Modelling Data Flow in Smart Claim Processing Using Time Invariant Petri Net with Fixed Input Data

Anokye Acheampong Amponsah[1†], Adebayo Felix Adekoya[2††] and Benjamin Asubam Weyori[3†††]
amponsah.acheampong@uenr.edu.gh Adebayo.adekoya@uenr.edu.gh Benjamin.weyori@uenr.edu.gh
University of Energy and Natural Resources, Department of Computer Science and Informatics,
P. O. Box, 214, Sunyani, Ghana

Summary
The NHIS provides free or highly subsidized healthcare to all people by providing financial fortification. However, the financial sustainability of the scheme is threatened by numerous factors. Therefore, this work sought to provide a solution to process claims intelligently. The provided Petri net model demonstrated successful data flow among the various participant. For efficiency, scalability, and performance two main subsystems were modelled and integrated – data input and claims processing subsystems. We provided smart claims processing algorithm that has a simple and efficient error detection method. The complexity of the main algorithm is good but that of the error detection is excellent when compared to literature. Performance indicates that the model output is reachable from input and the token delivery rate is promising.

Keywords:
NHIS, Petri net, claim processing, reachability, data flow

1. Introduction

The National Health Insurance Scheme (NHIS) is very beneficial to most people all over the world especially to those in developing countries. Many African countries including Ghana, South Africa, Zimbabwe, Tanzania, Kenya, and Uganda are providing free and/or highly subsidized healthcare to their citizens and residents. According to [1-2], Ghana was a pioneer in the NHIS and has succeeded relatively in the provision of quality universal healthcare in fulfilment of the World Health Organization's requirements and the Sustainable Development Goals (SDG) [3-4]. Goal 3 of the Sustainable Development Goals (SDG 3) is to secure healthy lifestyles and promote well-being for all people of all ages. The primary goal is to attain universal health coverage, which includes financial risk protection, access to high-quality essential healthcare services, and universal access to safe, effective, high-quality, and affordable necessary medications and vaccinations.

The WHO defines Universal Health Coverage (UHC) as guaranteeing that everyone can get the health care they require, regardless of time and location, without being impeded by money. It covers the entire spectrum of basic health services, from prevention to treatment, rehabilitation, and palliative care, [3, 5]. The UHC definition covers three broad objectives as follows – (i) Equity – accessibility available to all with no exception; (ii) quality healthcare – high-quality healthcare for all the healthcare recipients; (iii) financial bulwark – financial fortification covering all participants and not those who can afford. It is conspicuous enough to say that UHC theoretically provides for all categories of health care seekers without any boundary or impediments.

The ideal anticipations of SDG 3 and UHC as specified by the UN and the WHO respectively can be achieved practically if there exist enough finances to sustain the costs of seeking and delivering healthcare. However, it has been established by [6] that the NHIS, especially in Ghana, is faced with numerous challenges including financial and operational ones, and together with [7] suggest that taxes be introduced and/or increased to resolve the financial bottlenecks. The tax net involves charging the oil revenue, levying large and money-making companies, increasing Value Added Tax, improving the methods of revenue collection, introducing co-payments and capitation. Understandably, introducing new taxes or increasing old ones have a direct adverse effect on the poor households who are essentially seeking to be protected by the NHIS. According to [8], when revenue consumption is not taken into account, energy taxes may or may not have a disproportionate impact on low-income households, contrary to popular belief. Poor households are known to spend a big portion of their income on food [9], and they are also thought to spend a large portion of their budget on energy usage. The poor are also more likely to own older models of energy-consuming durable items like refrigerators and automobiles, which are less energy efficient than the newer versions used by wealthy households. The Bureau of Labor Statistics' 2014 Consumer Expenditure Survey confirms the popular belief that general energy taxes in the United States are regressive. It is established [8] that the average household

Manuscript received February 5, 2022
Manuscript revised February 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.2.53

in the lowest annual expenditure decile spends over 15% of its budget on electricity, natural gas, gasoline, and other fuels, whereas the average household in the top expenditure decile only pays 5%. Most African countries impose Valued Added Tax and NHIS Levy on products and services which go a long way to overburden the poor. Introducing more taxes or increasing existing ones may not help the poor and it is believed that no amount of taxes or level of revenue generation will suffice the expenditure of the NHIS if systems that ensure transparency, accountability, auditability, and corruption-free operations are not instituted. Digitalization of the processes of the NHIS is key but it must be done in a manner that guarantees little or no human involvement.

[10] acknowledge that the National Health Insurance Authority (NHIA) – the regulatory body of the NHIS has instituted numerous mechanisms to protect its purse and the cost of operations. However, there are still issues of overpayment of claims, fraud, and inefficiencies in the claims processing, and operations of the NHIA. As suggested by [6], the tax net can be expanded, and the existing ones increased to enlarge the financial base of the NHIA. However, if the operations are not protected from fraud and malicious personnel found in the NHIS claim processing lifecycle, corrupt practices will drain the monies into private accounts. In this light, the main objective of this work seeks to model a palpable solution to prevent fraud in the NHIS claims processing using a time-invariant Petri Net with fixed data input. The work models the data flow between and within the subsystems of the proposed model with the focus on the input subsystems and the claims processing subsystem. Descriptively, the model is designed to test the flow of data from input to output noticing the data delivery rate, conflicts, and any other issue. The benefits of this system include limiting the number of personnel involved in the NHIS claims processing lifecycle and demonstrating the transparent automation of the verification and approval of claims.

Petri net is known to develop workflow of simple and advanced business processes [11] and can even predict very important events [12]. It has been used in many domains from manufacturing systems [13], robotics, healthcare systems [14], personal lifestyle and safety [15], and system modelling and evaluation [16, 17, 18]. As much as known, no direct work has been done by simulating Petri net models in the insurance sector, and the claims processing to be specific. Therefore this work provides a novel model solution that is very interesting to many groups.

The remainder of the paper is organized as follows; section 2 highlights the contribution of this work. Section 3 also discusses basic concepts and definitions of the methods utilized in this work. Section 4 describes the system. It concentrates on the architecture, intelligent claims processing algorithm, and the subsystems before and after integration. Performance modelling is discussed in section 5 and the performance results are presented and analysed in section 6. The work is concluded in section 7 and the recommendations and future work are in section 8

2. Contribution of the Work

We propose a conceptually working model which presumably will reduce the claims expenditure, save some finances and ensure Universal Health Coverage and Sustainable Development Goal 3 as the expectations of the WHO and UN respectively. The main aim of this work is to theoretically and practically design and test a simple Petri Net model to simulate the palpable implementation of smart claims processing. By adopting Petri Net the proposed model is made more methodical and realistic. The local benefit of this implementation is the demonstration that little human involvement can be ensured in the current NHIS claims processing lifecycle, thereby eliminating or drastically reducing the rate and effect of fraud and corrupt practices. The patients whose claims are being filed are included in the lifecycle to verify the claims being submitted by their providers. This inclusion transparently checks and balances the providers and other potentially malicious personnel in the scheme.

Secondly, this work demonstrates that workflows can be used in place of smart contracts in blockchain, and as such this work provides the theoretical foundation for the development of a blockchain-based solution. Consequently, we model the NHIS insurance processes using time-invariant Petri Net.

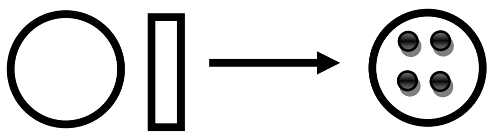
The findings of the work contribute to the findings of other researchers interested in the theories and applications of Petri nets. We propose a very simple and efficient error detection method to model a system with a finite set of places.

3. Concepts and Definitions

3.1 Basics of Petri Nets

Petri nets are a graphical and mathematical modelling tool for studying and describing information processing in systems with concurrency, asynchronicity, nondeterminism, distributed, stochastic, and/or parallel properties.

A Petri net can essentially be described as a directed graph with an initial marking M_0. The fundamental graph N of a Petri net is a bipartite, weighted, directed graph that has only two types of nodes, identified as places and transitions where arcs either output from a transition to a place or a place to transition. From the graphical perspective as in figure 1, a place is noted as a circle and a transition is drawn as a bar, rectangle (vertical or horizontal). An Arc is branded with weights with positive integers where a c-weighted arc can be interpreted as the set of c parallel arcs. A marking allocates a positive integer to each place. It can be discussed that a place p is marked with c tokens if a marking assigns a positive integer c to it. In figure 1(d), we assign $c = 4$ black dots (tokens) in the place p. A marking is denoted by M, an m-vector, where m is the over-all number of places. The number of tokens in place p is represented by the pth components of M, denoted by $M(p)$.



(a) (b) (c) (d)

Figure 1: Components of a Petri net (a) Place, (b) Transition, (c) Arc, (d) Place with 4 tokens

3.2 Formal Definition of Petri Net

In this subsection, we present a universally recognized definition of a Petri net that exists in the literature as the structure $N = (P, T, F, W)$ without any specific initial marking denoted by N. A Petri net with an indicated initial marking is denoted by (N, M_0). Therefore,

A Petri net is a 5-tuple, $PN = (P, T, F, W, M_0)$ where:

$P = \{p_1, p_2, p_3, \ldots, p_n\}$ is a fixed set of places,

$T = \{t_1, t_2, t_3, \ldots, t_n\}$ is a fixed set of transitions,

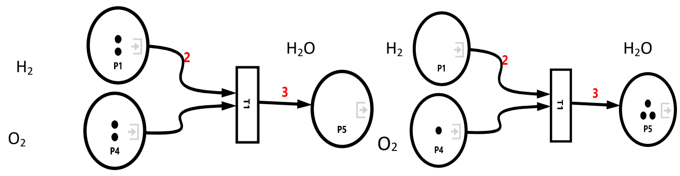
$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation from source to destination),

$W: F \rightarrow \{1, 2, 3 \ldots\}$ is a weight function,

$M_0: P \rightarrow \{0, 1, 2, 3, \ldots\}$ is the initial marking,

$P \cap T = \emptyset$ and $P \cup T \neq \emptyset$

Figures 2(a) and 2(b) show the state of the net before and after firing respectively. The transition t_1 fires with the function in Eq. (1) changing the initial marking $M_0 = \{2, 2, 0\}$ to another marking $M_1 = \{0, 1, 2\}$.



(a) (b)

Figure 2: A demonstration of a transition (firing) rule: (a) Initial marking before firing the transition T_1. (b) The resulting marking after firing T_1

3.3 Definition 1 (Petri Net)

We define three different Petri nets – (i) Data Input Subsystem, (ii) Claim Processing Subsystem and (iii) Integrated i and ii. The Data Input Subsystem Petri net is defined as PN = $\{P, T, F, W, M_0\}$ where; Place $P = \{p_1, p_2, p_4, p_5, p_6\}$. Out of these places, we have operation place = $\{p_3\}$; control place = $\{p_4, p_5\}$; resource place = $\{p_1, p_6\}$ and Transition T = $\{t_1, t_2, t_3\}$.

The Claims Processing Subsystem Petri net is also defined as PN = $\{P, T, F, W, M_0\}$ where; Place $P = \{p_{17}, p_{18}, p_{19}, p_{20}, p_{21}, p_{22}, p_{23}, p_{24}, p_{25}\}$ with operation place = $\{p_{19}, p_{22}\}$; control place = $\{p_{18}, p_{21}, p_{24}\}$; resource place = $\{p_{17}, p_{20}, p_{23}, p_{25}\}$ and Transition T = $\{t_{10}, t_{11}, t_{12}, t_{13}, t_{14}\}$.

In the integrated system there exist input places IP = $\{p_{27}, p_{28}, p_{29}\}$, output places $OP = \{p_{26}\}$ and subsystems = $\{s_1, s_2, s_3, s_4\}$ where s_1, s_2, and s_3 represent the data input models for the provider, patient and director respectively, and s_4 represent the claims processing model. Input places in PN feed the net with fixed user data. These data are acquired from the provider, patient, and director respectively. The operation place momentarily stores the processed tokens until all the input tokens needed to fire a transition as in the case of t_3, t_6, and t_9 are obtained.

For efficiency, scalability and performance, the system was demarcated into two main subsystems – the data input and claim processing subsystems. Part of the load is handled by the input processing subsystem – update, correction, or deletion of claim data. The demarcation enhanced the performance of the claim processing subsystem and the error detection scheme.

4. System Description

4.1 Model data Input Specification

The proposed model requires the same number of data inputs from the various participants using Eq. (2). For example, a maternity home P_{home} serves 200 clients and enters their claims data. The model requires these clients or their representatives to validate the claim inputs from the provider which amounts to 200 inputs from the patients. Again the director needs to approve and authorize the payment of all the claims. In the end, the output from the equation must be a positive whole number. Any other value signifies the occurrence of deadlock or unprocessed claims.

$$\alpha = m_0 \xrightarrow{t_1} m_1 \quad (1)$$

$$\mathfrak{D} = \left(\sum_{i=1}^{n} p_i + \sum_{i=1}^{n} m_i + \sum_{i=1}^{n} d_i\right) / k \quad (2)$$

$$f(\mathfrak{D}) = \begin{cases} conflict \;\; x = x.y \\ conflict - free, \; x = W \end{cases} \quad (3)$$

4.2 Smart Claim Processing Algorithm

We modified the original algorithm for smart claims processing to conform to the objective of this work. The modification was necessary because the aim was to test the data flow in the system. Therefore from algorithm 1 if the total quantity of tokens submitted by the provider is not the same as the total quantity of tokens submitted by the patient then the p_{19} *mod* p_{20} |= *0* and consequently the patient or the representative can be notified to verify by the claim submitting a token. Similarly, if the total quantity of tokens verified by the patients is not the same as the total quantity of tokens approved by the director then p_{22} *mod* p_{23} |= *0* then the director or representative is notified to submit the approval token. After checking these conditions, if a claim token is submitted by the provider, a verification token is submitted by the patient, and an approval token is submitted by the director as in line 5 then the token is fired to the output place in line 6.

Table 1: Smart Claim Processing Algorithm

	Algorithm 1: Smart Claim Processing Algorithm
1	**Procedure SmartClaimProcessing (p_{17}, p_{20}, p_{23}) WHILE True (so long as there is a claim token) DO**
2	If $p_{19} = \sum_{i=1}^{n} p_i$ mod $p_{20} = \sum_{i=1}^{n} m_i$ |= 0 then flag pat;
3	If $p_{22} = \sum_{i=1}^{n} m_i$ mod $p_{23} = \sum_{i=1}^{n} d_i p_{23}$ |= 0 then flag dir;
4	**FOR** (int i=p_{17}; i>0; i--)
5	**IF** p_i == m_i == d_i == TRUE Then
6	Send token to Output; // (add to Output Place)
7	**END IF**
8	**END FOR**
9	**END WHILE**

4.3 Conceptualized Composition of the Petri Net Model

Figure 3 presents the architecture of the Petri net model depicting the system integration. There is an intentional delay in the firing of tokens from s_1, s_2, s_3 to s_4. The implementation of the delay was purposeful to allow time for anomalies to be rectified before data are sent to s_4 for processing. When claims are successfully submitted by the providers, duly verified by the patients, and approved by the director, they are sent to the processed claims output.

4.4 Data Input Subsystem Model

The data input subsystem allows providers, patients, and directors to send their respective data in the claims processing subsystem via s_1, s_2, and s_3 respectively. Figure 4 shows the net for the provider. PTHRESH (p_3) is implemented to accommodate a threshold which when attained will trigger t_3 to fire into s_4. As indicated earlier, p_4 and p_5 are control places that enable the firing of t_1 and t_2. The control places play a crucial role in ensuring flexibility in the management and programming of the Petri net.

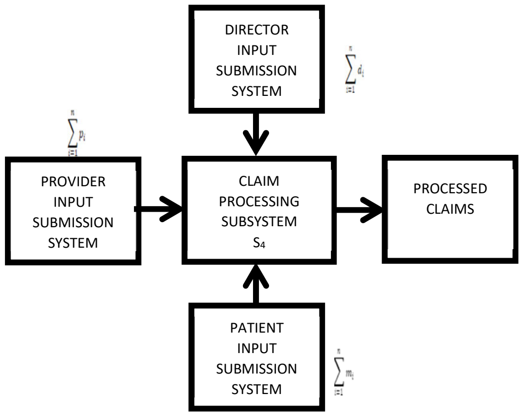


Fig 3: Conceptualized composition of the Petri net model

4.5 Claim Processing Subsystem Model

The claim processing subsystem (s_4) in figure 5 accepts inputs from the data input subsystems and processes them accordingly. *In1*, *In2*, and *In3* represent the input ports for the provider, patient, and the director respectively. *S4* processes the claims serially from the provider inputs through the patient to the director. The tokens are originally submitted by the provider after which the patient and the director send validating and approval tokens. Algorithm 1 in section 4.2 plays a momentous role in this system. For example, if the output of line 2 is not equal to 0, then the output number of tokens remains in operation place (p_{19}). Similarly, if the output of line 3 is not equal to 0, then the output number of tokens remains in operation place (p_{22}). This implementation ensures swift rectification by directing the participant involved to the exact location of the net.

However, if the output of both lines 2 and 3 are 0 then all input tokens are fired by the transitions t_{11}, t_{13}, and t_{14} to the final sub destination (p_{25}) and later forwarded through the output port *out1* to p_{26} (processed claims place). Figure 5 graphically demonstrates these processes.

4.6 The Overall Proposed System

s_1, s_2, and s_3 are the data input subsystem and s4 is the claims processing subsystem discussed in sections 4.4 and 4.5. Places p_{27}, p_{28}, and p_{29} are the input consoles of the participants. Theoretically, these places are input places, however, they can be either the keyboard of a personal computer or mobile phone or input from SMS short code. During the system implementation and testing, the threshold for the input systems was set to 10 tokens and the system was tested with different numbers of tokens. The design of the system after integration is presented in figure 6. The testing results are also presented in table 2.

5. Performance Modelling and Experimental Setup

We evaluated the Petri net marking vector components using reachability and conflict case characteristics. Unless otherwise stated, the experiments were done with the following Oscillogram options: GraphicDistance (30), GraphicHeight (50), NumbersPerSection (1), SectionWidth (25), ShowIndexes (False). The Response option had the following options: EndTime (1000). The following subsections explain the concepts and results.

5.1 Reachability

It is the case that after a series of firing, the initial marking of the Petri net results in different markings. Consequently, a marking M_n is considered reachable from marking M_0 if there exists a sequence of firings that transform M_0 to M_n. A firing sequence is symbolized as $\sigma = M_0\ t_1\ M_1\ t_2\ M_2 \dots t_n\ M_n$. Hence M_n is reachable from M_0 by σ and this can be simplified as $M_0\ [\sigma > M_n$. The set of possible markings reachable from M_0 in a net (N, M_0) is denoted by $R(N, M_0)$. The set of all possible sequences of firing from M_0 in a net (N, M_0) is also denoted by L(N, M_0). So the problem of reachability in Petri nets hinges on finding if $M_n \in R(M_0)$ for a given marking M_n in an (N, M_0) net.

5.2 Reachability Graphs

The reachability graph of the entire system was huge and for the sake of space, the graph focused on the claims processing subsystem and it was modelled to portray the system as the architecture in figure 5. Figure 7(a) and (b) show the state of the net before and after firing the transitions respectively. Figure 7 (c) is the reachability graph where the blue vertices represent the vanishing state and the red vertex represents the tangible or final state. 7 (d) is a

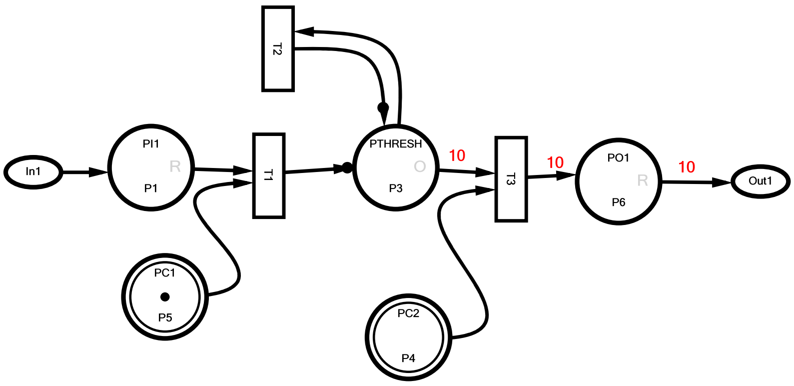


Fig 4: Data Input Subsystem Model

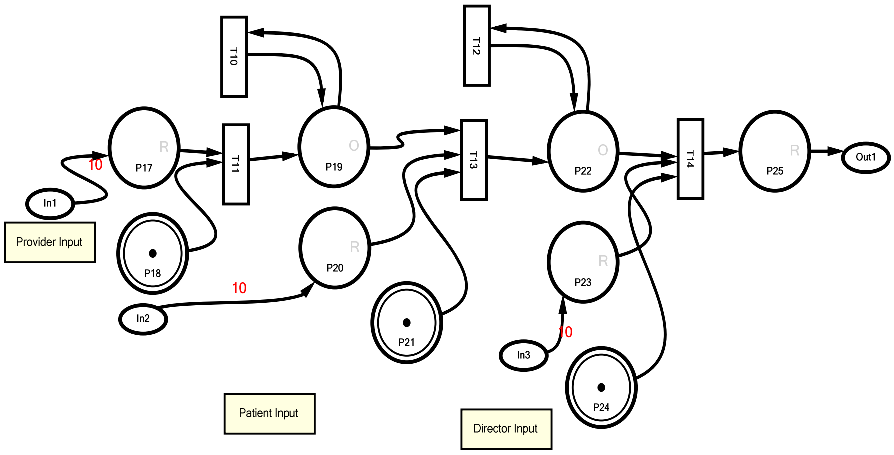


Figure 5: Claim Processing Subsystem Model

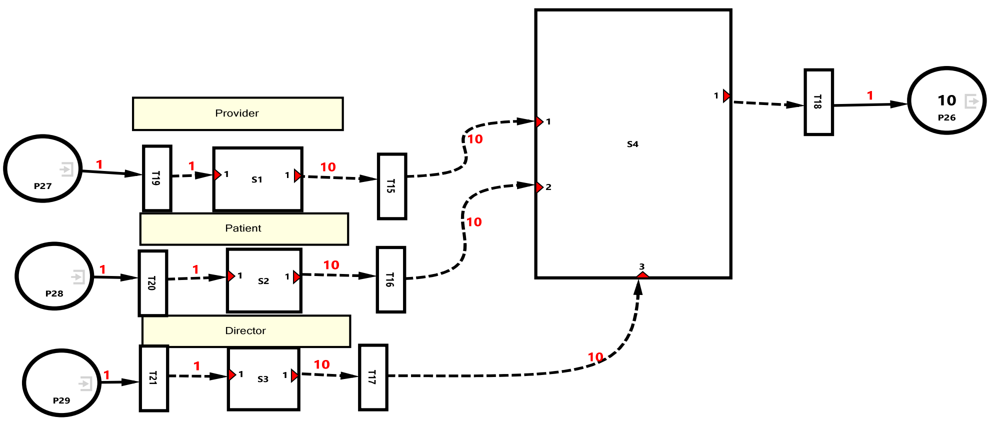
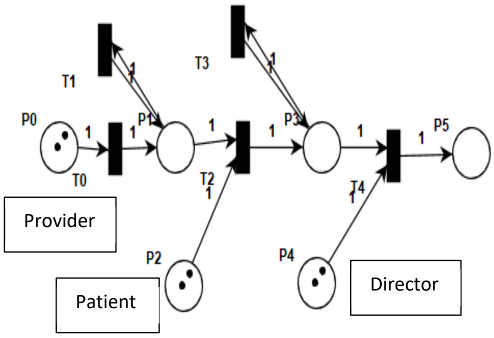


Figure 6: Unified Claim Processing system

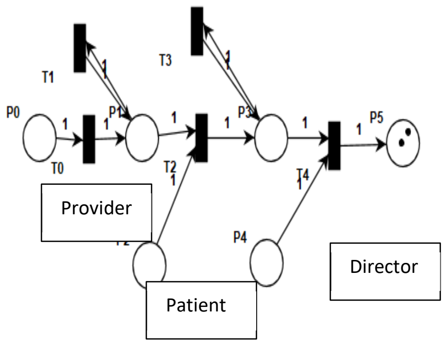
tabular view showing the state markings from s_0 to s_9. The reachability graph clearly shows that s9 (the final marking of the model) is reachable from s_0 (initial marking of the model). Computations results show that the graph was generated in 0.032 seconds and it was constructed in 0.122 seconds. Approximately the entire process lasted a total time of 0.154 seconds. For simplicity, 2 input tokens were supplied to the model.

5.3 Missing Data Error

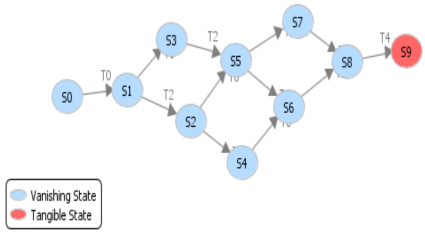
According to [19-21], missing data errors if not checked and corrected can result in system exceptions and dangling. These authors define missing data error to happen when an originally uninitiated data element is being accessed and this can be likened to a variable being used without its definition or initialization in computer programming. In this line, we propose a simple and efficient method to resolve this missing data error in our work in Eq. (2). In finite-state systems as in the case of this work, our proposed algorithm first checks the summation of all inputs before submission into the claim processing subsystem made up of a finite set of places and transitions. As already indicated the summation of all summed inputs modulus constant results in an error or conflict if the output is greater than zero. This method points to the exact location of the system where the data is missing. The results of this algorithm are shown in section 5.4 and table 2 indicating the data flow tests when the total number of inputs equals the total number of outputs.



(a)



(b)



(c)

P0	P1	P2	P3	P4	P5	
2	0	2	0	2	0	**S0**
1	1	2	0	2	0	**S1**
1	0	1	1	2	0	**S2**
0	2	2	0	2	0	**S3**
1	0	1	0	1	1	**S4**
0	1	1	1	2	0	**S5**
0	1	1	0	1	1	**S6**
0	0	0	2	2	0	**S7**
0	0	0	2	2	0	**S8**
0	0	0	0	0	2	**S9**

(d)

Figure 7: Reachability Testing (a) architecture before firing (b) architecture after firing (c) the reachability graph (d) state markings

5.4 Token delivery Rate

In the context of this work, a token can be described as a single claim data submitted by a provider, verification and approval data submitted by the patient and director respectively. In this weight, the flow of tokens from a source to the destination nodes is in tandem with the concept of reachability. Consequently, the model was tested with diverse tokens (10, 100, and 1000) for the delivery or failure rate. Table 2 shows the number of input tokens and the number of output tokens for all the subsystems. Subsystems (s_1, s_2, s_3) are the input subsystems for the provider, patient, and director while the subsystem (s_4) is the claim processing subsystem. Output tokens refer to the provider tokens that are verified and approved by the patient and the director. The case where a token is not verified and/or approved by the respective participants has been described.

Table 2: Results of the data flow tests for S_1, S_2, S_3, and S_4 subsystems and the unified system

Subsystem	*n* token	Input Token	Output Token	Success Rate (%)
	10	10	10	100
S_1, S_2, S_3	100	100	100	100
	1000	1000	1000	100
	10	10	10	100
S_4	100	100	100	100
	1000	1000	1000	100
	10	10	10	100
The Unified System	100	100	100	100
	1000	1000	1000	100

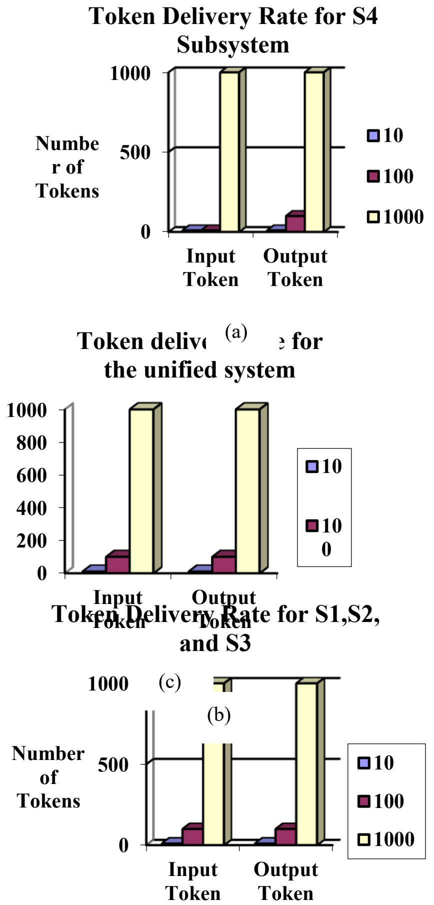


Figure 8: Data flow tests results (a) Data Input subsystems (b) Claim Processing subsystem (c) Integrated system

5.4 Conflict Scenario

Conflict can occur when a claim remains unverified or unapproved. For example, a Poly Clinic provider (P1) serves twenty patients and files their claims resulting in twenty tokens. The tokens will be delivered into S_4 and will be expecting twenty individual tokens from the patients after which twenty tokens will be expected from the director. In this case, Eq. (2) results in $\mathfrak{D} = (20 + 20 + 20) / 3$ and $\mathfrak{D} = 20$. On the other hand, for example if two patients are unable to verify, $= (20 + 18 + 18) / 3$ and $\mathfrak{D} = 18.67$. The output of decimal numbers or any other number greater than 0 results in conflict and essentially conflict is defined as there exist unprocessed claims in the S_4 – Claims Processing System.

6. Performance Results Analyses

This section presents the data flow results and resource utilization.

6.1 Data Flow

Figure 9 shows the simulation results after testing the model with real tokens (*n*). For simplicity and ease in reporting, $n = 10$ tokens each is inputted in places p_{27}, p_{28}, and p_{29}. These are input places and are engaged at 0 NumbersPerSection to feed the model with data (tokens). Transitions t_{19}, t_{20}, t_{21} fire a token each so long as the firing rules evaluate to true. The number of tokens decreased steadily and consistently to zilch and after the tenth firing, the line remains unperturbed.

Another set of equally crucial places is p_3, p_8, p_{13} also known as PTHRESH, MTHRESH, and DTHRESH respectively. These places started firing their corresponding transitions to move tokens at 2 NumbersPerSection. The graph in figure 10 shows that these places under observation commenced receiving tokens from the input places discussed in figure 9 above. The places gained tokens steadily and consistently at number 12 and the places stopped receiving tokens resulting in a levelled and unperturbed line. The data flows into subsystems s_1, s_2, and s_3 respectively as shown in figure 6 which depicts the overview of the system after integration.

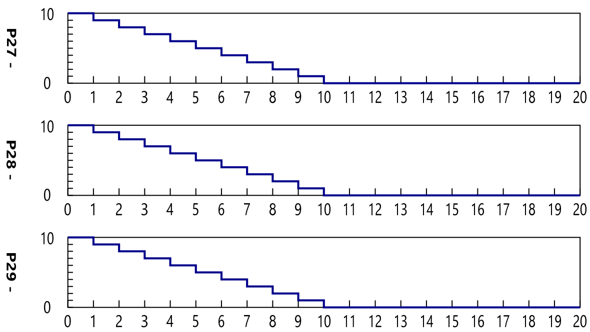


Figure 9: Data outflow results from P_{27}, p_{28}, p_{29}

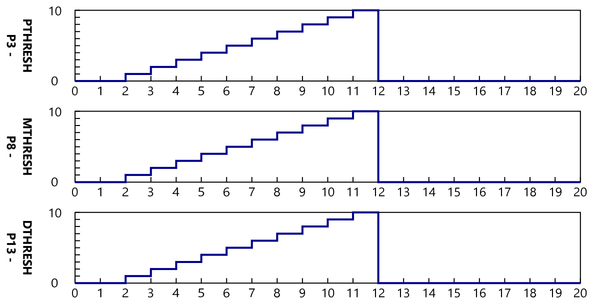


Figure 10: Data inflow results to P_3, p_8, p_{13}

Figure 11 shows the operation of the control places. P4 - PC2 is the second control place for the provider and the figure shows that it did not activate during the testing period. Similarly, P9 – MC2 and P14 – DC2 are the second control places for the patient and the director respectively. The chart indicates that they did not fire during the testing period with the testing logic. However, P5 – PC1, P10 – MC1, and P15 – DC1 are the first control places for the provider, patient, and the director respectively. These three control places concurrently started from point 2 to point 12, indicating the flow of the 10 tokens from the input to the output places respectively.

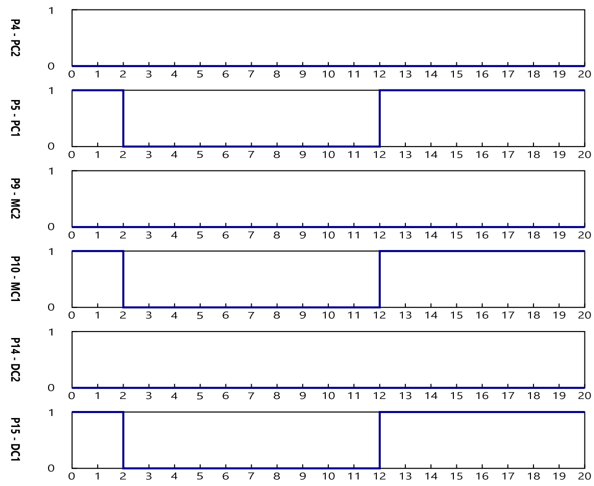


Figure 11: Activation of Control Places

Figure 12 depicts the data flow for places p_{17}, p_{20}, and p_{23}. These places receive the claims data created by the providers, patients, and director in *s4*. In this phase of the simulation, the number of claims represents the number of tokens which in this case is *n* (10). We assume that all claims are duly verified by the patients (members) and subsequently approved by the director resulting in $n = 10$ for all participants. In this manner, then all 10 claims are created, verified, and approved successfully. These places are augmented with control places {p_{18}, p_{21}, p_{24}} to manage the tokens in the case a token (claim) gets unverified and/or unapproved by their respective participants. Consequently, all the places received their tokens concurrently at number 13 in figure 12. However p_{17} dissipated the token immediately from 14, while p_{20} started from number 15, and p_{23} started from number 16. This is the case because the control place p_{18} was programmed to release a token immediately after a token arrives in p_{17} and once a token remains in there. A token from this step is fired by t_{11} into p_{19}. p_{19} requires tokens from p_{20} and p_{21} to fire t_{13}. Again, p_{21} is a control place and it is programmed to release a token once the other tokens are available. In the case that other tokens (p_{20}, p_{21}) are not available t_{10} ensures that the awaiting token(s) remains in p_{19} by looping. It makes it easy to locate the unprocessed claim at the exact level – thus either patient or director for swift rectification.

Places p_{22}, p_{23} and p_{24} are to fire t_{14} once the firing rule evaluates to true. Similarly, in the case that a claim (token) is unapproved by the director, the token remains in a loop in p_{22} using t_{12}. In figure5, In1, In2, and In3 are the input ports for the providers, patients, and the director respectively.

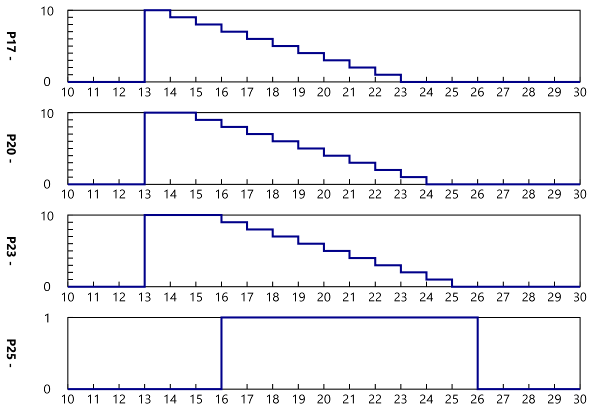


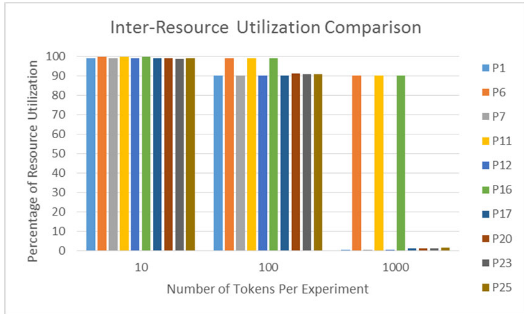
Figure 12: Claim being processed during the testing phase

6.2 Resource utilization

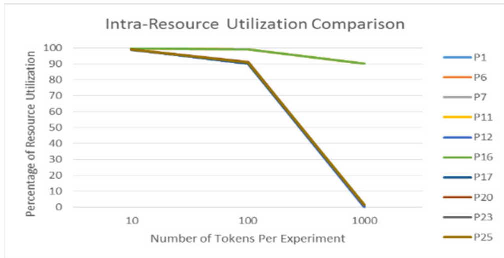
The experiment was set up to end the simulation after 1000 discretization intervals using the same number of tokens as in the data flow experiment. Table 2 and figure 13 show the results of the resource utilization in tabular and graphical presentations respectively. 10 tokens yielded the maximum usage of the resources with a minimum value of 989 and a maximum value of 999 representing 98.9 percent and 99.9 percent respectively. This is followed by n=100 with minimum and maximum values of 900 and 991 representing 90 and 99.1 percent respectively. The last experiment n=1000 yielded very interesting results indicating the least resource utilization in the Petri net model. The highest value is 901 representing 90.1 percent and the least value is 1 representing 0.1 percent. On average, places p_6, p_{11}, and p_{16} were utilized the most by the Petri net with average percentages of 93.3, 96.4, and 96.4. Figures 13 (a) and (b) throws more light on the poor resource utilization when the number of tokens was 1000 by comparing the inter-resource and intra-resource consumption statistics. This work has shown that it is possible to predict resource consumption using Petri net models as agreed by [12].

Table 3: Integrated model resource utilization

Res. / No. of Tokens	P_1	P_6	P_7	P_{11}	P_{12}	P_{16}	P_{17}	P_{20}	P_{23}	P_{25}
10	99	99.9	99	99.9	99	99.9	99	98.9	98.8	98.9
100	90	99	90	99.1	90	99.1	90	91	90.9	90.9
1000	0.1	90.1	0.1	90.1	0.1	90.1	1.3	1.3	1.3	1.5
Avg	**63**	**93.3**	**63**	**96.4**	**63**	**96.4**	**63.4**	**63.7**	**63.6**	**63.8**



(a)



(b)

Figure 13: Integrated Model Resource Utilization Results (a) Inter-Resource Comparison (b) Intra-Resource Comparison

6.3 Complexity of the Algorithm

In this section, we compare the algorithm complexity of our conflict resolution method to that of [19]. It is shown in figure 14 that our method with O(1) performs better than that of Lui et al with $O(n^2)$. Although the overall complexity of our proposed Smart Claims Processing System is $O(n^2)$, it is relatively better as it is the complexity of the overall system.

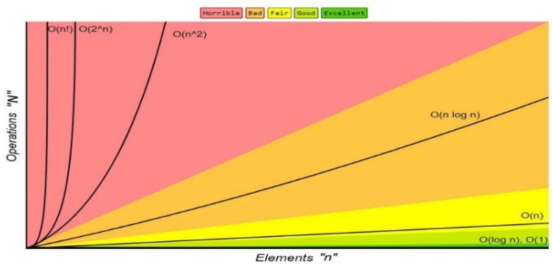


Figure 14: Algorithm Complexity

7. Conclusion

The main objective of this work was to model the flow of data within the smart claim processing system. The relevance of the work hinges on the palpable solution provided to solve the concerns of fraud and corrupt operation in the NHIS claims processing lifecycle. In this work, data are received from three different participants – provider, patient, and director. These participants are modelled with three different input subsystems which fire tokens into the main claims processing subsystem. The reachability of the integrated system was extremely huge to generate therefore, the main system was modelled according to its components or subsystems and the reachability graph was presented. Similarly, the delivery rates of all the tokens were modelled as a crucial aspect of the system. Consequently, Eqs. (2) and (3) were produced to detect and prevent conflict cases. It was discovered that in a case that all claims are successfully created, verified, and approved by the various participants, all the claims get processed and stored in the output system. Table 2 shows the simulated results with varying numbers of tokens. With the proposed equations in operation, the exact locations of conflicts are identified and solved with ease and swiftness. Descriptively, conflicts can occur at transitions t_{10} and t_{12} and places p_{19}, and p_{22}. This happens when the number of claims submitted by the provider is more than the verifications done by the patients and/or the verifications are not equal to the approvals done by the director. The conflict case in the system is not fatal but easily resolvable by verifying or approving the unverified or unapproved claim(s). The system utilized the resources in the system interestingly. Simply put, the higher the number of claims, the lower the rate of resource utilization. Table 3 and figure 13 show the number of claims and their rates of resource utilization. The interesting resource utilization results are expected as the tokens from the various participants are threshed before being submitted into the main claims processing system. Consequently, the higher the number of tokens and the higher the resources in the claims processing system get utilized. It must be emphasized that disparity in the results poses no operational threats to the system. It has been shown that a Petri net model can be used to simulate the NHIS claims processing. The results obtained in this work are very promising.

8. Recommendation and Future Work

This work has propounded an easily implementable solution to solve the tendency of fraud in the NHIS claims processing and to reduce the cost of operations associated with the process. The following recommendations are made based on this work – (i) Claims submitted must be swiftly verified and approved to ensure successful processing and prevent conflicts (ii) claims should be submitted in batches with minimal files to be processed. This ensures the maximum usage of all resources in the system. Regarding the resource utilization results obtained in figure 13, we found that the data follows a Weibull distribution. Consequently, more experiments will be conducted and the Weibull analysis will be used to extensively understand the behaviour of the system beyond the scope of this work. Again, the models will be enhanced further using Stochastic, Stochastic Timed, and other transitions to make available a comprehensive, adaptable, and resource balancing system to operate well in numerous standards.

References

[1] Dixon, J., Tenkorang, E. Y., and Luginaah, I.: *Ghana's National Health Insurance Scheme: a national level investigation of members' perceptions of service provision*. BMC International Health and Human Rights, 13(1), 1-10 (2013).

[2] Preker, A. S., Lindner, M. E., Chernichovsky, D., and Schellekens, O. P.: *Scaling up affordable health insurance: staying the course*. World Bank Publications, (Eds.) p 24 (2013)..

[3] Dake, F. A.: *Examining equity in health insurance coverage: an analysis of Ghana's National Health Insurance Scheme*. 10 (2018).

[4] United Nations.: *Full Report of the Open Working Group of the General on Sustainable Development Goals. Document number: A/68/970*. 2014. http://undocs.org/A/68/970. Accessed 2 November 2021.

[5] The World Health Organization.: *Universal Health Coverage. 2021*. https://www.who.int/health-topics/universal-health-coverage#tab=tab_1 accessed 2 November 2021.

[6] Alhassan, R. K., Nketiah-Amponsah, E., and Arhinful, D. K.: *A review of the National Health Insurance Scheme in Ghana: what are the sustainability threats and prospects?* PloS one, 11(11) (2016), e0165151.

[7] Abiiro, G. A. and McIntyre, D.: *Universal financial protection through National Health Insurance: a stakeholder analysis of the proposed one-time premium payment policy in Ghana.* Health policy and planning, 28(3), 263-278 (2013).

[8] Pizer, W. A., and Sexton, S.: *The distributional impacts of energy taxes*. Review of Environmental Economics and Policy, volume 13, issue 1, pp. 104–123 (2020). doi: 10.1093/reep/rey021

[9] Williams III, R. C., Gordon, H., Burtraw, D., Carbone, J. C., and Morgenstern, R. D.: *The initial incidence of a carbon tax across income groups*. National Tax Journal, *68*(1), 195-213 (2015).

[10] Wang, H., Otoo, N., and Dsane-Selby, L.: *Ghana National Health Insurance Scheme: improving financial sustainability based on expenditure review*. The World Bank (2017)..

[11] Du, S., Wu, P., Wu, G., Yao, C., and Zhang, L.: *The collaborative system workflow management of industrial design based on hierarchical colored petri-net.* IEEE Access, *6*, 27383-27391 (2018).

[12] Teixeira, M., Lima, R., Oliveira, C., and Maciel, P.: *Performance evaluation of service-oriented architecture through stochastic petri nets*. In 2009 IEEE International Conference on Systems, Man and Cybernetics, pp. 2831-2836 (2009). IEEE.

[13] Kaid, H., Al-Ahmari, A., Nasr, E. A., Al-Shayea, A., Kamrani, A. K., Noman, M. A., and Mahmoud, H. A.: *Petri net model based on neural network for deadlock control and fault detection and treatment in automated manufacturing systems*. IEEE Access, 8, 103219-103235(2020).

[14] Hsieh, F. S.: *A hybrid and scalable multi-agent approach for patient scheduling based on Petri net models*. Applied Intelligence, *47*(4), 1068-1086 (2017).

[15] Santana, J. A. D., Orozco, J. L., Lantigua, D. F., Furka, D., Furka, S., and Cruz, A. G.: *Using integrated Bayesian-Petri Net method for individual impact assessment of domino effect accidents.* Journal of Cleaner Production, 294, 126236 (2021).

[16] Shailesh, T., Nayak, A., and Prasad, D.: *An UML based performance evaluation of real-time systems using timed Petri net*. Computers, 9(4), 94 (2020).

[17] Babu, K. R., and Samuel, P.: *Petri net model for resource scheduling with auto scaling in elastic cloud*. International Journal of Networking and Virtual Organisations, 22(4), 462-477 (2020).

[18] Li, X. Y., Liu, Y., Lin, Y. H., Xiao, L. H., Zio, E., and Kang, R.: *A generalized petri net-based modeling framework for service reliability evaluation and management of cloud data centers*. Reliability Engineering & System Safety, 207, 107381 (2021).

[19] Liu, C., Zeng, Q., Duan, H., Wang, L., Tan, J., Ren, C., and Yu, W.: *Petri net based data-flow error detection and correction strategy for business processes.* IEEE Access, *8*, 43265-43276 (2020)..

[20] Ling, S., and Schmidt, H.: *Time Petri nets for workflow modelling and analysis*. In Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0 (Vol. 4, pp. 3039-3044) (2000). IEEE.

[21]Li, J., Fan, Y., and Zhou, M.: *Timing constraint workflow nets for workflow analysis.* IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 33(2), 179-193 (2003).