

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person should be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 27-11-2001		2. REPORT TYPE FINAL		3. DATES COVERED (From - To) 01-08-2001 - 26-11-2001	
4. TITLE AND SUBTITLE An Intuitive (But Analytical) Approach to the Effectiveness of Kill Chain Automation				5a. CONTRACT NUMBER N/A	
				5b. GRANT NUMBER N/A	
				5c. PROGRAM ELEMENT NUMBER N/A	
6. AUTHOR(S) Major James H. Baker and Dr Eva Cza				5d. PROJECT NUMBER N/A	
				5e. TASK NUMBER N/A	
				5f. WORK UNIT NUMBER N/A	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) SAF/AQID, 1500 Wilson Boulevard, Rosslyn, VA				8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) SAF/AQID, 1500 Wilson Boulevard, Rosslyn, VA				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT NO LIMITATION ON DISTRIBUTION					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>The Time Critical Targeting (TCT) process (also referred to as the "Kill Chain") depends heavily on information for success. Increasing "fast" and "reliable" information flow are the two MOEs most often used to increase kill chain effectiveness. This paper explores both of the MOEs, by modeling forward and backward error rates for generic kill-chain components as a Markov chain. The value of the discrete Markov model lies in its simple and intuitive nature, the measurability of the independent variables, its wide applicability, and its analytical tractability.</p> <p>We briefly sketch the results of a simple simulation, which extend the Markov model and address some of its limitations. Multiple simulation runs were conducted and analyzed over several conditions to statistically evaluate impacts of error rates on the kill-chain timeliness and probability of mission success.</p> <p>Application of this model may provides a robust approach to allow system-of-systems architects to evaluate potential trade-space options for the kill chain as a whole. It allows an intuitive feel into the relative importance of system interoperability; data checking; and command, control and communications (C3) links, including estimates of how forward and backward error rates influence both the ultimate time to engage and the probability of correct engagement.</p>					
15. SUBJECT TERMS Kill Chain, Combat Identification, Information Rework, Error Propagation, Markov chain, SIMAN, ARENA, system architecture					
16. SECURITY CLASSIFICATION: UNCLASSIFIED			17. LIMITATION OF ABSTRACT NONE	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
b. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (include area code)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Sta. Z39-18
298-102

20020904 057

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available.

(e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final. Etc. If applicable, enter inclusive report dates (e.g. 10 June 87 – 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C – Contract	PR – Project
G – Grant	TA – Task
PE – Program Element	WU – Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. Of ...; To be published in ... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD – See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE – See authorities.

NASA – See Handbook NHB 2200.2.

NTIS – Leave blank.

Block 12b. Distribution Code.

DOD – Leave blank.

DOE – Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports..

NASA – Leave blank.

NTIS – Leave blank.

Block 13. Abstract. Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (*NTIS only*).

Block 17. – 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e. UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

An Intuitive (But Analytical) Approach To The Effectiveness Of Kill Chain Automation Measures

Major Jim Baker (SAF/AQID)
Dr. Eva Czajkowski (Veridian)
1060 Air Force Pentagon
Washington, DC 20330-1060
703-588-6424 (Work)
james.baker@pentagon.af.mil
eva.czajkowski@veridian.com

69th MORS Symposium

Working Group 15
12-14 June 2001
26 November 2001

ABSTRACT

The Time Critical Targeting (TCT) process, sometimes also referred to as the "Kill Chain" depends heavily on information for success. The better information can flow through this process, the "better" the kill chain. Thus, increasing "fast" and "reliable" information flow are the two MOEs most often used to increase kill chain effectiveness.

This paper explores both of the MOEs for the TCT process. It does so by modeling forward and backward error rates for generic kill-chain components as a discrete Markhov process. We include estimates of how forward and backward error rates (our independent variables) influence both the ultimate time to engagement and the probability of success (our dependent variables, or measures of merit). The value of the discrete Markhov model lies in its simple and intuitive nature, the measurability of the independent variables, its wide application within the Operations Research community, as well as its analytical tractability.

We briefly sketch a simple simulation and its results, which extend the Markhov model and address some of its limitations. Multiple simulation runs were conducted and analyzed over several conditions to statistically evaluate impacts of error rates on the kill-chain timeliness and probability of mission success.

Application of this model may help demonstrate the relative impact of new technologies or new processes upon the kill chain. It provides a robust approach to allow system-of-systems architects to evaluate potential trade-space options for the kill chain as a whole.

The straightforward model provides an intuitive feel into the relative importance of system interoperability; data checking; and command, control and communications (C3) links, including estimates of how forward and backward error rates influence both the ultimate time to engage and the probability of correct engagement.

INTRODUCTION

“Automation” of the kill chain can involve either new technology or changes in tactics, techniques, and procedures (TTP). Either or both of these are intended to reduce the time between sensor and shooter and/or increase the reliability of that data. The wide variety of models, simulations, and levels of fidelity, and acceptance complicate tradeoff analysis by various acquisition and user organizations. We propose a simple yet robust framework to aid in evaluating potential tradeoffs in the effectiveness of different automation measures.

One Measure of Effectiveness (MOE) for the kill chain is the time it takes for information to pass from initial sensor detection to decision maker to shooter for release of weapons. Reducing this transmittal time is critical to meeting a single minute digit detection to kill chain. A second MOE is the potential for error by the shooter, either by receiving invalid information, or failing to act on valid information.

This paper formulates and analyzes a generic model of the kill chain “organization” (i.e., people, systems, and workflow or process of information and products) behavior under differing levels of information quality.

KILL CHAIN INFORMATION MODEL OVERVIEW

Optimizing the Time Critical Targeting (TCT) process is critical to maximally employ overwhelming military power, whether by aerospace, naval, or land against mobile or fleeting target sets. The TCT process is also often referred to as the “Kill Chain”. One generic set of steps commonly used to describe the kill chain is: detect, identify, track, and destroy. In the most

general case, different persons and weapon systems support different aspects of the kill chain. As a result, information gathered by one participant must be passed to another. The faster and more reliably the information can flow through this process, the “better” the kill chain.

We propose to look at the two most common measures of merit given for an effective kill chain. The first measure is the time it takes for information to be transformed and transmitted by each node the process. The second measure is the potential for error by the shooter, or equivalently, the probability of an incorrect or invalid message being received at the terminal node of the chain.

The Information Rework Model

Motivation

The kill chain is a highly complex system. Two challenges immediate pose themselves to model this ‘system’. First, what level of abstraction should be used? In other words, does one represent simply a generic sensor and shooter? Does one model the interaction of a high-range resolution radar (HRR) sensor with its environment? What level of abstraction should be used, and does the model apply at different levels? What about the interface between a pilot and the information displays in the cockpit, or the level of crew coordination within an M1? Or the voice communications between a Forward Air Controller (FAC) and an Air Support Operations Center (ASOC)?

Second, what elements and aspects of the model can be measured? If we can’t measure the input variables effectively, we can’t understand their impact on the performance. How do we ensure that experimental hypotheses about future performance based on a system or tactics, techniques and procedures (TTP) enhancements can actually be proven?

Both of these concerns arise because several kill chain nodes possess highly specific and validated models (e.g. the interaction of surveillance radar with ground clutter.) Others are not as clear, or have not been as readily verified (what constitutes the effectiveness of an Operations Center?)

We approached the problem as primarily one of information flow. How accurately is information sensed, fused, transmitted, and received within kill chain nodes? To address these two concerns, a generic kill chain model must (1) be able to appropriately measure the variables that can affect performance for all the nodes in the chain, and the impact of that those variables on overall kill chain effectiveness and (2) be scalable to different and multiple levels of abstraction. These are our two criteria for success. We suggest than an "information rework" model meets these requirements.

Modeling Information Flow

Communication between kill chain elements often are ambiguous, or are perceived with some level of uncertainty by the elements who are in communication. Ambiguity can arise out of degradation in the communication channel, use of multiple and different protocols, elements using different contexts for interpreting information, or just as a function of human communication. A mock radio transcript between an attack platform being routed by voice to a target by a surveillance platform gives an example of how ambiguity can arise in a voice communication:

SURVEILLANCE: Find town X

ATTACK: I can see town X

SURVEILLANCE: Look for the road leading northwest out of the town

ATTACK: The two lane or four lane road?

SURVEILLANCE: Two lane

ATTACK: Got it

SURVEILLANCE: Go two bridges along that road

ATTACK: Got it

SURVEILLANCE: Assume the length of the bridge is one unit, look three units south and five units west, and look for a small clearing

ATTACK: (pause) Say again, five south and three west to clearing?

SURVEILLANCE: (pause) No, THREE units south and FIVE west to clearing

ATTACK: Counting three south and five west, no clearing

SURVEILLANCE: (pause) You don't see a clearing?

ATTACK: There's no !!! clearing at three south and five west

SURVEILLANCE: (long pause)

ATTACK: Repeat no clearing at three south and five west

SURVEILLANCE: Which bridge did you use as the unit?

ATTACK: The bridge furthest west

SURVEILLANCE: OK, look seven units south and nine units west for the clearing

ATTACK: I have the clearing

SURVEILLANCE: Roger that, your target is directly south of the clearing

ATTACK: Five movers?

SURVEILLANCE: Five movers -- that's your target

ATTACK: Roger, I have the target

Such ambiguity doesn't only arise between human elements. Machine to machine

communication also experiences ambiguities and errors, although they often have protocols that make discovery and rectification of the error or ambiguity easier.

Ambiguity and errors, when discovered, cause 'rework', as nodes communicate with each other to increase their confidence in the information. Note that ambiguity, even where there is no error, also requires the messages to be reworked.

The simple, intuitive model described in the next section can act as a building block for more complicated models. By understanding its behavior, insights can be gained into the more complex behaviors of the entire kill chain, determinations can be made as to some of the measurable physical parameters that influence the overall performance, and system or procedural recommendations can be explored with regards to kill chain performance.

Elements of the Model

The proposed information rework kill chain model consists of three elements:

1. **Messages**, which are passed from one node to the next. The model makes no definition of the length, or duration of the message, only the perceived correctness of the content. Messages can represent the output of a radar signal processor of a target detection by a sensor, a target track passed by a C3 link, verbal control messages from one platform to another, and/or verbal target coordinates.
2. **Nodes**, which act to process information. Information processing nodes. Nodes act to read, interpret, fuse and transmit messages from other nodes. Physically, nodes could represent different levels of hierarchy (from wing man to flight

lead), or generic kill chain components (sensor, shooter, decision maker), or even different organizational elements within a command structure.

3. **Processing errors**, which are introduced by the nodes as they receive, translate, transform, or send messages. Processing errors represent clear or garbled transmissions, the probable result of a noncooperative target identification (NCTI) algorithm output, the effectiveness of visual identification of a target, the classic probability of detection (Pd) and/or probability of false alarms (Pfa) associated with a radar, or the result of a coordinate translation inside a TCT cell. A more robust physical interpretation of these errors and their impact on kill-chain effectiveness will be explored later.

The key underlying assumption of this model is that the content of all the data and/or information held by all the nodes is not completely transparent and instantaneously available to all players in the kill chain. Instead, information must be passed from one node to another, from an information source to the final end user (information sink), with perhaps an additional node(s) in between. The model assumes the promise of a 'global information grid,' where any requested information is received correctly, nearly instantaneously, and in a usable format, will not be realized in the near future.

Modeling the Information Content Inside the Kill Chain

We begin by trying to develop the simplest model possible that meets our two criteria for success.

Consider first a model with three nodes,

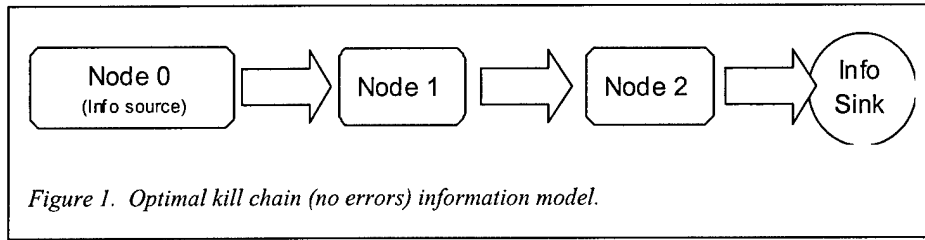


Figure 1. Optimal kill chain (no errors) information model.

as shown in Figure 1. The first node (node 0) acts as an information source, emitting messages to node 1.

The first node reviews the message, transforms it into another message, and forwards it onto another node, which performs another transformation prior to sending it onto the information sink. Assume for a moment that there are no processing errors.

Of the two measures of merit we adopted for kill chain effectiveness, this model meets the “reliability” MOE 100%. In this sense, it reflects the ideal kill chain. There is no chance of error introduced into the messages along the way. Our other measure of merit for kill chain effectiveness was time for messages to move from beginning to end. Thus, efforts to improve kill-chain performance for this model would focus solely on either reducing the number of nodes that separate the information source from the end user and/or reducing the time it takes for each node to process the message.

Currently, the US Air Force (USAF) pursues both of these approaches, sometimes simultaneously. For example, introducing Link-16 onto fighter aircraft as well as sensor platforms such as JSTARS decreases the amount of time to transmit messages by going directly to data displays instead of via voice.

It also reduces the number of nodes, since target

location data may be transmitted directly from the JSTARS sensor to the

F-16 multi-function display (MFD). This approach to optimization reduces the number of nodes and the time it takes for nodal processing. The sensor data are no longer first interpreted by a JSTARS operator, transmitted to an ABCCC operator by voice, who copies the data down manually, and then relays them by voice to a fighter aircraft.

Is focusing solely on reducing node count or decreasing the time a message spends at a node sufficient? Is the assumption of little or no error in the transmission of messages along nodes a good assumption? Probably not—yet this is the model most often assumed. However, it does not capture the hidden costs of additional time due to information rework, due to finite probability of errors being introduced.

We may extend this model by allowing for error in transmission between each node, with probability alpha [$\alpha(i)$] at each step, where “i” is defined as the node number. See Figure 2. In other words, the information source outputs correct messages with probability $1-\alpha(i)$, and incorrect messages with probability $\alpha(i)$. The first agent performs a correct transformation (i.e. doesn’t introduce an error) with similar probabilities, and so on. Each message transformation is assumed to be independent

of the results of the previous node.

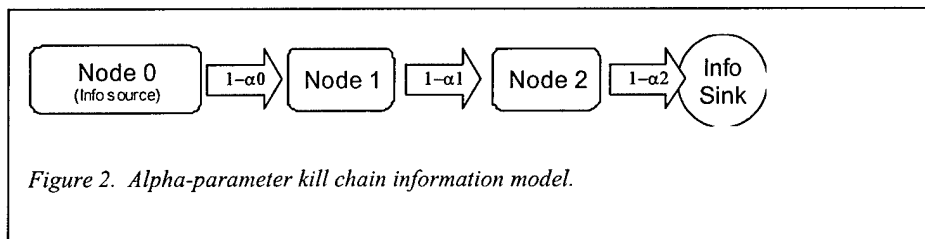


Figure 2. Alpha-parameter kill chain information model.

This approach is better than our initial model. However, it also has a conceptual flaw. It assumes that there is no way for an erroneous message to be fixed before being passed from one node to the next. That doesn't represent the real world very well.

Typically, system designers provide for communication protocols for

recovery from invalid messages. Nodes that represent human interpretation of data usually will also possess a "protocol" for recovery from invalid messages, although it is normally not as robust as the machine-based protocols. For example, a weapons controller who receives an incorrect latitude coordinate because of a transposition of numbers will probably catch that error before transmitting it further. A pilot looking at an MFD display of target location and status will cross-reference that data with other on- or off-board sensors before shooting. A Joint Surveillance Target Attack Radar System (JSTARS) operator may wait and use a history of tracks to

identify a moving track as enemy, and so avoid passing on an erroneous interpretation of sensor data. Other examples come readily to mind, where actions are purposeful in order to clarify messages and remove errors. Such actions are both critical and time consuming, as there are numerous opportunities for re-keying, data conversion, or simple misinterpretation. Thus, our initial model is extended to allow for "reworking" of erroneous messages passed by previous nodes.

We introduce the possibility of rework is another variable beta [$\beta(j)$] in Figure 3, where "j" is defined as the node number. $\beta(j)$ represents the probability that an error introduced in a previous step will NOT be caught.

So, with probability $1-\beta(j)$, the error is caught and the message is returned

to the previous agent (or to the information node) for correction.

Note that small $\beta(j)$ are the desirable characteristics for any information processing system. For any number of nodes and processing times, $\beta(j) = 0$ for all j will optimize kill chain performance. No errors will be transmitted along the chain.

In terms of information rework, this model is superior to our previous models but still not sufficient. Using it would assume that no erroneous interpretations could take place.

A node may request clarification of a message even when in fact the message contained no

error. This probability is defined as gamma $\gamma(k)$, where "k" is defined as the node number.

This parameter accounts for the case where the message is "returned" to a previous node, when in fact the message is correct. See Figure 4.

The system would experience rework when none was necessary.

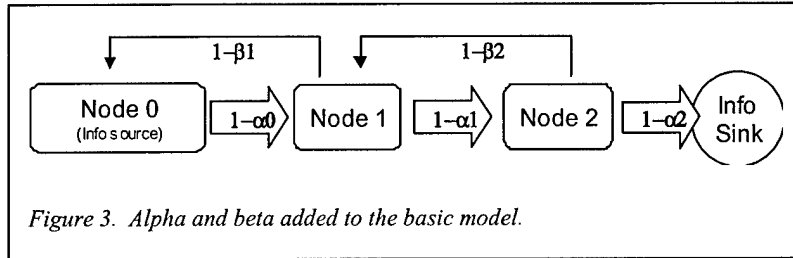


Figure 3. Alpha and beta added to the basic model.

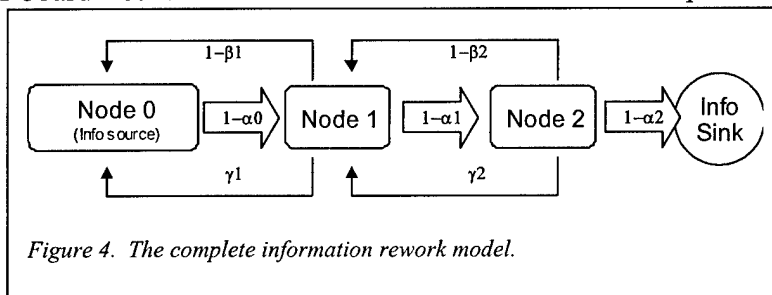


Figure 4. The complete information rework model.

The physical interpretation of this parameter is either (a) the level of 'confidence' that a node downstream has in the messages it receives, or (b) the criticality of action taken based a bad message. A node with high $\gamma(k)$ will act to often confirm message content, even when the probability of a message being in error is in fact low.

In some kinds of systems, one could reasonably assume that $\gamma \ll \beta$. For example, given the domain knowledge of Air Operations Center (AOC) workers (even in the midst of a stressful campaign), the chance of inadvertent rework should be much less than the chance of intentional rework.

Some counterexamples also readily come to mind. For example, verbal communications protocols between aircrew and air traffic controllers characterize a system with a relatively low α and β , but a high γ , because the consequences of any misinterpretation can be so severe.

In essence, γ and β serve the same role in the information flow model as Type I and Type II errors in hypothesis testing.

Again, note that errors with small α , β , and γ are representative of a "good" kill chain. If $\alpha = \beta = \gamma = 0$ (i.e., an error-free processing model), this reduces to the initial almost "ideal" kill chain.

Strengths and Limitations

There are a number of strengths and limitations with this type of model representation. One strength is that it is simple and intuitive, yet represents the way real information transactions are handled between nodes.

The key strength of this model, however, is that the key parameters (α , β , and γ) model can be measured, modeled, and then tested

in a real world system. These key parameters can be determined regardless of whether a node represents an aircraft display, a duty officer on a sub, the probability of detection for a radar system, or the verbal commands of C3 platform to a shooter.

A side benefit is that we may derive analytical results that roughly relate the model parameters (α , β , and γ) to the kill chain MOEs e.g., time to kill and reliability of the final message. These results are presented in the next section.

There are also several limitations to this approach. For example, we assumed that messages couldn't be checked for error more than one node "downstream". This is probably not the case in a real-world environment. Checks performed two or more nodes downstream may well catch previous errors, depending on other contextual information available to them. Also, for a given node, α , β , and γ are the same for every type of message in the system. Additionally, these parameters don't change over time and are completely deterministic. This limitation can be addressed by choosing different parameters to represent different message sets. For example, $\alpha(i, \zeta)$ would represent the probability of error for node i and message type ζ . Or we can vary the parameters via a distribution function over time – they need not remain a constant number, e.g. $\alpha(i, \zeta, t)$. These types of enhancements are readily handled in most simulation tools

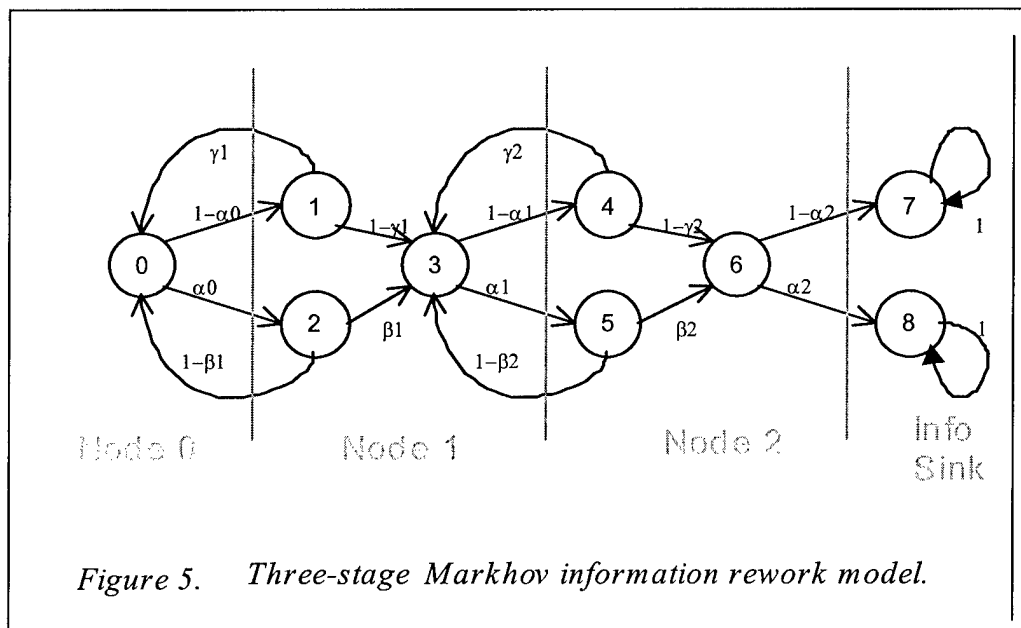
Additionally, because we intend to represent this model as a discrete Markov system, is no way to examine the distribution of message creation, the queue length of any node, or the overall impact and relationship

of task time to the error probabilities introduced.

Some of these limitations of this approach will be tackled later in the paper with the use of a computer simulation. First though, insights can be gleaned by analyzing the α - β - γ model as a discrete Markhov chain, and then seeing how our three parameters influence the kill chain MOEs.

Kill Chain Information Flow Model Analysis

A Markhov chain for the α - β - γ kill chain information flow model is constructed and shown in **Figure 5**.



The states for this model are defined as follows:

	Definition
0	Information sink. Emits inaccurate messages with probability α_0 .
1	Node 1 processing for accurate messages.
2	Node 1 processing for inaccurate messages.
3	Node 1 transmittal of new message to Node 2. Inaccurate messages transmitted with probability α_1 .
4	Node 2 processing for accurate messages.
5	Node 2 processing for inaccurate messages.
6	Node 2 transmittal of new message to Information Sink. Inaccurate messages transmitted with probability α_2 .
7	Information Sink for accurate messages.
8	Information Sink for inaccurate messages.

The end-states that represent the sink are assumed to be absorbing. Our assumption is that once a message enters the sink, no error recovery is possible from the final stage. All other transitions should be self-explanatory.

The transition matrix (P) for this chain is:

$$P := \begin{pmatrix} 0 & 1 - \alpha_0 & \alpha_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \gamma_1 & 0 & 0 & 1 - \gamma_1 & 0 & 0 & 0 & 0 & 0 \\ 1 - \beta_1 & 0 & 0 & \beta_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 - \alpha_1 & \alpha_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_2 & 0 & 0 & 1 - \gamma_2 & 0 & 0 \\ 0 & 0 & 0 & 1 - \beta_2 & 0 & 0 & \beta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 - \alpha_2 & \alpha_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

where:

$$\alpha(i) = P(\text{error})$$

$$1 - \beta(j) = P(\text{rework}|\text{error})$$

$$\gamma(k) = P(\text{rework}|\text{no error})$$

Note again the physical interpretation of the various probabilities in the model is clear and unambiguous. The forward error rate is defined by $\alpha(i)$, $1 - \beta(j)$ is the backward error correction rate, and $\gamma(k)$ is the backward error "confusion" rate.

Information Rework Model Results

There are two measure of merit for kill chain performance available to as a result of using this model; these measures will be examined as a function of the α - β - γ variables.

Those measures are:

- Number of errors introduced in output messages. This measure relates to how often the nodes pass erroneous messages, as a percentage of the total messages processed. In the Department of Defense (DoD), one often specifies performance of errors introduced by any one particular system. One rarely specifies overall error performance of the 'system of systems,' including the influence of operator training and TTP. However, this measure applies to the performance of the kill chain as a whole, not any particular part.
- Number of steps to absorption (and hence, indirectly, overall time a message spends in the process). Although using the discrete Markov process approach does not allow for exact timing determinations [since a transition from any one step to another is assumed to take the same (dimensionless) unit of time], one can get a rough measure of the rework by comparing the minimum number of steps throughout the model with the expected number of

steps given certain forward (α) and backward (β, γ) error rates.

Of particular interest is the analytical relationship of these two measures to $\alpha, \beta,$ and γ . How does changing the forward and backward error rates affect the overall kill-chain performance?

Both of these MOEs are influenced by all three error rates. Intuitively, an increase in $\alpha(i)$ will strongly affect the probability of initially introducing an error, combined with a large $\beta(j)$ will result in the error being undetected. The $\gamma(k)$ effect will contribute in a secondary fashion, since mistaking an error-free message for containing an error will result in the possibility of a real error being introduced [with probability $\alpha(i)$]. For the number of steps to absorption, all three error probability types will clearly influence the outcome. The exact nature of the relationship of these factors on the measures will be examined in the next two sections.

Number of Errors Introduced

Since the simplified model leaves each node as independent from other stages, it suffices to consider the relationship between any two nodes in the cycle where rework is possible. In order to calculate the absorption probabilities, an extra state in the model is introduced (Figure 6), thus:

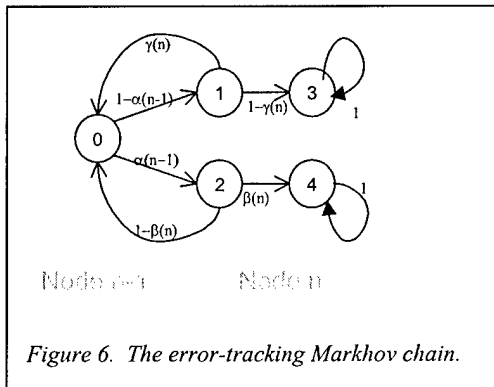


Figure 6. The error-tracking Markov chain.

	Definition
0	Information source from stage n-1. Emits inaccurate messages with probability $\alpha(n-1)$.
1	Node n processing for accurate messages.
2	Node n processing for inaccurate messages.
3	Error free message processed.
4	Error introduced and not caught.

By applying standard Markov chain analysis techniques, we derive the probability of a message leaving the node with an error is defined as:

$$p^{(n)}_{error} = \frac{\alpha(n-1) \cdot \beta(n)}{1 - \gamma(n) + \gamma(n) \cdot \alpha(n-1) - \alpha(n-1) + \alpha(n-1) \cdot \beta(n)}$$

So:

$$p^1_{error} = \frac{\alpha 0 \cdot \beta 1}{1 - \gamma 1 + \gamma 1 \cdot \alpha 0 - \alpha 0 + \alpha 0 \cdot \beta 1}$$

$$p^2_{error} = \frac{\alpha 1 \cdot \beta 2}{1 - \gamma 2 + \gamma 2 \cdot \alpha 1 - \alpha 1 + \alpha 1 \cdot \beta 2}$$

$$p^3_{error} = \frac{\alpha 2 \cdot \beta 3}{1 - \gamma 3 + \gamma 3 \cdot \alpha 2 - \alpha 2 + \alpha 2 \cdot \beta 3}$$

By definition in a three node model, $\beta 3 = 1$ and $\gamma 3 = 0$. Therefore, $p^3_{error} = \alpha 2$ for the limiting case we consider here.

Examining this function more closely, the limiting case of $\alpha(n-1) = 0$ (i.e., no errors introduced), and $\beta(n) = 0$ (i.e., every error found is corrected), then $p^{(n)}_{error} = 0$. Also, if $\gamma(n) = 0$ and $\beta(n) = 1$ (i.e., no feedback on the messages), $p^{(n)}_{error} = \alpha(n-1)$.

Since each stage is independent, the probability of exactly m errors (where m varies from 0 to 3 in this model) in n stages (where $n = 3$ here) is the familiar binomial (where q_m is the probability of m errors):

$$q_0 = P(\text{errors}=0) = (1 - p_{1\text{error}}) * (1 - p_{2\text{error}}) * (1 - p_{3\text{error}})$$

$$q_1 = P(\text{errors}=1) = p_{1\text{error}} * (1 - p_{2\text{error}}) * (1 - p_{3\text{error}}) + (1 - p_{1\text{error}}) * p_{2\text{error}} * (1 - p_{3\text{error}}) + (1 - p_{1\text{error}}) * (1 - p_{2\text{error}}) * p_{3\text{error}}$$

$$q_2 = P(\text{errors}=2) = p_{1\text{error}} * p_{2\text{error}} * (1 - p_{3\text{error}}) + p_{1\text{error}} * (1 - p_{2\text{error}}) * p_{3\text{error}} + (1 - p_{1\text{error}}) * p_{2\text{error}} * p_{3\text{error}}$$

$$q_3 = P(\text{errors} = 3) = p_{1\text{error}} * p_{2\text{error}} * p_{3\text{error}}$$

For the three-stage model, one can look at both the probability distribution function and expected number of errors in a three-stage model. The expected number of errors is

$$E(N_{\text{errors}}) = \sum_{m=0}^3 m \cdot q_m$$

This function is plotted below (Table 1 shows all the cases run and Figures 7 – 15 show sample results for a constant expected value of number of errors) for various values of $\alpha(i)$, $\beta(j)$, and $\gamma(k)$. The expected number of errors form constant contours, plotted against $\alpha(1)$, $\beta(1)$. $\gamma(k)$ increases as the figures go from right to left, and the $\alpha(i)$, $\beta(j)$ decrease as a proportion of $\alpha(1)$, $\beta(1)$ as the figures go from top to bottom.

Application

With these results in hand, we make the following observations:

- As α_0 (the probability of error from node 0) increases and $1 - \beta_1$ (the probability of reworking the error in node 1) decreases, then the expected number of errors increases. For example, if a sensor produces a large error due to an undetected system malfunction and the associated surveillance operator that interprets the data does not question them due to the sensor's historically high accuracy, the error could propagate through the system-of-systems. An error in the message to the terminal node could possibly give the shooter an erroneous "shoot/no-shoot" decision, if that error was not caught in any of the subsequent nodes.
- As $\gamma(k)$ increases for node 2 or 3 (the probability of reworking correct data increases) then the number of expected error also increases. The physical interpretation is that the operators may not be comfortable with a new sensor/system and may routinely question the data validity. Each time they handle a message, there is an additional opportunity to introduce an error. Thus, the error propagation increases.
- As α_1 decreases and is less than α_0 , then the error propagation correspondingly decreases. The physical interpretation of this is that node 1 "catches" some of the errors that were processed through from node 0. For example, if node 0 is the malfunctioning sensor example again and node 1 is the surveillance operator again, node 1 now will question the data that are being received based on some additional data or track history information he has available to him. As a result, he will correct the data that are being forwarded,

thereby reducing the forward error into node 2.

- As α_2 decreases and is less than α_1 which in turn is less than α_0 (i.e., there's a smaller chance of error being introduced as messages move from one node to the next) then the error propagation correspondingly decreases. From the above example, the physical interpretation of this is that node 1 "catches" some of the errors that were processed through from node 0. Again, node 1 will correct the data that are being forwarded, thereby reducing the forward error into node 2. Node 2 will, in turn, question some of these data as well based on some additional "on-board" information he has available to him. For example, node 2 could be a FAC or a fighter pilot that will question the data that are being received based on some additional data he has available to him. As a result, he will correct the data that are being forwarded to the "information sink" node, e.g., the actual weapon system.

A system architect can use this type of analysis to simplify first-level trades. The interesting tradeoff questions and outcomes and brought into relief. For example, given a reduction in one of the error rates, what is the overall impact for the system of system performance? A first order answer follows.

For the sake of clarity, assume that all the α 's are equal, all the β 's are equal, and all the γ 's are equal.

- For the case where α is reduced by a factor of N and $\beta, \gamma \rightarrow 0$, the reduction in $p_{1\text{error}}$ and $p_{2\text{error}}$ goes as $(1 - \alpha)/(N - \alpha)$; or as $1/N$ for small α . The reduction in $p_{3\text{error}}$ also goes as $1/N$.

- For the case where β is reduced by a factor of N and $\alpha, \gamma \rightarrow 0$, the reduction in $p_{1\text{error}}$ and $p_{2\text{error}}$ goes as $1/N$. $p_{3\text{error}}$ is not a function of β .
- For the case where γ is reduced by a factor of N and $\alpha, \beta \rightarrow 0$, the reduction in $p_{1\text{error}}$ and $p_{2\text{error}}$ goes as $[N*(1 - \gamma)]/(N - \gamma)$, which isn't much of a reduction in terms of N . There is no effect on the third node since $p_{3\text{error}}$ is not a function of γ either.
- For the case where both α and β are reduced by a factor of N and $\gamma \rightarrow 0$, the reduction in $p_{1\text{error}}$ and $p_{2\text{error}}$ goes as $(1 - \alpha + \alpha\beta)/(N^2 - \alpha N + \alpha\beta)$, which essentially goes as $1/N^2$, in terms of N . The reduction in $p_{3\text{error}}$ goes as $1/N$.
- For the expected number of errors, each of these reductions are amplified since $E(N_{\text{errors}})$ go essentially as $m*(\text{reduction})^m$.

The implication of these results is that minimizing α and β error rates is much more effective for decreasing the error number of errors (or error propagation) MOE than minimizing γ . System architects could apply these results, depending on the relative cost (for systems) or ease of implementation (for procedures) to determine an optimum mix of new technologies and TTPs.

Table 1. Test Cases Run

CASE	$\alpha 0$	$\alpha 1$	$\beta 1$	$\gamma 1$	$\alpha 2$	$\beta 2$	$\gamma 2$
1	0 ... 0.5	$\alpha 0$	0 ... 0.5	0	$\alpha 0$	$\beta 1$	$\gamma 1$
2	0 ... 0.5	$\alpha 0$	0 ... 0.5	0.05	$\alpha 0$	$\beta 1$	$\gamma 1$
3	0 ... 0.5	$\alpha 0$	0 ... 0.5	0.1	$\alpha 0$	$\beta 1$	$\gamma 1$
4	0 ... 0.5	$\alpha 0$	0 ... 0.5	0.25	$\alpha 0$	$\beta 1$	$\gamma 1$
5	0 ... 0.5	$\alpha 0$	0 ... 0.5	0.5	$\alpha 0$	$\beta 1$	$\gamma 1$
6	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0	$\alpha 0/2$	$\beta 1/2$	$\gamma 1/2$
7	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0.05	$\alpha 0/2$	$\beta 1/2$	$\gamma 1/2$
8	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0.1	$\alpha 0/2$	$\beta 1/2$	$\gamma 1/2$
9	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0.25	$\alpha 0/2$	$\beta 1/2$	$\gamma 1/2$
10	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0.5	$\alpha 0/2$	$\beta 1/2$	$\gamma 1/2$
11	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0	$\alpha 0/4$	$\beta 1/2$	$\gamma 1/2$
12	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0.05	$\alpha 0/4$	$\beta 1/2$	$\gamma 1/2$
13	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0.1	$\alpha 0/4$	$\beta 1/2$	$\gamma 1/2$
14	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0.25	$\alpha 0/4$	$\beta 1/2$	$\gamma 1/2$
15	0 ... 0.5	$\alpha 0/2$	0 ... 0.5	0.5	$\alpha 0/4$	$\beta 1/2$	$\gamma 1/2$

Figure 7. Error Propagation Case 1

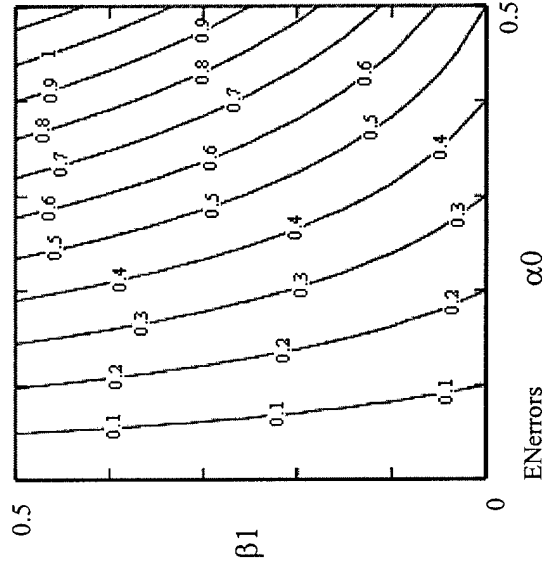


Figure 8. Error Propagation Case 3

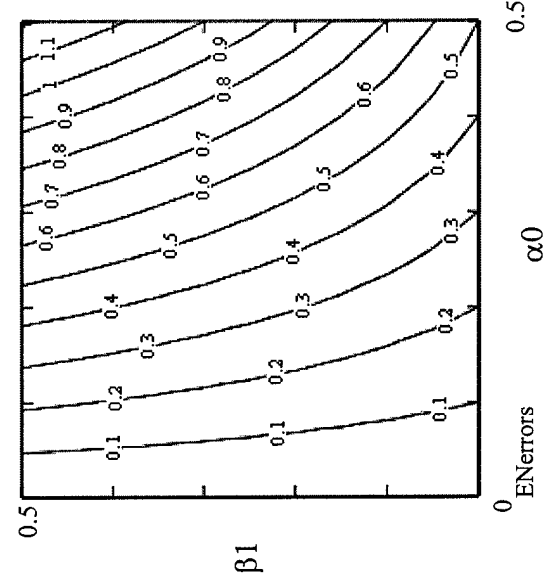


Figure 9. Error Propagation Case 5

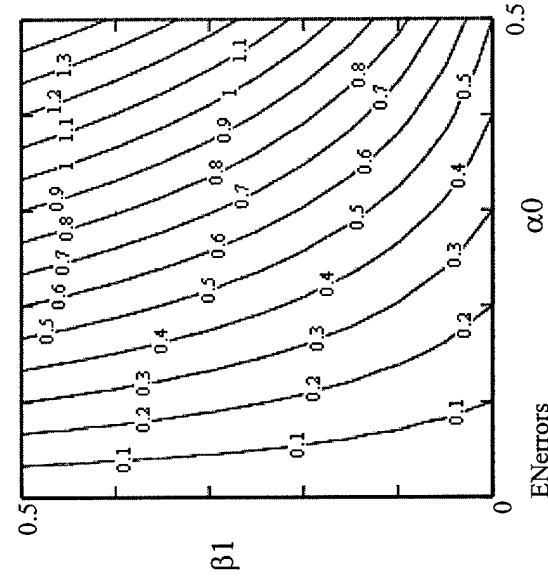


Figure 10. Error Propagation Case 6

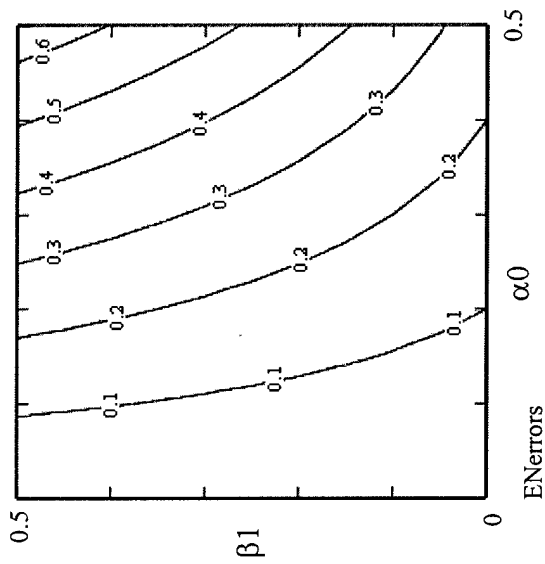


Figure 11. Error Propagation Case 8

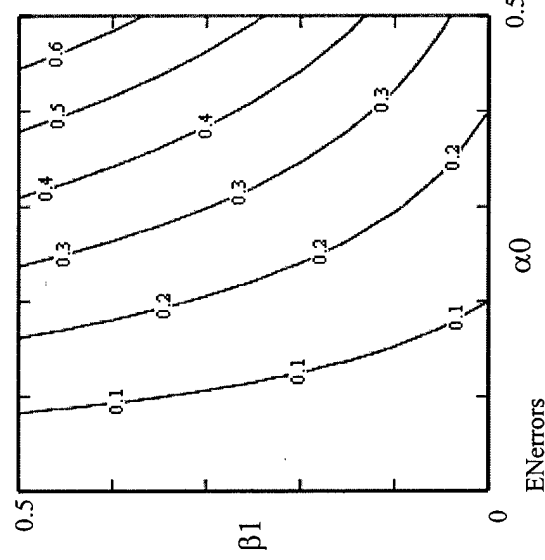


Figure 12. Error Propagation Case 10

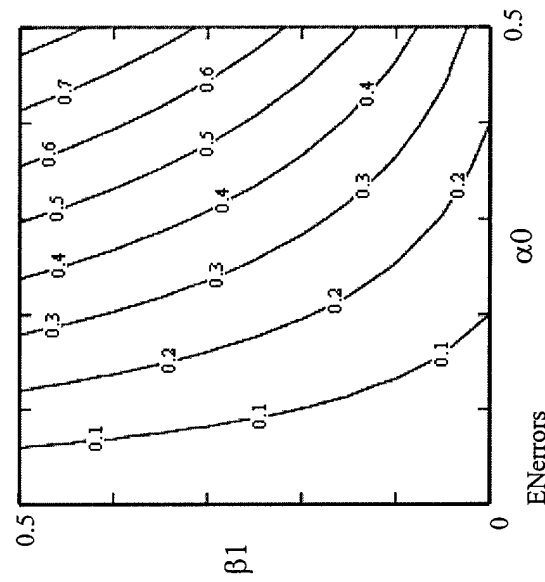


Figure 13. Error Propagation Case 11

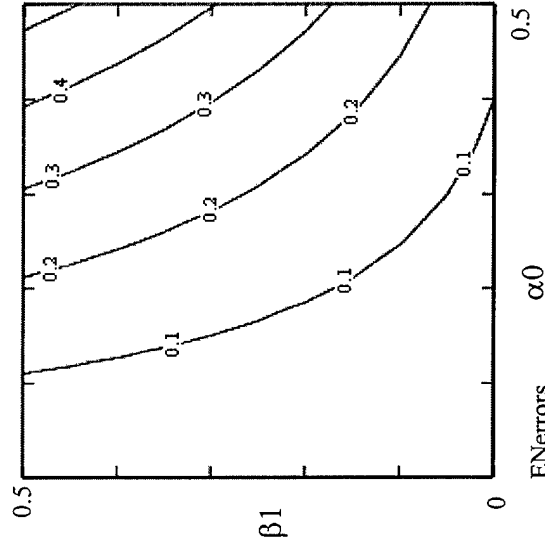


Figure 14. Error Propagation Case 13

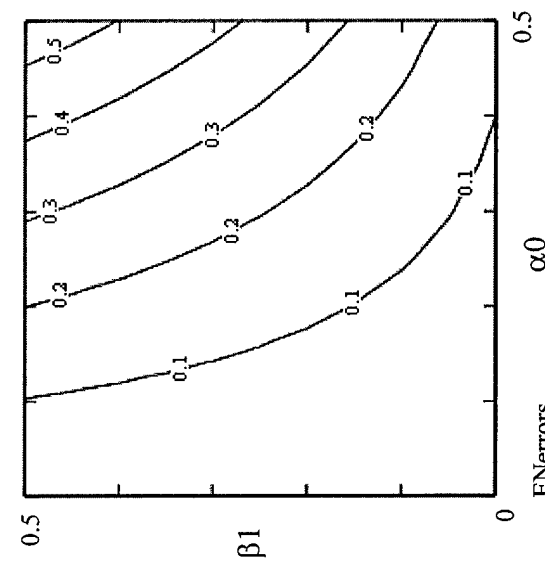
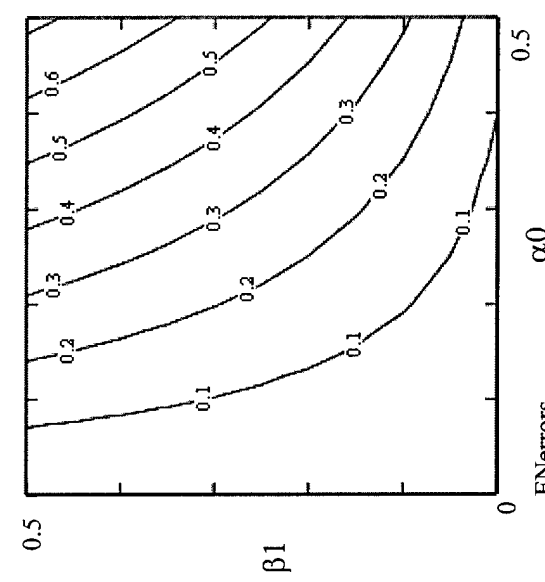


Figure 15. Error Propagation Case 15



Number of Steps to Absorption (Rework Cycles)

The total number of rework cycles is equal to the expected number of steps before moving to the information sink (i.e., the absorbing states). Although this is only an indirect measure of the delay time measure of effectiveness, it also provides some interesting insights.

Again, following the standard procedure for dealing with Markhov absorbing chains, the matrix is divided into the standard form, with P11 consisting of all the transient states, Q consisting of the absorbing states, and P12 consisting of the transition between them. In this case, the absorbing states are the information sink states (number 7 and 8). See Figure 16.

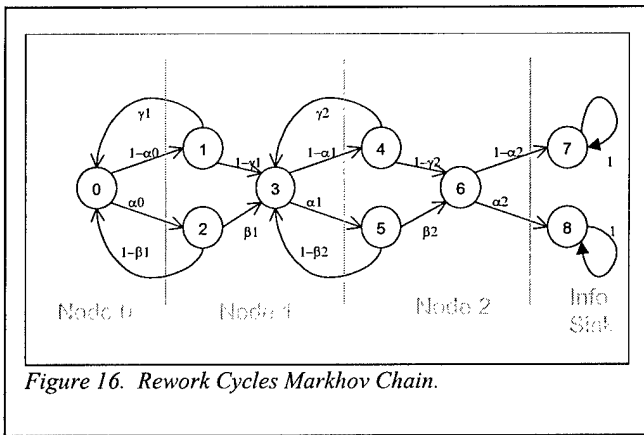


Figure 16. Rework Cycles Markhov Chain.

Note that by definition, $\beta_3 = 1$ and $\gamma_3 = 0$.

The expected number of steps to absorption (in either state 7 or state 8) is given by

$$N_{absorbing} = (I - P11)^{-1} \cdot v$$

where v is the unity vector.

The columns of $(I - P11)^{-1}$ represent the average number of steps spent in that state (given that one starts in the state represented by the row).

Our only interest is in the expected number of steps from state 0 to state 7 or state 8, therefore, the sum of the elements of row one is used:

$$N_{absorbing} = n11 + n12 + n13 + n14 + n15 + n16 + n17$$

where:

$$n11 = \frac{1}{1 - \alpha_0 + \beta_1 \cdot \alpha_0 - \gamma_1 + \gamma_1 \cdot \alpha_0}$$

$$n12 = \frac{1 - \alpha_0}{1 - \alpha_0 + \beta_1 \cdot \alpha_0 - \gamma_1 + \gamma_1 \cdot \alpha_0}$$

$$n13 = \frac{\alpha_0}{1 - \alpha_0 + \beta_1 \cdot \alpha_0 - \gamma_1 + \gamma_1 \cdot \alpha_0}$$

$$n14 = \frac{1}{1 - \gamma_2 + \gamma_2 \cdot \alpha_1 - \alpha_1 + \alpha_1 \cdot \beta_2}$$

$$n15 = \frac{1 - \alpha_1}{1 - \gamma_2 + \gamma_2 \cdot \alpha_1 - \alpha_1 + \alpha_1 \cdot \beta_2}$$

$$n16 = \frac{\alpha_1}{1 - \gamma_2 + \gamma_2 \cdot \alpha_1 - \alpha_1 + \alpha_1 \cdot \beta_2}$$

$$n17 = 1$$

Below are plots of $N_{\text{absorbing}}$ (Figures 17 – 22) for various $\alpha(i)$, $\beta(j)$, and $\gamma(k)$. The expected number of steps to an absorbing state are again plotted as constant contours form against $\alpha(1)$, $\beta(1)$. $\gamma(k)$ increases as figures go from right to left, and the $\alpha(i)$, $\beta(j)$ decrease as a proportion of $\alpha(1)$, $\beta(1)$ as the figures go from top to bottom.

Application

From these figures (although not all cases are presented here), one comes to the following conclusions:

- As α_0 increases and $1-\beta_1$ (the probability of reworking the error in node 1) increases, then the expected number of steps to absorption increases. That is to say that as there are more errors and more rework to correct those errors, there are more steps to absorption required, which is to be expected. Physically, this gets back to the “malfunctioning sensor” example in node 0, however, this time the surveillance operator in node 1 is questioning the error and reworking the problem so as to pass correct data forward. Because of this, there are more steps involved.
- As $\gamma(k)$ increases for either node, then the steps to absorption also increases since more time is spent on reviewing correct data. This also is to be expected. Physically, this means that operators are not comfortable, for whatever reason, with the output of a sensor/source of information, even though it is producing correct data, and are thus reworking the data each time. This wastes time on reworking correct data as opposed to reworking incorrect data, which is where their focus should be. As a result, more steps are required for absorption, or more time is needed, on average, to find and kill targets.
- As α_1 decreases and is less than α_0 , then the steps to absorption correspondingly decrease, as expected. Again, the physical interpretation of this is that node 1 “catches” some of the errors that were processed through from node 0. For example, if node 0 is the malfunctioning sensor example again and node 1 is the surveillance operator again, node 1 now will question the data that are being received based on some additional data or track history information he has available to him. As a result, he will correct the data that are being forwarded, thereby reducing the total number of messages that might require rework from node 2, and reducing the overall number of rework cycles.
- Consider now as α_2 decreases and is less than α_1 , which in turn is less than α_0 (the probability of error from node 0), i.e., there’s less error being generated as the data move from one node to the next. In this case, the steps to absorption have no change due to α_2 as the theory shows that $N_{\text{absorbing}}$ is not a function of α_2 , but rather of α_0 and α_1 . This is because the final “information sink” node absorbs all the data that come out of node 2. For example, if node 0 is the sensor, node 1 is the surveillance platform, node 2 is the fighter aircraft, and the “information sink” is the weapon system/computer, the weapon system will accept what the fighter aircraft/pilot will send to it. (Note, this set of plots is not presented here, however, $N_{\text{absorbing}}$ for Cases 11 – 15 are identical to those for Cases 6 – 10.)

Let’s again examine some limiting cases, where all the α ’s, β ’s and γ ’s are again equal.

UNCLASSIFIED

For this case, the number of steps to absorption reduces to:

$$N_{\text{absorbing}} = \frac{3(1 - \gamma + \gamma\alpha - \alpha + \alpha\beta) + 2}{1 - \gamma + \gamma\alpha - \alpha + \alpha\beta}$$

As expected, the number is strongly influenced by all three variables, with the α effect still predominant. Also note that the minimum number of steps would be achieved with $\alpha = \gamma = \beta = 0$, and is equal to 5.

As with our previous derivation on errors introduced, the system architect could use this function to get an intuitive feel for the tradeoffs involved in reducing different error rates.

- For the case where α is reduced by a factor of N and $\beta, \gamma \rightarrow 0$, the reduction in $N_{\text{absorbing}}$ goes as $[(1 - \alpha) * (5N - \alpha)] / [(N - \alpha) * (5 - \alpha)]$.

- For the case where β is reduced by a factor of N and $\alpha, \gamma \rightarrow 0$, the reduction in $N_{\text{absorbing}}$ goes as 1 - namely, there is no effect.
- For the case where γ is reduced by a factor of N and $\alpha, \beta \rightarrow 0$, the reduction in $N_{\text{absorbing}}$ goes as $[(1 - \gamma) * (5N - \gamma)] / [(N - \gamma) * (5 - \gamma)]$.
- And, for the case where both α and β are reduced by a factor of N and $\gamma \rightarrow 0$, the reduction in $N_{\text{absorbing}}$ goes as $[(5N^2 - \alpha N + \alpha\beta) * (1 - \alpha + \alpha\beta)] / [(N^2 - \alpha * N + \alpha\beta) * (5 - \alpha + \alpha\beta)]$.

As with the previous measure of merit, system architects could weight the likely cost of new technologies or TTPs against the reduction seen in these factors.

Figure 17. Steps to Absorption Case 1

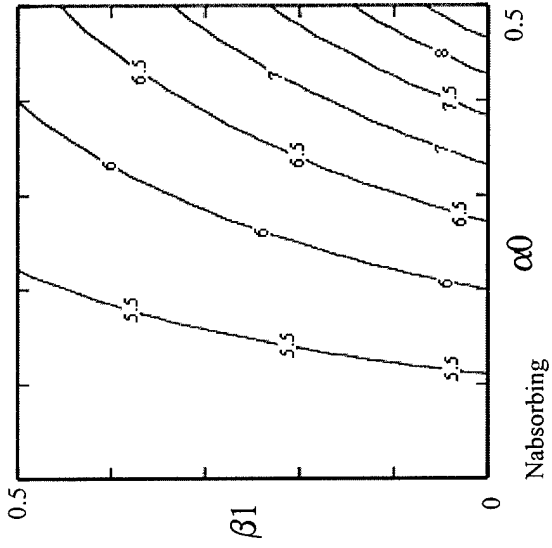


Figure 18. Steps to Absorption Case 3

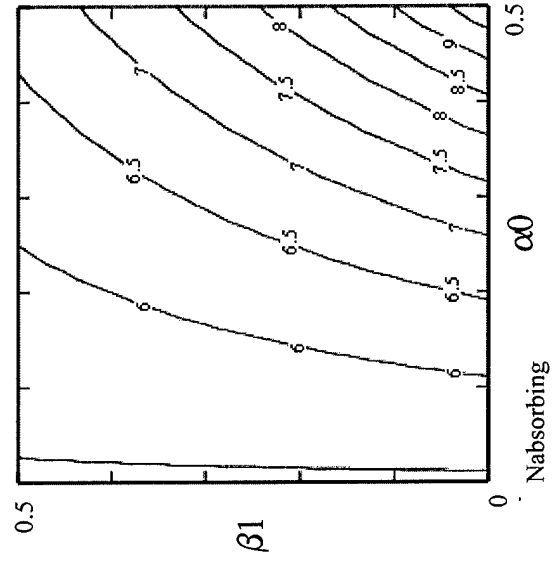


Figure 19. Steps to Absorption Case 5

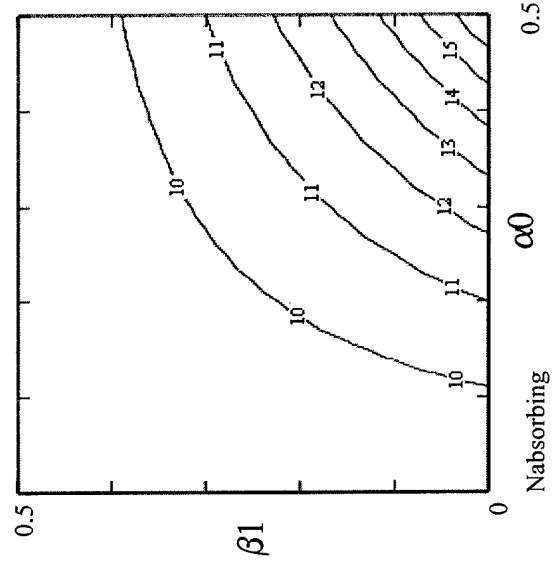


Figure 20. Steps to Absorption Case 6

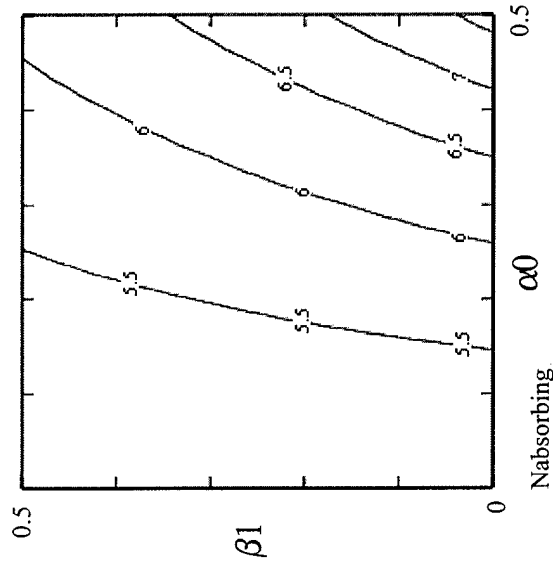


Figure 21. Steps to Absorption Case 8

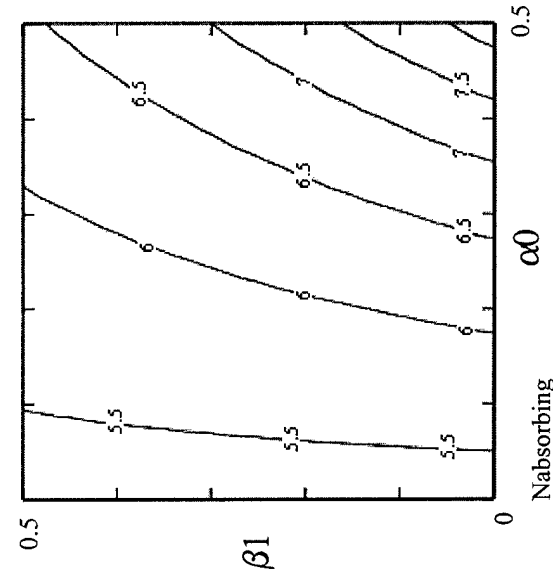
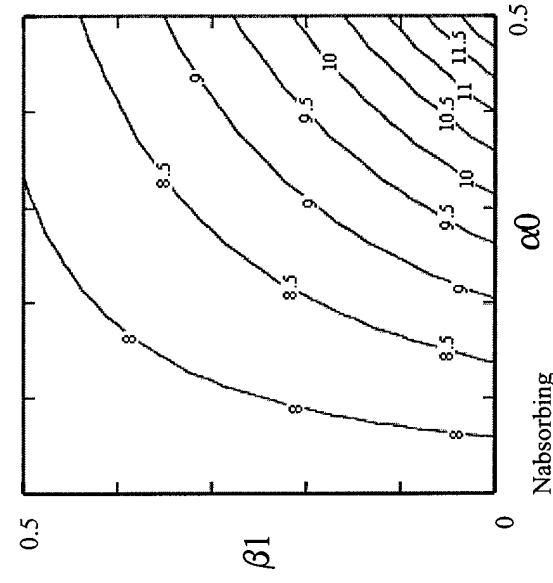


Figure 22. Steps to Absorption Case 10



SIMULATING THE REWORK CHAIN

Using a Markhov chain has provided several interesting insights into the effect of imperfect information, imperfect processing, and rework cycles. However, the limitations to pursuing this approach are high, with several factors making continuing this approach untenable:

- Inability to handle resource overload and queuing questions. The α - β - γ -model essentially assumes that each message is processed sequentially through the system, so that queuing cannot take place nor resource utilization be measured.
- Inability to handle the time that tasks can take. In a discrete Markhov model, all tasks take the same amount of time. One could add states to simulate additional time being spent by a node, but the time taken would still be deterministic. Information processing tasks have variability in duration. Attacking this limitation is critical to moving the model from its current indirect measure of time information spends getting from sensor to shooter (a la rework cycles) to a more robust measure.

Addressing these factors requires a more powerful tool than the simple analytic model. We used the ARENA®/SIMAN® simulation application to create a more detailed kill chain model, based on the understanding of the system description and the extra insight from the rework Markhov chain model. We'll use this simulation to address the first limitation associated with queuing problems and resource utilization by varying the interarrival time of messages into the

system. Further research will be needed to introduce variation into task duration.

For this baseline study, all error variables were held constant throughout each simulation run so their effect could be more readily seen. This is again a limitation to increase our intuition about the model. As we have said, α , β , and γ are easily measurable, and could be modeled as random variables with a joint distribution associated with a given node, message type, operator skill, time, etc. We also did not introduce any change into the task duration of each node.

Two cases were simulated via ARENA. The first one included a constant interarrival time (i.e., time delta) between new message inputs to node 0 for the 'system of systems'. These time intervals with three different values. The second case assumed an exponential time distribution for interarrival of messages with the same three values. The processing time of each subsequent node, however, remained as a constant value of 1 nondimensional time unit. The $\alpha(i)$'s, $\beta(j)$'s, and $\gamma(k)$'s were as previously presented in Table 1, however, only select cases are presented here for illustration purposes.

The physical interpretation of these cases is thus each node can process messages at a rate of 1/minute (or second), and the first node begins receiving messages at a mean interarrival rate of 5, 10 or 30/minute (or second).

Tables 2 – 5 present the comparison of the error propagation between the theoretical calculation and the ARENA runs.

Table 2. Error Propagation for $\alpha_0 = 0.2$ and $\beta_1 = 0$.

CASE	E(errors) (Theory)	Time Delta	a0 (Model)	a1 (Model)	a2 (Model)	a3 (Model)	E(errors) (Model)
1	0.2	30	0.810	0.190	0	0	0.190
		10	0.803	0.192	0	0	0.192
		5	0.802	0.198	0	0	0.198
3	0.2	30	0.801	0.199	0	0	0.199
		10	0.800	0.200	0	0	0.200
		5	0.812	0.188	0	0	0.188
5	0.2	30	0.819	0.181	0	0	0.181
		10	0.798	0.202	0	0	0.202
		5	---	---	0	0	---

Table 3. Error Propagation for $\alpha_0 = 0.5$ and $\beta_1 = 0.5$.

CASE	E(errors) (Theory)	Time Delta	a0 (Model)	a1 (Model)	a2 (Model)	a3 (Model)	E(errors) (Model)
6	0.660	30	0.457	0.438	0.095	0.010	0.658
		10	0.459	0.436	0.097	0.008	0.654
		5	0.464	0.431	0.099	0.006	0.647
8	0.688	30	0.438	0.451	0.101	0.010	0.683
		10	0.442	0.443	0.105	0.010	0.683
		5	0.441	0.436	0.118	0.005	0.687
10	0.850	30	0.327	0.516	0.150	0.007	0.837
		10	0.341	0.491	0.158	0.010	0.837
		5	0.334	0.490	0.158	0.018	0.860

Table 4. Error Propagation for $\alpha_0 = 0.2$ and $\beta_1 = 0$.

CASE	E(errors) (Theory)	Time Delta	a0 (Model)	a1 (Model)	a2 (Model)	a3 (Model)	E(errors) (Model)
1	0.2	EXPO(30)	0.794	0.206	0	0	0.206
		EXPO(10)	0.795	0.205	0	0	0.205
		EXPO(5)	0.805	0.195	0	0	0.195
3	0.2	EXPO(30)	0.804	0.196	0	0	0.196
		EXPO(10)	0.804	0.196	0	0	0.196
		EXPO(5)	0.801	0.199	0	0	0.199
5	0.2	EXPO(30)	0.806	0.194	0	0	0.194
		EXPO(10)	0.802	0.198	0	0	0.198
		EXPO(5)	---	---	0	0	---

Table 5. Error Propagation for $\alpha_0 = 0.5$ and $\beta_1 = 0.5$.

CASE	E(errors) (Theory)	Time Delta	a0 (Model)	a1 (Model)	a2 (Model)	a3 (Model)	E(errors) (Model)
6	0.660	EXPO(30)	0.481	0.410	0.104	0.005	0.633
		EXPO(10)	0.482	0.424	0.092	0.002	0.614
		EXPO(5)	0.476	0.428	0.096	0.006	0.638
8	0.688	EXPO(30)	0.447	0.436	0.114	0.003	0.673
		EXPO(10)	0.444	0.437	0.109	0.010	0.685
		EXPO(5)	0.458	0.421	0.113	0.008	0.671
10	0.850	EXPO(30)	0.320	0.501	0.164	0.015	0.874
		EXPO(10)	0.323	0.509	0.151	0.017	0.862
		EXPO(5)	0.331	0.493	0.167	0.009	0.854

Table 6. Steps to Absorption for $\alpha_0 = 0.2$ and $\beta_1 = 0$.

CASE	E[N absorption] (Theory)	Time Delta	N (min)	N (ave)	N (max)
1	6.000	30	5.00	5.97	17.00
		10	5.00	5.93	15.00
		5	5.00	6.78	20.00
3	6.556	30	5.00	6.52	19.00
		10	5.00	6.54	19.00
		5	5.00	7.50	24.00
5	11.000	30	5.00	11.02	39.00
		10	5.00	11.65	49.00
		5	---	---	---

Table 7. Steps to Absorption for $\alpha_0 = 0.5$ and $\beta_1 = 0.5$.

CASE	E[N absorption] (Theory)	Time Delta	N (min)	N (ave)	N (max)
6	6.128	30	5.00	6.14	15.00
		10	5.00	6.18	21.00
		5	5.00	6.86	19.00
8	6.438	30	5.00	6.48	15.00
		10	5.00	6.50	21.00
		5	5.00	7.30	21.00
10	8.200	30	5.00	8.18	23.00
		10	5.00	8.37	43.00
		5	5.00	12.82	73.00

Table 8. Steps to Absorption for $\alpha_0 = 0.2$ and $\beta_1 = 0$.

CASE	E[N absorption] (Theory)	Time Delta	N (min)	N (ave)	N (max)
1	6.000	EXPO(30)	5.00	6.25	18.16
		EXPO(10)	5.00	7.25	22.10
		EXPO(5)	6.00	13.64	81.32
3	6.556	EXPO(30)	5.00	6.82	18.09
		EXPO(10)	5.00	8.40	56.13
		EXPO(5)	5.00	15.98	75.06
5	11.000	EXPO(30)	5.00	11.71	51.78
		EXPO(10)	5.00	18.31	101.78
		EXPO(5)	---	---	---

Table 9. Steps to Absorption for $\alpha_0 = 0.5$ and $\beta_1 = 0.5$.

CASE	E[N absorption] (Theory)	Time Delta	N (min)	N (ave)	N (max)
6	6.128	EXPO(30)	5.00	6.39	13.80
		EXPO(10)	5.00	7.61	24.65
		EXPO(5)	5.00	13.28	65.72
8	6.438	EXPO(30)	5.00	6.77	18.00
		EXPO(10)	5.00	8.29	49.96
		EXPO(5)	5.00	13.86	56.86
10	8.200	EXPO(30)	5.00	8.72	32.54
		EXPO(10)	5.00	10.78	55.72
		EXPO(5)	5.00	39.74	270.59

Table 10. Amount of Times Nodes are Busy for $\alpha_0 = 0.2$ and $\beta_1 = 0$.

CASE	Time Delta	Node 2 Busy (ave)	Node 1 Busy (ave)	Node 0 Busy (ave)
11	30	0.03710	0.11263	0.04217
	10	0.11086	0.33777	0.12686
	5	0.22331	0.67013	0.24690
13	30	0.03919	0.11667	0.04413
	10	0.11769	0.34977	0.13209
	5	0.23251	0.71511	0.28269
15	30	0.05051	0.16743	0.08356
	10	0.15118	0.50115	0.24992
	5	0.30169	0.98684	0.48574

Table 11. Amount of Times Nodes are Busy for $\alpha_0 = 0.5$ and $\beta_1 = 0.5$.

CASE	Time Delta	Node 2 Busy (ave)	Node 1 Busy (ave)	Node 0 Busy (ave)
11	30	0.04107	0.11910	0.04467
	10	0.12396	0.35898	0.13497
	5	0.24640	0.71240	0.26600
13	30	0.04240	0.12473	0.04897
	10	0.12726	0.37429	0.14697
	5	0.25460	0.73840	0.28380
15	30	0.05253	0.15308	0.06718
	10	0.15738	0.45993	0.20250
	5	0.31543	0.91356	0.39848

Table 12. Amount of Time Nodes are Busy for $\alpha_0 = 0.2$ and $\beta_1 = 0$.

CASE	Time Delta	Node 2 Busy (ave)	Node 1 Busy (ave)	Node 0 Busy (ave)
11	EXPO(30)	0.03638	0.10952	0.04061
	EXPO(10)	0.10633	0.32091	0.11861
	EXPO(5)	0.22027	0.66977	0.25017
13	EXPO(30)	0.04010	0.11874	0.04498
	EXPO(10)	0.11833	0.35329	0.13553
	EXPO(5)	0.23291	0.71464	0.28026
15	EXPO(30)	0.05084	0.16943	0.08456
	EXPO(10)	0.14696	0.51135	0.26373
	EXPO(5)	0.28231	0.97386	0.49658

Table 13. Amount of Time Nodes are Busy for $\alpha_0 = 0.5$ and $\beta_1 = 0.5$.

CASE	Time Delta	Node 2 Busy (ave)	Node 1 Busy (ave)	Node 0 Busy (ave)
11	EXPO(30)	0.03913	0.11470	0.04352
	EXPO(10)	0.12734	0.36844	0.13898
	EXPO(5)	0.24567	0.71860	0.27288
13	EXPO(30)	0.04247	0.12320	0.04749
	EXPO(10)	0.13227	0.38203	0.14786
	EXPO(5)	0.25893	0.75366	0.29181
15	EXPO(30)	0.05495	0.16002	0.07041
	EXPO(10)	0.15851	0.45810	0.20114
	EXPO(5)	0.30459	0.87806	0.38345

As the Time Delta decreases in value from 30 to 5, indicating that messages are arriving on average more quickly, the error propagation values, although they differ, do not really change appreciably. This is also true for the exponential distribution of the input time intervals. The overall error propagation is not sensitive to this parameter.

Tables 6 – 9 present the comparison of the steps to absorption between the theoretical calculation and the ARENA runs. This time, as the Time Delta decreases in value from 30 to 5 (to include the exponential distribution), the absorption tends to increase in general, though not linearly – there sometimes is a slight decrease on part of this curve. As such, the absorption is more sensitive to this parameter – the average value somewhat and the standard deviation more appreciably. Additionally, although the tables do not present the numbers, there is no difference between Cases 6 – 10 and 11 – 15, where α_2 varies between the two. This confirms the fact that the steps to absorption are not a function of α_2 .

Tables 10 – 13 present the amount of time that the individual nodes are kept busy during this ARENA simulation of the 'system-of-systems' process. As the Time Delta decreases in value from 30 to 5 (to include the exponential distribution though the 'busy-ness' factor is not overly sensitive to the type of distribution, just the value), the amount of time the nodes are busy does increase as the time interval decreases. This is expected since as the time interval between messages decreases, more and more messages are moving through the system that the nodes must process.

CONCLUSION

The model provided an understanding of these two independent variables and their interaction within the kill chain. Multiple simulation runs were conducted and analyzed over a wide variety of conditions to statistically evaluate their impacts on the total kill-chain timeliness and probability of mission success.

Application of this model may help demonstrate the relative impact of new technologies or new processes upon the kill chain. It provides a robust approach to allow system-of-systems architects to evaluate potential trade-space options for the kill chain as a whole.

The straightforward model provides an intuitive feel into the relative importance of system interoperability; data checking; and command, control and communications (C3) links, including estimates of how forward and backward error rates influence both the ultimate time to engage and the probability of correct engagement.

Although the problem definition for this paper was motivated by an examination of the kill chain against mobile targets, the results obtained are applicable to any problem where timely and accurate transmission of information is critical to mission success.