REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
May 2012	JOURNAL ARTICLE (Post Print)	MAR 2010 – MAY 2012

4. TITLE AND SUBTITLE

MODELING OPERATIONAL ROBUSTNESS AND RESILIENCY WITH HIGH-LEVEL PETRI NETS

5a. CONTRACT NUMBER
In-House

5b. GRANT NUMBER
FA8750-09-2-0157

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)

AFRL: Timothy E. Busch

La Salle University: Tavana Madjid

University of Texas at Austin: Eleanor L. Davis

5d. PROJECT NUMBER
S2TS

5e. TASK NUMBER
IH

5f. WORK UNIT NUMBER
04

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

La Salle University
Philadelphia, PA 19141

University of Texas at Austin
1 University Station C0200
Austin, TX 78712

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

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

Air Force Research Laboratory/Information Directorate
Rome Research Site/RISC
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TP-2012-17

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA Case Number: 88ABW-2010-3929
DATE CLEARED: JAN 2010

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Military operations are highly complex workflow systems that require careful planning and execution. The interactive complexity and tight coupling between people and technological systems has been increased in military operations, which leads to both improved efficiency and a greater vulnerability to mission accomplishment due to attack or system failure. Although the ability to resist and recover from failure is important to many systems and processes, the robustness and resiliency of workflow management systems has received little attention in literature. The authors propose a novel workflow modeling framework suing high-level Petri nets (PNs). The proposed framework is capable of both modeling structure and providing a wide range of qualitative and quantitative analysis. The concepts of self-protecting and self-healing systems are captured by the robustness and resiliency measures proposed in this study. The proposed measures are plotted in a Cartesian coordinate system; a classification scheme with four quadrants (i.e., possession, preservation, restoration, and devastation) is proposed to show the state of the system in terms of robustness and resiliency. The authors introduce and overall sustainability index for the system based on the theory of displaced ideals.

15. SUBJECT TERMS
resiliency, robustness, modeling

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					SHELBY BARRETT
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (*Include area code*)
U	U	U	UU	47	N/A

Modeling Operational Robustness and Resiliency with High-level Petri Nets

Madjid Tavana, Ph.D.*
Professor, Management Information Systems
Lindback Distinguished Chair of Information Systems
La Salle University
Philadelphia, PA 19141, U.S.A.
Phone: +1.215.951.1129
Fax: +1.267.295.2854
Email: tavana@lasalle.edu
URL: http://lasalle.edu/~tavana

Timothy E. Busch, Ph.D.
Air Force Research Laboratory
AFRL/IFSB
525 Brooks Road
Rome, NY 13441
Phone: (315) 330-1486
Email: timothy.busch@rl.af.mil

Eleanor L. Davis
Institute for Computational Engineering and Science
University of Texas at Austin
1 University Station C0200
Austin, Texas 78712
Email : eleanorldavis@gmail.com

***Corresponding Author**

UNCLASSIFIED

Approved for Public Release
88ABW-2010-3929

ACKNOWLEDGEMENT

This research was supported by the U.S. Air Force Research Laboratory grant number FA8750-09-2-0157

Accepted Article in Press
International Journal of Knowledge-Based Organizations

Modeling Operational Robustness and Resiliency with High-level Petri Nets

ABSTRACT

Military operations are highly complex workflow systems that require careful planning and execution. The interactive complexity and tight coupling between people and technological systems has been increasing in military operations, which leads, on the one hand, to improved efficiency and on the other, a greater vulnerability to mission accomplishment due to attack or system failure. Although the ability to resist and recover from failure is important to many systems and processes, the robustness and resiliency of workflow management systems has received little attention in literature. In this study, we propose a novel workflow modeling framework using high-level Petri nets (PNs). The proposed framework is capable of both modeling structure and providing a wide range of qualitative and quantitative analysis. The concepts of self-protecting and self-healing systems are captured by the robustness and resiliency measures proposed in this study. The proposed measures are plotted in a Cartesian coordinate system; a classification scheme with four quadrants (i.e., possession, preservation, restoration, and devastation) is proposed to show the state of the system in terms of robustness and resiliency. We also introduce an overall sustainability index for the system based on the theory of displaced ideals. We demonstrate the application of our methodology in the evaluation of an air tasking order generation system at the United States Air Force.

Keywords: High-level Petri Net; Workflow Management System; Resiliency; Robustness; United States Air Force.

Modeling Operational Robustness and Resiliency with High-level Petri Nets

Madjid Tavana, La Salle University, U.S.A.
Timothy E. Busch, Air Force Research Laboratory, U.S.A.
Eleanor L. Davis, University of Texas at Austin, U.S.A.

INTRODUCTION

A workflow management system is a set of activities involving the coordinated execution of multiple tasks performed by different processing entities (Casati et al., 1995). Different techniques may be used for workflow modeling depending on the goals and objectives. There are two general categories of workflow management systems (Mentzas et al., 2001), communication-based and activity-based techniques. The communication-based techniques assume that the objective of business process reengineering is to improve customer satisfaction (Winograd and Flores, 1987). Activity-based techniques focus on modeling the tasks involved in a process and their dependencies (McCarthy and Sarin, 1993). Despite their popularity and wide-spread application, workflow management systems still suffer from lack of standards and an agreed-upon modeling method (Salimifard and Wright, 2001). Van der Aalst et al. (1994) have criticized that workflow management systems have (i) no needed functionality, (ii) no clear set of definitions, and (iii) no general conceptual model.

Although the ability to resist and recover from failure is important to many systems and processes, the robustness and resiliency of workflow management systems has received little focus in literature. The idea of self-protecting and self-healing systems is frequently discussed in relation to computer networks, but it has not been addressed thoroughly in other contexts despite its potential relevance (Dragoni et al. 2009). For example, military operations are highly interactive and complex systems that require efficient and effective command and control to be

successful. The interactive complexity and tight coupling between people and technological systems has been increasing in military operations, which leads to unpredictability of operations and inevitably to failures.

Robustness is a property intimately associated with the organization's capacity to avoid failure while resiliency is the organization's ability to recover from failure. A deep understanding of robustness and resiliency has emerged from the study of many high-reliability organizations such as nuclear power production, aviation, space exploration, healthcare, air traffic control and chemical production (Gauthier et al., 2006; Perrow, 1999). The major interest in high-reliability organizations comes from their capacity to achieve high performance while operating in hazardous conditions (Weick and Sutcliffe, 2001). Robustness and resiliency are not technological or organizational properties but a combination of both. They are the combination of technological features, such as redundancy, protection systems, and good engineering design (Leveson et al., 2009), with organizational features such as sense-making and training (Weick, 2001). The recent studies on robustness and resiliency emphasize the integration between the organizational and technological views in complex socio-technical systems (Hollnagel et al., 2006). A good balance between robustness and resiliency should be envisaged (Nomura et al., 1998). We define robustness as the capacity of a system to avoid failure in the face of unexpected events, and resiliency as the ability of the system to recover from failure after it occurs. Robustness is important to keep the organization under control; resiliency is necessary to react to hazards.

Recently, Petri nets (PNs) have been used for workflow modeling (van der Aalst et al., 2000; van der Aalst et al., 1998). Although different modeling techniques can be used for workflow modeling, PNs are the only formal technique capable of both modeling structure and

providing a wide range of qualitative and quantitative analysis. PNs allow a graphical representation to ease the understanding of the modeled system. They can also be used to formally analyze, verify, and validate the model (van der Aalst, 1997; Desel, 2000).

Zisman (1977) pioneered the concept of workflow management systems by applying PNs to represent office procedures. His work was followed by Ellis (1979), who introduced an extension of the classical PNs called information control nets, and applied it to model office information systems. Since the early 1980s, workflow modeling based on PNs has captured a great deal of interest and is still an active area of research in workflow management system (van der Aalst, 1998).

Van der Aalst (1998) has shown that a PN is a suitable tool for modeling the process, case, and resource dimensions of a workflow management system. Different classes of PNs can be used depending on the modeling objectives. A classical PN is used for modeling the structural aspects of a single workflow; high-level PNs are those expanded with notions of time, color, and other features. A timed PN is used for analyzing the time-dependent behavior of a workflow, while a colored PN is used for a workflow management system where a number of instances of the same or different workflows has to be modeled and traced. Using PNs, each workflow task is represented by a corresponding transition. Places represent pre- and post-conditions or resources needed for a task to be performed. Arcs represent logical relationships between tasks and the flow of the work. In addition to ordinary PNs, timed PNs, and colored PNs, stochastic PNs and fuzzy PNs are also used to model engineering and business systems (Chen et al., 2001; Huang et al., 2008; Lee et al., 2001; Li and Lara-Rosano, 2000; Liu et al., 2007; Murata, 1989). PNs are well-suited for the design, specification, and formal verification of complex systems (Sakthivel and Tanniru, 1988-89). PNs, with their graphical and precise nature and their firm

mathematical foundation, are commonly used to model many complex systems. Their graphical appearance allows for models that are easy to understand, while their formal semantics allow for precise and unambiguous descriptions. PNs are particularly considered a rich, versatile, and dynamic graphical tool in the development and validation of workflow management systems (Mehrez et al. 1995, Wong 2001).

PNs were initially defined by Carl Adam Petri (Petri 1962) and later refined and named after him by Holt (1971). Peterson (1981) elegantly discusses the dynamic behavior of PNs, while Murata's tutorial review paper (Murata 1989) provides a thorough review of the history and applications of PNs. PNs are abstract, formal models used to describe and analyze the flow of information and control in systems, particularly systems that exhibit asynchronous and concurrent activities, with conditions for the performance of events within the system (Bullers 1991). They have been proven to be useful for the modeling and analysis of several classes of systems including web-based systems (Huang et al. 2008, Zhovtobryukh 2007), communication systems (Berthelot and Terrat 1982), knowledge-based systems (Jantzen 1980), simulation systems (Piera et al. 2004), and process control systems (Bruno and Marchetto 1986).

The remainder of this paper is organized as follows. The next section presents PN principles and formalism followed by a discussion of the PN-based workforce management system. We go on to propose expansions to the traditional PNs relevant to our discussion. We then define our measures of robustness and resiliency for the PNs and provide a graphical representation of the model. We next present a numerical example and an application of the proposed model to an air tasking order generation process of the U.S. Air Force. Finally, we present our conclusions and future research directions.

BACKGROUND

PN principles and formalism

Classical PNs as defined by Petri (1962) and further discussed by Peterson (1981) are identified by 5-tuples (P, T, I, O, μ) where $P = \{p_1, p_2, ..., p_m\}$ is a set of places; $T = \{t_1, t_2, ..., t_n\}$ is a set of transitions; $[I \subseteq P x T]$ is the input function from P to T; $[O \subseteq T x P]$ is the output function from T to P; and μ, called marking, is a function that defines a mapping from a set of places P to Z (here Z denotes the set of all nonnegative integers).

$\mu : P \rightarrow Z \; where \; \mu_i = \mu(p_i) \in Z, p_i \in P \; ; i = \{0,1,...,m\}.$

Mathematically, a PN is a directed bipartite graph with two different types of node called *places* and *transitions*. A place *p* is represented by a circle and a transition *t* is represented by a rectangle. The nodes are connected through directed *arcs*. Directed arcs from *p* to *t* create *input places*, while directed arcs from *t* to *p* create *output places*. Input places are a set of places that can fire a transition, while output places are a set of places that are associated with the results (outputs) of a transition. Only the static properties of a system are represented by a PN structure, however, dynamic properties result from PN execution. Execution requires the use of tokens or markings (denoted by dots) associated with places. Each place contains zero or many tokens drawn as black dots; the marking of the PN is the distribution of its tokens. The execution of a PN may affect the number of tokens in a place. A transition is called *enabled* when each of its input places has enough tokens. A transition can be *fired* only if it is enabled. When a transition is fired, tokens from input places are used to produce tokens in output places. We will use the PN shown in Figure 1 to illustrate the classical PN.

Insert Figure 1 Here

We begin with one token in p_1. This enables transition t_1, which fires, producing tokens in p_2 and p_3. Transitions t_2 and t_3 are both enabled; either may fire. We choose to examine the case in which t_2 fires. Since the tokens in p_2 and p_3 are consumed, transition t_3 is no longer enabled. The firing of t_2 produces tokens in p_1 and p_4, returning the system to its initial state with the addition of a token in p_4. Transition t_1 is again enabled and fires. Transitions t_2 and t_3 are then enabled once more by the tokens in p_2 and p_3. This time, we consider the case in which t_3 fires. The resulting token in p_5, combined with the previously produced token in p_4, enables transition t_4. Firing t_4 consumes these tokens and produces a token in p_6, the output of the system.

The PN in Figure 1 consists of six places $\left(P = \{p_1, p_2, p_3, p_4, p_5, p_6\}\right)$ and four transitions $\left(T = \{t_1, t_2, t_3, t_4\}\right)$. The input and output mapping of this PN is given as:

$I(t_1) = \{p_1\}$ $\qquad$ $O(t_1) = \{p_2, p_3\}$

$I(t_2) = \{p_2, p_3\}$ $\qquad$ $O(t_2) = \{p_1, p_4\}$

$I(t_3) = \{p_3\}$ $\qquad$ $O(t_3) = \{p_5\}$

$I(t_4) = \{p_4, p_5\}$ $\qquad$ $O(t_4) = \{p_6\}$

Any PN can be specified in matrix form as a *D*- Matrix with *m* rows and *n* columns, where *m* is the number of *transitions* and *n* is the number of *places* in the PN. For each position $[i, j]$ in the matrix, a 1 is placed in the position if transition *i* receives input from place *j*. A 0 is placed if transition *i* does not receive input from place *j*:

$$D- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Similarly, a $D+$ Matrix with m rows and n columns can be constructed, where m is the number of *transitions* and n is the number of *places* in the PN. For each position [i,j] in the matrix, a 1 is placed in the position if transition i produces output to place j. A 0 is placed if transition i does produce output to place j:

$$D+ = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The Composite Change Matrix (Matrix D) can be computed by subtracting D- from $D+$:

$$[D+]-[D-]=[D]$$

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}$$

In addition, a $1 \times m$ matrix, representing the firing of the PN, can be constructed. In each position $[1, j]$ we place the number of times transition j is to fire. The first Transition Matrix for our PN is:

Transition Matrix $= [1 \quad 0 \quad 0 \quad 0]$, t_1 firing because of the token in p_1.

Finally, a $1 \times n$ matrix is constructed showing the current marking of the PN. In each position $[1, j]$, the identified number of tokens in position j is placed. The Marking Matrix for our PN is:

$\text{Marking Matrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$, given one token in p_1.

The marking of the PN after the transition specified in the Transition Matrix (Next Marking) can be found as:

$$[\text{Transition Matrix}][D] + [\text{Marking Matrix}] = [\text{Next Marking}]$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

From a modeling perspective, PNs can be characterized as a "conceptual model with analytical qualities," where a "conceptual model" generally represents a graphical approach, while an "analytical model" expresses functional and mathematical relationships (Mehrez et al. 1995). As a hybrid modeling approach, PNs can display several important properties, including the ability to model situations for simulation analysis and describe system behavior (Kim et al. 2001).

There are a number of properties associated with PNs. Some of these properties describe the overall structure of the PN, while others are related to specific markings. Reachability, liveness, and boundedness are among the most commonly discussed properties. One marking is said to be reachable from another if there is a firing sequence that leads from the first marking to the second. For a live initial marking, no matter what marking has been reached from it, it is ultimately possible to fire any of the net's transitions. A PN is bounded if its set of reachable markings is finite. If a PN is persistent or non-interrupted, for any two enabled transitions, the firing of one cannot disable the other (if it can, the two transitions are in conflict). Depending on the PNs studied, it may be relevant to discuss the pureness, fairness, conservativeness,

consistency, or many other features of the net. See Atherton and Borne (1992), Valk and Girault (2003), and Cassandras and Lafortune (1999) for a detailed description of these properties.

PN-based workflow management

Workflow management systems are complex artifacts that are expensive to build and difficult to validate, especially when the components of the system exhibit a variety of situations including sequential execution, conflict, concurrency, synchronization, merging, confusion, or prioritization (Balduzzi et al. 2000, Mehrez et al. 1995). *Sequential execution* refers to the processing of precedence constraints; *conflict* refers to mutually exclusive activities or results; *concurrency* refers to simultaneous task operation; *synchronization* refers to multiple resource usage in a single operation; *merging* refers to multiple precedence constraints; *confusion* refers to the combination of conflict and concurrency; and *prioritization* refers to the determination of the priorities of activities.

In our study, we use high-level timed PNs expanded in several ways. The ability of PNs to model the workflow primitives identified by the Workflow Management Coalition was shown by van der Aalst (1996). We expand those primitives to include alternate splits, where the dotted paths serve as secondary routes to be taken if there is an error in the primary path (e.g. a broken arc or a failed transition). The modeling of these "back-up plans" is useful in the study of the robustness and resiliency of the PN (See Figure 2).

Insert Figure 2 Here

While classical PNs have no concept of time, timed PNs introduce delay times associated with the transitions and/or places (Atherton and Borne, 1992). We assign best-case (Ta_{t_i}), worst-case (Tb_{t_i}), and most-likely (Tm_{t_i}) times to each transition. We also introduce the idea of repair times, the amount of time needed to fix the transition if a problem is encountered.

PN WORKFLOW EXTENSIONS

Probability of occurrence consideration

The existence of alternate paths leading to an output increases the robustness of the network. If part of the network fails, a back-up route can be taken. Failures of proper behavior in transitions or places can be modeled by the incapacitation of arcs. An arc from place i to transition j is denoted l_{p_i,t_j}, while one in the opposite direction would be written l_{t_j,p_i}. We do not distinguish between arcs from places to transitions and arcs from transitions to places; for example, the notation Pa_{p_i,t_j} is considered to cover both cases. In general, the smaller the overall expected incapacitation probability, the more robust the network. Several measures are considered in the calculation of this overall probability:

- Optimistic breakage probability (Pa_{p_i,t_j}): The optimistic-case scenario considers the lowest probability that an arc can be broken ($0 \le Pa_{p_i,t_j} \le 1$).

- Pessimistic breakage probability (Pb_{p_i,t_j}): The pessimistic-case scenario considers the highest probability that an arc can be broken ($0 \le Pb_{p_i,t_j} \le 1$).

- Most-likely breakage probability (Pm_{p_i,t_j}): The most-likely-case scenario considers the most-likely probability that an arc can be broken ($0 \le Pm_{p_i,t_j} \le 1$; $Pa_{p_i,t_j} \le Pm_{p_i,t_j} \le Pb_{p_i,t_j}$).

- Expected breakage probability (Pe_{p_i,t_j}): We assume a beta probability distribution for the probability estimates. For a beta distribution the expected value for each arc can be approximated using the following weighted average:

$$Pe_{p_i,t_j} = (Pa_{p_i,t_j} + 4\,Pm_{p_i,t_j} + Pb_{p_i,t_j})/6. \quad (1)$$

If the breakage probabilities of the arcs are known, the optimistic, pessimistic, and most-likely breakage probabilities are not necessary. They should only be used as tools to help approximate the expected breakage probabilities.

The incapacitation probability of a network is the probability that the network will be rendered nonfunctional. It can be calculated by examining the structure of the network and the breakage probabilities of its constituent arcs. If every arc must be complete for the network to function, then the incapacitation probability is easily calculated by subtracting the probability that none of the arcs are broken from one (i.e., certainty). The (expected) probability that none of the arcs are broken is the product of $1 - Pe_{p_i,t_j}$ for all expected breakage probabilities Pe_{p_i,t_j}, i.e. :

$$P_{\text{incapacitation}} = 1 - \prod (1 - Pe_{p_i,t_j}) \quad (2)$$

This calculation becomes more complex when there are alternate paths to the output and so the breakage of one arc does not necessitate the failure of the entire network. Instead, the network is incapacitated only when all of the paths are broken. To simplify this calculation, we recommend dividing the PN into zones surrounding each transition. Zones should contain either primary or back-up arcs, but not both. A failure in any of the arcs in each zone will incapacitate at least one path. By considering the probability of the failure of each individual zone and various combinations of zones, the incapacitation probability can be found. See the numerical example.

Completion time consideration

The process completion time is often vital information when studying workflow systems. For a timed PN, each transition takes a certain amount of time. Given optimistic, pessimistic, and

most-likely running times for each transition, we can determine the expected running time of the system as a whole, as well as the best- and worst-case scenarios. The greater the difference between the expected and pessimistic values, the less robust the network, since the repercussions of problems are greater.

- Optimistic transition time (Ta_{t_i}): The optimistic-case time is the shortest time the transition can occur in.
- Pessimistic transition time (Tb_{t_i}): The pessimistic-case scenario is the longest time the transition may occur in.
- Most-likely transition time (Tm_{t_i}): The most-likely-case scenario is the time that it will most likely take the transition to occur ($Ta_{t_i} \le Tm_{t_i} \le Tb_{t_i}$).
- Expected transition time (Te_{t_j}): We assume a beta probability distribution for the time estimates. For a beta distribution the expected value for each transition can be approximated using the following weighted average:

  $$Te_{t_i} = (Ta_{t_i} + 4\,Tm_{t_i} + Tb_{t_i})/6 . \quad (3)$$

  The variance for the transition time is given by:

  $$v_{t_i} = [(Tb_{t_i} - Ta_{t_i})/6]^2 . \quad (4)$$

The system time is the time that it takes to go from the initial input to the final output. Along each possible path from the input to the output, the path time Tp is the sum of the expected transition times of the transitions along that path. If there is only one path to the output, the path time is the system time. If there are concurrent transitions that are all required, the expected system time is the longest of the concurrent path times. Adding all of the transition time

variances along a path will give the variance associated with that path time. This information can be used to determine the probability that the actual system time will be less or more than the expected system time.

If there are alternate paths to the output, a weighted average of the resulting path times can be taken to find the overall expected system time. In this calculation, it is assumed that some path to the output will be found; network incapacitation is not considered, since that would imply an infinite time until completion. The probability of taking each path is multiplied by its path time. These values are then added together, giving the overall expected system time:

$$Ts = Tp_1 \times P(\text{path}_1) + \ldots + Tp_n \times P(\text{path}_n) \quad (5)$$

Similarly, we can define best-case system times and worst-case system times, using the best- and worst-case path times rather than the expected path times.

Repair time consideration

We introduce another property to the transitions in order to study PN resiliency. In addition to best-case, worst-case, and most-likely running times, we consider the repair time and its best, worst, and most-likely estimates.

- Optimistic repair time (Ra_{t_i}): The optimistic-case time is the shortest time the transition can be repaired.
- Pessimistic repair time (Rb_{t_i}): The pessimistic-case scenario is the longest time it may take to repair the transition.
- Most-likely repair time (Rm_{t_i}): The most-likely-case scenario is the time that it will most likely take to repair the transition ($Ra_{t_i} \le Rm_{t_i} \le Rb_{t_i}$).

- Expected repair time (Re_{t_j}): We assume a beta probability distribution for the repair time estimates. For a beta distribution the expected value for each repair can be approximated using the following weighted average:

$$Re_{t_i} = (Ra_{t_i} + 4\,Rm_{t_i} + Rb_{t_i})/6\,. \tag{6}$$

The variance for each repair time is given by:

$$v_{t_i} = [(Rb_{t_i} - Ra_{t_i})/6]^2 \tag{7}$$

We define path and system repair times similarly to the times defined previously. Path repair times Rp are the times to complete repairs in all of the transitions along that path (this assumes failure of every transition). The path repair time variance can be found by adding all of the repair time variances along that path. To calculate the system repair time, we use a weighted average of the path repair times with the probability of taking each path. The overall expected system repair time is:

$$Rs = Rp_1 \times P(\text{path}_1) + \ldots + Rp_n \times P(\text{path}_n) \tag{8}$$

We can also define best-case, worst-case, and most-likely system repair times Rs_{best}, Rs_{worst}, and $Rs_{\text{most-likely}}$ using those values.

ROBUSTNESS AND RESILIENCY MEASURES

Robustness measure

The network incapacitation probability is combined with the system time to create an overall robustness measure. The less vulnerable the system time is to problems, the more robust the network will be. Therefore, we examine the ratio of the best-case system time to the worst-case system time, $\frac{Ts_{\text{best}}}{Ts_{\text{worst}}}$.

The smaller the value of this ratio, the greater the potential ramifications of disruption to the system, and so the less robust the system is. We define κ as the combination of time and probability used to measure robustness:

$$\kappa = \frac{Ts_{\text{best}}}{Ts_{\text{worst}}} - P_{\text{incapacitation}} \tag{9}$$

Here, κ ranges from -1, least robust--where the worst-case time is infinitely greater than the best-case time and failure is certain--to 1, most robust, where the best-case time and worst-case time are equal and failure is impossible. This measure can be adjusted depending on its application. For example, an organization may wish to discard the time factor or give it a minimal weight if its goal is simply to complete the process regardless of the time cost.

Resiliency measure

We combine the expected system time Ts with the expected system repair time Rs to measure resiliency γ. The system repair time assumes failure in each transition, regardless of the probability of failure associated with it. This is reasonable, since resiliency is concerned with the ease of recovery after disaster has already occurred, no matter how likely that disaster was. Resiliency is defined as:

$$\gamma = \frac{Ts - Rs}{Ts + Rs} \tag{10}$$

Like our robustness measure κ, γ ranges from -1 (the repair time is infinitely greater than the system time) to 1 (the system time is infinitely greater than the repair time). If the repair time is longer than the system time, γ will be negative and the system will not be particularly resilient. If the repair time is shorter than the system time, γ will of course be positive; the system is better able to cope with disaster and is thus more resilient.

A GRAPHICAL PERSPECTIVE

When both robustness and resiliency values have been calculated, the ordered pair (κ, λ) can be plotted on the plane with range and domain $[-1,1]$. By choosing threshold values $\hat{\gamma}$ and $\hat{\kappa}$, the plane can be divided into four quadrants. These threshold values are left to the discretion of the user, since the decision of whether certain results indicate robustness or resiliency is situation-dependent. For example, in some cases, if the repair time is more than twice that of the expected system time, the system will not be considered resilient, so $\hat{\gamma} = -\frac{1}{3}$.

After the thresholds have been chosen, the plane will be divided into the Possession Quadrant, the Preservation Quadrant, the Restoration Quadrant, and the Devastation Quadrant (See Figure 3).

Insert Figure 3 Here

- **Possession Quadrant:** Networks in this quadrant are both robust and resilient. They are unlikely to encounter obstacles that will disable the system; if they do, the system will recover without significant difficulty.
- **Preservation Quadrant:** In this quadrant, networks are robust but not resilient. These networks are unlikely to fail. However, if an unforeseen disaster occurs, the road to recovery will be long and hard.
- **Restoration Quadrant:** Restoration Quadrant networks are resilient but not robust. They may fail again and again, but repairs are quick. Before long, it will be back on its feet.
- **Devastation Quadrant:** These networks are neither robust nor resilient. They are very susceptible to failure, and once they have failed, recovery is difficult. Significant changes are needed to improve these systems.

As both γ and κ range from -1 to 1, we have the ideal point (1,1), where both robustness and resiliency are at their maximum, and the nadir (-1,-1) at which both are at their minimum values. The displacement of a given point from the ideal point is given by the Euclidean distance formula, $Distance = \sqrt{(\gamma - \gamma_0)^2 + (\kappa - \kappa_0)^2}$. We define $(\kappa_0, \gamma_0) = (1,1)$. The overall sustainability index (ω) is the ratio of this distance to the maximum distance possible (i.e. the distance between the ideal point and the nadir).

$$\omega = \frac{\sqrt{(\gamma - 1)^2 + (\kappa - 1)^2}}{2\sqrt{2}} \tag{11}$$

The lower the sustainability index, the closer the system resiliency and robustness are to the ideal.

APPLICATION

Numerical example

In the network shown in Figure 4, if any arc is broken, the network will fail. The probabilities given for each arc are the expected breakage probabilities. The other probabilities from which the expected values are derived are given in Table 1. These are provided as an example of a potential data set, although only one probability value is required for each arc.

Insert Figure 4 and Table 1 Here

We can then calculate the incapacitation probability (probability of network failure) as:

$$P_{\text{incapacitation}} = 1 - \prod (1 - Pe_{p_i, t_j}) = 1 - .9 \times .8 \times .7 \times .6 = .6976$$

The times above the transitions in Figure 4 are the expected transition times, while the times below the transitions are the expected repair times. Since $Te_{t_1} = 2$ and $Te_{t_2} = 6$, the expected system time is 8. Similarly, the expected system repair time is 15. In Table 2, we can

see the best-case, worst-case, and most-likely times associated with each transition. These values give a best-case system time of 3 and a worst-case system time of 3.4.

In Figure 5, transitions t'_1 and t'_2 are added as alternate transitions. If the arcs to or from transitions t_1 and t_2 are broken, the alternate arcs to the alternate transitions will be taken instead. It is clear from the figure that multiple arcs can be broken without incapacitating the network. There are four possible paths from the input to the output. We define them as follows: Path 1 goes through zones 1 and 2; path 2 goes through zones 1 and 4; path 3 goes through zones 3 and 2, and path 4 goes through zones 3 and 4. The network is incapacitated only if arcs in all of these paths fail.

Insert Figure 5 and Table 2 Here

We first divide the network into sections to calculate the probability of network failure. Clearly, there are four possible zones in which the failure of any arcs incapacitates a possible path (See Figure 5). These zones are sets of arcs immediately adjacent to each transition. If, for example, either arc l_{p_1,t_1} or l_{t_1,p_2} is broken (or both), zone 1 has failed and the final path must pass through transition t'_1. It is not necessary for all four zones to contain failures for the system to fail.

For each zone, it is simplest to find the probability that neither arc will fail, then subtract that result from one.

- Probability that zone 1 fails: $1-.9\times.8=.28$
- Probability that zone 2 fails: $1-.7\times.6=.58$
- Probability that zone 3 fails: $1-.4\times.6=.76$
- Probability that zone 4 fails: $1-.5\times.3=.85$

Finally, we consider the seven possible ways for the network to fail, since the failure of each zone on its own is not an independent event. All of the zones could fail, exactly three zones could fail, or exactly two zones could fail.

- Probability that zones 1,2,3,4 fail: $.28 \times .58 \times .76 \times .85 = .105$
- Probability that zones 1,2,3 fail while zone 4 does not: $.28 \times .58 \times .76 \times .15 = .017$
- Probability that zones 1,2,4 fail while zone 3 does not: $.28 \times .58 \times .24 \times .85 = .033$
- Probability that zones 1,3,4 fail while zone 2 does not: $.28 \times .42 \times .76 \times .85 = .076$
- Probability that zones 2,3,4 fail while zone 1 does not: $.72 \times .58 \times .76 \times .85 = .270$
- Probability that zones 1,3 fail while zones 2,4 do not: $.28 \times .42. \times 76 \times .15 = .013$
- Probability that zones 2,4 fail while zones 1,3 do not: $.72 \times .42 \times .24 \times .15 = .011$
- The probability of network failure is then the sum of these probabilities:

$$P_{\text{incapacitation}} = .105 + .017 + .033 + .076 + .270 + .013 + .011 = .525$$

Since in this example there are no concurrent transitions, simply alternate processes, to calculate the expected system time we need only to calculate the probabilities and expected times of each path. The path times are easily calculated for this basic example: $Tp_1 = 8$, $Tp_2 = 7$, $Tp_3 = 10$, and $Tp_4 = 9$.

In order to calculate the probability of taking each path, we assume that at least one path will work. For example, we make the probability of passing through zone 2 the complement of passing through zone 1: if zone 1 is incapacitated, zone 2 will not be, because otherwise the network will fail.

The probability of taking path 1 is then the probability that both zones 1 and 2 will not be disabled: $.72 \times .42 = .3024$

The probability of taking path 2 is the probability that zone 1 will not be disabled and zone 2 will be: $.72 \times .58 = .4176$

The probability of taking path 3 is the probability that zone 1 will be disabled and zone 2 will not be: $.28 \times .42 = .1176$

The probability of taking path 4 is the probability that both zones 1 and 2 will be disabled $.28 \times .42 = .1624$

Using the probabilities as weights, we can then calculate the overall expected system time:

$$Ts = Tp_1 \times P(\text{path}_1) + Tp_2 \times P(\text{path}_2) + Tp_3 \times P(\text{path}_3) + Tp_4 \times P(\text{path}_4) = .3024 \times 8 + .4176 \times 7 + .1176 \times 10 + .1624 \times 9 = 7.98$$

In addition to the overall expected system time, we can ascertain the best-case and worst-case times using the optimistic and pessimistic transition times. The lowest path time calculated using the optimistic transition times is the best-case system time, while the highest path time using the pessimistic transition times is the worst-case system time.

$$Ts_{\text{best}} = Tp_{1,\text{best}} \times P(\text{path}_1) + Tp_{2,\text{best}} \times P(\text{path}_2) + Tp_{3,\text{best}} \times P(\text{path}_3) + Tp_{4,\text{best}} \times P(\text{path}_4) = .3024 \times 3 + .4176 \times 3 + .1176 \times 3 + .1624 \times 3 = 3$$

$$Ts_{\text{worst}} = Tp_{1,\text{worst}} \times P(\text{path}_1) + Tp_{2,\text{worst}} \times P(\text{path}_2) + Tp_{3,\text{worst}} \times P(\text{path}_3) + Tp_{4,\text{worst}} \times P(\text{path}_4) = .3024 \times 13.4 + .4176 \times 16 + .1176 \times 25 + .1624 \times 21 = 17.08$$

Similarly, we determine the system time variance by adding the variances along each path, then finding the weighted sum.

$$Tv = .3024 \times 1.94 + .4176 \times .6 + .1176 \times 7.22 + .1624 \times 5.88 = 2.6412$$

We can use this value and its square root, the standard deviation, to find the probability of various ranges of system times. For example, assuming a normal distribution, there is a 98% probability that the system time will be less than three standard deviations above the expected system time, i.e., 12.8556. The same calculations can be done with the repair time variances.

The repair times associated with the network are given in Table 3. The expected system repair time is:

$$Rs = Rp_1 \times P(\text{path}_1) + Rp_2 \times P(\text{path}_2) + Rp_3 \times P(\text{path}_3) + Rp_4 \times P(\text{path}_4) =$$
$$.3024 \times 15 + .4176 \times 7 + .1176 \times 13 + .1624 \times 5 = 9.8$$

Insert Table 3 Here

From the values calculated above, we can find the robustness and resiliency of the system:

$$\gamma = \frac{Ts - Tr}{Ts + Tr} = \frac{7.98 - 9.8}{7.98 + 9.8} = -.102$$

$$\kappa = \frac{Ts_{\text{best}}}{Ts_{\text{worst}}} - P_{\text{incapacitation}} = \frac{3 - 17.08}{3 + 17.08} = -.701$$

With $(\kappa, \gamma) = (-.701, -.102)$, the sustainability index is:

$$\omega = \frac{\sqrt{(-.102 - 1)^2 + (-.701 - 1)^2}}{2\sqrt{2}} = .717.$$

This shows that the system is 71.7% displaced from the ideal.

If we assume that the network is functioning without difficulties, we can model it mathematically using the Composite Change Matrix and other elements introduced earlier by excluding the backup paths from these elements. Allowing the possibilities of breakages increases the complexity of mathematical representation. That issue is not addressed in this study but is rather left as an area for future research.

Air tasking order generation case study

The Air and Space Operations Center is the senior element in the United States Air Force command and control system and is generally comprised of five divisions. The combat operations division is responsible for the generation of the Air Tasking Order (ATO). The Air

Tasking Order lays out the specific tasks and timelines necessary to achieve the commander's objectives.

There are several steps involving various teams in the generation of the ATO. Initially, the commander expresses his objectives and guidance in the form of an Air Operations Directive (AOD). This directive articulates a desired set of conditions for a given point in time and the purpose those conditions will support. The other input to the ATO generation process is the target nominations list. Targeting is the process of selecting and prioritizing targets and matching the appropriate response to them, considering operational requirements and capabilities. This targeting process is integrated across the land, sea, and air components.

The Target Effects Team (TET) correlates target nominations. It screens nominated targets and ensures that once attacked, they create the desired effects that meet the commander's guidance as delineated in the AOD and verifies that chosen measures of effectiveness will accurately evaluate progress and can be observed. They also prioritize nominated targets based on the best potential for creation of the commander's desired effects and the components' priorities and timing requirements. The product of this effort when approved by the commander is the Joint Integrated Prioritized Target List (JIPTL). The JIPTL provides the basis for weaponeering assessment activities which recommend aimpoints, weapon systems and munitions, fusing, target identification and description, desired effects of target attack, probability of creating the desired effect and collateral damage concerns. This information is passed to the Master Air Attack Plan (MAAP) team, which matches available resources to the prioritized target list. It accounts for all support requirements such as air refueling and other factors that might affect the plan.

Once the MAAP is approved, the ATO production team develops the detailed schedule taking into account airspace control measures (ACO) and special instructions (SPINS). The level of detail balances combat effectiveness with the safe, orderly, and expeditious use of the airspace. The completed and approved ATO is then disseminated to the Combat Operations division and relevant forces for execution. The traditional time for this process is 72 hours and there are 3 ATOs in production at any given point in time.

Figure 6 depicts a simplified PN of this process. It is worth noting that the ATO generation process is dependent on networks of software applications and databases. These in-turn are dependent on the underlying communications networks. While the case presented here focuses on the high level workflow and the notions of robustness and resiliency with regard to the organizational aspects of the system, we are equally interested in the cross layer problem as well. Figure 7 shows the system with notional backup processes in place.

Insert Figures 6 and 7 Here

Tables 4 and 5 give the transition and repair times associated with this network; the breakage probabilities for each link are shown within the figures. Using the data provided, we can find the robustness and resiliency of both the initial and modified systems. Because of the greater complexity of this example (there are in this case 16 different paths from input to output to consider), only the final results will be given in Table 6.

Since the numbers in this example are chosen solely for demonstration purposes, we also choose arbitrary threshold values of $\hat{\gamma} = 0$ and $\hat{\kappa} = 0$. In Figure 8, we can see that the initial system was neither robust nor resilient. The robustness and resiliency values both increased with the addition of backup transitions with low repair times. These modifications move the system from the Devastation Quadrant to the Possession Quadrant.

Insert Tables 4-6 and Figure 8 Here

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Although robustness and resiliency are essential attributes of good system design, there has been very little discussion of these properties in workflow management system literature. The necessity for readiness and the ability to cope with the possibility and reality of failure in complex systems makes this an important area for future workflow management studies. In this paper, we showed the utility of PNs in mathematical and graphical representation of such systems and introduced an expanded PN framework with alternate paths and repair times. We also proposed measures of robustness and resiliency applicable to such expanded PN systems. The measures proposed in this study are structured and yet flexible enough to be adjusted for specific circumstances and systems, since resiliency and robustness are relative, not absolute.

The framework proposed in this paper could be extended to:

- Account for a wider range of economical and non-economical factors beyond those covered in this study through the consideration of multiple incomparable and often competing factors such as cost of resources and opportunity costs
- Account for other concepts relating to resiliency and robustness such as differing levels of importance in transitions.
- Consider the interdependency of resiliency and robustness. The two features may not necessarily function independently as changes to one may affect the other.
- Incorporate interval and fuzzy data. Although crisp data are fundamentally indispensable for determining the robustness and resiliency measures, the observed values in the real-world problems are often imprecise or vague. These imprecise or vague data can be suitably characterized with fuzzy set theory.

- Formulate mathematical representation of the expanded PN models, since allowing the possibilities of breakages increases the complexity of matrix representation.
- Encompass a computer implementation of the proposed framework. An automated system will provide the capability for continuous monitoring of the resiliency and robustness in large systems.

In this study we have laid the groundwork for the consideration of resiliency and robustness in workflow management systems. We hope that the concepts introduced here will provide inspiration for future research.

REFERENCES

Atherton, D.K., & Borne, P. (1992). *Concise encyclopedia of modeling and simulation*, Pergamon Press, Oxford.

Balduzzi, F., Giua, A., & Menga, G., (2000). First-order hybrid Petri nets: a model for optimization and control, *IEEE Transactions on Robotics and Automation*, *16*(4), 382-399.

Berthelot, G., & Terrat, R. (1982). Petri-net theory for correctness of protocols. *IEEE Transactions on Communication*, *30*(12), 2497–2505.

Bruno, G., & Marchetto, G. (1986). Process-translatable Petri nets for the rapid prototyping of process control systems. *IEEE Transactions on Software Engineering*, *12*(2), 346–357.

Bullers, W.I. (1991). A tripartite approach to information systems development, *Decision Sciences*, *22*(1), 120-135.

Casati, F., Ceri, S., Pernici, B., & Pozzi, G. (1995). *Conceptual modelling of workflows*, Lecture Notes in Computer Science, vol. 1021, pp. 341–354, Springer, Berlin.

Cassandras, C.G., & Lafortune, S. (1999). *Introduction to discrete event systems*, Kluwer Academic Publishers, Dordrecht.

Chen, S.C., Ke, Y.L., & Wu, J. S. (2001). Coloured Petri-nets approach for solving distribution system contingency, *Proceedings of the IEEE*, *148*(5), 463-470.

Desel, J. (2000). Validation of process models by construction of process nets, in W.M.P. van der Aalst, J. Desel, A. Oberweis (Eds.), *Business process management: models, techniques and emperical studies*, Lecture Notes in Computer Science, vol. 1806, pp. 110–128, Springer, Berlin.

Dragoni, N., Massacci, F., & Saidane, A. (2009). A self-protecting and self-healing framework for negotiating services and trust in automatic communication systems, *Computer Networks*, *53*, 1628-1648.

Ellis, C.A. (1979). Information control nets: a mathematical model of office information flow, in Proceedings of the Conference on Simulation, Measurement and Modelling of Computer Systems, ACM Press.

Gauthier, A., Davis, K., & Schoenbaum, S. (2006). Achieving a high-performance health system: High reliability organizations within a broader agenda, *Health Services Research*, *41*(4), 1710–1720.

Hollnagel,E., Woods, D., & Levenson, N. (2006). *Resilience engineering: Concepts and precepts*, Hashgate, Hampshire, England.

Holt, A.W. (1971). Introduction to occurrence systems. In *Associate Information Techniques*, Jacks, L. (ed.), pp. 175–203, New York: American Elsevier.

Huang, Y-M, Chen, J-N, Huang, T-C, Jeng, Y-L, & Kuo, Y-H (2008). Standardized course generation process using Dynamic Fuzzy Petri Nets, *Expert Systems with Applications*, *34*, 72-86.

Jantzen, M (1980). Structures representation of knowledge by petri nets as an aid for teaching and research. Lecture Notes in Computer Science. Berlin: Springer Verlag.

Kim, C.H., Yim, D.S., & Weston, R.H. (2001). An integrated use of IDEF0, IDEF3 and Petri-net methods in support of business process modeling, *Proceedings of the Institution of Mechanical Engineers*, *215*(4), 317-329.

Lee, J, Pan, J.I., & Kuo, J.Y. (2001). Verifying scenarios with time Petri-nets. *Information and Software Technology*, *43*(13), 769–781.

Leveson, N., Dulac, N., & Marais, K. (2009). Moving beyond normal accidents and high reliability organizations: A systems approach to safety in complex systems, *Organization Studies*, *30*(2–3), 227–249.

Li, X., & Lara-Rosano, F. (2000). Adaptive fuzzy Petri nets for dynamic knowledge representation and inference, *Expert Systems with Applications*, *19*(3), 235-241.

Liu, R., Kumar, A., & van der Aalst, W. (2007). A formal modeling approach for supply chain event management, *Decision Support Systems*, *43*, 761-778.

McCarthy, D., & Sarin, S. (1993). Workflow and transactions in InConcert. *Data Engineering Bulletin*, *16*(2), 53–56.

Mehrez, A., Muzumdar, M., Acar, W., & Weinroth, G. (1995). A Petri-net model view of decision making: an operational analysis, *Omega - the International Journal of Management Science*, *23*(1), 63-78.

Mentzas, G., Halaris, C., & Kavadias, S. (2001). Modelling business processes with workflow systems: an evaluation of alternative approaches, *International Journal of Information Management*, *21*(2), 123-135.

Murata, T. (1989). Petri Nets: properties, analysis and applications, *Proceedings of the IEEE*, *77*(4), 541-580.

Nomura, T., Hayashi, K., Hazama, T., & Gudmundson, S. (1998). Interlocus: Workspace configuration mechanisms for activity awareness, in Conference on computer-supported cooperative work, Seattle, Washington.

Perrow, C. (1999). *Normal accidents, living with high-risk technologies*, Princeton University Press, Princeton, New Jersey.

Peterson, J., L. (1981). *Petri Net Theory and the Modeling of Systems*, Morristown, NJ: Prentice-Hall, Inc.

Petri, C. A. (1962). *Kommunikation mit Automaten*. University of Bonn. English Translation by C.F. Greene (1965) Communication with automata, Supplement to Technical Report RADC-TR-65-377, Vol. 1, Rome Air Development Center, Griffith Air Force Base, Rome, New York.

Piera, M.A., Narciso, M., Guasch, A., & Riera, D. (2004). Optimization of logistic and manufacturing systems through simulation: a colored Petri net-based methodology, *Simulation*, *80*(3), 121–129.

Sakthivel, S., & Tanniru, M. R. (1988-89). Information verification and validation during requirement analysis using Petri nets. *Journal of Management Information Systems*, *5*(3), 33-52.

Salimifard, K., & Wright, M. (2001). Petri net-based modelling of workflow systems: An overview, *European Journal of Operational Research*, *134*(3), 664-676.

Valk, R., & Girault, C. (2003). *Petri nets for systems engineering: a guide to modelling, verification and applications*, Springer-Verlag, Berlin.

van der Aalst, W.M.P. (1996). Petri-net-based workflow management software, in A. Sheth (Ed.), Proceedings of the NFS Workshop on Workflow and Process Automation in Information Systems, Athens, Georgia, pp. 114-118.

van der Aalst, W.M.P. (1997). *Verification of workflow nets*, Lecture Notes in Computer Science, vol. 1248, pp. 407–426, Springer, Berlin.

van der Aalst, W.M.P. (1998). The application of Petri nets to workflow management, *The Journal of Circuits Systems and Computers*, *8*(1), 21–66.

van der Aalst, W.M.P., De Michelis, G., & Ellis, C.A. (1998). *Proceedings of workflow management: net-based concepts, models, techniques and tools*, Lisbon, Portugal.

van der Aalst, W.M.P., Desel, J., & Oberweis, A. (2000), *Business process management: Models, techniques and empirical studies*, Lecture Notes in Computer Science, vol. 1806, pp. 1-15, Springer, Berlin.

van der Aalst, W.M.P., van Hee, K.M., & Houben, G.J. (1994). Modelling workflow management systems with high-level Petri nets, in G. De Michelis, C. Ellis, G. Memmi (Eds.), Proceedings of the Second Workshop on Computer-Supported Cooperative Work, Petri nets and related formalisms, pp. 31–50.

Weick, K. (2001). Making sense of the organization, Blackwell, Oxford.

Weick, K., & Sutcliffe, K. (2001). *Managing the unexpected: Assuring high performance in an age of complexity*, Jossey-Bass, San Francisco.

Winograd, T., & Flores, R. (1987). *Understanding computers and cognition*, Addison-Wesley, Reading.

Wong, M.L. (2001). A flexible knowledge discovery system using genetic programming and login grammars, *Decision Support Systems*, 31, 405-428.

Zhovtobryukh, D. (2007). A Petri net-based approach for automated goal-driven web service composition. *Simulation*, *83*(1), 33–63.

Zisman, M.D. (1977). *Representation, specification and automation of office procedures*, Ph.D. Thesis, University of Pennsylvania, Wharton School of Business.

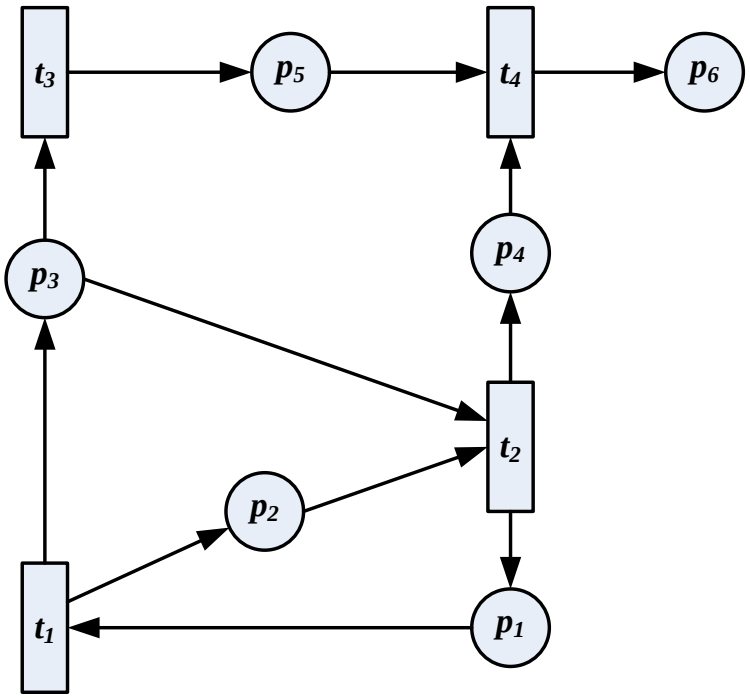
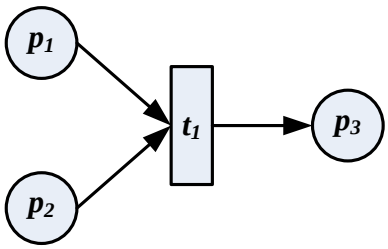
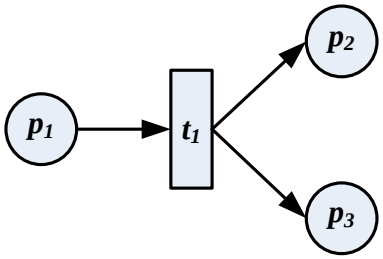


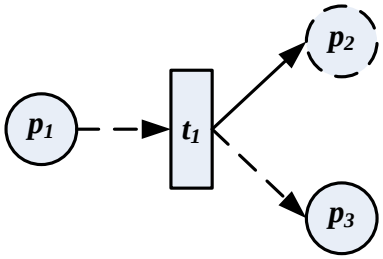
Figure 1: Simple Petri Net Example



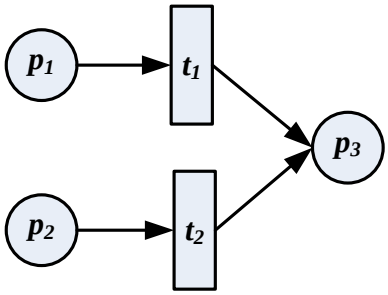
2.1. AND-Join



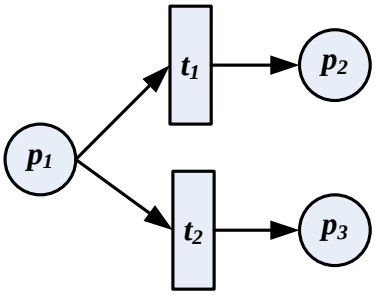
2.2. AND-Split



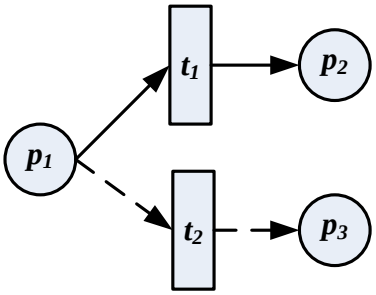
2.3. Alternate-AND-Split



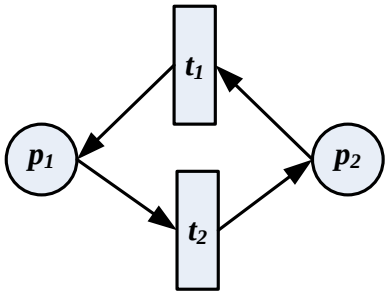
2.4. OR-Join



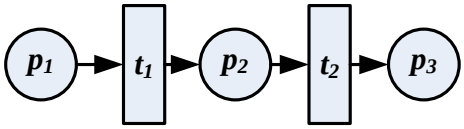
2.5. OR-Split



2.6. Alternate-OR-Split



2.7. Iteration



2.8. Causality

Figure 2: Workflow Primitives

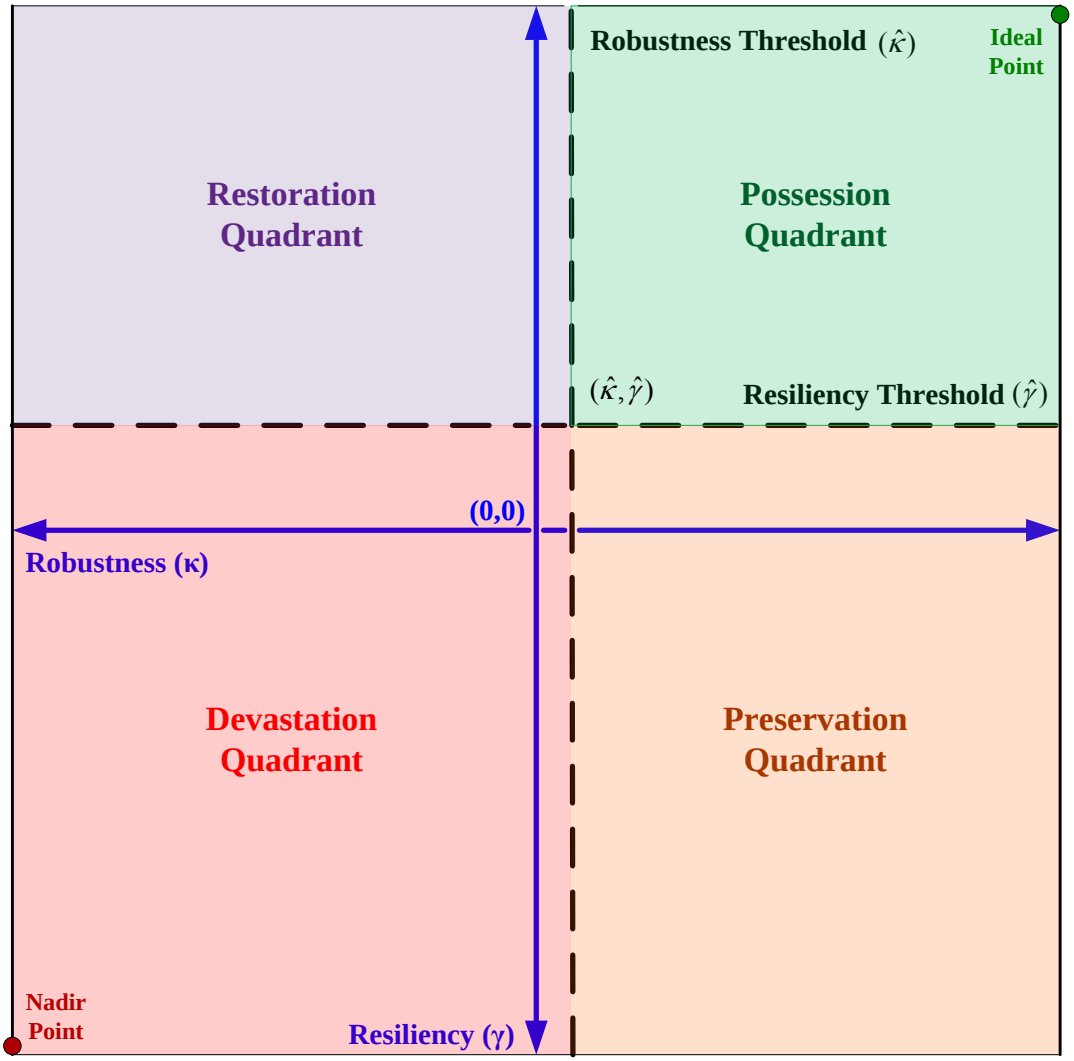


Figure 3: Robustness-resiliency plane

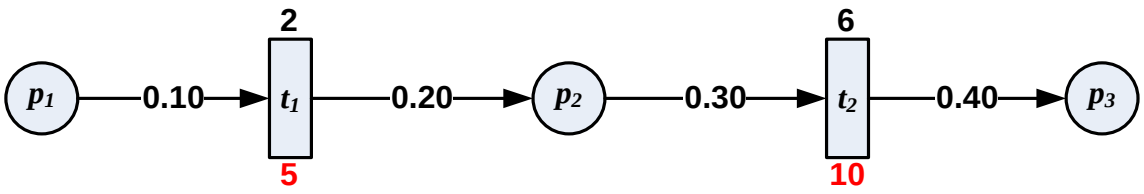


Figure 4: Simple workflow example

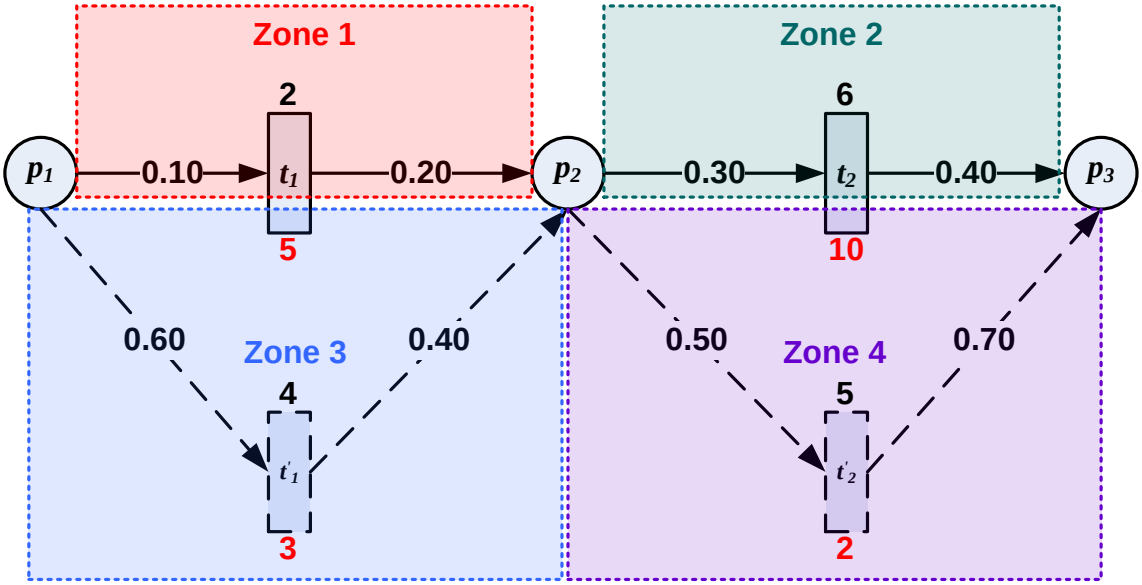


Figure 5: Simple workflow example (with alternate transitions)

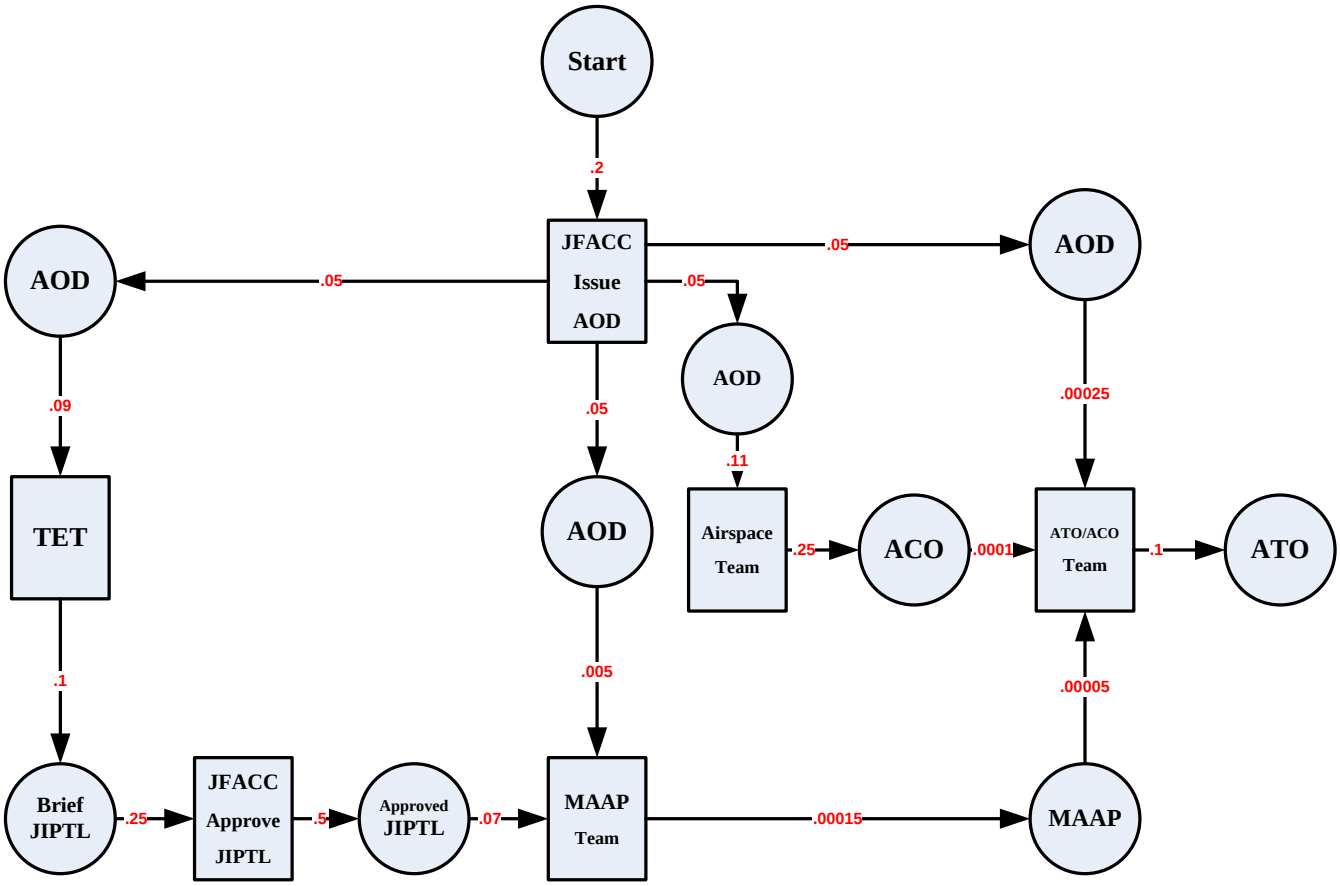


Figure 6: Air tasking order generation workflow

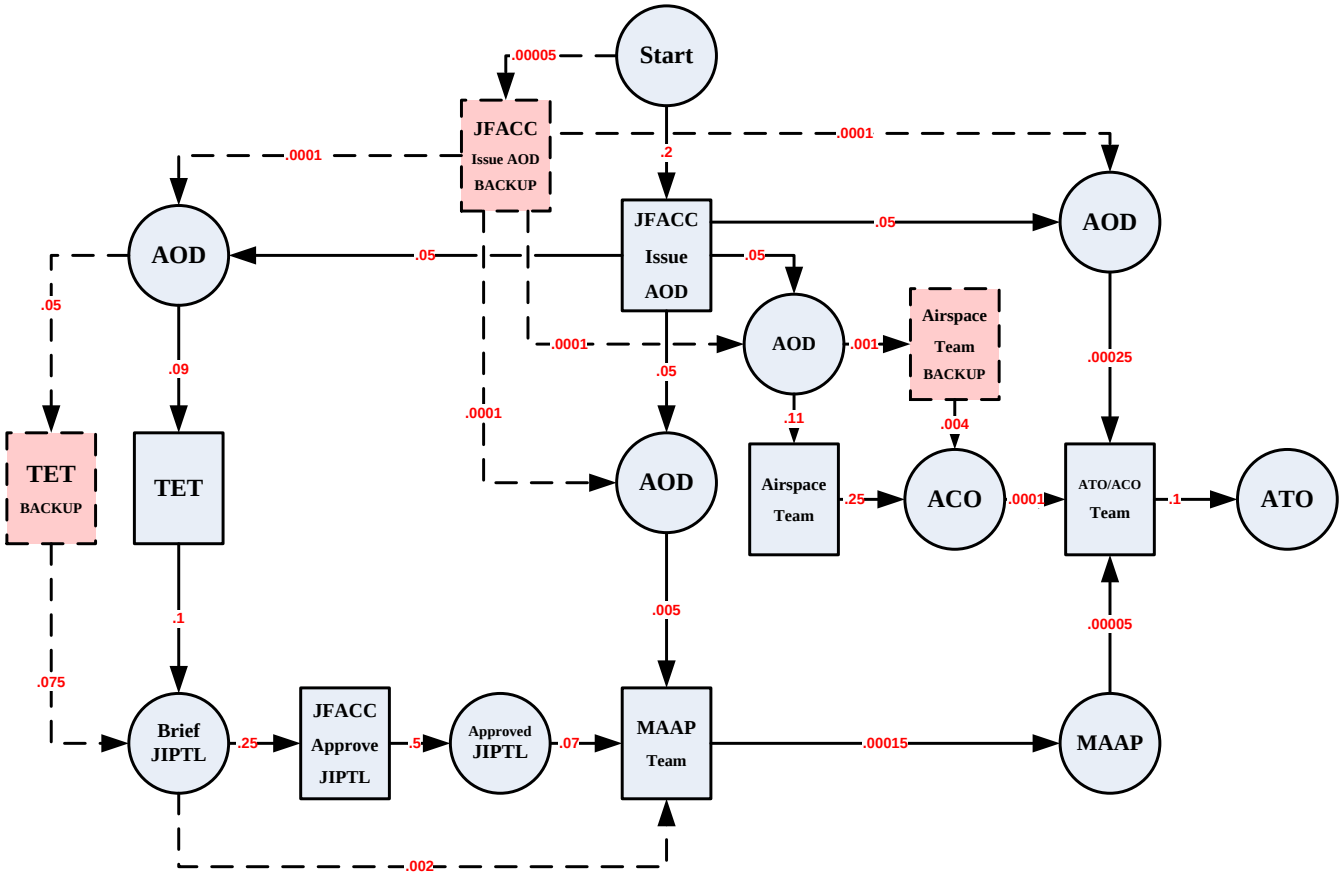


Figure 7: Air tasking order generation workflow with backup

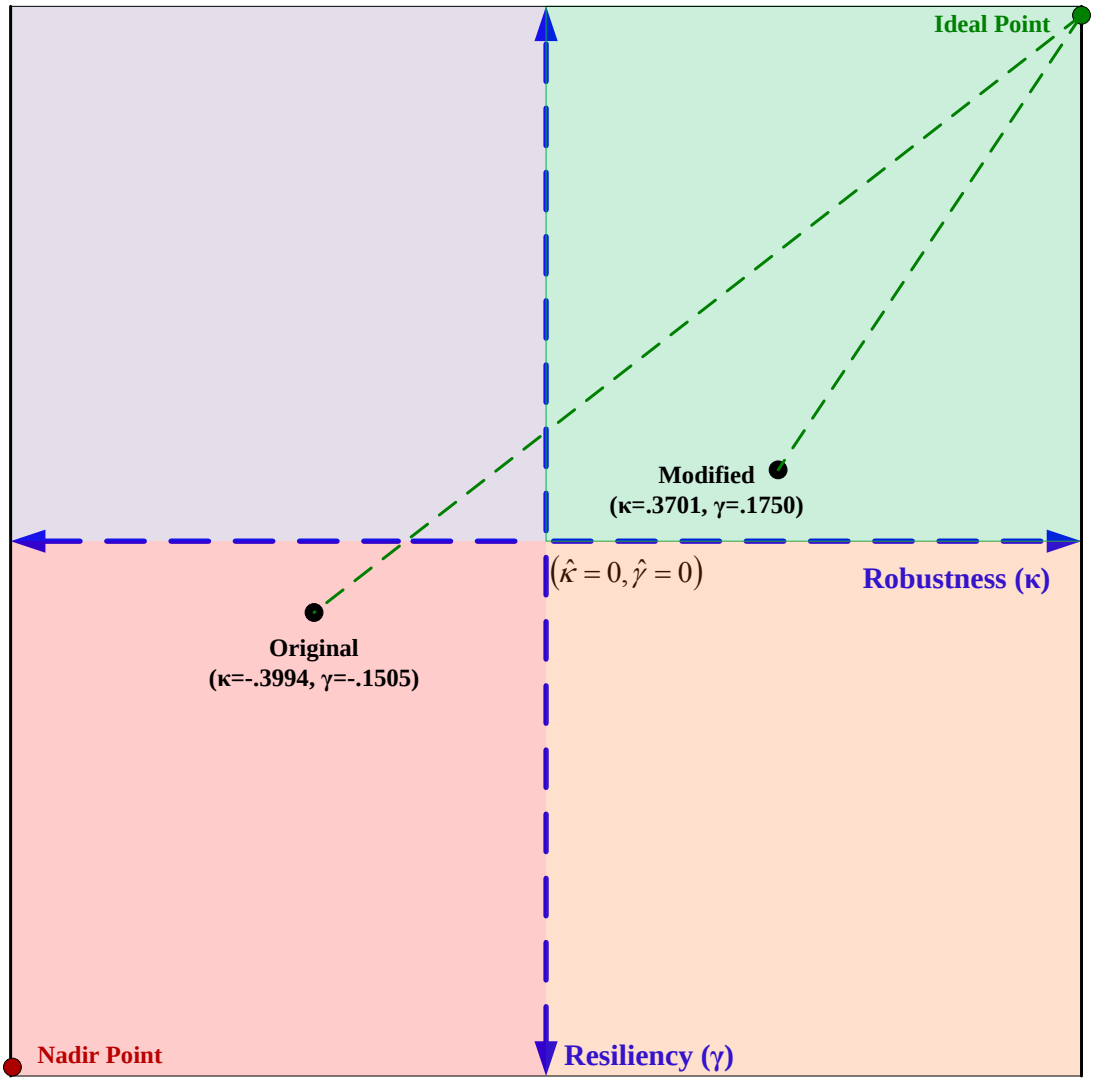


Figure 8: Original and modified case study robustness-resiliency plane

Table 1: Breakage probabilities for the example

Arc	Best-case Breakage Probability	Most likely Breakage Probability	Worst-case Breakage Probability	Expected Breakage Probability
l_{p_1,t_1}	0.8	0.1	0.12	0.1
l_{t_1,p_2}	0.01	0.0475	1	0.2
l_{p_2,t_2}	0.04	0.35	0.36	0.3
l_{t_2,p_3}	0.4	0.4	0.4	0.4

Table 2: Transition times for the example

Transition	Best-case Transition Time	Most likely Transition Time	Worst-case Transition Time	Expected Transition Time	Expected Transition Time Variance
t_1	1	1.9	3.4	2	0.16
t_2	2	6	10	6	1.78
t'_1	1	2	15	4	5.44
t'_2	2	5.5	6	5	0.44

Table 3: Repair times for the example

Transition	Best-case Repair Time	Most likely Repair Time	Worst-case Repair Time	Expected Repair Time	Expected Repair Time Variance
t_1	0.1	0.225	29	5	23.20
t_2	3	8	25	10	8.93
t'_1	3	3	3	3	0
t'_2	1	2	3	2	0.11

Table 4: Case study transition times

Transition	Best-case Transition Time	Most likely Transition Time	Worst-case Transition Time	Expected Transition Time
JFACC Issue AOD	17	35	60	36.1667
TET	24	29	40	30
JFACC Approve JIPTL	3	12	18	11.5
MAAP Team	30	40	50	40
Airspace Team	24	45	65	44.8333
ATO/ACO Team	65	95	116	94
JFACC Issue AOD	85	125	141	121
BACKUP	60	86	95	83.1667
TET BACKUP	90	133	240	143.6667

Table 5: Case study repair times

Transition	Expected Repair Time
JFACC Issue AOD	64
TET	82
JFACC Approve JIPTL	100
MAAP Team	10
Airspace Team	47
ATO/ACO Team	30
JFACC Issue AOD BACKUP	1
TET	2.5
BACKUP	12

Table 6: Original and modified case study system results

System Property	Original	Modified
$P_{\text{incapacitation}}$	0.8888	0.1274
Ts_{best}	139 hours	176.772 hours
Ts_{worst}	284 hours	354.930 hours
Ts	211.167 hours	262.817 hours
Rs	286 hours	184.537 hours
κ	-0.3994	0.3701
γ	-0.1505	0.1750
ω	0.6405	0.3670