Network Science Center
West Point

Advancing the Study of Network Science

United States Military Academy, Network Science Center 2009.12.1

Secure Computer Systems:

Extensions to the Bell-La Padula Model[1]

By

John R. James, Frank Mabry, Kevin Huggins, Michael Miller, Thomas Cook, Florian Tamang, Sam Abbott-McCune, Howard Taylor and William J. Adams

The Network Science Center's Research Reports presents work in progress by researchers affiliated with the Center. It is the mission of the Center to bring together service members, civilians, U. S. Military Academy cadets, and the academic community to contribute to the emerging discipline of network science and, in doing so, to address specific Army needs and related network science challenges. Through its work, the Center also provides important, and relevant, educational opportunities for West Point cadets and faculty. To learn more about the Center and its work, visit www.netscience.usma.edu.

Network Science Center at West Point
Thayer Hall, Room 119
West Point, NY 10996
1-845-938-0804

www.netscience.usma.edu
net-sci@usma.edu

ISBN 978-1-934808-08-5 1-934808-08-3

[1] This interim report for work performed as part of the Flowing Valued Information Project is a publication of the United States Military Academy's Network Science Center. This material is based upon work supported by the U.S. Army Research Office under Grant Award Number MIPR9FDATXR048. The views expressed in this report are those of the authors and do not reflect the official policy or position of the United States Military Academy, the Department of the Army, or the Department of Defense.

ABSTRACT

This report provides a summary of initial results of a project investigating solutions to problems in flowing valued information among coalition partners. The research objectives of the Flowing Valued Information project include: (1) improving our capability to enable automated understanding of command intent and (2) improving our capability to provide automated support of a command decision to share information. Initial investigations have indicated a need to extend the mathematical foundations provided by D. Elliott Bell and Leonard J. La Padula which applied early system theory to enable building formal systems for proving security results for distributed computing systems.

Our extensions are in two areas: (1) we discuss application of current system theory results in modeling compositions of continuous and discrete systems, and (2) we discuss mathematical foundations for adding support for a commander's decision to share information. The motivation for the extensions is grounded in two continuing shortfalls in science and technology available for decision support: (1) the inability of current system models to predict future state of complex systems and (2) the continued difficulty in enabling automated support for a commander's decision to share information in order to meet mission requirements. We believe that useful extensions are achievable for building more accurate models of complex system interactions for small unit operations since general system theory has advanced since the work of Bell and La Padula and we believe that explicit extensions for sharing information are needed (and possible) for information which needs to be shared while simultaneously protecting information which must remain protected.

Many of the available solutions for sharing information have successfully created multi-level secure systems (networks of systems) which follow access control rules (many based on the Bell-La Padula security model) in which access to information is granted to a given level of classified information once confirmation is achieved that a given subject has the required clearance (mandatory access control). However, the current implementation of mandatory access controls and role-based access controls does not support mission success for those missions that require sharing information on an ad hoc basis, especially at the lowest tactical level for operations which require social and cultural awareness of local populations and non-government agencies as well as local support in achieving mission success. Thus, there is a need to explicitly enable categories of information which can be be labeled "need-to-share". We note that similar needs are present in commercial systems where proprietary information needs to be protected while marketing information needs to be shared. As in the case of the early work by Bell and La Padula, we begin with an introductory section to bridge the gap between systems theory and practical problem solving.

We then extend the Bell-La Padula model to include continuous and discrete system states as is done for current general system theory for control of complex, distributed systems. We also extend the Bell-La Padula definitions for "security" and "compromise" to include a definition of a "need-to-share" and a "failure to share". A basic result concerning security in computer systems, using the precise notions for "security", "compromise", "sharing", "need-to-know", and "need-to-share" is then given.

Finally, we demonstrate via a platoon-level scenario why compositions of continuous and discrete (hybrid) systems models are needed to reason about command intent and why implementation of a "need-to-share" information is needed to help achieve command intent.

PREFACE

The early paper by Bell and La Padula applied general systems theory available at that time. Since then, system theory has been extended to include not only the discrete-valued finite state systems considered by Bell and La Padula but also compositions of continuous and discrete systems. A central argument of our project is that building secure systems today requires an underlying modeling framework which supports both formal logical analysis of the current and future state of the system under discussion as well as predictive analysis of analytical components which must obey the laws of physics. Since modeling large networks of interacting, complex systems is not feasible, the issue at hand in our investigations is to model enough of the continuous dynamics to capture physics of interest while maintaining the ability to reason about the logical state of the system (the set of discrete components). Thus, analysis of secure computer systems for current and future military operations (net-centric or net-enabled operations) requires a framework which admits composition of discrete and continuous systems. This argument is not new in general system science. Indeed, for several years every student majoring in either electrical engineering or computer science at the University of California at Berkeley has taken a course which requires considering systems as compositions of continuous and discrete components (Lee & Varaiya, 2000).

The extensions to the Bell and La Padula model reported in this paper lay the groundwork for: (1) building more accurate models of the complex operational environments of today and tomorrow, and (2) providing automation support for a commander's decision to share information while simultaneously maintaining the security of information which must not be compromised. Indeed, concerning the issue of sharing information, we take the position that at least one category of data, metadata, should be shared continuously with everyone, all of the time, and in every area (e.g. information metadata for all categories of information should be shared with all categories of users).

TABLE OF CONTENTS

ABSTRACT ii
PREFACE iii
SECTION I INTRODUCTION 1
General Systems 1
Systems Modeling 1
Flowing Information 3
Secure Computer Systems 4
Problems of Security 5
Summary 5
SECTION II FOUNDATIONS OF A MATHEMATICAL MODEL FOR SECURING AND SHARING INFORMATION. 6
Elements of the Model 6
States of the System 7
State-Transition Relation 8
Summary 9
SECTION III A FUNDAMENTAL RESULT 9
Compromise, Security of Information and Sharing of Information 9
Constraints 10
Basic Security and Sharing Theorem 11
Summary 11
SECTION IV CONCLUSION 12
Introduction 12
Problem Reformulation 12
Access Control 15
Data Base Sharing 17
Summary 17
REFERENCES 17
Appendix A: Earthquake Scenario ..... A – 1
Operations Order ..... A - 1
Discussion ..... A - 2

Command Intent: ..... A - 3
Discrete and Continuous Variables ..... A - 3
Need-to-Know and Need-to-Share ..... A - 4
Available Technologies: ..... A - 6

SECTION I INTRODUCTION

General Systems

As indicated in the abstract, general system theory has advanced considerably since the very capable summary of the subject was provided by Bell and La Padula over 35 years ago (Bell & LaPadula, 1973). The discussion below concerning creating a general system model for complex systems follows the development of the text used by Professors Edward Lee and Pravin Varaiya for educational programs for electrical engineers and computer scientists (Lee & Varaiya, 2000). The primary distinction to be discussed is that while Bell and La Padula considered a system in its most general form to be a relation on abstract sets, the modern system theorists add consideration of continuous systems as well as compositions of discrete, set-based, systems and continuous systems. Functional concepts of a mapping from one state space (the domain) to another (the range) remain the same. That is, while Bell La Padula considered the expression $S \subseteq X \times Y$ where the system S is a relation on the abstract sets X and Y, Lee and Varaiya (and others) consider the general system S to have elements which are members of abstract sets and also elements which are members of general functional spaces (Lee & Varaiya, 2002).

Systems Modeling

General systems theory has been a very valuable tool for advancing our understanding of complex systems. Unfortunately however, the science of systems modeling still lags the complexity of large-scale networks of systems of interest (e.g. power generation and distribution networks, telecommunications networks, and economic networks) in the sense of being unable to predict future behaviors of networks of systems (BAST, Board on Army Science and Technology, 2005). For purposes of this paper, we restrict the complexity of systems under consideration to those whose behaviors can be modeled by current systems theory. The practical application of the theory to real-world problems for any given system then depends upon the predictions of future system state available from the model being "close enough" to the actual future states of the system of interest.

For purposes of this paper, we are interested in (1) extending the models of the systems being analyzed to include what are described today as "complex systems" and (2) extending the existing Bell-La Padula model for defining a failure to secure information (a security compromise) to include defining a failure to share information (a sharing compromise).

Following the development in the original Bell and La Padula paper, we assume a system, S, to be adequately approximated as an input-output relation. That is, we consider the behaviors of S to be represented as:

$$S \subseteq V \times X$$

where S is a function from V to X (S: V $\rightarrow$ X), and it is natural to consider S to be a functional system. In this case, it is convenient to consider the elements of V to be inputs and the elements of X to be outputs (the state of the system) so that S expresses a functional input-output relationship. However, while Bell and La Padula assumed that V and X are members of abstract sets, capturing the complexity of

networked systems of interest requires that the domain and range of the functions of interest be expanded to include real-valued variables as well as discrete-valued variables. Following the development of hybrid control theory as discussed by Lygeros, Pappas, and Sastry (Lygeros, Pappas, & Sastry, 1999), we consider the functional behavior (input output mapping) of a complex system, S, to be closely approximated by a *hybrid automaton*, *S*, which captures the logical and physical constraints on system evolution: $S = (X, V, Init, f, Inv, R)$ where

X is a finite collection of state variables. We assume

$X = (X_D \cup X_C)$ with X_D countable and

$X_C \in \Re^n$;

V is a finite collection of input variables. We assume

$V = (V_D \cup V_C)$ with V_D countable and $V_C \in \Re^n$;

$Init \subseteq X$ is a set of initial states;

$f: \ X \times V \to X_C$ is a vector field, assumed to be

globally Lipschitz in X_C and continuous in X_C ;

$Inv \subseteq X \times V$ is an invariant set;

$R : X \times V \to 2^X$ is a reset relation.

We refer to $x \in X$ as the state of S and to $v \in V$ as the input of S.

Associated with this model are rigorous definitions of continuous and discrete states and associated models of continuous behaviors and discrete behaviors and hybrid (combination of continuous and discrete) behaviors. These behaviors consist of continuous, discrete and hybrid trajectories from a set of initial states to a set of final states. The complete power of the hybrid modeling approach is not needed for each component (and may not be desireable!). For some (maybe most) of the components, a discrete model such as that used by Bell and La Padula is sufficient. Likewise, for some components, a continuous-system model is sufficient. The hybrid model is used when the future states of the composed system includes parameters of interest which exhibit both discrete and continuous behaviors (evolutions). We are convinced that for our particular problem space, the hybrid model is generally required for capturing the range of parameter values of interest for complex system evolution. Our problem space of interest in this paper is that which can adequately represent tactical-level military operations where success in humanitarian assistance/disaster recovery (HADR) operations requires reasoning about trustworthiness of information elements to be flowed between distributed information nodes in a manner which (1) increases the value of information available for goal-oriented decisions in

accordance with the intent of the commander taking into account that some of the information elements vary continuously with time and space, and (2) which complies with a command decision to share information. It is interesting to note that addressing item one above (flowing valued information) was a subject of discussion at the time the creators of the original Bell-La Padula model were working on their model (Bell D. E., 2005), (Landwehr, Heitmeyer, & Mclean, 1984), (Denning, 1976), at least in terms of seeking to analyze information security in terms of information flow. While this paper seeks to extend the framework of Bell and La Padula in terms of a formal treatment of general systems modeling and information sharing, we remark that the implementation details, in addition to following the Bell-La Padula extensions in terms of information security and sharing, will also be achieved as extensions to the current military messaging systems in terms of information flow between network nodes. As indicated by John McLean, there has long been considerable interest in fashioning the treatment of security in the same manner as Shannon had done for information theory by establishing the science for determining channel capacity (McLean, 1990). McLean's treatment of information flow considers bi-directional flow of information as preserving security for causal systems if the security state of the information object of interest is considered at different instances of time. However, McLean's treatment does not consider continuous values in time and space and also does not consider the case in which information value decays over time or distance from where it is most useful. Bell's review in 2005 of the Bell-La Padula model states: "Consideration of access modes led to the unexpected identification of a hard-to-name information flow property, the * -property. The relation ***W*** that conceptualized allowable changes of state was not constructive and was therefore insufficient for the analysis and formulation of core system calls that change the security state. (Bell D. E., 2005)" The * -property refers to the basic constraint of information flow across a security level in the Bell-La Padula model as allowing "no read-up, no-write-down" operations (Figure 1 and Figure 2 of Bell D.E., 2005). Thus, decision support tools available to commanders today continue to rely on security models which restrict analysis to parameters whose values are members of sets. This restriction does not enable reasoning about parameters of interest whose values change continuously. An example of situations in which reasoning about continuously-varying parameters is essential for mission success is provided in the appendix.

Flowing Information

As indicated by a key individual in implementing the current security features available in Java, Li Gong, "Fred Schneider, a key member of the Java Security Advisory Council, together with his PhD student at Cornell, Ulfar Erlingsson, proposed Inline Reference Monitors, which promised not only a mechanism to completely separate security policy from enforcement (via bytecode rewriting) but also a theoretical proof that the solution was extremely expressive – it is able to encode all enforceable policies (Gong, 2009)." The idea of incrementally adding security features as information flows through a system has also been investigated by Tse and Zdancewic: "In addition to allowing more expressive security policies, run-time principals enable the integration of language-based security mechanisms with other existing approaches such as Java stack inspection and public key infrastructures. We sketch an implementation of run-time principals via public keys such that principal delegation is verified by certificate chains (Tse & Zdancewic, 2007)." Incremental manipulation of signals over time is also an attribute of causal systems which are the category of systems considered by the controls community. The hybrid automaton modeling approach has been developed within the control community for analysis, design and

implementation of distributed (networked) control systems. The technology enables a more rigorous analysis of the Service Oriented Architecture (SOA) or middleware approach for distributed system development whereby applications use well-defined interfaces to access services from other local and distributed applications (the service or middleware) in order to enable achieving desired functionality.

By appropriate choice of the domain and range of the functional system (hybrid automaton) (and a set Z to represent outputs when necessary), one can closely represent some situation of particular interest and reach significant conclusions about that situation.

Secure Computer Systems

As well stated by Bell and La Padula and extensively developed since their paper, a large number of systems have been implemented which address the general problem of security in some form and to some extent. For some implementations of secure systems, privacy of data is the principal objective; in others, the prime objective is access control, and in others availability of system resources and/or capabilities may dominate tradeoff decisions between risk and functionality (Ross, Katzke, Johnson, Swanson, & Stoneburner, 2008). As was the case when Bell and La Padula made their contribution in 1973, for the security criteria which we shall establish, however, no existing system of which we are aware is adequate. That is, to our knowledge, no one has extended the Bell and La Padula model to either explicitly include compositions of continuous and discrete system components nor has anyone extended the model to include support for declaration of a “need to share” information.

We accept the Bell and La Padula definition of a secure computer system. That is, we mean one which satisfies some definition of "security" where our interest in security is in the usual military and governmental senses in which security relating to information elements is determined in terms of a range of security classifications (UNCLASSIFIED, CONFIDENTIAL, SECRET, TOP SECRET, …) for those information elements and also in terms of a user’s “need-to-know” those information elements. However, in addition to the above notion of a secure computer system, we add discussion of the “need to share” information elements. That is, our interest in sharing is in the usual military and governmental senses in which sharing decisions regarding information elements are made in terms of a range of security classifications (UNCLASSIFIED, CONFIDENTIAL, SECRET, TOP SECRET, …) for those information elements and also in terms of a commander’s decision of a “need-to-share” selected information elements with selected users and/or groups.

Also, while we shall investigate a bounded subset of the general problem of computer security for networked devices, the problem is chosen to be representative of the more general problem of improving our understanding (prediction) of the future states of sets of interdependent complex networks of infrastructures (James, Dodge, Graham, & St. Leger, 2009) and the populations which use them (Thompson, 2006). Indeed, the more general problem for operationally-significant information elements is to value the information relative to its trustworthiness and temporal and spatial relevance (James J. R., Thoughts on Information Operation Detection as a Nonlinear, Mixed-Signal Identification Problem: A Control Systems View, 2000; James & Mabry, Building Trustworthy Systems: Guided State Estimation as a Feasible Approach for Interpretation, Decision and Action Based on Sensor Data, 2004) (James & McClain, Tools and Techniques for Evaluating Control Architecture, 1999). That is, the issue of

valuation and sharing of battlespace data is directly tied to estimating trust among the participants in providing information artifacts concerning battlespace state as well as in estimating the temporal and spatial degradation of the relevance of the information between different points among the battlespace dimensions.

As in the case of the Bell-La Padula paper, our interest in this paper is in a bounded subset of the general certification problem (Ross, Katzke, Johnson, Swanson, & Stoneburner, 2008) which is to provide an approach to certify security status within a single computer (i.e. a single component of an information system comprised of a network of devices). That is, we seek to certify that a security compromise has not occurred within a given computer. In addition, we also seek to certify that a sharing compromise (failure to share elements of information which have been marked as "need to share") has not occurred within a given computer. As with Bell and La Padula, the entities with which we shall deal, then, are those appropriate for consideration on a single computer: applications, data, algorithms which control access to data, classifications of data elements and applications, the "need-to-know" status of computer entities, and the "need-to-share" status of computer entities.

Problems of Security

Absolute security is not known to be achievable with available science and technology. The approach taken by the National Institute of Standards and Technology (NIST) under the authority of the Federal Information Systems Management Act (FISMA) is to make explicit decisions regarding balancing functionality with risk and to certify systems for operation after putting acceptable controls in place to achieve the selected level of risk (Ross, Swanson, Stoneburner, Katzke, & Johnson, 2004). For military operations, there are situations, such as Humanitarian Assistance/Disaster Recovery (HADR) operations, where mission success requires that relevant operational information be shared with individuals and groups not normally among those with whom we share operational information. A widely-recognized operational shortfall is an inability to provide automated support to operational decisions to share information. One need is to provide automation support to small unit commanders to exercise military judgment and choose what information to share with whom and when to share the information. At the same time, provision of that support to automatically share relevant information need for mission success, must be implemented in such a fashion as to not lead to security compromises of information which remains in a "need to know" status.

Following Bell and La Padula, we consider a security compromise to be unauthorized access to information, where unauthorized means that an inappropriate clearance or a lack of need-to-know is involved in the access to the information. The approach taken by Bell and La Padula (Bell & LaPadula, 1973) solved the problem within a single computing system concerning how to guarantee that unauthorized access (by an application) to information does not occur.

Summary

In section I we have provided a brief overview of general systems theory and discussed the need to protect (prevent security compromise) of information which needs to be protected and share (prevent sharing compromise) of information which needs to be shared.

SECTION II FOUNDATIONS OF A MATHEMATICAL MODEL FOR SECURING AND SHARING INFORMATION

Elements of the Model

Table I below is a modification of Table II in the original Bell and La Padula model (Bell & LaPadula, 1973) to include the capability of modeling compositions of continuous and discrete system components (the *objects* in the model) and also to include the ability to reason about a command decision to share information (the *signal* G and the *functions* f_5 and f_6).

We add an assumption regarding the need to share information to the set of Bell-La Padula assumptions concerning the computer system of interest. That is, we assume:

1. the system has multiple users operating concurrently on a common data base ,
2. the system operates with multi-level classifications for both users and data,
3. the system has need-to-know categories associated with both users and data, and
4. the system has need-to-share categories associated with both users and data.

Bell and La Padula discussed their model in terms of sets, elements of the sets, and an interpretation of the elements of the sets. Instead we follow the more general interpretation of the information flow problem as being associated with input and output *signals* (some signals may be sets) and *systems*, which transform signals.

Table I

Elements of the Model

Signal	Elements	Semantics
S	$\{S_1, S_2, \cdots, S_n\}$	*Subjects;* processes, programs in execution
O	$\{O_1, O_2, \cdots, O_m\}$	*objects*; data, files, programs, subjects. Note: as in Section I, an object can have either (or both) continuous and discrete attributes - $O_m = (O_{mD} \cup O_{mC})$ with O_{mD} countable and $O_{mC} \in \Re^n$
C	$\{C_1, C_2, \cdots, C_q\}$ $\{C_1 > C_2 \cdots > C_q\}$	*classifications*; clearance level of a subject, classification of an object
K	$\{K_1, K_2, \cdots, K_r\}$	*need-to-know categories;* project numbers, access privileges
G	$\{G_1, G_2, \cdots, G_i\}$	*need-to-share categories*; project numbers, access privileges,
A	$\{A_1, A_2, \cdots, A_p\}$	*access attributes*; read, write, copy, append, owner, control, …
R	$\{R_1, R_2, \cdots, R_u\}$	*requests*; inputs, commands, requests for

		access to objects by subjects
D	$\{D_1, D_2, \cdots, D_v\}$	*decisions*; outputs, answers, "yes", "no", "error"
T	$\{1, 2 \cdots, t, \cdots\}$	*indices*; elements of the time set; identification of discrete moments; an element *t* is an index to request and decision sequences
$P\alpha$	All subsets of α	*power set of* α
α^{β}	All functions from the set β To the set α	
$\alpha \times \beta$	$\{(a, b): a \in \alpha, b \in \beta\}$	Cartesian product of the sets α and β
F	$C^S \times C^O \times (PK)^S \times (PK)^O \times (PG)^S \times (PG)^O$ An arbitrary element of F is written $f = (f_1, f_2, f_3, f_4, f_5, f_6)$	*classification/need-to-know/need-to-share vectors;* f_1: subject-classification function f_2: object-classification function f_3: subject-need-to-know function f_4: object-need-to-know function f_5: subject-need-to-share function f_6: object-need-to-share function
X	R^T An arbitrary element of X Is written x	*request sequences*
Y	D^T An arbitrary element of Y Is written y	*decision sequences*
M	$\{M_1, M_2, \cdots M_c\}$ $c = nm2^p$ An element M_k of M is an $n \times m$ matrix with entries from PA ; the (i, j)-entry of M_k shows S_i's access Attributes relative to O_j	*access matrices*
V	$P(S \times O) \times M \times F$	*states*
Z	V^T an arbitrary element of Z is written z; $z_t \in z$ is the t-th state in the state sequence z	*state sequences*

States of the System

We again follow the Bell-La Padula model of system state (with the extension to consider both continuous- and discrete-valued variables as state variables) as being sufficient for determining both the security of a given computer system and also the compliance of the system with a commander's decision to share information.

A state $v \in V$ is a 3-tuple (b,M,f) where

$b \in P(S \times O)$, indicating which subjects have access to which objects in the state v;

$M \in M$ Indicating the entries of the access matrix in the state v;

and

$f \in F$, indicating the clearance level of all subjects, the classification level of all objects, the need-to-know associated with all subjects and objects in the state v, and the need-to-share associated with all subjects and objects in the state v.

We comment that a major difference in achieving automated assistance of a need to share compared to a need-to-know is that command decisions to share will associate individuals (subjects) not normally authorized access with specific information (objects). Thus, we assume (and any implementation must ensure), that the set of information objects annotated with need-to-know attributes and the set of information objects annotated with need-to-share attributes are disjoint sets.

State-Transition Relation

Let $W \subseteq R \times D \times V \times V$ (i.e the Cartesian product of the sets of requests, decisions, current state and prior state). The system $\Sigma(R, D, W, z_0) \subseteq \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$ is defined by

$(x, y, z) \in \Sigma(R, D, W, z_0)$ if and only if $(x_t, y_t, z_t, z_{t-1}) \in W$ for each $t \in T$,

where z_0 is a specified initial state usually of the form (ϕ,M,f), where ϕ denotes the empty set.

As discussed in Section I, in general W is a *hybrid automaton*: The system W can be considered as a mapping which transformsthe input to the output:

$$W = (X, V, Init, f, Inv, R)$$

For purposes of proving desired security and sharing properties, W has been defined as a relation. It can be specialized to be a function, although this is not necessary for the development herein. When considering design questions, however,W will be a function, specifying next-state and next-output. W should be considered intuitively as embodying the rules of operation by which the system in any given state determines the current decision for a current request and moves into a next state. Concerning the distinction between relations, which in this paper are declarative in nature versus functions, which may also be procedural in nature, it is information to be aware of the distinction between declarative and imperative functions (pages 56-57 of (Lee & Varaiya, 2002)). As pointed out by Lee and Varaiya, a declarative definition of the square root as:
$SquareRoot(x)$ *is the unique value of* $y \in \mathbb{R}eals$ *such that* $y^2 = x$ does not tell us how to calculate the square root. However, an imperative, or procedural implementation of the function $SquareRoot(x)$ would yield a value, ŷ, whose square, $\hat{y}^2$, might not equal x but would be approximately

equal to x. Thus, it is important to consider the distinction between logical assertions of relations which are provably correct and procedural implementations of declarative relations which may yield approximations of variables of interest. This will usually be the case for continuous variables whose values change over time and space since the dependent variable values change with infinitesimal temporal and spatial variations yet we make decisions over intervals of time and space for operational decisions.

Summary

In this section we have modified the Bell-La Padula model for analyzing security compromises (the basis for multi-level security systems) to add a capability to analyze compositions of discrete and continuous models and also added a capability to analyze information sharing compromises (failure to share information declared sharable by a commander).

SECTION III A FUNDAMENTAL RESULT

Compromise, Security of Information and Sharing of Information

Following Bell-La Padula, we define a compromise state as follows: $v = (b, M, F) \in V$ is a **compromise state** (**security or sharing compromise**) if there is an ordered pair $(S, O) \in b$ such that

(i) $f_l(S) < f_2(O)$, a security compromise, or

(ii) $f_3(S) \not\supseteq f_4(O)$, a security compromise, or

(iii)) $f_5(S) \not\supseteq f_6(O)$, a sharing compromise.

In other words, v is a compromise if the current allocation of objects to subjects (b) includes an assignment ((S,O)) with at least one of three undesirable characteristics:

(i') S's clearance is lower than O's classification;

(ii') S does not have some need-to-know category that is assigned to O, or

(iii) S does not have some need-to-share category that is assigned to O.

In order to make later discussions and arguments a little more succinct, we shall define a security and sharing condition. $(S, O) \in S \times O$ satisfies the **security and sharing condition relative to f**

(SC rel f) if

(iv) $f_l(S) \geq f_2(O)$, and

(v) $f_3(S) \supseteq f_4(O)$, and

(vi)) $f_5(S) \supseteq f_6(O)$.

A state $v = (b, M, F) \in V$ is a secure state if each $(S, O) \in b$ satisfies SC rel f.

In the remainder of this section we follow the development of Bell and La Padula but add the discussion needed to consider enabling a commander's decision to share some elements of information. As in Bell and La Padula's development, we present a table which summarizes some constraints on the system subjects and objects and then prove a basic security and sharing theorem.

Proposition: v $\in$ V is not a secure state iff v is a compromise.

A state sequence z $\in$ Z **has a compromise** if z_t is a compromise for some t $\in$ T. z is a secure state sequence if z_t is a secure state for each t $\in$ T. We shall call (x,y,z) $\in \Sigma$ (R,D,W,z_0) an **appearance** of the system. (x,y,z) $\in$ (R,D,W,z_0) is a secure appearance if z is a secure state sequence. The appearance (x,y,z) has a compromise if z has a compromise.

Σ (R,D,W,z_0) is a secure system if every appearance of Σ (R,D,W,z_0) is secure.

Σ (R,D,W,z_0) has a compromise if any appearance of Σ (R,D,W,z_0) has a compromise.

Proposition: z $\in$ Z is not secure iff z has a compromise.

Proposition: Σ (R,D,W,z_0) is not secure iff Σ (R,D,W,z_0) has a compromise.

Constraints

We make constraints (assumptions), as shown in Table III, which reflect a subset of requirements (actually a lack of requirements) to be imposed on the system. In Section IV we shall change some of these assumptions and observe the effect on the system.

Table II

Initial Requirements

	REQUIREMENTS	
	RAISE?	**LOWER?**
SUBJECT CLEARANCE	**NO**	**NO**
OBJECT CLASSIFICATION	**NO**	**NO**
	INCREASE?	**DECREASE?**
SUBJECT NEEDS-TO-KNOW	**NO**	**NO**
OBJECT NEEDS-TO-KNOW	**NO**	**NO**
	INCREASE?	**DECREASE?**
SUBJECT NEEDS-TO-SHARE	**NO**	**NO**
OBJECT NEEDS-TO-SHARE	**NO**	**NO**

We remark that the novel contribution of this paper to the previously-established treatment of multi-level security systems lies in the observation that, if we ensure that "classified objects" and "sharable

objects" are disjoint sets, then the same framework for treatment of classified objects can be used for shareable objects. Thus, the constraints of Table II, in effect, say that "no" is the answer to each of the questions "Is there a requirement to (raise / lower / increase / decrease) a (subject's / object's) (classification or clearance / needs-to-know / needs-to-share)?".

Basic Security and Sharing Theorem

Basic Security Theorem:

Let $W_t \subseteq R_t \times D_t \times V_t \times V_{t-1}$ be any relation such that $(R_i,D_j,(b^*,M^*,f^*),(b,M,f) \in W$

implies

(i) $f = f^*$ and

(ii) every $(S,O) \in b^* - b$ satisfies SC rel f^*.

Then: Σ (R,D,W,z) is a secure system for any secure state z .

Proof: Let $z_0 = (b,M,f)$ be secure. Pick $(x,y,z) \in \Sigma$ (R,D,W,z) and write $z_t = (b^{(t)},M^{(t)},f^{(t)})$ for each $t \in T$.

z_1 is a *secure state*. $(x_1,y_1,z_1,z) \in W$. Thus by (i), $f^{(1)} = f$. By (ii), every (S,O) in $b^{(1)} - b$ satisfies SC rel $f^{(1)}$. Since z is secure, every $(S,O) \in b$ satisfies SC rel f. Since $f = f^{(l)}$, every $(S,O) \in b^{(1)}$ satisfies SC rel $f^{(1)}$. That is z_1 is secure.

If z_{t-1} is secure, z_t is secure. $(x_t,y_t,z_t,z_{t-1}) \in W$. Thus by (i), $f^{(t)} = f^{(t-1)}$. By (ii), every (S,O) in $b^{(t)} - b^{(t-1)}$ satisfies SC rel $f^{(t)}$. Since z_{t-1} is secure, every $(S,O) \in b^{(t-1)}$ satisfies SC rel $f^{(t-1)}$. Since $f^{(t)} = f^{(t-1)}$, every (S,O) $\in$ b(t) satisfies SC rel f(t). That is, z_t is secure. By induction, z is secure so that (x,y,z) is a secure appearance. (x,y,z) being arbitrary, Σ (R,D,W,z_0) is secure.

Summary

In this section we have applied the mathematical model of Section II to the modeling of a secure computer system which also supports sharing information with coalition partners and non-government agencies. We have defined a secure system precisely, through the definitions of security and sharing compromises, and have given a rule of operation, W, which we have shown guarantees that the system is secure in its operation while also sharing information as authorized by a commander.

SECTION IV CONCLUSION

Introduction

In Section I we discussed the motivation and basis for this paper. We pointed out advancement of general system theory since the time of the original Bell-La Padula model and mentioned the need to support access to information based upon a commander's decision to share the information.

Subsequently, we extended the Bell-La Padula mathematical model of for study of secure computer systems, to include considerations of compositions of logical and continuous system components as well as considerations of meeting requirements for sharing information and the notion of a ***sharing compromise***.

We then applied the extended model, under a given set of assumptions, to the question of security of information and sharing of information (security compromise and sharing compromise). We gave a rule by which, for the assumptions given, the system would remain secure in its operation for information requiring a need-to-know while also enabling sharing of information in accordance with a commander's declaration of a need-to-share.

As in the case for the original Bell-La Padula security result, an important point for the security and sharing result is that the proof did not depend on the choice of elements for the access attributes(the set A). This means that any access set is acceptable and any access matrix is acceptable. Stated differently, the proof process has shown that, under the given assumptions, security of the system is independent of the access matrix and the rules (if any) by which the access matrix is changed.

Thus, to the extent that access can be made arbitrarily difficult, we have modeled the system in such a manner that complying with the model restrictions may result in a system which is not of practical use. This section will address some of the specific questions to be considered if a viable system is to be developed from the extended Bell-La Padula model.

Problem Reformulation

We seek to address problems which relate to compositions of discrete (logical) and continuous (physical) models whose behaviors approximate those of complex systems of interest. Following Bell and La Padula, we will first change the requirements listed in Table II and derive a result related to sharing information which has been declared as shareable while maintaining security of information which remains categorized as "need-to-know". from the changed assumptions. We will then discuss criteria to be met by the access control mechanisms in order to maintain the declared constraints on the disjoint sets of "need-to-know" and "need-to-share" information elements.

Table III

Requirements

	REQUIREMENTS	
	RAISE?	**LOWER?**
SUBJECT CLEARANCE	**YES**	**NO**
OBJECT CLASSIFICATION	**NO**	**YES**
	INCREASE?	**DECREASE?**
SUBJECT NEEDS-TO-KNOW	**YES**	**NO**
OBJECT NEEDS-TO-KNOW	**NO**	**YES**
	INCREASE?	**DECREASE?**
SUBJECT NEEDS-TO-SHARE	**YES**	**NO**
OBJECT NEEDS-TO-SHARE	**NO**	**YES**

Basic Security Theorem (revised with sharing):

Let $W_t \subseteq R_t \times D_t \times V_t \times V_{t-1}$ be any relation such that $(R_i,D_j,(b^*,M^*,f^*),(b,M,f)) \in W$

implies

(i) $f^*_1(S) \in f_1(S)$ for each $S \in S$,

$f^*_2(O) \in f_2(O)$ for each $O \in O$,

$f^*_3(S) \in f_3(S)$ for each $S \in S$,

$f^*_4(O) \in f_4(O)$ for each $O \in O$,

$f^*_5(S) \in f_5(S)$ for each $S \in S$,

$f^*_6(O) \in f_6(O)$ for each $O \in O$ and

(ii) every (S,O) $f^*_3(S) \in f_3(S)$ for each $S \in S$,

$f^*_4(O) \in f_4(O)$ for each $O \in O$ b^* – b satisfies SC rel f^*,

every (S,O) $f^*_5(S) \in b^*$ – b satisfies SC rel f^*.

Then $\Sigma(R,D,W,z_0)$ is a secure system for any secure state z_0.

Proof: Let $z_0 = (b,M,f)$ be secure. Pick $(x,y,z) \in \Sigma(R,D,W,z)$ and write

$z_t = (b^{(t)}, M^{(t)}, f^{(t)})$ for each $t \in T$.

z_1 is a secure state. $(x_1,y_1,z_1,z_0) \in$ W. By (ii), every (S,O) in $b^{(1)} - b$ satisfies SC rel $f^{(1)}$. Since z is secure, every (S,O) in b satisfies SC rel f; that is, $f_1(S) \geq f_2(O)$, $f_3(S) \supseteq f_4(O)$, and $f_5(S) \supseteq f_6(O)$. By (i), we have, for each (S,O) in $b^{(1)} - (b^{(1)} - b)$,

$f_1^{(1)}(S) \geq f_1(S) \geq f_2(O) \geq f_2^{(1)}(O)$,

$f_3^{(1)}(S) \supseteq f_3(S) \supseteq f_4(O) \supseteq f_4^{(1)}(O)$, and

$f_5^{(1)}(S) \supseteq f_5(S) \supseteq f_6(O) \supseteq f_6^{(1)}(O)$

so that each (S,O) in $b^{(1)}$ satisfies SC rel $f^{(1)}$. That is, z_1 is secure.

If z_{t-1} is secure, then z_t is secure. $(x_t,y_t,z_t,z_{t-1}) \in$ W. By (ii), every (S,O) in $b^{(t)} - b^{(t-1)}$ satisfies SC rel $f^{(t)}$. Since Z_{t-1} is secure, every (S,O) in $b^{(t-1)}$ satisfies SC rel $f^{(t-1)}$; that is,

$f_1^{(t-1)}(S) \geq f_2^{(t-1)}(O)$,

$f_3^{(t-1)}(S) \supseteq f_4^{(t-1)}(O)$, and

$f_5^{(t-1)}(S) \supseteq f_6^{(t-1)}(O)$.

By (i), we have for each (S,O) in $b^{(t)} - (b^{(t)} - b^{(t-1)})$,

$f_1^{(t)}(S) \geq f_1^{(t-1)}(S) \geq f_2^{(t-1)}(O) \geq f_2^{(t)}(O)$,

$f_3^{(t)}(S) \supseteq f_3^{(t-1)}(S) \supseteq f_4^{(t-1)}(O) \supseteq f_4^{(t)}(O)$,and

$f_5^{(t)}(S) \supseteq f_5^{(t-1)}(S) \supseteq f_6^{(t-1)}(O) \supseteq f_6^{(t)}(O)$, so that each (S,O) in b(t) satisfies SC rel $f^{(1)}$.

That is, z_t is secure. By induction, z is secure so that (x,y,z) is a secure appearance. (x,y,z) being arbitrary, Σ (R,D,W,z_0) is secure.

The revised theorem just proved indicates that dynamic

(i) raising of subject clearance;

(ii) lowering of object classification;

(iii) increasing of subject needs-to-know;

(iv) decreasing of object needs-to-know

(v) increasing of subject needs-to-share; and

(vi) decreasing of object needs-to-share

can be provided in the system without security compromise or sharing compromise. As with the original Bell-La Padula result, however, the proof is independent of what is happening in the access matrix. We will discuss the implications of the changes in the general systems model, the change in explicitly considering a security category devoted to sharing information, and consider the implications for flowing valued information via extensions to the military messaging system as the subject of the next section.

Access Control

In Section I we provided a hybrid system model as the most general model we will consider for this paper. The notation of sections II and III followed the notation of Bell and La Padula for purposes of extending the formal logic associated with defining and analyzing system security in terms of subject and object classification, need-to-know, and need-to-share. That is, sections II and III deal only with set-based approximations of complex systems (and we assume that sets of "need-to-know" objects and "need-to-share" objects are disjoint sets). For the issue of ***access control*** of information representing the *state of a complex system*, we revisit the hybrid system model and discuss issues associated with maintaining estimates of continuously-varying parameters associated with physical systems while making logical approximations at appropriate instants of time. For this purpose, we repeat the hybrid system model below.

We consider the functional behavior (input output mapping) of a complex system, S, to be closely approximated by a *hybrid automaton*, *S*, which captures the logical and physical constraints on system evolution: $S = (X, V, Init, f, Inv, R)$ where

X is a finite collection of state variables. We assume

$X = (X_D \cup X_C)$ with X_D countable and

$X_C \in \Re^n$;

V is a finite collection of input variables. We assume

$V = (V_D \cup V_C)$ with V_D countable and $V_C \in \Re^n$;

$Init \subseteq X$ is a set of initial states;

$f: \ X \times V \to X_C$ is a vector field, assumed to be globally Lipschitz in X_C and continuous in X_C ;

$Inv \subseteq X \times V$ is an invariant set;

$R : X \times V \to 2^X$ is a reset relation.

We refer to $x \in X$ as the state of S and to $v \in V$ as the input of S. A fundamental issue in building approximate models is the tradeoff between model accuracy and model complexity. While Sections II

and III provide a formal result, they are unfortunately limited to problems so simple (i.e. set-based models only) to be of limited use in building realistic decision support tools since every practical problem exists under continuous time and space constraints as well as continuous constraints on physical system evolution over time while set-based models ignore continuous constraints. Our challenge in applying the theory of Sections II and III to access information stored on computing and communications networks relating to real events and real physical objects is then to deal with issues of trust and viability. Trust constraints are associated with ensuring that the information being analyzed meets standards of confidentiality and integrity as well as metrics associated with trusting the original source of the information (real sensors have error models for building metrics for approximating reality and real people have trust issues which change over time and should be captured). Viability constraints are associated with ensuring that the composition of system logical components and physical components represents a physically-realizable system (many models are not physically realizable) (Aubin, 1991) (Deshpande & Varaiya, 1995). The issue of viability of existence of a solution to the composed problem is dealt with in the controls community by ensuring that "sufficiently close" approximations are available from the composed components used to generate the feedback control laws (digital and/or analog signal filters) used to move the current system state to some desired future state. Models used for this purpose must be grounded in viable solutions to approximating (predicting) future state from current state and current input (the system identification problem). Validity of such models is only assured for time-invariant (stationary) systems or, for slowly time-varying systems, for the time frame in which the system parameters have not changed "significantly". Future reports will deal with issues of trust and viability for scenarios of interest. The current scenario being studied for information sharing is contained in the appendix for a platoon-level Humanitarian Assistance-Disaster Recovery (HADR) operation.

Understanding the HADR problem (in the sense of building models which predict future values/states of parameters of interest) at the platoon level, exhibits the full range of complex systems analysis issues we are interested in addressing. This scenario of an earthquake in Afghanistan, requires a unit to move to a new location, coordinate with local leaders and non-government agencies to provide disaster relief, and provide local security in the area of operations. Given that the local government and economy has been affected and the disaster has occurred in an area being contested between the Government of the Islamic Republic of Afghanistan (GIROA) and the Taliban, then the scenario places the leader in the position of considering political, military, economic, social, infrastructure, and information (PMESII) outcomes related to whatever decisions are taken in attempting to assist in providing disaster relief. Furthermore, available analytical techniques for determining the human terrain (social and cultural aspects of social network analysis studies) are just now being investigated. While predictive models of human terrain are not expected to be available for some time, the modeling approach described above supports the broad range of hybrid system modeling techniques that have been used in the past for automatic control system identification (i.e. data analysis to determine the correct model type and assign model parameters to predict future system state), system control law design (i.e. feedback filter design to cause the closed-loop current system state to move to a desired future system state in response to inputs over time), and control law implementation and update (for adaptive control systems). These modeling techniques include state-space model (linear or nonlinear, statistical or

deterministic, stationary or non-stationary, frequency/spectral-domain or time-domain) models as well as wavelet models. The basic issues in each model effort includes matching model parameters to actual system data and determining the range of parameter values for which the model behavior is "close enough" to actual system behavior. For the range of PMESII problems being addressed by junior leaders on a daily basis, we are as yet unable to achieve acceptable ("close-enough") model performance.

Data Base Sharing

The original Bell-La Padula paper (Bell & LaPadula, 1973) discussed some of the issues associated with implementing the security results on a shared database since, at the time the paper was published, this was the method of implementation of sharing information. We observe here that implementations for sharing data over networks of sensing, communicating, and computing devices have grown exponentially over the intervening 36 years and will probably continue to grow for at least the next 15 years as Moore's Law continues to fuel Information Age expansions. Also, Bell subsequently observed that the multi-level security result obtained from the original paper applies to networks of devices (Bell D. E., 2005). Thus, our efforts to apply the extensions developed in Sections II and III above will focus on the trust and viability issues mentioned above. In that regard, we expect to extend the capabilities of the Android smart phone to prototype information sharing technology implementations.

We intend to initially prototype automation support which implements the logical constraints imposed on information transfer by the results of this paper. The information transfer will be appropriate for that authorized by a commander in executing a HADR operation and will simulate a commander creating a text string and authorizing the string to be shared with designated individuals over a designated area for a designated time. We will move the string (and subsequent responses to the string) across security boundaries among network nodes in accordance with a declaration of a "need-to-share" while maintaining constraints on other information whose movement is constrained by a "need-to-know".

Summary

We have provided extensions to the Bell and La Padula model which lay the groundwork for: (1) building more accurate models of the complex operational environments of today and tomorrow, and (2) providing automation support for a commander's decision to share information while simultaneously maintaining the security of information which must not be compromised.

Indeed, concerning the issue of sharing information, we take the position that at least one category of data, metadata, should be shared continuously with everyone, all of the time, and in every area (e.g. information metadata for all categories of information should be shared with all categories of users).

REFERENCES

Aubin, J. P. (1991). *Viability Theory.* Cambridge, MA: Birkhauser Boston Inc.

BAST, Board on Army Science and Technology. (2005). *Network Science.* Washington DC: National Academy Press.

Bell, D. E. (2005). Looking Back at the Bell-La Padula Model. *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC 2005)* (pp. 337-351). IEEE Xplore.

Bell, D. E., & LaPadula, L. (1973). Secure Computer Systems: Mathematical Foundations - Volume I. *Mitre Technical Report 2547* .

Denning, D. E. (1976). A lattice model of secure information flow. *Communications of the ACM, Volume 19, Number 5, May 1976* , 236-243.

Deshpande, A., & Varaiya, P. (1995). Viable Control of Hybrid Systems. In P. Antsaklis, W. Kohn, A. Nerode, & S. Sastry, *Lecture Notes In Computer Science; Vol. 999, Hybrid Systems II* (pp. 128-147). London, UK: Springer-Verlag.

Foley, S. (1989). A model for secure information flow. *Proceedings, 1989 Symposium on Security and Privacy* (pp. 248-258). IEEE.

Gong, L. (2009). Java Security: A Ten Year Retrospective. *Proceedings, 2009 Annual Computer Security Applications Conference.* Honolulu, HI: Conference Publishing Services.

Honda, K., & Yoshida, N. (2007). A uniform type structure for secure information flow. *ACM ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 29, Issue 6* .

James, J. R. (2000). Thoughts on Information Operation Detection as a Nonlinear, Mixed-Signal Identification Problem: A Control Systems View. *Proceedings, 2000 IEEE Symposium on CACSD* (p. 6). Anchorage, Alaska: IEEE.

James, J. R., & Mabry, F. (2004). Building Trustworthy Systems: Guided State Estimation as a Feasible Approach for Interpretation, Decision and Action Based on Sensor Data. *37th Hawaii International Conference on System Science* (p. 6). Kohola Coast, Hawaii: HICSS.

James, J. R., & McClain, R. (1999). Tools and Techniques for Evaluating Control Architecture. *Proceedings,10th IEEE International Symposium on CACSD* (p. 6). Kohala Coast, Hawaii: IEEE.

James, J., Dodge, R., Graham, J., & St. Leger, A. (2009). *Gap Analysis for Survivable PCS: Final Report.* I3P, http://www.thei3p.org/publications/ResearchReport14.pdf.

Landwehr, C. E., Heitmeyer, C. L., & Mclean, J. (1984). A Security Model for Military Message Systems. *ACM Transactions on Computer Systems, Vol. 2, No. 3, August 1984* , 198-222.

Lee, E. A., & Varaiya, P. (2000). Introducing Signals and Systems, The Berkeley Approiach. *First Signal Processing Education Workshop.* SPE.

Lee, E., & Varaiya, P. (2002). *Structure and Interpretation of Signals and Systems.* Addison-Wesley.

Lygeros, J., Pappas, G., & Sastry, S. (1999). An Introduction to Hybrid System Modeling, Analysis and Control. *Preprints of the First Nonlinear Control Network Pedagogical School*, (pp. 307-329). Athens, Greece.

McLean, J. (1990). Security Models and Information Flow. *1990 IEEE Symposium on Security and Privacy.* Oakland, : IEEE Press.

Ross, R., Katzke, S., Johnson, A., Swanson, M., & Stoneburner, G. (2008). *NIST SP800-39, Managing Risk from Information Systems An Organizational Perspective.* Gaithersberg, MD: NIST, http://csrc.nist.gov/publications/drafts/800-39/SP800-39-spd-sz.pdf.

Ross, R., Swanson, M., Stoneburner, G., Katzke, S., & Johnson, A. (2004). *Guide for the Security Certification and Accreditation of Federal Information Systems.* Gaithersberg, MD: NIST Special Publication 800-37, http://csrc.nist.gov/publications/nistpubs/800-37/SP800-37-final.pdf.

Thompson, K. R. (2006). "GENERAL SYSTEM" DEFINED FOR PREDICTIVE TECHNOLOGIES OF A-GSBT (AXIOMATIC-GENERAL SYSTEMS BEHAVIORAL THEORY). *IIGSS Academic Publisher: Scientific Inquiry, vol. 7, No. 1,* 10.

Tse, S., & Zdancewic, S. (2007). Run-Time Principals in Information-Flow Type Systems. *ACM Transactions on Programming Languages and Systems, Vol. 30,No. 1, Article 6,* .

Appendix A: Earthquake Scenario

Humanitarian Assistance/Disaster Recovery (HADR) Operation

Earthquake Scenario (With Information Sharing)

Operations Order

1. **Situation** Two earthquakes occurred early this morning in Nangahar Province North East of Kabul, Afghanistan. Approximately 200 families are homeless and 20 people have been killed. There is a need for food, temporary shelter, water, medical aid, and search teams. Our platoon will depart in four hours to provide HADR support to the Provincial Reconstruction Team (PRT) of Nangahar Province. We have been assigned to ?? Village, ?? District, Nangarhar Province, Afghanistan.

2. **Mission** Provide HADR support to the people of ?? Village, ?? District, Nangarhar Province from H Hour on D Day until relieved (relief in place expected in 72 hours).

3. **Execution** Phase 1 is movement to the village area. Phase 2 is securing the village and searching for additional victims. Phase 3 is providing recovery assistance.

   During Phase I the Platoon Leader will lead an advance party to the village while the Platoon Sergeant moves the platoon into the village area. Phase 1 ends when the Platoon Leader briefs the platoon on the advance party results and assigns security and search areas.

   During Phase 2 first and second squads will provide security of the village area while third and fourth squads assist villagers in searching for survivors and victims. As the search is underway the Platoon Sergeant will lead security activities and the Platoon Leader will lead search activities and coordinate with local leaders, government and non-government agencies concerning feasible recovery assistance activities. Phase 2 ends when the village leaders indicate all inhabitants are accounted for or that the search for survivors is completed. The Platoon Leader will then assign recovery activities to each

squad based upon the results of coordination with local leaders, government and non-government agencies.

During Phase 3 third and fourth squads will provide security of the village area while first and second squads execute assigned recovery activities. Phase 3 will continue until relieved.

4. **Command and Signal** Current command relations are unchanged. Current CEOI will remain in effect. There will be United Nations (UN) relief organization personnel in the area as well as other national and international relief organization personnel who may or may not be members of the UN Assistance Mission of Afghanistan (UNAMA).

5. **Administration and Logistics** Carry basic combat load, extra water and three days of rations. Battalion will be setting up a temporary combat outpost in the vicinity within 24 hours.

Discussion

Currently there is ***no underlying science*** for automatically moving valued information from one network node to another in accordance with a commander's ***intent for conduct of an operation***. Furthermore, there is ***no underlying science*** for automatically moving information across a security boundary in accordance with a commander's declaration of ***intent to share the information***. Thus networks of forces in Afghanistan and elsewhere are inundated with information which may not be valuable to the current operation and commanders are constrained to manually share information face-to-face with coalition partners who are unable or unwilling to obtain security clearances to work on available networks.

The Flowing Valued Information project will result in ***flowing valued information among network nodes to increase the value of shared information***. Consider what might be possible with information sharing technologies which will automatically flow valued information and will also execute movement of information across network nodes in accordance with a commander's declaration of intent to share information with an individual or with a group. The following assertions are made concerning possible results for HADR operations such as the one summarized in the above Operations Order (OPORD).

The flowing Valued Information project seeks to enable dynamic alteration of the movement of information across network nodes in response to both (1) the relative utility of the information to

meeting commander's intent and also (2) the expressed intent (perhaps recently expressed) of sharing information with a particular group and/or individual. In order to achieve these goals, a mechanism must be created to dynamically change the information being flowed across network nodes and to do so at multiple time scales and multiple distance scales.

Command Intent:

The Military Decision Making Process (MDMP) is a structured approach for generating alternative courses of action, selecting a course of action, generating written operation orders for the selected course of action, and executing the selected course of action. ***Command Intent*** for the selected course of action is not explicitly captured in the written documents associated with an OPORD but is expected to be understood and achieved in executing the selected course of action. Command intent is summarized in the *mission* and *execution* sections of the OPORD but may also include elements not included in these sections. Subordinate commanders are expected to understand ***command intent*** from development of the selected course of action during the planning process and to exercise ***military judgment*** in dynamically altering the details of the plan during execution in order to meet ***command intent***.

The OPORD sketched out above would normally not be written since the echelons below battalion level normally do not follow the MDMP and normally do not issue written OPORDs. Instead ***Troop Leading Procedures*** are followed in which the same basic decision flow of considering alternative courses is considered, a course of action is selected, and a verbal OPORD is created and delivered to subordinate commanders. However at both the higher echelons of tactical-level operations (Brigade and Battalion) and the lower echelons of tactical operations (Company, Platoon and Squad) ***command intent*** is developed and conveyed to other leaders during the decision-making process and subordinate commanders are expected to dynamically change the plan during the execution process to meet the intent of the commander.

Thus, in the mathematical sense, command intent is the ***system invariant*** around which other parameters vary during the execution process and information flow should be optimized to make information available at different communication and computing nodes in the unit network according to the role to be played by the unit associated with that node in meeting the ***intent of the commander***.

Discrete and Continuous Variables

Moreover, the kinds of parameters which may dynamically vary during the conduct of the operation consist of both discrete and continuous variables. For instance, soldiers are trained to make and continuously update a visualization of the battlespace (i.e. the ***state*** of the operational environment) which considers Mission, Enemy, Terrain and weather, Troops and support available, Time available, and Civil considerations (METT-TC). Variables which define the weather and vary continuously include air density, wind velocity, temperature, humidity, rainfall, and illumination. Terrain and time available vary continuously. The physical parameters which

affect the accuracy of every engagement process vary continuously (weapon and target position (latitude, longitude, and altitude), three dimensions of velocity, three dimensions of acceleration, projectile flight characteristics, atmospheric dynamics, projectile charge explosive characteristics,…). Commanders at higher levels may identify and specifically task intelligence personnel to identify values for the commander's critical information requirements (CCIR). CCIR are usually discrete-valued variables such as enemy strength, enemy location, and enemy intent. However, The Army's Field Manual for Operations also indicates that "METT-TC emphasizes the operational environment's human aspects. This emphasis is most obvious in civil considerations, but it affects the other METT-TC variables as well. Incorporating human factors into mission analysis requires critical thinking, collaboration, continuous learning, and adaptation. It also requires analyzing local and regional perceptions. Many factors influence perceptions of the enemy, adversaries, supporters, and neutrals. These include—

- Language.
- Culture.
- Geography.
- History.
- Education.
- Beliefs.
- Perceived objectives and motivation.
- Communications media.
- Personal experience. "

Need-to-Know and Need-to-Share

While some of the data on the tactical intranet (such as a written estimate of METT-TC for an upcoming operations) might be considered for sharing by a commander, the scenario in question provides an example of the case facing many commanders in which mission success requires asking questions of local leaders and non-government organizations (NGOs) and receiving answers. Consider the fact that mission success for the Platoon Leader, Platoon Sergeant, and four Squad Leaders of the platoon responding to the earthquake requires that they understand the tactical situation (e.g. the METT-TC estimate which is in the "need-to-know" information category) and prepare as best they can to help the village deal with the disaster. Questions which need to be answered immediately (and are in a "need-to-share" category) include:

- How many people are dead?
- How many people are missing?
- Is shelter available for those whose homes are destroyed?
- How much bedding and clothing are needed?

- Is the road network functional?
- Are the waterworks functional?
- Is electricity available?
- How many people need to be fed?

In addition, consider that the Platoon Leader, Platoon Sergeant and Squad Leaders might have the following information sharing needs for the different phases mentioned in the OPORD:

Phase 1 (movement into the operational area) – need to share location and activity data with

- Doctors Without Borders
- Local leaders
- UNAMA
- PRT (may not have TiGR)
- Brigade HTT (automatic)

Commander's critical information requirements include:

- Any new earthquake activity
- Any government and/or non-government agency providing HADR support in the platoon AO
- Any hostile activity in the area
- Any changes in the estimates for assistance in water, food, shelter, or medical support

Phase 2 (search for survivors) – need to share data with

- Doctors Without Borders
- Local leaders
- UNAMA
- PRT (may not have TiGR)
- Brigade HTT (automatic)
- Red Cross
- Red Crescent
- Another coalition partner

Phase 3 (recovery operations) – need to share data with

- Doctors Without Borders
- UNAMA
- Local leaders
- PRT (may not have TiGR)
- Brigade HTT (automatic)
- Red Cross
- Red Crescent
- Another coalition partner
- Government reconstruction agencies
- Local and international construction companies

Available Technologies:

US Forces (interacting with the existing military network (MILNET) – DISA has made great strides in implementing the service oriented architecture approach to enabling automation support for enterprise processes. Specific results which affect the future utility of project results include the Joint Cross Domain eXchange (JCDX) system to share data across the multilevel level secure system (MLS) and the classification Policy Decision Service (cPDS) system to enable discovery of labeled data for and subsequent automated identification of trust relations. These MILNET services enable automated building of the XML signature chains necessary for using the NetSMART project results to automatically implement policy decisions by commanders to share information across the Global Information Grid (GIG).

New tools available for the Development Process

Hardware development tool research has been working for some to move the automated tools available for electronic systems development from the resisistor-transistor-logic (RTL) level to the electronic systems level (ESL). As indicated in an article on system-level design by Rami Rachamim,:

> …the development process can be divided into three phases: Concept, ESL design, and RTL implementation. Each phase is derived from the previous one and feeds the next one.
>
> Concept (Vision): At the concept phase a system designer creates a conceptual description of the system without explicit software/hardware definitions or boundaries concept. The phase starts with a spec and a set of requirements that are mapped into algorithms and functions that can be validated. There are several languages that can serve this domain including UML and C/C++.
>
> ESL Design (Strategy): At the ESL design phase the designer needs to drive its strategies and map the conceptual description into hardware (RTL) and software (C/C++)

representations. This is where the ability to impact (variability) is the greatest, and iteration cycles are shorter. The design phase should define the hardware and software domains (partitioning) that can carry the concept and drive parallel hardware and software implementation flows. It should allow exploration and optimization when allocating and configuring hardware and software resources while making sure they interface appropriately.

The ESL design phase is where the designer creates and finds the best possible hardware and software system configuration that is functionally correct and can support the original system concept. The output of the design process should be a well-defined hardware structure at the RTL level (most likely VHDL or Verilog description) and some of the software layers. It is important to optimize systems against real software since the application greatly impacts the performance and power behavior of the system, and is reflected at the user experience level.

RTL Implementation (Tactics): At the implementation phase the designer actually executes upon the ESL guidelines and maps the hardware (RTL) into silicon. Automating the ESL design phase would not only make the RTL implementation task more efficient, it would also result in shorter verification cycles, since the integration of the main hardware and software blocks is already validated. RTL Verification would then focus solely on implementation related aspects, rather then system-level aspects (e.g. hardware/software interaction, protocol mismatches and data integrity) that are much harder to detect at RTL.

Software system-level design: The Flowing Valued Information project aims to flow information between the military network (MILNET) and other networks in accordance with Department of Defense (DoD) policy for sharing information. The DoD intent is to use SOA as the architecture-level approach for enabling sharing information among the US Armed Services as well as with our coalition partners. However, there is currently no underlying science available for flowing valued information among network nodes or for sharing information dynamically with coalition partners and non-government agencies. Thus, we specifically will investigate system-level design approaches for flowing valued information and sharing the information across security boundaries.

Software design and implementation: Also at the software level, new tools are available for implementing service-oriented architecture systems. Specifically, a promising set of open-source tools are being built under the Eclipse Swordfish project. As indicated on the project web site,"The goal of the Swordfish project is to provide an **extensible SOA framework** based on the proven Eclipse Equinox runtime technology. The framework is designed to be complemented by additional open source components such as a service registry, a messaging system, a process engine etc. to form a comprehensive open source SOA runtime environment based on both established and emerging open standards." The SOA project provides an open-source solution for using the Common Object Request Broker Architecture (CORBA) as the messaging component of an SOA and an extension of the NetSMART inference engine as the process engine to implement sharing policies.

Challenges: While new hardware and software technologies are available to perform system-level design and implementation , there are many unknowns and challenges in achieving the two primary goals of the project. Specifically:

1. Flowing valued information among network nodes in accordance with the **commander's intent**:

    a. How do we model **Complex Event Systems** (CES) with enough precision to predict future states of the system? A mathematical result over a century old establishes the existence of solutions to systems of equations which describe the complex event systems of interest (i.e. systems described by compositions of discrete-event components and continuous-time components). However, to date no method of discovering the solutions to such composed system models has been found. Thus, we will follow an approach of constructing components with known dynamics, composing those components, and experimenting with predicting future states of the composed systems.

    b. Which metrics are sufficient for capturing commander's intent and how do we measure system parameters to **estimate values** of those metrics?

    c. How do we accommodate the need of the commander to dynamically **change intent** of an operation while the operation is underway?

2. Moving information among network nodes in accordance with an expressed **intent to share**:

    a. How do we discover trust relations between entities distributed in time and space which normally are not connected as nodes in a communication network?

    b. How do we dynamically chain together trust relations to establish a "chain of trust" among components which are normally not connected?

    c. How do we prove that the trust policies of DoD for sharing information are satisfied by the system we implement for sharing information across security boundaries? For example, in the scenario above, **how do we prove** that we comply with DoD policy in our implementation of a solution for enabling the network to respond to a commander's declaration of an intent to share information X with user Y and group Z for period of time T?