International Journal of Advanced Engineering and Technology
www.allengineeringjournal.com
E-ISSN: 2456-7655
Received: 04-09-2026, Accepted: 04-10-2026, Published: 19-10-2026
Volume 10, Issue 4, 2026, Page No. 13-20

Development of an executable model for an inbound call centre using Hierarchical Timed Coloured Petri Nets

Akinlolu O Afolabi[1], Azeez O Ganiyu[2], Rafiu A Ganiyu[2], Elijah O Omidiora[2]
[1] Department of Cyber Security, LAUTECH Ogbomoso, Oyo State, Nigeria
[2] Department of Computer Engineering, LAUTECH Ogbomoso, Oyo State, Nigeria

Abstract
Inbound call centres are centralized offices which are operated by modern organizations to administer incoming product supports or information inquiries from consumers. In this paper, an improved model for an inbound call centre was developed using Hierarchical Timed Coloured Petri Nets (HTCPN). The inbound call section of Federal Road Safety Corps (FRSC) in Abuja, Nigeria was used as a case study in developing the model. The developed HTCPN model is made up of six (6) modules conceptualizing call distribution processes taking place in the centre, these areas; Call Arrival, Processes, Hang-up, Major Call, Minor Call and Enquiry. These modules modelled the dynamics of call classification, queuing and termination under overload conditions. By combining places, transitions, arcs and timed tokens, the HTCPN framework provides a structured, hierarchical representation of call flow processes. The developed HTCPN model can be easily modified through it associated modules to suit any future modification in the considered process or other related ones.

Keywords: Call centre, model, hierarchical timed coloured petri net, place and transition

Introduction

Call Centres are locations where calls are placed, or received, in high Call centres are locations where calls are placed, or received, in high volume for the purpose of sales, marketing, customer service, telemarketing, technical support or other specialized business activity (Amin *et al.*, 2021) [5]. A call centre is a set of resources such as communication equipment, employees and computers which enable the delivery of services via the telephone. Call centres have become an integral part of doing business as many companies use call centres for customer interactions such as selling products and services, providing product support and resolving billing issues (Chatterjee, *et al*.,2021) [8]. The three main components of any call centre are the customers (or calls), the technologies employed and the customer service representatives. Customers are typically categorized according to the flow direction of the call as either inbound call (or incoming call) or outbound call (or outgoing call). Call centres are categorized accordingly based on the calls they process, that is, as inbound call centres, outbound call centres, or mixed call centres.

Inbound calls are those initiated by customers to the centre (Koole, 2023) [13]. Call centres are used by many firms that operate in very different sectors, such as mail-order catalogue organizations, telemarketing companies, computer product help desks, banks, hotels and any large enterprises that use telephone as a medium of attending to customer services and products. In addition to answering incoming phone calls, reply e-mail messages and conduct marketing surveys such as questionnaire. Centre data is a vital resource for enhancing service accessibility and quality, essential for achieving the Millennium Development Goals (MDGs) in Nigeria. By effectively managing customer inquiries and complains, organizations can improve access to essential services, contributing to poverty alleviation and overall well-being of the society which is the core MDG objectives (Prahas *et al.*, 2015) [15].

High volumes of complains may highlight service quality issues that require urgent attention to meet targets related to health, education, and gender equality. Furthermore, analyzing this data, will aid decision-making, enabling better resource allocation and targeted initiatives. By effectively addressing customer requests, organizations foster community engagement and trust, essential for sustainable development, ultimately aligning their efforts with the broader vision of the MDGs and enhancing the quality of life for individuals in Nigeria. (Prahas *et al,*. 2015) [15]. In call centres, the cost of providing trained agents accounts for over 50% of total operations cost (Bayo *et al*, 2019) [7]. So it is crucial to determine correct strategies for staffing and workforce scheduling that directly affects the profitability and the efficiency. Call centres need for improved performance and the need for powerful decision-making tools to visualize, analyze and enhance call-centre business processes, the best available tool today to perform these functions are simulation tools (Afolabi *et al*., 2020) [3].

A real system like call centre system requires the use of modelling approach in studying the various operations in the subsystem effectively due to the complexity it entails. Many modelling languages have been suggested and used for system development, but Petri Nets has proven to be a strong modelling language for different types of discrete systems, due to the fact that its formalism offers a clear means to present simulation and control logic.

Colored Petri Nets is a discrete-event modelling language that combines classical Petri nets with the functional programming language (Ganiyu *et al*., 2014). Coloured Petri Nets (CPN) is an example of high-level Petri Nets which combines the strength of Petri Nets with the strength of functional programming language (Jensen *et al*., 2007). The inclusion of time in a CPN model resulted in a model called Time Coloured Petri nets (TCPN) model (Akingbade, 2015) [6].

A TCPN which supports modularity (hierarchy) is named Hierarchical Timed Coloured Petri nets (HTCPN), a

formalism used in modelling complex real systems which represent a system in a structured manner using substitution transitions which can also be modelled into details for the effective representation of the actual system (Adigun, 2022) [1]. Structurally, in HTCPN model, top page will represent the highest hierarchical level which provides the overview of the entire call flow processes while sub-modules will be represented at the lower level using double - bordered rectangles (substitution transitions) depicting other crucial sections of the call centre.

Research Methodology

1. The Modelling Approach

In this paper, a Hierarchical Timed Coloured Petri Net (HTCPN) formalism defined was utilized to develop an executable model for an inbound call centre. According to Odeniyi *et al*. (2021) [14], the HTCPN is defined as a nine-tuple = (S, SN, SA, PN, PT, PA, FS, FT, PP) where:

1. S is a finite set of pages;
   a. s is a non-hierarchical coloured Petri Nets.
   b. b. s1 ≠ s2 ⟹ (Ps1 ∪ Ts1 ∪ As1) ∩ (Ps2 ∪ Ts2 ∪ As2) = ϕ
2. SN ⊆ T is a set of substitution nodes.
3. SA is a page assignment function. No page is a subpage of itself.
4. PN ⊆ P is a set of port nodes.
5. PT is a port type function. PT: PN → {in, out, i/ o, general}.
6. PA is a port assignment function. It is defined from SN into binary relations:
   a. $\forall t \epsilon S$: $PA(t) \subseteq X(t) x PN$SA(t)
   b. b. $\forall t \epsilon SN$, $\forall (p1, p2) \epsilon PA(t)$: $PT(p2) \neq general \Rightarrow ST(p1,t) = PT(p2)$
   c. c. $\forall t \epsilon SN$, $\forall (p1, p2) \epsilon PA(t)$: $C(p1) = C(p2) \wedge I(p1) = I(p2)$
7. FS ⊆ PS is a set of fusion sets
   $\forall fs \epsilon FS, \forall (p1, p2) \epsilon fs$: $C(p2) = C(p2) \wedge I(p1) = I(p2)$
8. FT is a fusion type function. FT: FS → {global, page, instance}
9. PP ∈ SMS is a multi-set of prime pages.
10. R is a set of time values, also called time stamps
11. r 0 is an element of R called the start time.

2. Description of the case study

In developing a Hierarchical Timed Coloured Petri Nets (HTCPN) model for an inbound call centre, the call centre section of Federal Road Safety Corps (FRSC) Number 4 Maputo Street Zone 3, Wuse, Abuja, Nigeria was used. The organization's call centre section handled the incoming calls from different parts of the country. FRSC call centre is a centralized centre that handles inbound and outbound calls from public on 122 emergency (Toll-free) numbers. The centre has forty two (42) agents which operate twenty four hours services (24 hours), it operates two shifts: day (08:00hours -18:00hours) and night shift (18:00 hours - 8:00 hours) with twenty one (21) agents per shift. The calls are received based on automatic call distribution (ACD) where incoming client calls are automatically received and placed on queue to be distributed or routed to an available agent, while waiting, music will be listening to until the representative of the servicing department becomes available and the caller hangs up after the completion of the call. The delay associated with music will not be factored into the calls processing time. Other parameters that will not be included in the model are the agents and guests' personal attitudes which might lead to call abandonment, queue abandonment, as well as outbound calls. Figures 2.1 depict process under study.

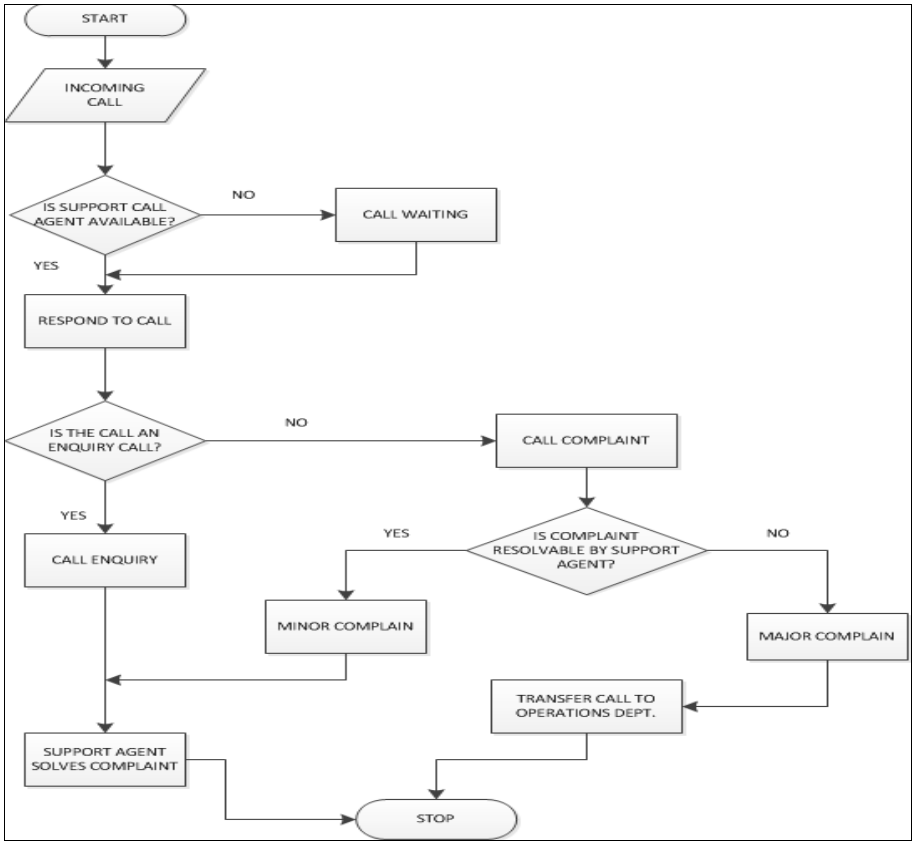


Fig 1: Flowchart of the Inbound Call Centre of FRSC Call Centre Abuja.

3. Data collection

Data collection is a vital stage because the results and findings of a simulation study are as important as the information inputted. The data used in developing the HTCPN model for an inbound call centre distribution process were obtained from the call centre section of Federal Road Safety Corps (FRSC) via printout. The data was used to generate input parameters for the developed HTCPN model and simulate the model as well. Four weeks data were collected which equivalent to a month were printed out. The data include the following information:

1. The total number of incoming calls to the call centre
2. The total number agents in the inbound section of the call centre
3. The inter arrival time of incoming calls

4. Development of the model for an inbound call centre

The model developed consisting of six(6) modules (sub-models) namely call Arrival module, Processes module, hang-up Module, Major call Module, Minor Module and Enquiry Module. Each module comprises places, transitions, arcs, arc inscriptions, transition guards and Places (represented as circle or eclipse) represent the state or condition that serves as a place holder for the entities in the system. Tokens are associated with places and used to indicate whether certain conditions have been met before firing can occur. Transition (represented as rectangles) abstracts an action of the system (events that may occur in the system) that may cause a change in the calls flow processes under consideration, and it is normally represented by a rectangular or bar shape. An arc can only connect a place to a transition or vice versa, the arrows are called arcs. When a transition occurs, they are determined using the arc expressions which are the textual inscriptions positioned next to the individual arcs. A transition and a place can also be connected by double-headed arcs. A double headed arc is shorthand for two directed arcs in opposite directions, between a place and transition which both have the same arc expression. This implies that the place is both an input place and an output place for the transition.

The arc expressions are written in the Coloured Petri Nets Modelling Language (CPN ML) and are built from typed variables, constants, operators and functions. The declaration of the colour- sets, variables, constant, tokens, arc, transition guards and arc inscriptions are contained in the index part of the screen. Places were used to model the states of call in the call centre call waiting, Major call and minor call. Tokens were used to denote resources that are dynamic, such as agents. Transitions were used to represent varying activities such as type of call, call arrival processes. Directed arc were used to illustrate how the state of call changes when the transition is enabled or fired. Enabling and firing of transitions denotes when the conditions required are met and activities are ready to take place (for example, arrival of call). Duration of each activity was modelled using time stamp (t). Coloured formalism represents different attributes of data (call) without the use of standard black tokens. Using HTCPN formalism, it means the developed model was structured in hierarchical level, according to the flow processes of calls across the call centre.

At the highest level, the structure of the HTCPN model consist of a top page (call flow processes) and six sub-pages (modules): call Arrival module, Processes module, hang-up Module, Major call Module, Minor Module and Enquiry Module. The top page of the developed model represents the highest hierarchical level and provides an overview of the entire flow processes. Each module was represented in a top page by a special transition called substitution transition (rectangle with double bordered lines). In the top page of the developed HTCPN model, there were socket places and one output place

The socket place represents the interim states of calls in the inbound call centre's flow processes while the output place represents the state of completed flow processes. Also, the top page consists of two (2) places. The places (Arrival and incoming call) are the conditions that hold for the states of the HTCPN model. Places Arrival holds the number of incoming call (i.e. customer) to be attended to for either request, enquiry or complain while incoming calls is the state of queues in the model before further action will be taken. Major Module takes care of major complains, Minor Module is for minor complains and Enquiry Module for enquiry. The top page of the developed model consists of Six (6) transitions modules out of which one (1) is the hang up module and 3 (substitution) transitions, that are rectangles with double border lines. All these, describing the events or action going on in all the call changes states form one place to another. Tokens were used to depict whether certain conditions have been met which may require firing of transition. Token on place new token indicates when a new call arrives at the arrival call. The colour set for the place is a timed colour set and the time stamp is attached to the token on the place to determine when transition for the next call can occur. The execution of the next call would create and put the call in its output place.

Table 1: Description of Major Places in HTCPN model

Place	Description
Next Call	Model entry of new incoming call
Call Waiting	Indicates call queue
Agent	Models the agent support
Busy	Indicates the support agent is busy
Major call	Indicates a complain as major
Call Type	Indicates if the complaint as minor or major
Type of Call	Indicates if the call is an enquiry
Hang up Call	Indicates exit of a customer call

Table 2: Description of Major Transition in HTCPN model

Transition	Description
Call Arrival	Execution of this transition models the arrival of a call
Call Type	Execution of this transition models the type of incoming call
Incoming Call	Execution of this transition models the start of a new call
Support Call Response	Execution of this transition models the response of the support agent to the call

Enquiry	Execution of this transition models the enquiry of a customer
Complain	Execution of this transition models the customer complaint
Minor Call (Complain)	Execution of this transition models a minor complain to be attended to by the support agent
Major Call (complain)	Execution of this transition models a major complain to be attended to by support agent

The inscriptions 1'( )@0.0 on the arc from the place newtoken to the transition arrival was used to generate the addition of new call to the queue for checking the arrival of new call (if any). Also, time stamps were denoted by notation @0+++ which was added to the arc expression to represent the time it took a process to complete. For instance, ( )@++((uniform(0.0,1.0))) was attached which was used to determine the rate at which call arrived at the call arriving module. By combining places, transitions, arcs, and timed tokens, the Top Page of the HTCPN model accurately simulates the behaviour of inbound calls in a call centre environment, supporting performance analysis and resource allocation studies

4.1 Development of a Call Arrival Processes Module

Call Arrival Processes Module was designed to model the generation and timing of incoming calls into the system. In this module, the socket place newtoken was used to initialise the process with a token 1( )@0.0`, which represents the starting point of the simulation at time zero. This ensures that the system begins with a defined state from which calls can be generated. The place Arrival models the entry of calls into the system. Tokens here are generated with a time stamp ( )@++((uniform(0.0, 1.0))), meaning that calls arrive at random intervals between 0 and 1 time units. This stochastic timing reflects the unpredictable nature of real call arrivals. The transition Next further controls arrivals by checking the condition [[time( ) <= 180.0]], ensuring that calls are only generated within the simulation horizon of 180 time units. Each new token created carries its own arrival time, maintaining the flow of calls across the simulation period.

4.2 Development of Call Processes Module

The Call Processes Module was designed to model how incoming calls are classified and routed into different service categories. In modelling this process, the socket place incomingcalls was used to represent all calls entering the system. Tokens in this place are defined as tuples (c, ar, ct), where c denotes the type of call, ar represents the arrival time, and ct is an intermediate variable used for calculating the timing of the call. This ensures that each call arriving into the system carries its own attributes for identification and timing. From the incomingcalls place, three transitions were defined to represent the routing of calls into different service modules. The transition Major Module directs calls classified as major issues into the place associated with the Major Page. The transition Equiry Module (linked to the Complain Page) handles calls that are enquiries or complaints, ensuring they are processed separately. The transition Minor Module routes calls of lesser importance into the Minor Page. Each transition is connected by arcs that carry tokens forward, ensuring that calls are distributed according to their type. The arc expressions attached to these transitions ensure that calls are routed correctly based on their classification. For example, tokens (c, ar, ct) are matched against conditions that determine whether a call is major, enquiry, or minor. The transitions fire only when the right tokens are available, thereby enforcing proper categorisation. This mechanism ensures that calls are not misrouted and that each type of call is handled by the appropriate service module.Finally, the flow from incomingcalls → Major Module / Equiry Module / Minor Module provides a realistic representation of call traffic in an inbound call centre. Calls are generated, classified according to their type, and then routed into the correct service page for handling. By combining places, transitions, arcs, and timed tokens, the Call Processes Module ensures that the HTCPN model accurately simulates the behaviour of inbound calls, supporting performance analysis and resource allocation across different service categories.

4.3 Development of Hang up Processes Module

The Hang up Processes Sub Module was created to model the behaviour of calls that cannot be admitted into the queues due to system constraints. In this design, the socket place incoming call represents all calls entering the system. Tokens here are defined as tuples (c, ar, ct), where c is the call type, ar is the arrival time, and ct is an intermediate variable used for calculating timing. These tokens flow into the sub module, where conditions are checked to determine whether the call can be admitted into a queue or must be terminated. The transition hangup governs the termination of calls. The arc expression [length(qs)+length(qe)+length(qc) >= 15] attached to this transition specifies that when the combined length of the queues, major (qm), enquiry (qe), and Minor (mc) reaches or exceeds 15, any new incoming call is disconnected. This condition ensures that the system respects its maximum capacity and prevents overload. The marking 1'[] indicates that the hang up transition consumes one token when the condition is met, effectively removing the call from the system. The places maqueue, minqueue, and enqqueue represent the different queues available for calls. Tokens are directed into these queues when capacity allows, with variables qm, qmi, and qe representing the respective queue states. The transitions labelled Fusion 1, Fusion 2, and Fusion 3 manage the distribution of calls into these queues, ensuring that calls are routed correctly according to their type. When the queues are not full, calls are admitted when the queues are saturated, the hang up transition fires. Finally, the flow from incomingcall to hangup / majqueue / minsqueue / enqqueue provides a realistic representation of call centre behaviour. Calls are either admitted into the appropriate queue or disconnected when capacity limits are reached. By combining places, transitions, arcs, and conditional expressions, the Hang up Processes Sub Module ensures that the HTCPN model accurately simulates call termination under overload conditions, supporting performance analysis and resource management in the inbound call centre.

4.4 Development of the Major Call Sub Module

The Major Call Sub Module was developed to model the handling of calls classified as major within the inbound call centre. In this design, the socket place incomingcall represents all calls entering the system. Tokens here are defined as tuples (call, ar, ct), where call denotes the call

object, ar is the arrival time, and ct is an intermediate variable used for calculating timing. These tokens are checked against conditions to determine whether they should be admitted into the major call queue or rejected.
The transition Accept/Enqueue Major governs the admission of major calls into the queue. The arc expression [(length(qs) < 10) andalso (#calltype(call) = Major)] ensures that only calls classified as major are admitted, and only if the length of the majorc all queue (qs) is less than 10. When this condition is satisfied, the token is appended to the queue using the expression qs^^ [call], which adds the call to the existing list of queued calls. This mechanism ensures that capacity limits are respected while allowing major calls to be processed.The transition joinqueue (Fusion 1) models the actual entry of the call into the major queue. Tokens are updated with the expression call::qs, which places the incoming call at the head of the queue. Once admitted, the transition getnext retrieves the next call for processing, represented by (k, call)@++proctime, where k is the resource (agent) assigned and proctime is the scheduled processing time. The transition process then models the handling of the call by the agent, with the tuple (k, call) representing the agent–call pairing. The transition completed finalises the service of the major call. The code segment attached to this transition calculates the processing time using val proctime = (uniform(4.0, 10.0)) / (Real.fromInt(#2 k)), which distributes service times randomly between 4 and 10 units, adjusted by the number of agents available. The call is then moved into the place Done, representing successfully completed calls. If the queue length condition is not satisfied, the transition reject Major fires, governed by the expression [(length(qs)>= 10) andalso (#calltype(call) = Major)], which disconnects the call when the queue is full.

4.5 Development of Minor Call Sub Module

The Minor Call Sub Module was developed to model the handling of calls classified as minor within the inbound call centre. In this design, the socket place incomingcall represents all calls entering the system. Tokens here are defined as tuples (call, ar, ct), where call denotes the call object, ar is the arrival time, and ct is an intermediate variable used for calculating timing. These tokens are checked against conditions to determine whether they should be admitted into the minor call queue or rejected.The transition Accept/Enqueue Minor governs the admission of minor calls into the queue. The arc expression [[length(qc) < 5] and also (#calltype(call) = Minor)] ensures that only calls classified as minor are admitted, and only if the length of the minor call queue (qc) is less than 5. When this condition is satisfied, the token is appended to the queue using the expression qc^^{call}, which adds the call to the existing list of queued calls. This mechanism ensures that capacity limits are respected while allowing minor calls to be processed.The transition joinqueue (Fusion 5) models the actual entry of the call into the minor queue. Tokens are updated with the expression call::qc, which places the incoming call at the head of the queue. Once admitted, the transition getnext retrieves the next call for processing, represented by (k, call)@++proctime, where k is the resource (agent) assigned and proctime is the scheduled processing time. The transition process then models the handling of the call by the agent, with the tuple (k, call) representing the agent call pairing.The transition completed finalises the service of the minor call. The code segment attached to this transition calculates the processing time using a uniform distribution, ensuring service times vary realistically. The call is then moved into the place Done, representing successfully completed calls. If the queue length condition is not satisfied, the transition Reject Minor fires, governed by the expression [[length(qc) >= 5] and also (#calltype(call) = Minor)], which disconnects the call when the minor queue is full. The place agents defines the available resources, marked as 1'("Record Officers", 4)@0.0, ensuring that four record agent are available to handle calls.

4.6 Development of Enquiry Process Sub Module

The Enquiry Process Sub Module was developed to model the handling of calls classified as complaints or enquiries within the inbound call centre. In this design, the socket place incoming call represents all calls entering the system. Tokens here are defined as tuples (call, ar, ct), where call denotes the call object, ar is the arrival time, and ct is an intermediate variable used for calculating timing. These tokens are checked against conditions to determine whether they should be admitted into the enquiry queue or rejected. The transition Accept/Enqueue Complain governs the admission of enquiry calls into the queue. The arc expression [(length(qe) < 5) and also (#calltype(call) = Complain)] ensures that only calls classified as complaints are admitted, and only if the length of the enquiry queue (qe) is less than 5. When this condition is satisfied, the token is appended to the queue using the expression qe^^[call], which adds the call to the existing list of queued calls. This mechanism ensures that capacity limits are respected while allowing enquiry calls to be processed The transition joinqueue (Fusion 2) models the actual entry of the call into the enquiry queue. Tokens are updated with the expression call::qe, which places the incoming call at the head of the queue. Once admitted, the transition getnext retrieves the next call for processing, represented by (k, call)@++proctime, where k is the resource (agent) assigned and proctime is the scheduled processing time. The transition process then models the handling of the call by the agent, with the tuple (k, call) representing the agent–call pairing. The transition completed finalises the service of the enquiry call. The code segment attached to this transition calculates the processing time using val proctime = (uniform(4.0, 15.0)) / (Real.fromInt(#2 k)), which distributes service times randomly between 4 and 15 units, adjusted by the number of agents available. The call is then moved into the place Done, representing successfully completed calls. If the queue length condition is not satisfied, the transition Reject Complain fires, governed by the expression [(length(qe)>=5) andalso(#calltype(call) = Complain)], which disconnects the call when the enquiry queue is full. The place agents defines the available resources, marked as 1'("Record Officers", 4)@0.0, ensuring that four record officers (agent) are available to handle calls development of Call Processes Module The place Count models the counting of calls as they arrive. It begins with an initial marking 11, and the integer variable i is incremented (i + 1) each time a call is generated. This mechanism keeps track of the total number of calls that have entered the system. The output expressionoutput(call); action newcall();` ensures that each generated call is recorded and passed forward for further processing. Finally, the place

incomingcall represents calls that have successfully entered the system and are waiting to be routed. The label Calls attached to this place indicates that tokens here correspond to actual call objects ready for classification and handling. The flow from newtoken → Arrival → Next → Count → incomingcall ensures that calls are generated, timed according to a uniform distribution, counted sequentially and then moved into the system for service. This combination of places, transitions, arcs and timed token provides a realistic representation of call traffic within the HTCPN framework

Results

Figure 3.1 shows the developed HTCPN model for inbound call centre of the considered establishment. The developed HTCPN model was structured in hierarchical levels. At the highest hierarchical level, the structure of the HTCPN model consists of a top page (Call centre) and two sub-pages (modules): the Call Arrival processes and the Call processes. The top page of the developed HTCPN model provides an overview of the call processing system in the entire inbound call centre being considered. The Call Arrival Processes module shown in Figure 3.2 modelled the process of arriving calls. The Call Processes module depicted in Figure 3.3 consists of four sub-modules which include Hangup Process, Major Complain, Minor Complain and Enquiry. The Hangup process sub -module shown in Figure 3.4 modelled the process of hanging up calls. The Major Complain sub-module as depicted in Figure 3.5 modelled the process of receiving call .The Minior Complain sub-module as shown in Figure 3.6 modelled the process of receiving call in the call centre under consideration. The Enquiry sub-module as depicted in Figure 3.7 modelled the process of receiving call that it an enquiry.

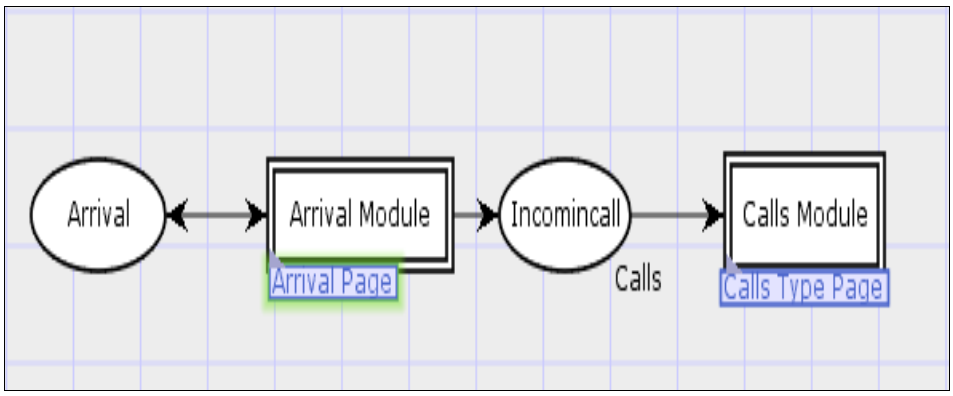


Fig 1: The Developed HTCPN Model (Top Page) of Inbound Call Centre

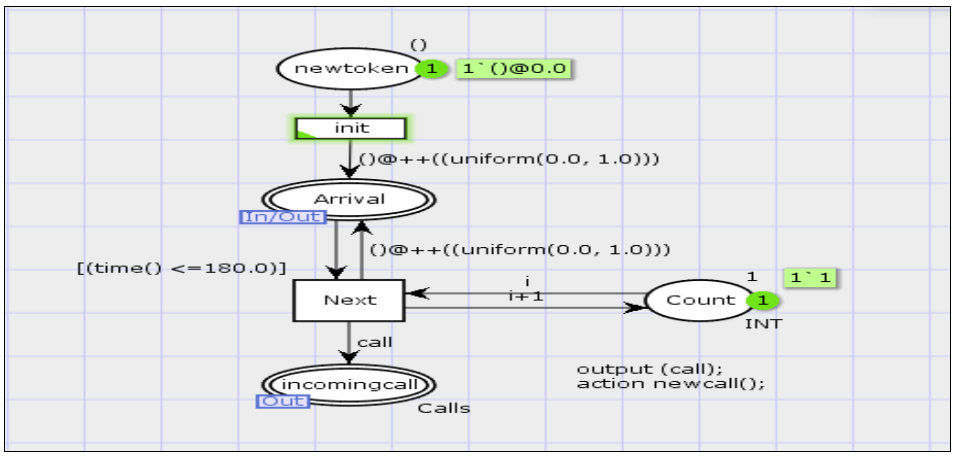


Fig 2: The Call Arrival Processes Module of the Developed HTCPN Model

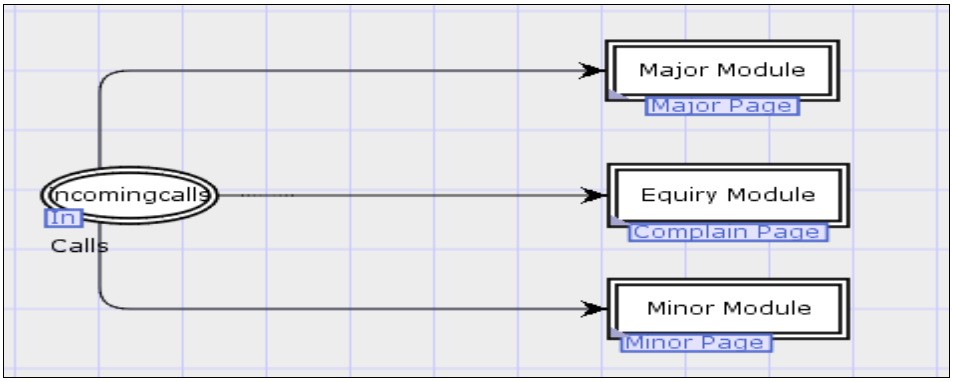


Fig 3: The Call Processes Module of the Developed HTCPN Model

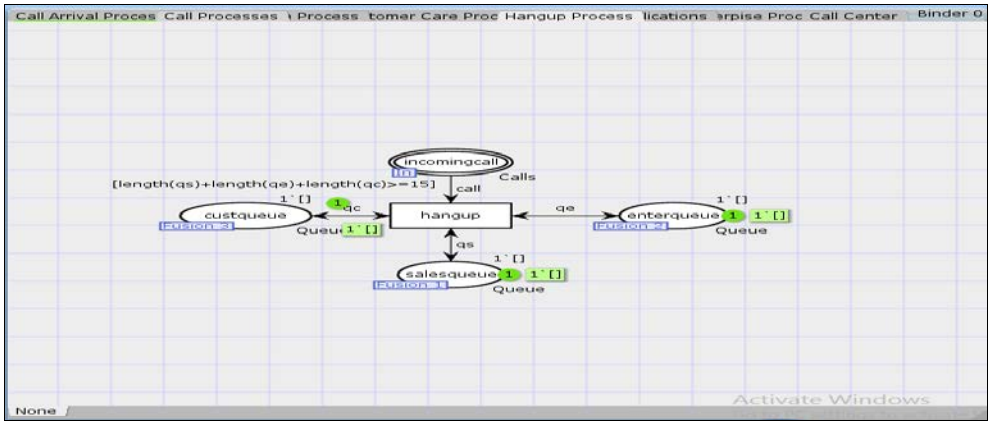


Fig 4: The Hang up Processes Sub- Module of the Developed HTCPN Model

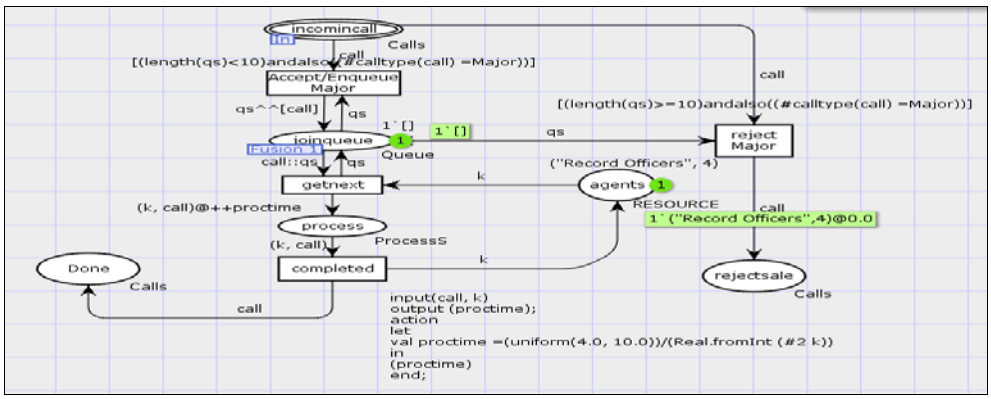


Fig 5: The Major Call Sub- Module of the Developed HTCPN Model.

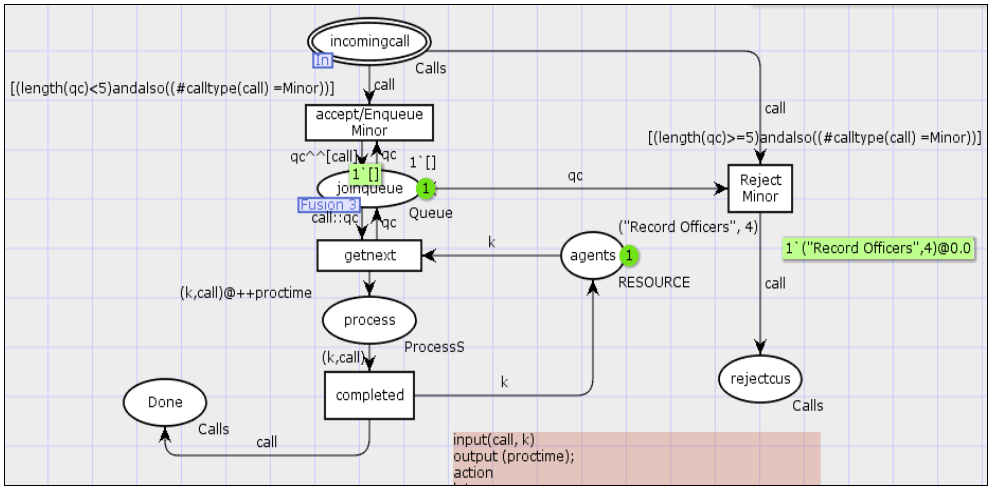


Fig 6: The Minor Call Sub-module of the Developed HTCPN Model

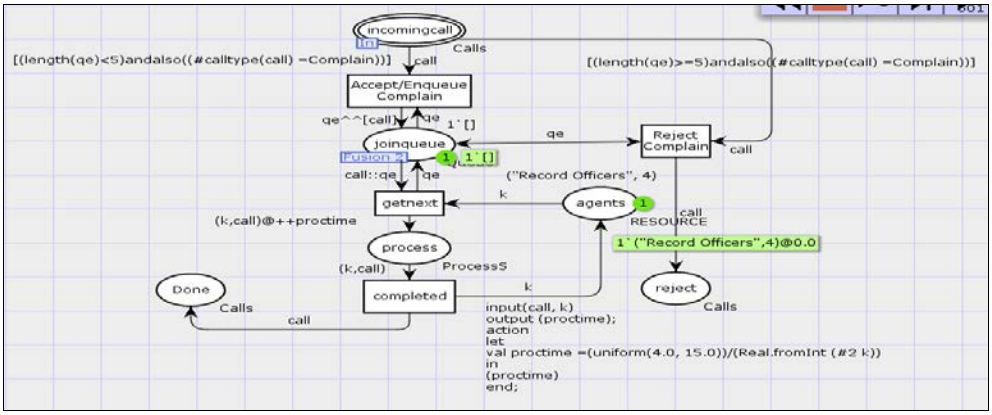


Fig 7: The Enquiry Process sub- Module of the Developed HTCPN Model

Conclusion and Future Work

In this paper, we have been able to develop an executable Hierarchical Timed Coloured Petri Nets (HTCPN) model for an inbound call centre using call centre section of Federal Road Safety Corps (FRSC) Number 4 Maputo Street Zone 3, Wuse, Abuja, Nigeria as a case study. The developed HTCPN model can be easily modified through its associated modules to suit any future modification in the call centre distribution process under consideration. Furthermore, it is recommended that future research may be geared towards validating and analyzing the performance of the developed HTCPN model through simulation based analysis technique.

References

1. Adigun SO. An executable model for student registration system usingtimed coloured Petri nets. FUDMA Journal of Sciences,2022:9(11):Pp3760. https://doi.org/10.33003/fjs-2025-0911-3760
2. Adigun MO, Ojo SO, Emuoyibofarhe OJ, Dehinbo J. E-Health Management: A Partnership and Collaboration Model in Developing Countries. In Proceedings 1st All Africa Technology Diffusion Conference, Johannesburg, 2006, 12-14.
3. Afolabi AO, Ganiyu RA, Arulogun OT, Ganiyu AO. Development of a model for an inbound call centre using Hierarchical Timed Coloured Petri Nets. Asian Journal of Research in Computer Science,2020:5(3):39–51. https://doi.org/10.9734/AJRCOS/2020/v5i330137
4. Alamu F, Emuoyibofarhe OJ, Oke A. Developing a robust multimedia picture archiving and communication system (PACS). Int J Comput Appl,2011:34(4):12-25.
5. Amin A, Alkattan A, Alageel A, Salah M, Almutairi M, Sagor K, *et al*. Onsite versus remote working: The impact on satisfaction, productivity, and performance of medical call center workers. INQUIRY: The Journal of Health Care Organization, Provision, and Financing,2021:58:1–7. https://doi.org/10.1177/00469580211056041 (doi.org in Bing)
6. Akingbade OY, Ganiyu RA, Olabiyisi SO, Badmus TA. Development of a timed coloured Petri nets model for health centre patient care flow processes. International Journal of Engineering and Computer Science,2015:4(1):9954–9961
7. Bayo A, Larraza M. Cost structures in call centre operations: An efficiency analysis. Journal of Service Management,2019:30(4):567–582.
8. Chatterjee S, Rana NP, Tamilmani K, Sharma A. The effect of AI-based CRM on organization performance and competitive advantage: An empirical analysis in the B2B context. Industrial Marketing Management,2021:97:205–219. https://doi.org/10.1016/j.indmarman.2021.07.005 (doi.org in Bing).
9. Ganiyu RA, Adepoju O, Salami R. Timed coloured Petri nets for performance analysis. Simulation Modelling Practice and Theory,2015:55:1–15.
10. Balakrishnan SN, Weil RD. Neurocontrol: A literature survey. Mathematical and Computer Modelling,1996:23(1-2):101-117.
11. Jensen K, Kristensen L, Wells L. Modelling elements of a Petri net. Concurrency and Computation: Practice and Experience,2021:33(12):5679.
12. Bernstein ML, McCreless T, Cote MJ. Five constants of information technology adoption in healthcare. Hospital Topics,2007:85(1):17-25.
13. Koole G, Li S. Workforce planning in call centers: A queueing perspective. Operations Research Perspectives,2023:10:100262.
14. Odeniyi LA, Ganiyu RA, Omidiora EO, Olabiyisi SO, Ganiyu AO. “Development of a Model for Restaurant Food Serving Process using Hierarchical Timed Coloured Petri Nets”, IOSR Journalof Computer Engineering (IOSR-JCE),2021:23(5):01–08.
15. Prahas R, Adeyemi T, Bello K. Call centre data and Millennium Development Goals in Nigeria. Nigeria MDGs End Point Report 2015. Office of the Senior Special Assistant to the President on MDGs (OSSAP MDGs) / UNDP, 2015