Into the Ballistic Vulnerability/Lethality Taxonomy." ARL-TR-715, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, March 1995.

zum Brunnen, R. L., R. W. Kunkel, Jr., and J. G. Reza. "Target Description Specifications for the Conduct of Integrated Analysis." Proceedings of the BRL-CAD Symposium '95, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, 1995.

NO. OF COPIES	ORGANIZATION
2	DEFENSE TECHNICAL INFO CTR ATTN DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	HQDA DAMO FDQ ATTN DENNIS SCHMIDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460
1	US MILITARY ACADEMY MATH SCI CTR OF EXCELLENCE DEPT OF MATHEMATICAL SCI ATTN MDN A MAJ DON ENGEN THAYER HALL WEST POINT NY 10996-1786
1	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL CS AL TP 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL CS AL TA 2800 POWDER MILL RD ADELPHI MD 20783-1145
3	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL CI LL 2800 POWDER MILL RD ADELPHI MD 20783-1145
	ABERDEEN PROVING GROUND
2	DIR USARL ATTN AMSRL CI LP (305)

NO. OF COPIES ORGANIZATION

1 OSD OUSD AT
STRT TAC SYS
ATTN DR SCHNEITER
3090 DEFNS PENTAGON RM 3E130
WASHINGTON DC 20301-3090

1 ASST SECY ARMY RESEARCH
DEVELOPMENT ACQUISITION
ATTN SARD ZD RM 2E673
103 ARMY PENTAGON
WASHINGTON DC 20310-0103

1 ASST SECY ARMY RESEARCH
DEVELOPMENT ACQUISITION
ATTN SARD ZP RM 2E661
103 ARMY PENTAGON
WASHINGTON DC 20310-0103

1 ASST SECY ARMY RESEARCH
DEVELOPMENT ACQUISITION
ATTN SARD ZS RM 3E448
103 ARMY PENTAGON
WASHINGTON DC 20310-0103

1 ASST SECY ARMY RESEARCH
DEVELOPMENT ACQUISITION
ATTN SARD ZT RM 3E374
103 ARMY PENTAGON
WASHINGTON DC 20310-0103

1 UNDER SEC OF THE ARMY
ATTN DUSA OR
RM 2E660
102 ARMY PENTAGON
WASHINGTON DC 20310-0102

1 ASST DEP CHIEF OF STAFF
OPERATIONS AND PLANS
ATTN DAMO FDZ RM 3A522
460 ARMY PENTAGON
WASHINGTON DC 20310-0460

1 DEPUTY CHIEF OF STAFF
OPERATIONS AND PLANS
ATTN DAMO SW RM 3C630
400 ARMY PENTAGON
WASHINGTON DC 20310-0400

NO. OF COPIES ORGANIZATION

1 ARMY RESEARCH LABORATORY
ATTN AMSRL SL
PROGRAMS AND PLANS MGR
WSMR NM 88002-5513

1 ARMY RESEARCH LABORATORY
ATTN AMSRL SL E
MR MARES
WSMR NM 88002-5513

1 ARMY TRADOC ANL CTR
ATTN ATRC W
MR KEINTZ
WSMR NM 88002-5502

1 ARMY TRNG & DOCTRINE CMND
ATTN ATCD B
FT MONROE VA 23651

ABERDEEN PROVING GROUND

1 CDR USATECOM
ATTN: AMSTE-TA

2 DIR USAMSAA
ATTN: AMXSY-ST
AMXSY-D

4 DIR USARL
ATTN: AMSRL-SL,
J WADE (433)
M STARKS (433)
AMSRL-SL-C, J BEILFUSS (E3331)
AMSRL-SL-B, P DEITZ (328)

1 CDR CBDCOM
ATTN: TECHNICAL LIBRARY
BLDG E3330

1 DIR CBIAC
BLDG E3330, RM 150

NO. OF COPIES	ORGANIZATION
1	COMMANDER U.S. ARMY MATERIEL COMMAND ATTN AMCAM 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	DIRECTOR USA ADV SYS RSCH & ANAL OFC ATCOM ATTN AMSAT R NR AMES RESEARCH CENTER MOFFETT FIELD CA 94035-1000
1	COMMANDER U.S. ARMY MISSILE COMMAND ATTN AMSMI RD CS R, DOCUMENTS AMSTA CG REDSTONE ARSENAL AL 35898-5000
1	DIRECTOR U.S. ARMY TRADOC ANAL CMD ATTN ATRC WSR WSMR NM 88002-5502
1	COMMANDANT US ARMY INFANTRY SCHOOL ATTN ATSH WCB O FT BENNING GA 31905-5000
1	DIRECTOR USA BALLISTIC MIS DEFNS SYS CMD ATTN ADV TECH CENTER PO BOX 1500 HUNTSVILLE AL 35807-3801
1	COMMANDER USA STRAT DEFNS CMD ATTN CSSD SL C CSSD SL S HUNTSVILLE AL 35807-3801
1	COMMANDER USA AVIATION SYS CMD ATTN AMSAV ES 4300 GOODFELLOW BLVD ST LOUIS MO 63120-1798
1	COMMANDER CECOM R&D TECH LIB ATTN ASQNC ELC IS L R FT MONMOUTH NJ 07703-5000

NO. OF COPIES	ORGANIZATION
1	COMMANDER USA RESEARCH OFC ATTN TECHNICAL LIBRARY PO BOX 12211 RESEARCH TRIANGLE PARK NC 27709-2211
1	COMMANDER USA LOGISTICS MGMT CTR ATTN DEFNS LOGISTICS STUDIES FT LEE VA 23801
1	COMMANDANT USA CMD & GENL STAFF COLLEGE FT LEAVENWORTH KS 66027
1	COMMANDER USA NGIC ATTN AMXST MC 3 220 SEVENTH ST NE CHARLOTTESVILLE VA 22901-5396
1	DIRECTOR INST OF DEFENSE ANALYSES ATTN LIBRARY 1801 BEAUREGARD ST ALEXANDRIA VA 22311
7	DIRECTOR USARL ATTN AMSRL SL EA R SHELBURNE AMSRL SL EG J PALOMO AMSRL SL EM R FLORES AMSRL SL EP D ALVAREZ AMSRL SL ES T ATHERTON AMSRL SL EV K MORRISON AMSRL SL C R SUTHERLAND WSMR NM 88002-5513
1	DIR USARL ATTN AMSRL SL EI J NOWAK FT MONMOUTH NJ 07703-5000
1	JEFF HANES 1447 HARFORD SQ DR EDGEWOOD MD 21040

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
1	CDR, USATECOM ATTN: AMSTE-TC
1	DIR, ERDEC ATTN: SCBRD-RT
1	CDR, CBDA ATTN: AMSCB-CII
1	CDR, USACSTA ATTN: STECS
70	DIR, USARL ATTN: AMSRL-SL, J. SMITH AMSRL-SL-BA, J. MORRISSEY L. ROACH AMSRL-SL-BG, A. YOUNG AMSRL-SL-BL, M. RITONDO AMSRL-SL-BS, D. BELY T. KLOPCIC AMSRL-SL-BV, R. SANDMEYER W. MERMAGEN, JR. M. MUUSS P. TANENBAUM W. BAKER AMSRL-SL-I, R. REITZ D. BASSETT D. KIRK M. VOGEL E. PANUSKA DR. D. HASKELL DR. J. FEENEY R. ZIGLER AMSRL-SL-C, L. D'ELICIO AMSRL-SL-CM, D. FARENWALD B. RUTH (15 CPS) L. DAVIS R. KUNKEL R. JOLLIFFE R. TYTUS R. ZUM BRUNNEN E. FIORAVANTE M. MAR J. SOLN MAJ JEROME GILMAN

NO. OF COPIES	ORGANIZATION
	AMSRL-SL-CO, D. BAYLOR J. NEALON R. LAUGHMAN A. BEVEC R. PARSONS J. CAPOBIANCO D. DAVIS J. ANDRESE R. PROCHAZKA AMSRL-SL-CS, J. BEILFUSS T. FLORY B. SMITH DR. M. SMITH J. FRANZ T. MAK M. KAUFMAN D. MANYAK R. POLIMADEI M. BUMBAUGH J. MYERS R. WEISS A. PRICE T. MALLORY J. KELLEY

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-1222 (Ruth) Date of Report November 1996

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT ADDRESS

_____
Organization

_____
Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

_____
Organization

_____
Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)

(DO NOT STAPLE)