

# Modeling Man and Machine Interactions for Virtual Vulnerability Defense

Hoh Peter In, *Member, IEEE*, Sung-Oh Jung, Marshall Scott Poole, and Steve Liu, *Member, IEEE*

**Abstract** – Information automation is a key to higher productivity and lower costs. The downside of automation is its virtual vulnerability ( $V^2$ ), which refers to system malfunctions or failures caused by the relative ease of exploiting the system vulnerability by Internet users from distance. Email virus, denial of service attacks, unauthorized information access and operations are some examples of  $V^2$ . Other examples of  $V^2$  include O.S. backdoor, common mode of failures (e.g., Y2K clock representation), etc. In this paper, we propose a man-machine interaction model to characterize the dynamics between attacks and defense of  $V^2$ . Our objective is to quantify the threat and damage of  $V^2$ , and other costs to machines due to human misbehavior. We emphasize on the group behavior of defense teams, assuming that attacks are predictable. Our model allows for more realistic characterization of the attack-defense process, and it will help optimizing resource allocation to information assurance investment.

**Index terms** – Security, virtual vulnerability, hybrid Petri-net, group behavior model, man-machine interaction

## I. INTRODUCTION

Protection of critical information systems and global networking infrastructure is a vital on-going process that evolves around technologies, human organizations, and policies changes. Networking technologies allow novice users to enjoy ever better communication quality, but they also make security threats as common as car wrecks. Wide-spread hacking activities make *virtual vulnerability* ( $V^2$ )—security breaching caused by malicious remote software attacks—such a critical issue that it leaves government agencies and business enterprises little choice but to confront it with due diligence. Quantified, scientific methodologies are in great need to understand the nature, costs, and the social problems it spawns.

The sources of virtual vulnerability are a combination of technology and users' misbehavior. Therefore, in

technology and human behavior lay the resolutions to virtual vulnerability. While information assurance and security research dates back more than two decades, little has been done in the area of the human and machine behavior modeling and analysis. It is important to identify bottlenecks in the information protection process in order to make the best investment in approaches to prevent and defuse threats. A particularly critical factor in response to  $V^2$  is the behavior of the incident response team. The internal dynamics of the response team introduce a potential human bottleneck into the system. Teams that are poorly coordinated, not alert, or have been exhausted by repeated massive attacks are unlikely to be able to respond effectively no matter how talented their members and how many tools the organization provides.

Vulnerability assessment is one of the most important issues that one must address [Bis99] to help users determine the best approaches for preventing attacks. In view of the complexity of modeling human behavior, we must start with certain hypotheses about the different motivations of attackers, i.e., fame, money, and privacy [GKB00, GKP01]. The approach proposed in [GKB00, GKP01] is to reduce risk in software life cycles by using a software security assessment instrument.

Incident response teams are commonly used to respond to attacks in large software development or user organizations [WB01, Yas01], but there are few descriptions of the behavioral dynamics that shape their actual behavior. Most studies of teams in crisis or high-pressure situations focus on teams engaged in physical activities, such as tank crews or surgical teams. In one of the few applicable studies, Smart and Vertinsky [SV77] present a model of crisis response unit behavior that describes the impacts of stress and surprise on unit information processing and implementation of solutions. Janis and Mann [JM97] have termed this process “hot cognition,” which refers to analysis and decision-making on high stakes issues while under pressure.

Hybrid (continuous-discrete) systems modeling using hybrid Petri-net makes it an important trend in control theory [CS98, CWS99]. Discrete Petri Nets and differential equations have been used for discrete and continuous control systems description, respectively

- 
- Hoh Peter In, Sung-Oh Jung, and Steve Liu are with the Department of Computer Science, Texas A&M University, College Station, TX 77843-3112. Email: {hohin,jungs,liu}@cs.tamu.edu.
  - Marshall Scott Poole is with the Department of Speech Communication and Department of Information and Operations Management, Texas A&M University, College Station, TX 77843-4234. Email: mspoole@neo.tamu.edu.

[CWS99]. Reachability [Mur89, HK99] provides a basis for studying the dynamic properties of mixed systems. As our best knowledge, however, there is no study on the hybrid modeling of man-machine interaction.

This paper proposes an integrated man-machine model to characterize the dynamics of the attack response team to make best use of human and technological resources in exigent situations. The interaction model is expected to contribute to: (1) developing a more realistic model of the attack-defense process by incorporating group dynamics into the system; (2) modeling of the attack-defense process as an on-going enterprise involving multiple units, enabling us to explore the dynamics of  $V^2$ ; (3) optimizing resource allocation to information assurance investment; and (4) understanding better group behavior model under emergency response situations.

The approach of the interaction model for  $V^2$  is to extend a hybrid Petri-net technique to capture continuous states of human group behavior model, discrete system states, and their interaction under the exigent situations. The hybrid Petri-net techniques are well established in control theory area to represent states of systems and simulate their interactions. However, they must be properly tailored for our purposes to capture unique characteristics of human behavior. The rest of this paper is organized as follows. Section II presents a general man-machine interaction model for  $V^2$ . We discuss a hybrid Petri-net model for representing and simulating the man-machine interaction model in Section III. A simple example of the modeling technique is presented in Section IV. The paper concludes in Section V.

## II. MAN-MACHINE INTERACTION MODELING

The goal of the man-machine interaction modeling is to explore the dynamic relationship between attackers, defenders (users), and information systems in order to create team defense resolutions to virtual vulnerabilities. The term virtual vulnerability is advanced to emphasize the fact that a poorly managed information system could be seriously compromised without intruders' physical presence. With amateur-style hacking rapidly evolving into state/corporate/organized-crime sponsored activities, it is important for users to understand the tradeoff between completeness of specification, operational costs, attack probabilities, and the costs of system recoveries and business losses.

Although the Internet does not have a centralized architecture, it can be reasonably organized into user communities, such as banks, military divisions, e-retailers, and education institutes, as shown in Figure 1.

Similarly, it is not difficult to categorize intruders based on their motives or organizations. When the levels of attacks escalate to global actions, well-developed teams with talented members, skills, mutually compatible values, and coordinated activities are critical to the success of defense against highly hazardous intrusions. It is clear from the *code-red* and *nimda* incidents that isolated firewalls, antivirus, and intrusion detection systems are inadequate to fence off such large-scale attacks. We note that it takes relatively little effort to produce this kind of software and infect computers worldwide. A reliable and effective coordination channel among network administrators, developers, and users is much needed to defend such global attacks.

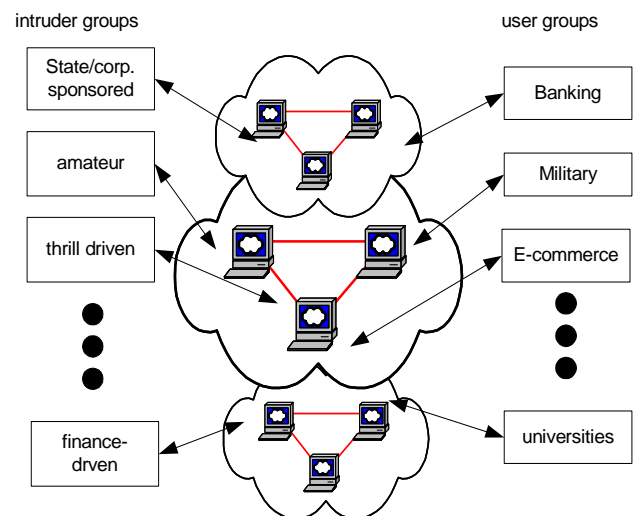


Figure 1. Overview of the Human-Machine Groups Involved in Computer Attacks

Cost-effective mitigation solutions for  $V^2$  are needed because secure software design is expensive. Especially when the size and complexity of the software systems exceeds certain threshold(s), the cost of system development escalates. It is important to understand the tradeoff in security requirements, rework costs, and other standard software economics concerns in the design of  $V^2$  defense strategies.

### A. Group Modeling

Current research on defender behavior treats the incident response team as if it were a smoothly operating machine. In real conditions, however, response teams vary greatly in terms of preparedness, ability to work together, and overall fitness of the team to rise to the challenge presented by the attackers. An adequate model of response will have to incorporate these human factors;

otherwise, an important source of errors and costs will be omitted.

The two main characteristics of attack behavior that is considered in the man-machine interaction model are attack frequency and creativity. While there is no definitive model of intruder behavior, the work by Jonsson and Olovsson [JO97] suggests that attack frequency follows an exponential distribution. We assume that creativity—directly correlated to attack severity—follows a normal distribution. On the basis of these two assumptions, we build a black box model of attack behavior, which is sufficient, given our focus on response teams.

We assume that defense teams will be explicitly formed in response to attacks. Such teams are commonly discussed in the literature on cyber incidents [WB01] and network forensics [Yas01]. To determine how teams contribute to the response in the man-machine model, it is necessary to develop a model of team dynamics that determine the development rate for response and quality of the response. Our group behavior model is shown in Figure 2.

The key variable in the model is *decision style*, which refers to the degree to which the team is systematic and rational in developing its response. One adaptive and two non-adaptive decision styles are considered. The adaptive response, a *vigilant* decision process, exhibits systematic, thorough making reflected in six characteristics [JM97]: (1) thorough search for information; (2) unbiased and thorough assimilation of information; (3) thorough canvassing of objectives and goals; (4) thorough canvassing of alternative solutions and responses; (5) careful evaluation of the positive and negative consequences of choices; and (6) thorough planning for implementation and contingencies. The first nonadaptive decision style, *nonresponsiveness*, is characterized by general lethargy and lack of response on the part of the team. In this style, the team does not engage in careful information processing and does not realize there is a situation that requires a response. The other nonadaptive decision style, *hypervigilance*, is at the other end of the scale and involves an overreaction to the situation characterized by frantic activity and hasty choice in which the team seeks an immediately available solution and then is willing to quickly abandon it for another seemingly

plausible solution, which it may then abandon, and so on in a cascade of ineffective activity. Hypervigilant teams are teams in a panic.

In terms of our two mediating outcomes, rate of response development and quality of the response, the decision styles have different results. Vigilance results in the most rapid rate of response development and the highest quality response. Hypervigilance leads to a lower rate of development and lower quality. Nonresponsiveness leads to the lowest rate of development and lowest quality because it results in no response at all or a tepid one at best.

A system of factors influences the decision style enacted by the team. *Information processing efficiency* is the degree to which the team can absorb, interpret, and route information into the proper category (attack, problem, solution, goal, consequence, requires no response). Each team is assumed to have an optimal efficiency level determined by the number of members and their talent level. *Preparedness* is the degree to which the team is ready to deal with attacks. *Stress* is the mean level of negative arousal the team has. *Overconfidence* is the degree to which the team assumes it can handle any problem without much difficulty. As the model indicates, information processing efficiency and preparedness are positively related to an adaptive decision style, whereas stress is related in an inverted U relationship and overconfidence negatively. Specifically:

- Vigilance is the decision style when information processing efficiency and preparedness are high, stress is moderate, and overconfidence is low.
- Hypervigilance is the style when information processing efficiency is low, preparedness is moderate, stress is high, and overconfidence is high.
- Nonresponsiveness is the style when information processing efficiency, preparedness, and stress are all low and overconfidence is high.

Tracing the system to the determinants of the four factors, information processing efficiency is related to information load (the amount of cues and messages to be processed) in an inverted-U curve. At low levels of load the team is likely to be bored and inattentive, but as load increases the team will involve more of its resources in processing until it achieves optimal level and all members are

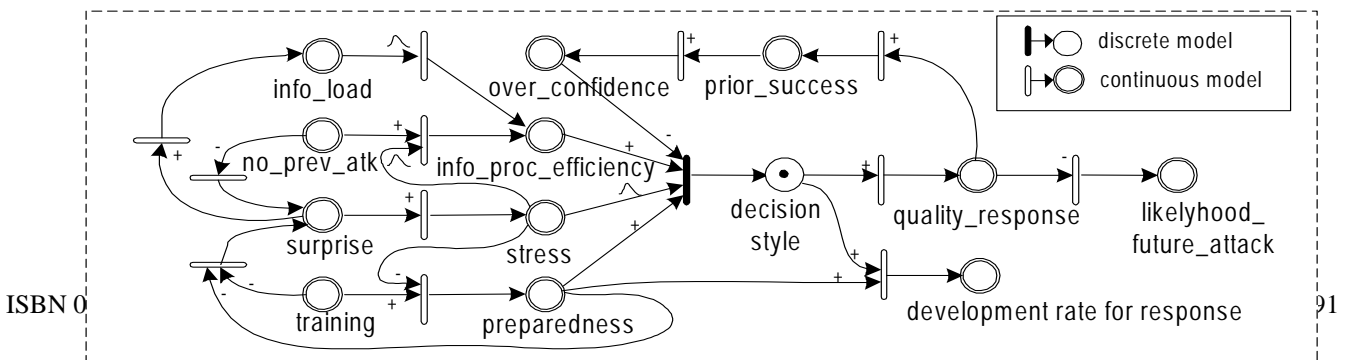


Figure 2. A Simple Model on the Mental States of a Group

engaged; however, at a certain point the team has too much information to process and it lowers standards and ignores or delays processing of some input, lowering efficiency. Stress is also related to information processing efficiency in an inverted U relationship. At low levels of stress the team is inattentive to cues and as stress rises to moderate levels the team is able to handle information effectively; however, high levels of stress cause the team members to exhibit stress reactions such as inability to focus, irritability toward other members, etc. which friction lowers efficiency. Finally, the number of prior attacks is positively related to information processing efficiency, as the team members are “on edge” and ready to recognize attacks. As attacks are fewer, the team is less likely to be ready to recognize cues signaling an attack. Both information load and stress are increased in a linear positive manner by surprise. Information load is increased because surprise causes the team to scan for information, picking up many relevant and irrelevant items. Stress is increased because surprise arouses the team.

Preparedness has a positive linear relationship with training. Teams that train for attacks are more likely to be prepared than those that do not. Stress reduces preparedness in a negative linear fashion because of stress reactions. Surprise is a conduit for two indirect effects on stress. The number of previous attacks reduces surprise; other things being equal, teams that have experienced prior attacks are less likely to be surprised. Also, preparedness reduces surprise as well, for obvious reasons. Overconfidence is a positive function of prior success, which is a function of quality of response. Teams that have handled problems effectively in the past tend to become confident. This is not a problem except for the high range of confidence (overconfidence).

This model directly influences the attack-defense process through its impact on rate and quality of solutions (response) developed. The attack-defense process influences the group dynamics model through increasing pressure and surprise, and through modifying confidence level based on success-failure rate of the group’s strategy.

### B. System Modeling

The key to modeling of machine behavior is achieving scalability and resolutions for simulation and computation of the analytical models. For our purposes, it is neither practical nor necessary to capture fully the complex details of the software making or operational modes. In the simplest form, we assume that three major types, *development*, *deployment*, and *operation*, are needed to characterize the major activities related to an information system (see Figure 3). In the system model, the

information system can be a business application suite, a networking protocol, an operating system, or even an attack tool. The development phase can be further refined into multiple action stages such as learning, architecture design, and programming. The deployment phase can be refined in a similar way. For instance, one could divide the types of the software deployment strategies into CD distribution, download, etc., each of which could have drastically different performance characteristics. The target system can be in one of many states: off-line, normal, or be in one of a number of possible non-normal states. These non-normal states are expected to lower business utility and are generalized as “degraded modes”. Case by case, each state and mode would have multiple input variables that would affect the state transitions and/or the software system’s productivities. Furthermore, development and deployment of the software systems are expected to occur in parallel with their on-line (and older) versions. From the modeling viewpoint, the different releases of the same software can be considered different software systems, and be treated as such in their mathematical representations. As needed, the system model can add more development stages easily.

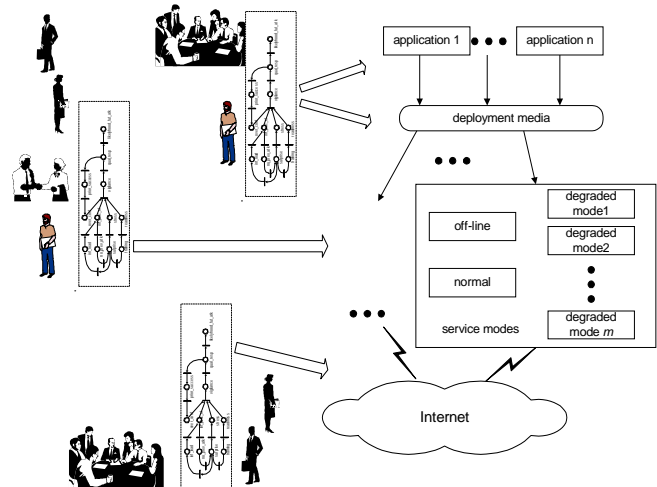


Figure 3. A simple man-machine model for development/management of software systems

After the state configuration for a system is defined, the next step is the creation of state transitions and their firing rules. Users and intruders use different software applications to access data or to affect other applications (i.e., attacks.) Their behavior determines the state transitions, including creation and destruction of various states. It is fairly easy to simulate machine behavior using a wide array of simulation and analytical tools, e.g., stochastic/static Petri-net, for different firing rules and time distributions of significant events.

The primary benefit of the proposed man-machine model is that it allows one to comprehend the impact of human skills on the overall information assurance quality. For certain cases, a common measure can be a simple and cost-effective solution; however, for others, a formal virtual warfare strategy may be more appropriate. In the next section, a stochastic, hybrid model is presented to represent and simulate the human-machine interaction model.

### III. THE APPROACH: HYBRID PETRI-NETS MODELING

Stochastic hybrid Petri-net models [PL95, CS98, DK98, CMB93] and simulation tools [Web02a, Web02b, Web02c] are adapted, which have been developed in the control system area to represent, understand, and manipulate complicated states of system components. In this paper, the model is extended to approximate the *positive* and *negative* relationships commonly used in the group behavior model. Our approach for the modeling is to develop basic components (e.g., continuous and discrete places/transitions) and their interconnections, i.e., firing rules for continuous-continuous, discrete-discrete, and continuous-discrete state transitions, so that one can freely develop one's own virtual vulnerability models in their domains.

In a continuous model, marks are considered as a real quantity by subdividing whole marks in infinitesimally small parts of the marks (called "tokens"), whereas marks are treated as integers in a discrete model [CS98]. Even if the mere passing of time does not have direct effect onto the state of a discrete event model, general Petri-net models are extended to a Discrete Petri Nets (DPNs) by introducing time variables in the firing vector,  $V$ , similar to the one proposed in [CS98]. In DPNs, state changes when a transition is fired and represented as

$$M(t^+) = M(t) + C^d \bullet V(t) \dots \dots \dots Eq. (1)$$

where  $M(t^+)$ , the next following marking function, is driven by the current marking function,  $M(t)$ , according to firing vector,  $V(t)$ . Firing speed is represented by  $V(t)$ .  $M(t^+)$  and  $M(t)$  are elements of a set of integer-number marking vectors  $\mathbf{M}$ . Initial marking,  $M_0$ , is defined by  $M(t)$  at time  $t = 0$ . Driven by an event,  $V(t)$  determines an instant transition with zero duration.  $C^d$  is an incident matrix of DPNs corresponding to the weights of the links (or arcs). Thus, in DPNs, the amount of marking change caused by a state change,  $M(t^+)$  minus  $M(t)$ , is  $C^d \bullet V(t)$ .

Several approaches to defining Continuous Petri-Nets (CPNs) are presented in [PL95, CS98, DK98, CS99, CMB93, RFS97], depending on their compatibility with

DPNs. Instead of firing the transitions at certain instants with zero duration, our approach is a continuous firing with flow  $V(M(t), t)$  that may be externally generated by an input signal and may also depend on the continuous marking vector  $M(t)$  [PL95]. The amount of marking change caused by a state change, in CPNs, is described as a nonlinear differential equation in Eq. (2) [PL95].

$$\dot{M}(t) = C^c (M(t)) V(M(t), t), \quad M(t) \geq 0 \dots \dots Eq. (2)$$

where  $C^c$  is the incidence matrix corresponding to the continuous weights. A transition is continuously fired with flow speed,  $V(M(t), t)$ , if the markings of all places into this transition are greater than zero. Note that Eq. (2) is a *differential equation* for representing the marking change amount. In order to represent positive and negative relationships shown in the group behavior models (Figure 2), we use  $\dot{M}_{in}(t)$  and  $\dot{M}_{out}(t)$  ( $\in \dot{M}(t)$ ), which are marking change amount in an input and output place, respectively. If  $\dot{M}_{in}(t) \dot{M}_{out}(t)$  is greater than zero, the firing vector represents a *positive* relationship between a place and the following one; however, if  $\dot{M}_{in}(t) \dot{M}_{out}(t)$  is less than zero, it represents a *negative* relationship. If  $\dot{M}_{in}(t) \dot{M}_{out}(t)$  is equal to zero, it means the states of one place or both places did not change (i.e., constant value at time  $t$ ). Note that positive relationship means that the input flow amount of the following place increases (or decreases) as the output flow amount of the previous place increases (or decreases). Likewise, a negative relationship means that the input flow amount of the following place decreases (or increases) as the output flow amount of the previous place increases (or decreases).

In addition, some relationships in the group behavior model are expected to have time delay ( $=\Delta$ ) in transitions as shown in Figure 4 (a). In Figure 2, for example, *training* is expected to take positive effect on *preparedness* after group members experience some mistakes in a limited period that will be translated into time delay between the target states. In Figure 4, we present *Asynchronous CPNs* in Eq. (3) as well as *Synchronous CPNs* in Eq. (2).

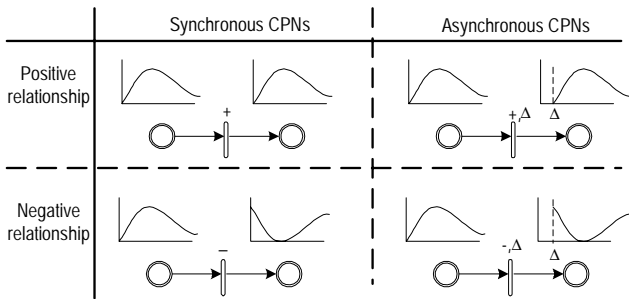
$$M(t^+) = M(t) + C^c \bullet V(t, \Delta) \dots \dots \dots Eq. (3)$$

Time delay  $\Delta$  is initially specified for each relationship, but it will be elaborated and verified according to experiments in the future. Each quantitative amount of place (e.g., *quality\_response*) may be different from that of others (e.g. *likelihood\_of\_future\_attacks*). Instead of using the operator "=" only, a general operator "φ" is

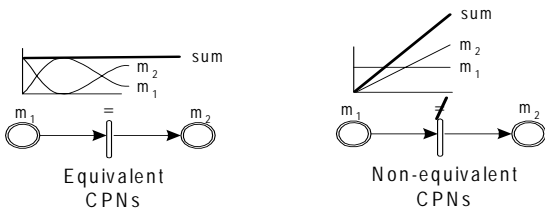
proposed, where  $\phi$  can be any operator out of the operators “<, ≤, =, ≠, >, ≥”.

$$M(t^+) \phi M(t) + C^c \bullet V(t) \dots\dots\dots Eq. (4)$$

Eq. (4) is called *Non-equivalent CPNs*, whereas *Equivalent CPNs* are shown in Eqs. (1) - (3). The sum of marks in places has constant values in Equivalent CPNs; however, the sum of marks in places is not constant in Non-equivalent CPNs, as shown in Figure 4 (b).



(a) Synchronous/Asynchronous CPNs

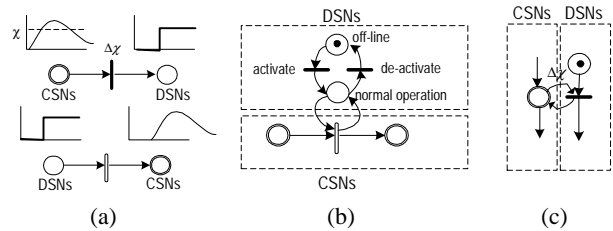


(b) Equivalent/Non-equivalent CPNs,

**Figure 4. Various Types of CPNs/DPNs**

Various interfaces between continuous and discrete models are shown in Figure 5. Continuous places (states) can be transformed into discrete places through state quantization techniques, or vice versa, as shown in Figure 5 (a). For example, decision styles of *vigilance* are discrete states transformed from continuous states such as *overconfidence*, *information process efficiency*, *stress*, and *preparedness*. Different quantization techniques are used to this type of transformation. An example [PL95] of the simple quantization transformation is that the threshold of the intermediate arc ( $\Delta\chi$ ) decides the amount of the corresponding jump, and is used to interpret discrete marking as a quantization of the continuous making. The development of quantization methods is out of our focus in this paper. However, more sophisticated methods can be found in the literature, including fix-rate scalar quantization [Ree38], feedback vector quantization [DG81], multi-stage vector quantization [JG82], and

universal quantization [Kie93]. The interface shown in Figure 5 (b) enables us to control the flow of continuous states by discrete states (just like “on/off” switches). Off-line or normal service modes shown in Figure 3, for example, are discrete system states that can turn on or off continuous group factors such as *information load*. Another type of interface shown in Figure 5 (c) is used to represent discrete states affected by continuous states. This type of interface, for example, can model *decision style*.



**Figure 5. Interfaces between CPNs and DPNs**

#### IV. AN EXAMPLE: DISTRIBUTED DENIAL OF SERVICE (DDoS) ATTACKS

This section presents a hybrid man-machine interaction model for distributed denial of service (DDoS) attacks using the proposed model. The hybrid DDoS model is composed of three model components: monitoring, control, and group behavior modules. The monitoring module shown in Figure 6 (a) models the process of monitoring the input traffic patterns of service systems from the Internet using backward propagation feedback control algorithm, presented in [XLS01], as an initial intrusion detection process. The algorithm detects abnormal traffic patterns, called “hot spots,” inside a machine (upper dot-line box) shown in Figure 6 (a-1) and outside the machine (bottom dot-line box) shown in Figure 6 (a-2). Total traffic rate (total\_traffic\_rate) is determined by these two inputs.

The control module depicted in Fig. 6(b) is to control throttle of network (throttling\_on/off) via traffic control indicator (under\_attack/normal) of the real system states through human verification of DDoS attacks. The verification is performed based on total traffic rate in the monitoring module. Through the notification transition (notify) to the control module, if DDoS attacks really happened, the systems’ state is changed to ‘under attack’ (i.e., pass a token into under\_attack). Under attack, administrators turn on the throttling to reduce network traffic (i.e., pass a token in throttling\_on from throttling\_off) to block suspicious packets. This is one of degraded modes under attack, as shown in Figure 3.

The group behavior module is presented in Figure 6 (c), which interacts with the system modules such as control module. Typical group behavior or to-do list under DDoS attacks is studied in [Eri02]. The administrators verify potential victim systems when the monitoring software makes warning flags, then identify, test, install, and execute effective defense mechanisms. If the large number of systems (measured by *num\_cur\_atk*) is attacked at the same time, both *information load* (*info\_load*) and *surprise* of administrators increases, and they finally affect *decision style*. Based on three styles of the decisions, quality of response (*quality\_response*) and rate of response development (*development\_rate\_for\_response*) are determined, which are key factors of how effectively the group responded with defensive mechanisms and how fast the systems can get back to normal operations by deactivating the throttling from on to off (i.e., moving a token from *throttling\_on* to *throttling\_off*), respectively.

## V. CONCLUDING REMARKS

Our contribution is to develop a man-machine interaction model using hybrid (continuous-discrete) Petri-net techniques to understand human group behavior, machine states, and their interactions under emergency situations such as  $V^2$  attacks. To the best of our knowledge, this is the first innovative effort to incorporate group behavior dynamics into the man-machine model of information vulnerability. The man-machine model enables us to assess not only vulnerability from machines, but also vulnerability from human behavior such as human misbehavior or lack of experience.

Simulating the proposed model is currently under development. Simulation results of tradeoff analysis will help to understand bottlenecks in the attack-defense process under the dynamics of  $V^2$ , thus help optimize human and system global resource allocation. We will also develop  $V^2$  economics assessment models in order to further assess secure software development, operation, and maintenance costs, and understand their tradeoffs using the man-machine model.

## VI. REFERENCES

- [Bis99] M. Bishop. Vulnerabilities Analysis, *Proceedings of the Recent Advances in Intrusion Detection*, pp. 125-136, September 1999.
- [CMB93] G. Chiola, M. Marsan, G. Balbo, and G. Conte. Generalized Stochastic Petri Nets: A Definition at the Net Level and Its Implications, *IEEE Trans. On Software Engineering*, Vol. 19, No.2, pp. 89-106, Feb 1993.
- [CS98] M. Chouikha and E. Schnieder. Modelling of Continuous-Discrete Systems with Hybrid Petri Nets, *Proceedings of the IEEE International Conference on Computational Engineering in Systems Applications (CESA '98)*, P. Borne, M. Ksouri and A. El Kamel, Eds., IEEE, Hammamet, pp. 606-612, 1998.
- [CWS99] M. Chouikha, S. Wegele, and E. Schnieder. Modeling and Analysis of Continuous-Discrete Systems with Hybrid Petri Nets, *The fourteenth World Congress of IFAC (International Federation of Automatic Control)*, Beijing, Hammamet, Vol. C, pp. 509-514, 1999.
- [DG81] J. G. Dunham and R. M. Gray. Joint Source and Noisy Channel Trellis Encoding, *IEEE Trans. Information Theory*, Vol. 27, pp. 516-519, July 1981.
- [DK98] I. Demongodin and N. Koussoulas. Differential Petri Nets: Representing Continuous Systems in a Discrete-Event World. *IEEE Trans. Automatic Control*, Vol. 43, No. 4, pp. 573-578, 1998.

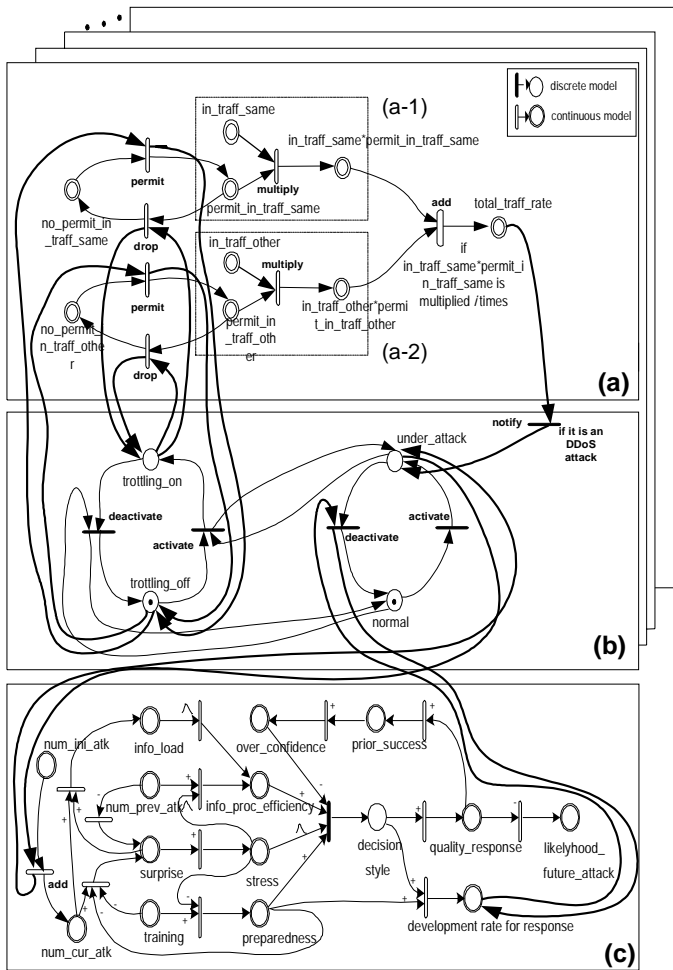


Figure 6. An Example of Man-Machine Interaction Model

- [Eri02] Eric Cole, *Hackers Beware*, New Riders, 2002.
- [GKB00] D. Gilliam, J. Kelly, and M. Bishop. Reducing Software Security Risk Through an Integrated Approach, *Proceedings of the Ninth IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 141-146, June 2000.
- [GKP01] D. Gilliam, J. Kelly, J. Powell, and M. Bishop. Development of a Software Security Assessment Instrument to Reduce Software Security Risk, *Proceedings of the tenth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 144-149, 2001.
- [HK99] G. Horton and M. Kowarschik. Discrete-Continuous Modeling Using Hybrid Stochastic Petri Nets, *Proceedings of European Simulation Symposium (ESS '99)*, Erlangen, Germany, SCS Publishing House, 1999.
- [JG82] B. H. Juang and A. H. Gray, Jr. Multiple Stage Vector Quantization for Speech Coding, *Proc. Intl. Conf. On Acoust. Speech, and Signal Processing*, Vol. 1, pp. 597-600, Paris, April 1982.
- [JO97] E. Jonsson and T. Olovsson. A Quantitative Model of the Security Intrusion Process Based on Attacker Behavior, *IEEE Transactions on Software Engineering*, Vol. 23, No. 4, April 1997.
- [JM97] I. L. Janis and L. Mann. *Decision Making*, New York: Free Press, 1977.
- [Kie93] J. C. Kiefer. A Survey of the Theory of Source Coding with a Fidelity Criterion, *IEEE Trans. Information Theory*, Vol. 39, pp. 1473-1490, September 1993.
- [Mur89] T. Murata. Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, Vol. 77, Issue 4, pp. 541-580, April 1989.
- [PL95] S. Pettersson and B. Lennartson. Hybrid Modelling Focused on Hybrid Petri Nets, *The Second European Workshop on Real-time and Hybrid systems*, Grenoble, France, 1995.
- [Ree38] A. H. Reeves. French Patent No. 852,183, 3 October 1938.
- [RFS97] M. Rauterberg, M. Fjeld, and S. Schluep. Goal Setting Mechanism in Petri Net Models of Human Decision Making, *IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, Vol. 3, pp. 2696-2701, 1997.
- [SV77] C. Smart and I. Vertinsky. Designs for Crisis Decision Units. *Administrative Science Quarterly*, 22, pp. 640-657, 1977.
- [WB01] B. J. Wood and J. F. Bouchard. Improving Government-wide Emergency Response to Cyber Incidents, *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY 5-6, pp. 195-198, June 2001.
- [XLS01] Yong Xiong, Steve Liu, and Peter Sun, "On the Defense of the Distributed Denial of Service Attacks: an On-Off Feedback Control Approach", *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, Vol. 31, Issue 4, pp. 282-293, July 2001.
- [Yas01] A. Yasinsac. Policies to Enhance Computer and Network Forensics, *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY, pp. 289-295, 5-6 June 2001.
- [Web02a] *Petri Nets Tools and Software website*, <http://www.daimi.au.dk/PetriNets/tools/>
- [Web02b] *A Collection of Modelling and Simulation Resources website*, <http://www.idsia.ch/~andrea/simtools.html>
- [Web02c] *ARGESIM Simulation Links website*, <http://eurosim.tuwien.ac.at/hotlinks/>