A Timed Petri Net-Based Model for Air Defense System Analysis

André N. Costa[1,2]
[1]Institute for Advanced Studies
Brazilian Air Force
São José dos Campos, SP, Brazil
anegraoc@gmu.edu

Roberto C. Follador[1]
[1]Institute for Advanced Studies
Brazilian Air Force
São José dos Campos, SP, Brazil
follador@ieav.cta.br

Paulo C. G. Costa[2]
[2]C4I & Cyber Center / SEOR Dept.
George Mason University
Fairfax, VA, USA
pcosta@gmu.edu

***Abstract*— The application of sound mathematical modeling and simulation techniques to support command and control of time and mission critical operations has been an active area of C4I research. This is especially true for C2 operations in the presence of adversarial players, where inadequate responses, as well as the lack or delay of appropriate responses, might have drastic consequences with loss of life. One typical example of such high-stake operations is Air Defense and their associated C2 Systems. This work focuses on the application of a specific combination of mathematical modeling and simulation techniques to support C2 operations. More specifically, we apply a combination of Bayesian networks and Petri nets to perform discrete event computer simulations in the analysis of a time and mission critical C2 system. We illustrate our findings by applying this combination of techniques to an air defense system that must generate responses to potential airborne threats. The threats are posed by initially unidentified objects acquired by the system sensors, going through three levels of track acquisition: detection, classification, and identification. In face of any given track, the system is supposed to generate a proper response, which in some cases must have to be cleared by competent authorities. In a more tangible manner, through the proposed methodology, it is possible to determine the sensor configuration (mostly radars), which allows for addressing these threats on a timely fashion.**

Keywords—air defense system; colored petri nets; networks; threat assessment; systems engineering

I. Introduction

The process of properly addressing airborne threats poses a challenge to military modelers worldwide. Most computer-based simulation tools currently available rely on deterministic techniques to model C2 operations, mostly due to their adequate performance in supporting time and mission critical operations. Unfortunately, deterministic methods are not adequate for highly dynamic changing scenarios in the presence of adversarial, hard-to-predict behaviors. These methods involve supplied pre-programmed artificial intelligence capable of capturing the important aspects related to motion and combat decisions on a tactical level, but are less suitable for capturing the inherently probabilistic nature of adversarial C2 operations (a.k.a. the fog of war). In addition, the complex interactions between various military echelons and their associated decision-making processes are also more suitable for probabilistic modeling techniques.

The command and control in air defense systems properly illustrates the above issues, as it involves both a highly complex level of interactions between blue forces and the need to properly react against adversarial players. Our research aims at applying both deterministic and probabilistic approaches to address these complex C2 environments. In this paper, we provide an initial step in demonstrating the feasibility of our ideas with a simulation experiment within a typical C2 scenario. The scenario involves an air defense system that must provide military pilots and analysts with a planning framework capable of analyzing the planned mission paths to determine mission survivability and success metrics.

For a scenario to be representative of the inherent complexity of these operations, it must contain realistic models of air defense systems characterizing the physical aspects of aircraft operation in detail, the dynamics of air operations as the scenario unfolds, as well as the uncertainty involved in the interactions between adversaries in an airspace theater of operations. Our approach employs Colored Petri Nets (CPNs) to achieve the first two aspects and Bayesian Networks (BNs) to obtain the latter.

With this approach, we aim to evaluate whether a given air defense system generates responses efficiently. If it does not, this would indicate that i) the information is not being acquired in time; ii) information is not being properly processed; or iii) the authorizations are taking too long to be given. We assess the first using the concept of radar layers, which is related to the number of radars allocated around a protected area. The area's geometry is the basis for defining the possible lines of detection of incoming threats. The other two potential causes of failure are quantified using probabilities within the CPN model itself.

We illustrate the multi-disciplinary approach mentioned above with a proof of concept prototype that implements some of our ideas in an air defense C2 planning example. The overall approach is laid out in this paper as follows. After the introduction above, Section II presents the main concepts under our application of BNs and Petri Nets, together with a partial but representative account of the related efforts in this area. In Section III we describe the underlying model as implemented in the prototype we built to run our experiments, which are then explained and discussed in Section IV. Finally, we bring our conclusions in Section V.

II. Background and Related work

BNs are a probabilistic representation technique in which a Joint Probability Distribution (JPD) is represented as a Direct Acyclic Graph (DAG). Nodes in the DAG represent random variables of a probabilistic model and its edges represent the probabilistic relationships among these variables. Conditional Probability Distributions (CPDs) represent the strength of these relationships. Although the technique can be traced back to the work of reverend Thomas Bayes more than 2 centuries ago, its applicability only became possible with the advent of computers and the work of various researchers in the late eighties, most predominantly Judea Pearl ([1], [2], [3]). BNs are applied in many domains of applications, mostly as a probabilistic expert systems approach. Within the C2 domain we focused in the present work, a proposed method to increase the verisimilitude of the models was to include Bayesian reasoning on the response generation process within the simulation, as seen in [4]. An air defense system is modeled on a physics-grounded simulation environment in a way that multiple simulation runs can define areas of higher vulnerability, as well as assess whether a mission has satisfactory chances of success. BNs are used to extract and refine knowledge from this simulation.

However, to simulate air operations in a realistic way, it is not only necessary to realistically replicate entity behaviors and decision-making processes to generate actions, but also to accurately model their processing times. The latter define the necessary conditions that must be met before executing each phase within the response generation process. One potential approach for achieving this would be dynamic Bayesian networks (DBNs), which extend BNs to capture probabilistic relationships as they evolve in time ([5], [6]). In our initial work, we considered the use of DBNs, and actually did some initial attempts to produce DBN models, but, given the modeling goals and performance requirements we had for this aspect of the modeling process, we opted for using a different technique, stochastic Petri nets.

Petri nets are a modeling language for describing discrete event systems. They are defined as directed bipartite graphs, in which the nodes represent places and transitions linked by directed arcs. Arcs are responsible for describing which places are pre- and/or post- conditions for which transitions [7].

Petri nets may include graphical markings on its places to enable the transitions of which they are pre-conditions. We explored this feature by using tokens forming the markings in the Petri net to represent incoming objects, which are possible threats to an existing air defense system. Also, when the transitions occur, time attributes are assigned to these tokens based on several probabilistic distributions, adding the desired variability to the system. Petri nets that contain time attributes are called timed Petri nets [8] and impose more restrictions to the firing of specified transitions. For the transition to occur, the simulation time must meet the token time, which can be increased throughout the simulation. Tokens may have data values attached to them, which are called colors. Petri nets that contain places that support tokens of a specific type (data values) are called Colored Petri Nets and these types are defined as color sets.

The tokens are spawned through a random object generator and are all considered to be able to be further identified in the model, *i.e.* all created entities within the model are detectable objects. Moreover, all objects are airborne, being detected by radars. After the detection phase, the system moves to the next two levels of acquisition: classification and identification. Classification is the process of determining the object's class, as specified in Section III, whereas identification relates to the definition of the object's stance. These levels of acquisition are commonly used in this field, as observed in ([9], [10], [11]). With the full knowledge of the potential threats characteristics, the response generation process goes through a Bayesian threat assessment and a probabilistic clearance obtainment. Lastly, the data is output for analysis as specified in Section IV.

This system is intended not only to properly model a general air defense system, but also to identify its vulnerabilities as well as to propose and evaluate suitable alternatives. More specifically, the main goal is to determine how many radar layers are necessary to define a warning line capable of detecting incoming threats in a timely fashion.

The model was implemented on the software CPN Tools, which is a tool for editing, simulating and analyzing Colored Petri Nets (CPN) [12]. This tool from the University of Aarhus is based on advanced, post-WIMP (windows, icons, menus, pointer) interaction techniques, including bi-manual interaction, tool glasses and marking menus to make the modeling process faster and more intuitive [13]. Also, it allows for coding on SML (Standard ML) as a means to control the data flow in a more complex fashion, mimicking complicated processes from the reality.

A CPN model for an air defense system was also proposed in [14], however, without probabilities, which are one of the main contributions of this paper. They advocate that CPN are a suitable approach to model command and control systems, allowing for rapid prototyping, gaming, and simulation. Following up into that, we increased the complexity of their model, mainly on the acquisition and decision processes, focusing on the determination of the number of radars layers.

III. Air Defense System Model

The system consists of four major phases: Acquisition, Response Generation, Approval, and Analysis. Some of them contain several sub-phases, which are described in the following sub-sections and presented in Fig. 1.

A. *Acquisition*

This phase consists of detecting, classifying, and identifying flying objects that enter the airspace controlled by the air defense system. A key assumption is that the arrival of random objects follows a Poisson distribution, and therefore the interarrival times can be represented by an exponentially distributed random variable. A process following such characteristics is called a Markovian process, and possesses the property of being memoryless. This is a very desirable feature for modeling aircraft arrivals, since it makes the models much more efficient, and its use in a wide range models ([15], [16], [17]) comes as no surprise.

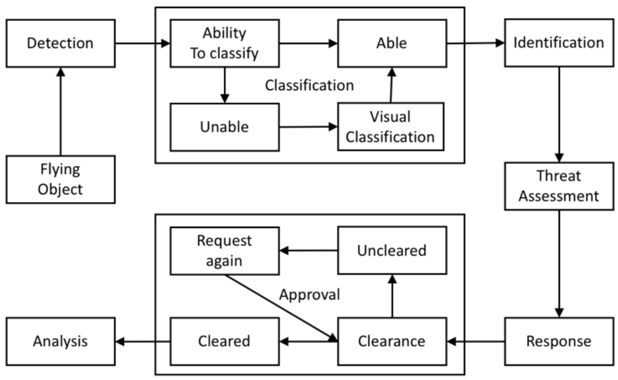

Fig. 1. Air Defense System Model

Besides the random interarrival times, the flying objects themselves are generated randomly. That is, their two basic characteristics, CLASS and STANCE, are assigned in a random and independent fashion. In our simulated system, the determination of these two characteristics is the primary goal of the acquisition phase. Of course, since it is a simulation, the analyst knows the actual value (i.e., the "ground truth") and can therefore compare what the system assesses in its acquisition phase against the ground truth. These characteristics are defined as follows:

1) CLASS: indicates the type of object that is entering the area under the air defense system's responsibility.

a) Combat Plane: a military airplane with the primary role of destroying enemy equipment, either in air combat or in bombing missions. This class includes fighters, bombers, as well as attack and electronic warfare aircraft.

b) Non-combat Plane: aircraft used in a supporting capacity for the military, covering a variety of functions including cargo transport, aerial refueling, search and rescue, evacuation, information-gathering, among many other roles. Although not designed for combat, non-combat aircraft may be equipped with weapons for self-defense when navigating through hostile areas or battlefields.

c) Helicopter: a type of aircraft that uses its blades, which are also called spinning wings, to fly. In our scenario, the main roles for helicopters are transport and attack.

d) Drone (Unmanned Air Vehicle - UAV): a small aircraft that can be either remotely controlled or autonomous. These aircraft can be used as surveillance systems, data collectors, cargo transporters, etc. Nowadays, they are also used in combat to attack ground targets.

e) Bird: a vertebrate animal that possesses feathers and wings and can fly. Bird is not a direct threat, but is considered in the Petri net system for evaluation purposes.

2) STANCE: indicates whether the incoming object is friend or foe.

a) Friendly: flying object considered as ally and potentially not a threat

b) Neutral: flying object considered as a potential threat

c) Opposing: flying object that pertains to a hostile group or nation, therefore considered as a threat

These two characteristics are displayed on a color set called UFO (unidentified flying object), that also contains the object's arrival time recorded through the function ATR (arrival time recorder). Color sets are specifications of the types of tokens that can reside on a particular place. In this case, the token possesses the two characteristics within it.

In our simulated environment, detectable flying objects are generated at the edge of the air defense system's detection range. That is, the moment the random generation of these objects occurs represents the time at which a detectable flying object goes within the air defense system's radar range. After its generation, it is just a matter of time for a given object to be detected. Based on the knowledge of subject matter experts (SME), the average time of 60 seconds was adopted as the average duration of the detection process, no matter what class of object.

Initially, all the average service times were modeled as exponential processes, following an M/M/1 queuing model. However, the second model adopted Erlang service times, with different parameters depending on the characteristics of the processes as discussed on Section IV. After going through the detection process a token is generated on the *Detected Object* place, representing that the system recognizes the existence of an object within range, not yet knowing its characteristics.

Following on with the levels of acquisition presented earlier in this paper, the classification phase includes an extra layer of complexity when compared to the detection phase, since it adds a probability assessment representing the ability to classify a detection without further investigation. This probability was arbitrarily set to 90%, stating that the clear majority of the objects can be classified by radar operators and analysts, and do not require a more detailed visual classification or the use of more sophisticated systems for disambiguation.

When classification was not achieved, the system includes an extra transition that leads to a time penalty on the overall process, representing a harder and more time-consuming classification process. During the evaluation of our results, we performed sensitivity analysis by modifying the classification probability and assessing its influence on the average processing time.

Following the above flow, depending on the sophistication of the classification methods, the system will be subjected to a penalty proportional to its performance. The *Classify* transition contains a service time modeled similarly to the *Detect* transition, *i.e.*, exponential at the beginning and Erlang on a further analysis, only adding an extra amount of time to the total processing time.

A token generated on the *Object Classified* place means that the CLASS of the object is finally known to the system. The next level of acquisition requires it to be identified so that an appropriate response can be generated. This is performed at another transition node called *Identify*, which again contains only a service time to be added to the object's processing time. However, to account for the fact that this process should be briefer when a friendly object is being identified, the definition of the extra time is conditioned to the object's STANCE.

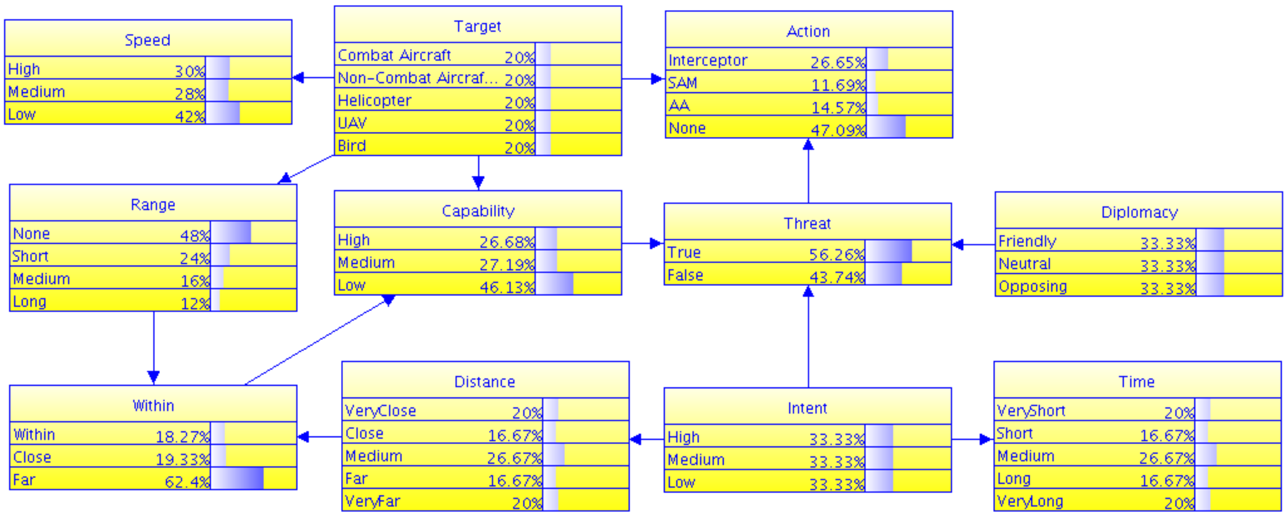

Fig. 2. Bayesian network for threat assessment.

B. Response Generation

Having the UFO as an *Identified Object*, the air defense system operators move to the next stage of the model, knowing both the CLASS and the STANCE of the incoming object. These characteristics are input to the *Threat Assessment (TA)* place, which contains Bayesian model probabilities of response for each of the possible combinations of these properties.

These probabilities are generated *a priori* by the Bayesian network displayed in Figure 2 and are stored in functions within the Petri net. Based on these probabilities for a particular object the response is picked and stored on the *Response* place, which has a color set called UFOR, indicating that it also contains the assigned response.

The BN depicted in Figure 2 follows the modeling approach proposed in [17], in which the variables present in an air defense scenario would be partitioned into proximity parameters, capability parameters, and intent parameters. Proximity parameters are those related to the distance between the aircraft executing the mission and its target (i.e. the defended asset, from the perspective of the model). In this case, the proximity parameter is the Distance node. Capability parameters are related to the ability of the enemy's air defense system to inflict damage to the defended asset, represented the BN nodes Range, Target, Time, and Speed. Finally, Intent parameters refer to the enemy's intentions towards the defended asset and are encompassed in the node Intent. In addition to the parameter nodes, the model includes the nodes Within, Capability, and Threat, which represent the results of the interactions between the parameter nodes. These interactions should represent some of the observed characteristics of the real system.

This Bayesian Network was created on UnBBayes, an open-source, Java-based, probabilistic graphical framework developed by the Artificial Intelligence Group (GIA) from the Computer Science Department at the Universidade de Brasília [18]. UnBBayes has a GUI and an API that provides support to various algorithms and techniques via a plugin-based architecture. This includes, but is not limited to, Bayesian inference, sampling, learning and evaluation, which yields some advantages compared to the other available software [19].

C. Approval

Following a similar pattern as in the classification process, the *Ask for Clearance* transition indicates the probability of a commander to ratify the selected response, giving the necessary clearance for the operations to occur. Again, this probability is arbitrarily set to 90%, indicating that the commander has a high propensity to accept the recommended action by the system. In the case the commander disagrees with the system's output, there is a reselection node, which replicates the *Threat Assessment* transition and leads to a new clearance requisition.

D. Analysis

Finally, the analysis stage simply calculates the processing time for a giving object and records it on a new color set called UFORT. The calculation is simply made by subtracting the recorded arrival time from the current token time. From the *Analysis* node, the data is extracted for further investigation on the software Excel [20].

IV. Methodology

For each simulation run, a limit of 1000 incoming objects was set for providing probabilistically significant results. Moreover, each scenario was simulated three times and the average was utilized for the purposes of comparison. It is important to note that, even though three runs may seem a small number of runs, the statistical significance comes from the fact that 1000 objects in each run is a number much larger than real scenarios would present. The analysis process was designed based on the goals laid out in Section I. Starting from the base model, modifications have been made to reduce the number of radar layers needed and to minimize the number of processing delays and leakers, being represented by the following metrics:

1) Number of radar layers: total of radars used to define the number of range (radii) that the incoming object has to go through until reaching the target. The number of radii is defined by inputting the number of radar layers on the expression 2n-1, meaning that a radar on a central position only accounts for a single radius, whereas the others account for two, *i.e.*, a diameter. This means that the radars are assumed to be tangent with each other and that they cover all directions from the central spot (position of the first radar, or first layer).

2) Average processing time: average time that each object takes to be processed by the air defense system, considering all possible classes.

3) Average processing delay per object class: average time that each object, from a given class, exceeds the processing time limit established by the number of radar layers and its ground speed. Each aircraft class has a particular average speed that determines a time in which the distance defined by the number of radar layers will be traveled.

4) Number of leakers: count of objects that are processed on a time that exceeds the time limit established by the number of radar layers, *i.e.*, objects that present processing delay.

This metrics are obtained from each of the following models:

A. Base Model

This is the exact model described in Section III, with no modifications. However, due to an elevated number of needed radar layers observed with this model, modifications were proposed and implemented in the following models.

Before considering the potential possibilities for reducing the number of the required radars, we decided to implement a design modification for modeling service times. This involved adopting an Erlang model, which we belief would reflect the actual operation of the system in a more realistic way.

B. Erlang Model

Some of the processes stated on the base model consist of a collection of sub-processes that would have to be modeled on a more complex way, resulting on a considerable increase on the number of places and transitions present on the Petri Net.

Since this increased complexity would depart from the scope of this research, which is to provide a high-level model for the analysis of processing times and leakers, the employed solution was to substitute the exponential service times to Erlang distributions. In this way, when varying the parameter k of the probability distribution, the model made it possible to represent an estimate of the number of sub-processes within each transition node.

C. Bypass Model

This alternative model was conceived by focusing on the most critical object on the base model: combat plane. This class of objects was clearly identified as a bottleneck, since the number of radars would go down to 2 if only the other three objects were analyzed.

To reduce the processing time of combat planes, so that it stays within the limits established by the plane's average speed, a bypass arc was created connecting the classification transition directly to the response place. The meaning of this modification in practical terms is that, as soon as an object is classified as a combat plane, an interceptor aircraft is activated and sent for further identification and possible response. The approval process is still maintained mostly for representing the case in which a friendly aircraft is inbound.

V. Results and Discussion

A. Base Model

After three simulation runs of the model described on Section III with no modifications, it was observed that the minimum number of radar layers for avoiding object processing delays on any given class was four layers.

B. Erlang Model Results

The results obtained from running this modification three times were similar to the ones from the base model, being quantitatively compared on subsection C of this section (*Bypass*). Qualitatively, the addition of sub-processes allowed for a greater variability on individual processing times. This is translated numerically to an increase from 215 to 236 on the number of leakers.

With the new Erlang base model at hand, some modifications were made to evaluate the classification process, as well as how the complacency of the clearance obtainment would affect the processing time, and, consequently, the number of leakers and radar layers.

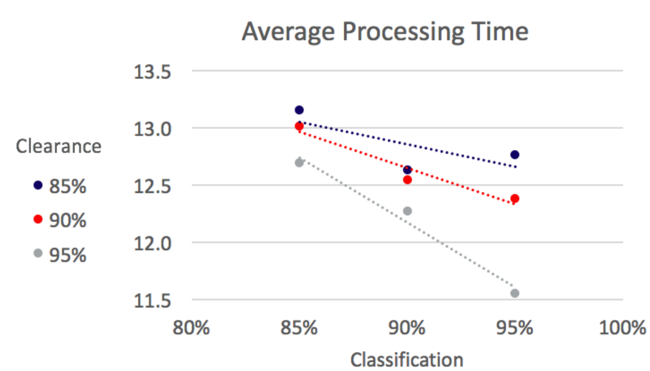

Fig. 3. Clearance and Classification influence on the Average Processing Time (in minutes).

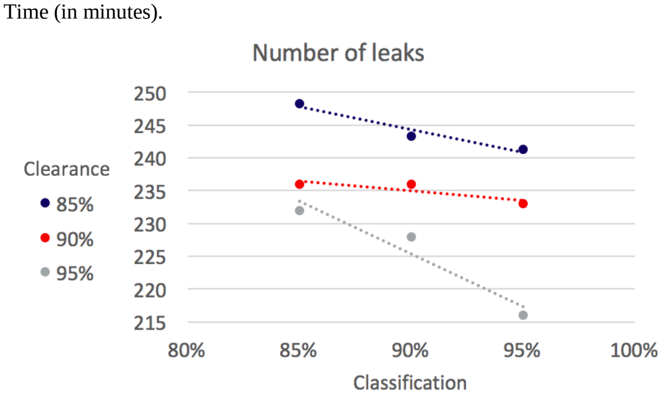

Fig. 4. Clearance and Classification influence on the Number of Leaks.

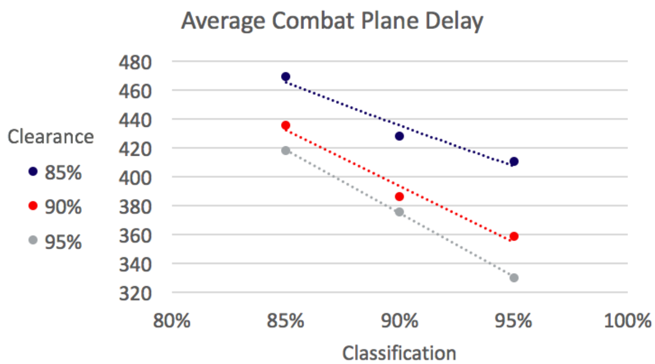

Fig. 5. Clearance and Classification influence on the Average Combat Plane Delay (in seconds).

As presented in Figures 3, 4, and 5, the higher probabilities for either the ability to classify or the clearance complacency yielded better results. This is due to the fact that the tokens will be held for a shorter period of time in these two nodes, and represents better sensors and a faster decision cycle. When analyzed as a whole – as in Table I – the average percentage variation of each modification of the probabilities was 4.91%. This number resulted from 4.95% for changes on the classification process and 4.87% for changes on the clearance process. As an insight on the meaning of these results, if one varies the probability to classify by 5%, any of the metrics' values should vary inversely by an average of 4.91%, for the analyzed range.

As an interesting fact, when each metric is analyzed individually it is possible to conclude that variations on the number of leakers are more evident when the clearance process is modified (Table II). On the other hand, the classification process showed a stronger influence on the average combat plane processing delay (Table III). Finally, the average processing time was affected very similarly by both the variations (Table IV).

TABLE I. TOTAL AVERAGE PERCENTAGE VARIATIONS

	Classification	Clearance
Total	4.95%	4.87%
Increase	4.41%	4.26%
Decrease	5.50%	5.48%

TABLE II. NUMBER OF LEAKERS PERCENTAGE VARIATIONS

	Classification	Clearance
Total	1.86%	4.05%
Increase	2.45%	4.13%
Decrease	1.27%	3.97%

TABLE III. COMBAT PLANE DELAY PERCENTAGE VARIATIONS

	Classification	Clearance
Total	9.48%	7.97%
Increase	7.78%	4.88%
Decrease	11.17%	11.06%

TABLE IV. AVERAGE PROCESSING TIME PERCENTAGE VARIATIONS

	Classification	Clearance
Total	3.52%	2.60%
Increase	2.99%	3.79%
Decrease	4.06%	1.42%

In spite of the significant results on the reduction of the processing times and number of leakers, the required number of radars for eliminating average delays was at most reduced to 3 on the best scenarios. This can be seen as a clear indication that a different approach for reducing this number is needed. We opted for implementing a few structural modifications to the analysis, since 95% was already a very high probability. Those modifications resulted in the bypass model explained below.

C. Bypass Model Results

Even though all the other object classifications lead to the same processes stated on the base model, the results lead to much lower metrics as showed in Figures 6, 7, and 8, being a very successful alternative to the base model.

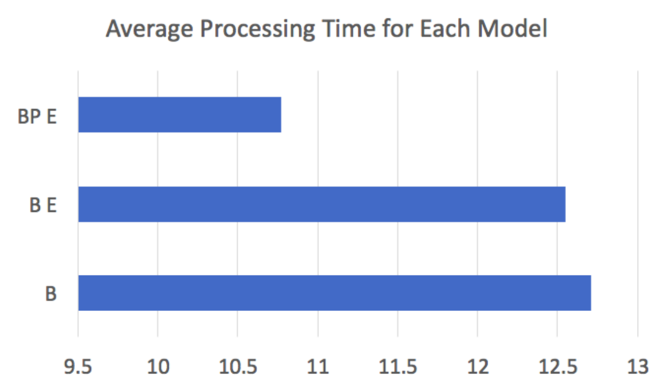

Fig. 6. Average Processing Time (in minutes) for Base (B), Erlang (B E), and Bypass (BP E) models.

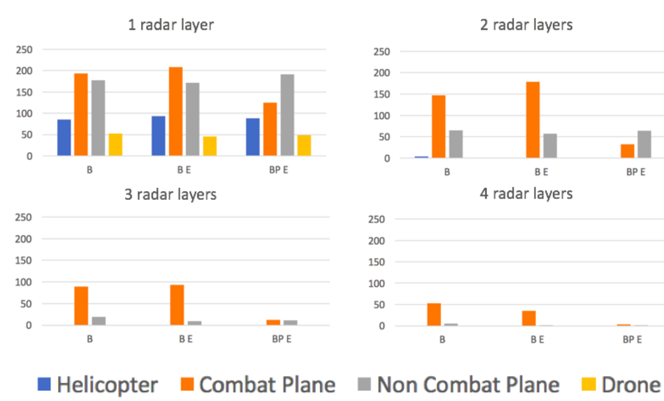

Fig. 7. Average Processing Delay (in seconds) for Base (B), Erlang (B E), and Bypass (BP E) models. Negative numbers indicate the processing delay, while positive indicate how many seconds in advance the threats were processed.

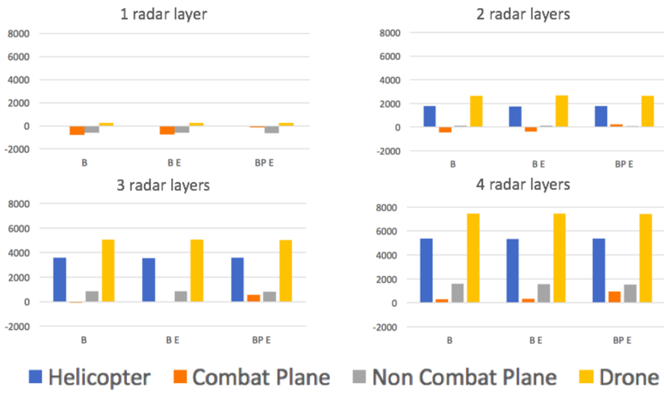

Fig. 8. Number of Leaks for Base (B), Erlang (B E), and Bypass (BP E) models.

VI. Conclusion and Future Work

This paper introduced our preliminary research on adopting a multi-disciplinary approach for evaluating time and mission critical C2 systems in the presence of adversarial behavior in an air defense system. While the vast majority of such systems employ a deterministic, rule-based approach, the prototype developed for this initial research employs a combination of Bayesian networks and Petri-Nets that leverages the main qualities of both deterministic and probabilistic representations to achieve an effective tool to model and evaluate complex systems.

In our experiments, we performed sensitivity analysis and other evaluation techniques to test the boundary conditions of our models and assess the impact of the various parameters, with the goal of determining potential approaches for reducing the cost and improving the efficiency of the system. We also were able to learn how to control the data flow by embedding standard ML codes into the system, as well as by using characteristics and properties of the Petri-nets. Finally, the model was designed in a way that allowed us to leverage features provided in the CPN tools software, such as the ability to fine-tune simulation and hierarchy tools to enhance the system's fidelity.

The results obtained so far, although preliminary, suggest that the combination of these approaches can produce models of the stochastic relationships and model dynamics with enough fidelity to allow for sophisticated analysis not previously warranted in current approaches. We intend to further explore these possibilities by extending the current models to include a physics-sound simulation to provide ground truth values to the current prototype (i.e instead of our SME-based average parameters). This would increase the ability to replicate unexpected behaviors and other aspects that are difficult to envision in the current setup.

Acknowledgment

The authors thank Dr. Abbas K. Zaidi for providing indispensable aid on the software CPN Tools and guidance during the elaboration of this work. Also, the authors acknowledge the fundamental role of Capt. Diego Geraldo from the Brazilian Air Force on providing subject matter expert knowledge regarding specially the object acquisition process.

References

[1] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artif. Intell.*, vol. 29, no. 3, pp. 241–288, Sep. 1986.

[2] J. Pearl, M. B. Morgan, S. Jowell, R. Van Genderen, D. Pearl, and L. Medoff, *Probabilistic reasoning in intelligent systems : networks of plausible inference*. .

[3] S. L. Lauritzen and D. J. Spiegelhalter, "Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems," *Source J. R. Stat. Soc. Ser. B*, vol. 50, no. 2, pp. 157–224, 1988.

[4] G. Johansson, F., Falkman, "A Bayesian network approach to threat evaluation with application to an air defense scenario," *Proc. 11th Int. Conf. Inf. Fusion*, vol. 0, pp. 1–7, 2008.

[5] P. Dagum, A. Galper, and E. Horvitz, *Temporal probabilistic reasoning: dynamic network models for forecasting.* Stanford University. Computer Science Department. Knowledge Systems Laboratory. KSL-91-64., 1991.

[6] N. Friedman, K. Murphy, and S. Russell, "Learning the Structure of Dynamic Probabilistic Networks." 1998.

[7] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*. Springer Science & Business Media, 2013.

[8] W. M. Zuberek, "Timed Petri nets definitions, properties, and applications," *Microelectron. Reliab.*, vol. 31, no. 4, pp. 627–644, Jan. 1991.

[9] L. A. Klein, *Sensor and Data Fusion: A Tool for Information Assessment and Decision Making (SPIE Press Monograph Vol. PM138)*. SPIE- International Society for Optical Engineering, 2004.

[10] National Research Council.*Automation in Combat Aircraft*. Washington, DC: The National Academies Press, 1982.

[11] J. P. Hutton, *Coast Guard: Opportunities Exist to Further Improve Acquisition Management Capabilities*. DIANE Publishing, 2011.

[12] A. V. Ratzer *et al.*, "CPN Tools for editing, simulating, and analysing coloured Petri nets," in *Applications and Theory of Petri Nets 2003: 24th International Conference, ICATPN 2003*, 2003, pp. 450–462.

[13] M. Beaudouin-lafon *et al.*, "CPN/Tools: A Post-WIMP Interface for Editing and Simulating Coloured Petri Nets," in *Coloured Petri Nets, Application and Theory of Petri Nets 2001, Proceedings of the 22nd International Conference, ICATPN 2001 Newcastle upon Tyne*, 2001, pp. 71–80.

[14] J. Liu and X. Li, "Application of Colored Petri Net in Command and Control System," in *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2009, pp. 323–326.

[15] I. I. Jessup, "Using Hybrid Simulation/Analytical Queueing Networks to Capacitate USAF Air Mobility Command Passenger Terminals," Mar. 2012.

[16] E. Mueller and G. Chatterji, "Analysis of Aircraft Arrival and Departure Delay Characteristics," in *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum*, American Institute of Aeronautics and Astronautics.

[17] N. Pyrgiotis, K. M. Malone, and A. Odoni, "Modelling delay propagation within an airport network," *Transp. Res. Part C Emerg. Technol.*, vol. 27, pp. 60–75, Feb. 2013.

[18] "Spreadsheet Software Programs | Excel." *Microsoft Excel 2016 Spreadsheet Software.* [Online]. Available: https://products.office.com/en-us/excel. [Accessed: 14-Sep-2017].