# Literature Review: The modeling of cheating in an examination enviroment using Stochastic Petri Nets

Greg Pennefather
Supervisor: Yusuf Motara

May 27, 2013

**Abstract**

Previous research into cheating and academic dishonesty has maintained a focus on the moral and social factors surrounding cheating. This research often overlooks the actual methods that are used in the process of cheating. My research project will investigate the use of Stochastic Petri Nets to model and analyse cheating methods with the intention of better understanding the conditions required for those methods to be effective.

## 1 Introduction

This paper explores the branch of modeling techniques based off the work of Carl Adam Petri known as Petri Nets. It also observes current trends in research of cheating in academic environments. Carl Adam Petri explored the idea of asynchronous components forming a connected and distributed system where multiple events could be handled concurrently. Petri Nets are a graphical formalisation to model such systems. Many different classifications of Petri Nets exist, with the most basic being Place/Transition Nets. More complex classifications include Stochastic and Generalised Stochastic Petri nets that increase the power of Place/Transition Nets by adding timings to the firing of transitions.

# 2 Petri Nets

## 2.1 Carl Adam Petri

Petri nets were created by Carl Adam Petri, an academic who published his PHD dissertation in 1961 called ”Communication with Automata”. In that dissertation he attempts to solve problems concerning the computation of recursive functions [3]. The problem was that the computational space required to complete a recursive function could not be known before the problem was solved. The only approach to solving this problem at the time was to allocate space and see if the function terminates. If it did not, more space would need to be allocated and the function would need to be executed again from the beginning [3].

Petri believed that starting the execution from the beginning was unnecessary. He proposed that it should be possible to allocate more space when necessary and continue computation. This did not match up with conventional computer architectures. These architectures required an increase in wire length to add additional capacity, resulting in a longer single transfer times. This meant that the clock speed would have to be reduced to accommodate the longer signal time. Perti's solution to this was by using asynchronous components that could be attached to components that were already part of the system [3]. These components were autonomous, allowing them to have their own clock, and were only capable of communicating with their immediate neighbors [3]. This lead Petri to believe that asynchronous systems were superior to synchronous ones. During the writing of this dissertation Petri developed the notation that would allow for the graphical and algebraic representation of these asynchronous systems [3].

Unfortunately for Petri, his work went largely unnoticed as the focus of the time was on the use of sequential processes. These processes made use of languages such as FORTRAN and large mainframe computers that operated off of a single global clock to solve numerical problems. Sequential programs were seen as superior to the slower and harder to master and parallel programs. [3]

## 2.2 Petri Nets

Petri Nets are used for the formal and graphical description of distributed systems. It is capable of handling the concurrency and synchronisation that makes up a core part of those systems [1]. They are able to do this by being made up of individual asynchronous elements that are only capable of interacting with its locally connected neighbors. This eliminates the idea of a global effect caused by a certain action [3]. This makes sense when considering examples of distributed computing systems such as a network of computers. A single computer cannot perform an action that will effect an entire network, rather it will perform an action that will effect only the machines on the network that it is directly connected to [3]. This makes it useful for modeling as individual components can be focused on.

Another aspect that makes Petri Nets a useful tool is the focus on the relationship between the conditions (places) and the actions (transitions) of a system [11]. At anytime it is possible to see which conditions in the system have been met, allowing an observer to see which conditions lead to which events occurring. This allows the model to describe the behavior of the system in a way that would be useful to a human being [3].

Due to the asynchronous nature of the Petri Nets they do not keep track of a timed sequence of events. Rather the events will be handled by the transitions in the order they enter the system. As an example an information processing system receives a piece of information and stores it in a queue where it waits to be removed and used by the processor. In a Petri Net the processor is able to process the information it is currently working on without having handle the information that is entering the queue. This is because a separate transition is handling the input. The two actions take place completely independently of each other. [11]

Petri Nets are also nondeterministic [11]. Only one transition can fire at a time, meaning that if two are activated at the same time there must be a way to determine which transition fires first. This is done using nondeterministic means such as randomness or factors that have not been modeled [11].

## 2.3 Place/Transition Net

A Place/Transition Net is the most commonly used form of Petri Net [12].

### 2.3.1 Graphical Representation

These Nets are made up of the following components:

**Places** are containers that can hold tokens. Places represent a modeled object or condition [12]. In graphical representations a place is depicted using the circle, labeled $t_0$ in Figure 1.

**Transitions** exist between places. When a required number of tokens is found at an input place a transition is activated. Activated transitions have the ability to fire, resulting in the destruction of tokens in the input place and the creation of tokens at the output place. In graphical representations a transition is depicted using a rectangle, labeled $t_2$ in Figure 1.

**Arcs** connect places to transitions or transitions to places, but never a place to a place or a transition to a transition. In graphical representations an arc is depicted using an arrow, labeled $t_3$ in Figure 1.

**Tokens** are counters that exist in places. They can be added and removed by transitions. Tokens give value to the objects represented by places. In graphical representations a token is depicted using a black circle, labeled $t_1$ in Figure 1.

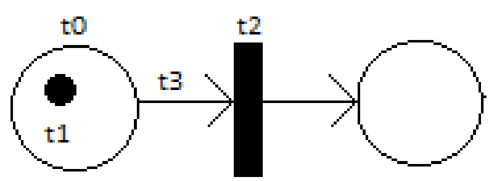


Figure 1: Basic Place/Transition Net

### 2.3.2 Formal Definition

A Petri net can be defined as a tuple made up of 5 different elements [13]

$$\rho \Rightarrow (P, T, M_0, I, O)$$

- $P = \{p_1, p_2, p_3, ..., p_n\}$ defines a set of places that is finite and contains at least one element.

- $T = \{t_1, t_2, t_3, ..., t_b\}$ defines a set of transitions that is finite and contains at least one element.
- $P \cap T = \emptyset$ as the two sets must be disjoined.
- $M_0$ defines the initial positioning of tokens. This is referred to as the initial marking.
- $I$ and 0 are functions that define the input and output places for a given transition. They are also known as arcs and are written as follows:

$$I(p_i, t_i) = n$$

$$O(p_o, t_o) = m$$

The input and the output functions come with an associated weighting, n or m. These weightings define the number of tokens that will be destroyed or created when a transition fires. The weighting of the input function also defines the threshold at which a transition will become enabled, making it capable of firing [1]. By convention, any unlabeled arc has a weighting of 1 [7].

### 2.3.3 Reachability, Boundedness, Liveness

The following are three properties that are used to analyse and qualify a Petri Net model. The reachability set for a Petri Net is a set of all the states it is possible for a Petri Net to execute into. A state can only be called reachable if it can be reached from the initial set of markings $M_0$. Given a current set of markings $m$ we can obtain the another set of markings $m'$ by activating and firing one of the transitions in the net. These two states are said to be reachable from on another, and represent a local change in the model. A problem that exists with Reachability sets and the use thereof is that they look at a net in a global sense. Petri nets are focused on local changes, and global focused changes will only obfuscate what is happening in the net. It takes away from the concurrency of the model. Despite this, there has still been a large amount of work done on the reachability set of Petri Nets [11]. A Petri Net is deemed to be live and have a Liveness property when none of its reachable markings can result in no transitions being able to fire again. This is a desirable as it prevents the system from becoming non-functional. [1] Boundedness is a property of a Petri Net that defines the maximum number of tokens that can exist in any of its reachable states at one time. This maximum is usually defined as one, but can have a higher limit depending on net. [1]

## 2.4 Timed Petri Nets

Petri Nets offer a great deal of functionality for the qualitative analysis of a system and its behaviors [1] but are unable to analyse the performance of the system. This is because the transitions of a Petri Net will fire when they are activated, giving no indication of the time that would be taken to complete a previous transition or prepare for the current transition [1]. To use Petri Nets for performance analysis a timed element must be made part of the net's description [1]. This is generally done in one of two ways:

**Timed Transition Petri Nets (TTPNs)** in which, once enabled, transitions attempt to fire after a set time interval. TTPNs are further divided into two groups, Preselection and Race models, that deal with the procedure for competing transitions for the same tokens. The Preselection model allows a transition to lock tokens for itself once activated, making them unusable until its timer expires and it destroys the tokens. The Race model does not allow for the locking of tokens, allowing them to be used on a first-come-first-served basis. This can cause the firing of one transition to deactivate another, as the tokens they were competing for have been destroyed. [1]

**Times Place Petri Nets (TPPNs)** in which tokens placed by the output of a transition are unusable for the input of a transition until a certain time has elapsed. Unlike TTPNs, TPPNs have no problem with race conditions as all transitions fire immediately when activated. [1]

Timed Petri Nets can be further classified depending on whether they use stochastic or deterministic timer intervals. The former are referred to as Stochastic Petri Nets while the latter are referred to simply as Timed Petri Nets [1].

## 2.5 Stochastic Petri Nets

### 2.5.1 Stochastic Theory

A system that is deterministic will always yield the same outputs for a given set of inputs[1]. However not many real world systems can be said to be deterministic. This is due to the fact that most systems are incapable of predicting what values they will be given and the time and sequence that they will be given them in. Newly inputted data may cause interrupts or changes to the system that will alter the outputs of the system, meaning that it can not be said to be deterministic [1]. We call these stochastic systems as they rely on a Stochastic process for their input values.

A Stochastic process is a set of random values that exist in a variable over a period of time. Together these values for a set called the state space of the function X(t), where X is the variable containing one of the random values and t is the time [1]. A formal definition is as follows [10]:

$$\{X(t), t \in T\}$$

where normally

$$T = [0, \infty)$$

A Stochastic process is space-discrete if the state space of X(t) is finite. This classification of process is also referred to as a chain [1]. A Stochastic process is time-discrete if the set of t is finite [1]

### 2.5.2 Markov Chains

A Markov process is a spesific clasification of Stochastic process that holds what is called the Markov property [1]. This Markov property is a conditional probability density function, the result of which is that future values of the process will be determined only by the present state. In systems where the output is based off of the current state and n previous states, the state space could be redefined to $N^n$. In this new state space each state will include a grouping of n sequential states [1]. Markov processes are used in the construction of discrete event systems despite most real life systems not holding the Markov property. This is due to the fact that they are less complex to analyse than other Stochastic processes [9].

## 2.6 Formal Definition

A Stochastic Petri net is a 2-tuple of conventional Place/Transition Petri nets and an additional set $\Lambda$.

$$SPN = (PN, \Lambda)$$

where

$$PN = (P, T, I, O, M_0)$$
$$\Lambda = (\lambda_1, ..., \lambda_n)$$

$\Lambda$ is a set of transition rates ($\lambda_n$). These rates are used in the following formula to calculate the firing interval for each transition [1]

$$F_{X_i}(x) = 1 - e^{-\lambda_n x}$$

Here the value of $x$ is a random variable existing at the time of calculation in a Stochastic Process. The function results in a random exponentially distributed interval time for each transition once it becomes activated [9].

In Figure 2 the initial markings result in transition $\beta$ firing after a random delay, causing it to destroy the token in OFF and add a token to ON. This token results in the activation of both $\alpha$ and $\mu$ who will each generate a randomly distributed exponential delay to wait before firing. This way it is possible for either transition to fire, depending on the random delays.

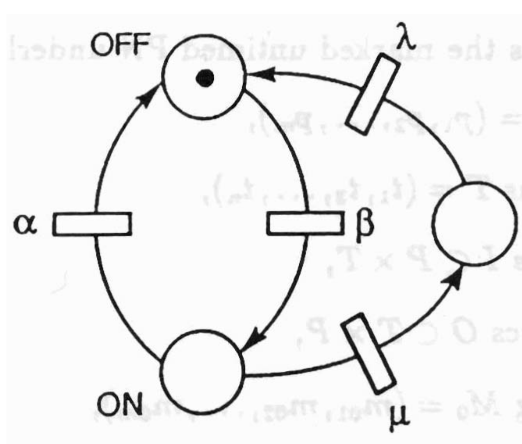


Figure 2: Stochastic Petri Net representing On, Off and Failed states [9]

In Firgure 3 the initial markings have a single token being places in P. In this Stochastic Petri Net transition $I(p, t_1) = 1$ and $I(p, t_2) = 2$. When a single token is found in P, $t_1$ is activated and its timer begins. At this point $t_2$ has not yet been activated and it appears that $t_1$ will be allowed to destroy the token. However, should $t_0$ fire a second token would be placed in P. This would activate $t_2$ and allow it to begin a timer that could be longer or shorter than the time remaining on $t_1$. This means that although the previous state meant that $t_1$ was activated and $t_2$ was not, it will not affect the number of possible outcomes for this current state. This makes Stochastic Petri Nets Markovian Chains as they contain a discrete number of possible states where only the current state effects the future states [9].

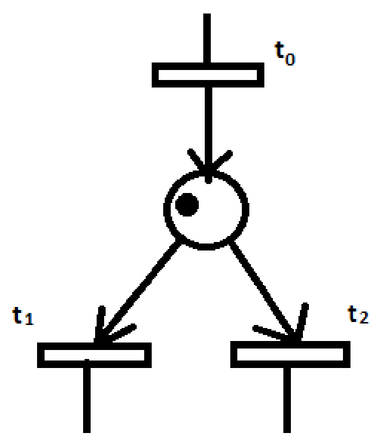


Figure 3: Weighted Stochastic Petri Net

### 2.6.1 Reachability:

Due to the fact that firing delays were calculated using a set of nonnegative real numbers there is a nonzero probability that each transition will fire once activates [9]. This can be seen in Figure 3, where it is possible for either transition to fire. This means that the reachability set of the SPN is the same as that of the PN that it is based off [9]. This also implies that the structural properties of the SPN will be the same as the structural properties of the PN [9].

### 2.6.2 Generalised Stochastic Petri Nets

Generalised Stochastic Petri Nets (GSPNs) add an additional classification of transition to Stochastic Petri Nets [1]. These transitions fire as soon as they are activated much like a transition in a Place/Transition Net. They are referred to as immediate transitions [1] and are represented graphically using a thin black bar [12]. Immediate transitions are used to represent activities that do not take an amount of time to complete that is worth considering. They take priority over timed transitions and make use of a probability mass function to decide the order when multiple immediate transitions are attempting to fire. [12]

GSPNs also add an additional arc classification named an inhibitor arc. An inhibitor arc can only connect a place to a transition and are represented graphically by a line with a circle on the transition side. An inhibited transition may not fire if the attached place contains more tokens then the weighting of the arc. [12]

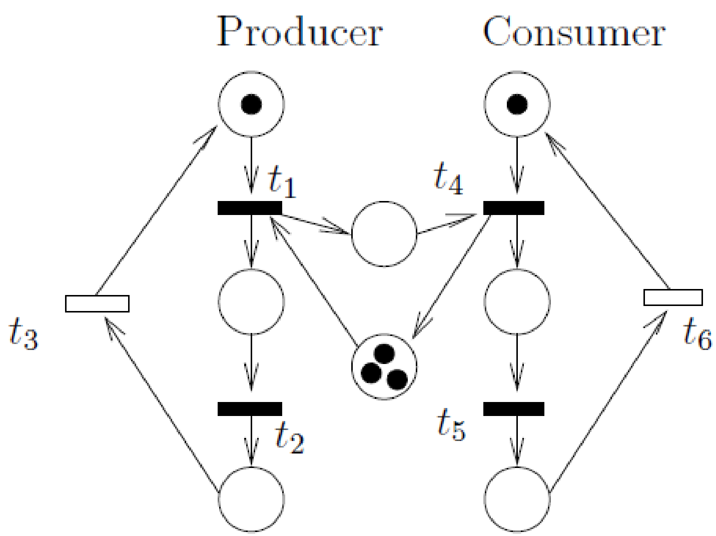


Figure 4: Generalised Stochastic Petri Net

Generalised Stochastic Petri Nets are defined in a 4-tuple that, like Stochastic Petri Nets, expands on Place/Transition Nets [4].

$$GSPN = (PN, T_1, T_2, W)$$

where

- $PN = (P, T, I, O, M_0)$ is the base Place/Transition Net.
- $T_1 \subset T$ and contains the timed transitions that can be found in a normal Stochastic Petri Net.
- $T_2 \subset T$ and contains immediate transitions added by the Generalised classification.
- $T_1 \cap T_2 = \emptyset$ and $T = T_1 \cup T_2$ means there are no intersection between $T_1$ and $T_2$ but together both make up T.
- $W = (w_1, ..., w_{|T|})$ where w is either the delay of a timed transition or the token weighting of an immediate transition.

## 2.7 Tools

Platform-Independent Petri Net Editor 2 (PIPE2) is an open source development and analysis tool for Generalised Stochastic Petri Nets [5]. The tool was created in Java by a team postgraduate students at Imperial College London in early later 2002 [5]. The tool allows for the creation of Places, Immediate Transitions, Timed Transitions, Inhibitors, Arcs and Tokens. It also allows you to individually step through the running of the net by firing off single or even multiple timed events. PIPE2 has been designed to be extensible [5]. It allows for further analysis functionality to be added by users with plugin modules [5].

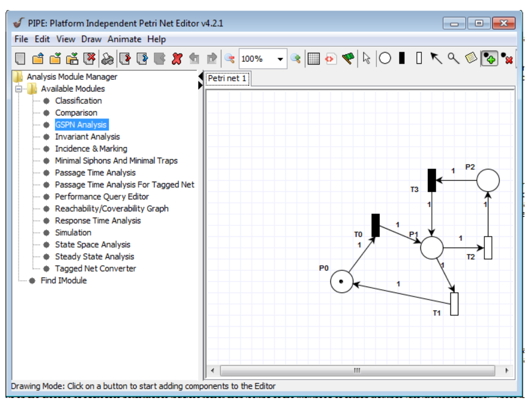


Figure 5: PIPE2

PIPE2 also has a query editor tool much like what can be seen in other query languages [5]. Queries are made using a formalism created at Imperial College London called Performance Trees [5]. Performance Trees provide a way to easily extract specific performance measures and determine specific properties of a system model [4].

# 3 Cheating in examinations

Western Universities have been aware of cheating and academic dishonesty for a sometime [2]. It is a problem that is hard to tackle due to its nature. In recent decades there have been a large number of different new evaluation methods put in place in order to help prevent cheating [6]. These can be broken up into three distinct groupings. They are preventing access to information during evaluation, evaluating those who share information equally and an increased accountability in the integrity of academic work. Prevention of access to information attempts to cut down on information sharing though individual examinations. These examinations can come in the form of closed book, open book and oral examinations where students are prevented from communicating with each other for the duration of the examination [6]. Group evaluations with larger workloads allow for those who share information to be given an equal mark [6]. Finally the use of plagiarism warnings and 'own work' declarations discourage cheating by making students aware of the consequences and repercussions [6].

However this has not managed to discourage students from cheating. Many students believe that cheating is socially acceptable [2] and do not discourage or report their peers who they find cheating [8]. More concerning is the number of academic supervisors who have ignored evidence of cheating [2]. Many supervisors choose not to do so because of the discomfort caused by reporting them to a university authority. Rather supervisors attempt to handle the students personally without involving the university [2].

Current studies in the field tend to focus on the moral and social issues surrounding cheating. They do not explore the way that students cheat or when students cheat. Instead they try to explain why certian students are cheating.

## 3.1 Previous Studies

In 1995 a Swedish-Finnish university performed a study that attempted to discover the frequency of confessed cheating, the kinds of cheating were most common, what the relation between cheating and sex is, and how their results related to British results [2].

The study distinguished between four types of cheating behaviors:

- Individual Opportunistic cheating
- Individual Planned cheating
- Active Social cheating
- Passive Social cheating

### 3.1.1 Fequency

Students participating in the study were given a questionnaire containing a list of different cheating methods and asked to mark off which they had engaged in. The list contained 23 different methods that could be used under different circumstances including coursework, research and examination environments. The study also classified the methods on the list according as either social or individual cheating and had a special classification for altruistic cheating [2]. The results of the questionnaire revealed that three quarters of students had engaged in at least one of the methods on the list [2]. However in a final question that asked whether they felt they had ever cheated overall only 63.5% of students felt they had [2]. This again indicates that students who engage in academic dishonesty sometimes don't believe that they done anything wrong [2].

### 3.1.2 Methods

For the purpose of this literature review we will focus on the methods of cheating used during examinations. According to the results of the questionnaire they are as follows:

- Copying during an exam
- Illicitly gaining advance information about the contents of an examination paper
- Taking unauthorised material into an examination (e.g. 'cribs')
- Premeditated collusion between 2 or more students to communicate answers to each other during an examination

- Lying about medical or other circumstances to get special consideration by examiner
- Taking an examination for someone else or having someone else take an examination for you

### 3.2 Room for further research

The Swedish-Finnish study tended to focus on the reasoning and social factors behind cheating, leaving a large number of questions about the methodology open. Most studies focus on the moral and social implications of cheating rather than addressing how cheating is done and how it can be prevented.

The questionnaire only gave one possibly opportunistic method for cheating, copying during the examination. However it did not specify how this would be done. It focused on the intention of the student to cheat, rather than discovering under what conditions and through what actions the student was able to cheat. In order to decrease the number of opportunistic cheating attempts it is necessary to know the conditions under which these situations arise. Further study should be done into the different methods students used to gain access to other students papers, and analysis should be done of the conditions required for those methods to be possible.

The questionnaire gave students a limited number of options to choose from. These are surely not the only means of cheating that are possible. At best they are archetypes that include many different methodologies much like was the case with opportunistic cheating. Further research should be done to discover these specific methodologies so they too can be modeled and analysed in order to limit the conditions under which they are possible.

## 4 Conclusion

The study of cheating in the academic world is one primarily focused on the moral and social implications. The small amount of study done on the methods used creates broad archetypes out of collections of methods such as 'copying from another student'. Individual methods should be explored and modeled in order to analyse them. This analysis will allow for deeper understanding of these methods, the conditions under which they are possible and the actions required to complete them.

# References


[1] Bause, F., and Iv, I. Stochastic petri nets an introduction to the theory, 2002.

[2] Bjorklund, M., and Wenestam, C.-G. Academic cheating: frequency, methods, and causes.

[3] Brauer, W., and Reisig, W. Carl adam petri and "petri nets".

[4] Brien, D. K., Dingle, N. J., Knottenbelt, W. J., Kulatunga, H., and Suto, T. Performance trees: Implementation and distributed evaluation. In *Proc. 7th Intl. Workshop on Parallel and Distributed Methods in Verification (PDMC08)* (2008).

[5] Dingle, N. J., Knottenbelt, W. J., and Suto, T. Pipe2: a tool for the performance evaluation of generalised stochastic petri nets. *SIGMETRICS Perform. Eval. Rev. 36*, 4 (Mar. 2009), 34–39.

[6] Hamilton, A. N., and Joyce, D. Abstract cheating, outsourcing, plagiarism: A growing problem?

[7] Jensen, K. Coloured petri nets and the invariant-method. *Theoretical Computer Science 14*, 3 (1981), 317 – 336.

[8] Lim, V. K. G., and See, S. K. B. Attitudes toward, and intentions to report, academic cheating among students in singapore. *Ethics & Behavior 11*, 3 (2001), 261–274.

[9] Marsan, M. A. Stochastic petri nets: an elementary introduction. In *Advances in Petri Nets 1989*. Springer, 1990, pp. 1–29.

[10] Marsan, M. A., Bobbio, A., and Donatelli, S. Petri nets in performance analysis: An introduction. In *Lectures on Petri nets I: basic models*. Springer, 1998, pp. 211–256.

[11] Peterson, J. L. Petri nets. *ACM Comput. Surv. 9*, 3 (Sept. 1977), 223–252.

[12] Trivedi, K. S., and Sun, H. Stochastic petri nets and their applications to performance analysis of computer networks. In *Proceedings of the International Conference on Operational Research* (1998).

[13] Valette, R. Analysis of petri nets by stepwise refinements. *Journal of Computer and System Sciences 18*, 1 (1979), 35 – 46.