https://dx.doi.org/10.4314/ijs.v24i1.11
Ife Journal of Science vol. 24, no. 1 (2022)

DEVELOPMENT OF COMPUTATIONAL MODELS FOR THE PRODUCTION OF FERMENTED AFRICAN LOCUST BEANS USING PETRI NET

Atoyebi, J. O.[1,*], Bello, S. A.[1], Owolarafe, O. K.[2]
[1]Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria
[2]Department of Agricultural and Environmental Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria
Corresponding Author's Email: jeliliatoyebi3124@gmail.com
(Received: 2nd December, 2021; Accepted: 28th March, 2022)

ABSTRACT

Production of *iru*, a fermented condiment from African locust bean is characterized by low productivity and drudgery. The hierarchical timed coloured Petri nets (HTCPN) formalisms were used to design computational models on unit operations of the *iru* production process. The developed models for the traditional *iru* production process (TIPP) and mechanized *iru* production process (MIPP) were simulated using CPN tools varying production scenarios. In this paper, the simulation results revealed that the measured quantities of 16, 32, 48, 80 kg of raw locust bean (*iyere*) gave production times of 4644, 5094, 5559 and 6493 mins and 4540, 4728, 4910 and 5267 mins for TIPP and MIPP, respectively. The production ratios of *iyere*:*iru* (per kg) for the TIPP were observed to be 16:12.8; 32:25.6; 48:38.4 and 80: 64; and for MIPP were 16:12.64; 32:25.28; 48:37.92 and 80:63.2; respectively. The resource (water) usage ratio of TIPP to MIPP (in percentage) was 6.7:10 for the washing operation; 56.67:60 for the long-cooking operation and 23:23 for the short-cooking operation, respectively. The individual dehulling and sieving operations in the TIPP took 1.70 and 11.7, respectively. Meanwhile, combined operations of dehulling and sieving in the MIPP gave 7.00. Conclusively, the mechanized process was found to utilize lesser quantity of water and shorter production time. Hence, more *iru* was obtained from traditional process. The developed computational models could serve as reference models for future process automation, further studying and improving *iru* production.

Keywords: African locust bean, Petri net, Computational model, *Iru* production process

INTRODUCTION

Since the early 1960s, the computer system and its associated software have become accepted in the domains of mechanisation, control and automation of production processes (Webb & Hoben, 1983). Mechanisation implies the use of machines and technology to carry out operations and replacing the traditional methods involving human or animal labour. The aim of the mechanised productions is to increase productivity, in order to meet the needs of the growing population within the timely operations (Owolarafe, *et al*., 2010). In recent time, extensive research has been done on the fuzzy model as a valuable tool to achieve multiple objectives and its diverse applications to modelling and control (Johanen & Babuska, 2003 and Takagi & Sugeno, 1985). Fuzzy algorithms and fuzzy models are considered applicable to a servo motor controller that are controlled by a microprocessor which further require more accurate and fast response compared with other industrial production processes (Li & Lau, 1989). A fuzzy system, turing machine and other similar technologies have played a significant role in modelling and control considering the use of words to replace numbers or symbols for describing real-world problems (Wang & Qiu, 2003). Likewise, control and automation of production processes involve the use of an electronic processing device. As a computer system is an electronics processing device, it serves as a platform to implement software developed for remote monitoring, adjusting operational conditions and controlling of a complex system (Odejobi & Owolarafe, 2004). As part of the acceptance of computer software to the industrial production process, the concept of modelling and simulation has been explored extensively to solve complex problems (Webb &Hoben, 1983).

Modelling is a process of creating symbols or making graphical notations for an abstract representation of a real-life system (Hussein, 2014). A model can be classified as mathematical, physical or computational. Each of these classes of a model could be static or dynamic (An *et al*., 2018). An abstract representation of a real-life system could be modelled with Petri Nets (Hack, 1976) and Snoopy (Heiner *et al*., 2012) tools, amongst others. Simulation is a process of imitating real-life system operations by evaluating the performance of the model developed for the system with different inputs to study the change in system behaviour. According to Hill *et al.* 2001, a computational model is a computer-based model used to formulate, simulate and study the behaviour of complex processes for better understanding and predicting the outcomes with a specific set of input parameters. A computational model has been widely accepted in neuroscience, earth simulator, weather forecasting, molecular protein folding, flight simulator, and neural network models (Dayan, 1994). Simulation of an executable model for industrial productions could give insight into the behavioural changes in the system (Webb &Hoben, 1983).

Recent research has revealed that the global sales of processed food had exceeded more than \$2 trillion (Mahalik 2014). In today's world, food industries are subjected to periodic change, caused by the operations' cost and the increasing demand by the teeming population. This constant change neccessitates the need to provide automated solutions and IT equipment that can enable the industry to become more agile and leaner with fewer hurdles (Mahalik, 2014). Food process automation replaces manual operations with electronics devices such as computer-controlled devices. Process automation is reported to reduce human resources, and material wastage found in manual handling and increases profit by increasing the production rates and productivity, which in turn boosts efficiency and reliability (Jijo & Ramesh Kumar, 2014). As a result, food industries are experiencing a steady degree of automation (Rousu *et al.*, 2003). As High-level Petri nets have been successfully deployed to model chemical processes, portray breweries and scheduling of a sugar milling plant (Hubert *et al.*, 2016). Thus, this study explored the process of production of fermented locust bean in the context of computational modelling.

The fermented African locust bean is a household food condiment consumed daily, though in small quantity but the frequency of consumption necessitates the high demand. It is used to flavour traditional soups and stews in West Africa. This fermented food condiment obtained from African locust bean (*iyere*), shown in Figure 1 is referred to as *iru* by Yorubas (Southern Nigerians), *dawadawa* by Hausas (Northern Nigerians) and *ogiri* by Igbos (Eastern Nigerians) (Owolarafe *et al.*, 2010). *Iru* is also famous as soumbala in Guinea, Burkina Faso, Cote d'Ivoire and Mali (Sarkar *et al.*, 2002). *Iru*, a rich source of protein that contributes a useful amount of lysine and riboflavin for human body system (Campbell- Platt, 1980) and also contains easily digestible calcium (Musa, 1991). The yellowish powder from the dried seed pulp substance is used to make gruel for feeding pigs. The fruit shell is desirable for extracting a substance that helps to harden the natively made residence floors and it is also a prime source of tannin for leather processing (Farayola *et al.*, 2012). Production of this staple food condiment in large quantities is found in Osun, Ekiti, Oyo, Niger, Kwara and Ondo States of Nigeria. This fact may not be unconnected with the climate of that region that favours the plantation of the trees of African locust bean. The tree is commonly found in South-East Asia, Africa and South America.

Figure 1: African locust bean

The production of this fermented locust bean requires low capital in small scale businesses and thus serves as a source of income to an enormous number of West African women (Olaoye, 2016). Production of *iru* after harvesting, depodding and depulping of the bean involves several unit operations. These processes include sorting, washing, long-cooking, dehulling, sieving, short-cooking, draining and fermentation (Akande *et al.*, 2010). Fermentation is the last unit operation of *iru* production. The fermentation operation is an essential food processing technology which solves the problems of food spoilage and food borne diseases world wide (Adeniran *et al.*, 2013). Low-skilled women are usually involved in *iru* production. Presently, *iru* is produced manually by either the pure traditional (purely human) method or the semi-mechanized method using a motorized device (Owolarafe *et al.,* 2011 and 2013). The traditional way follows eight processing stages while the semi-mechanized way consists of seven stages having merged two distinct stages of dehulling and sieving. However, both production processes are characterized by many problems (Akande *et al.*, 2010; Owolarafe *et al.*, 2011 and Farayola *et al.*, 2012). These problems, among others, are:

- Labour-intensive production process;
- The requirement of skilled and experienced labour for the production process;
- Difficulties in achieving consistency of product quality;
- Poor control and monitoring mechanism; and
- Human exposure to processing hazards such as boiling water, excessive heat radiation, fire outbreak, and so on.

This study employs a Petri net-based approach to develop computational models for the two existing (traditional and mechanized) *iru* production processes. This attempt will evaluate the two processes towards future automation of the *iru* production process. This study will give a clear and better understanding of the relationship between raw beans (input) and fermented beans (product), water (resource) usage rate and the production time involved in the entire process.

RELATED WORKS

Petri Net is a typical graphical tool for modelling and carried-out performance analysis of concurrent systems (Girault & Reisig, 1982 and Jensen, 1997). This makes its ideal to model parallel, asynchronous, non-deterministic and distributed processes (Murata, 1989a and 1989b). However, Petri net is insufficient in modelling modern real-time systems found in flexible manufacturing, logistics, production processes, communication and information processing systems (Van der Aalst, 1993). In order to solve these problems, extensions were incorporated into the basic Petri nets (He & Murata, 2005). Coloured Petri Nets (CP Nets) by Jensen, 1981 ; Jensen, 1993 and Jensen, 1997 employed colours for tokens to represent resources in the modelled systems. The Interval Timed Coloured Petri Net (ITCPN) model of Van der Aalst, in 1993 has features for the modelling of the dynamic behaviour of large and complex systems, without losing the possibility of formal analysis. The Timed Coloured Petri Nets reported by Morasca *et al.* in 1991 and the Hierarchical Timed Coloured Petri Net (HTCPN) model by Huang and Chung, 2010 that used explicit quantitative time as a token. Petri nets have been applied in analyzing computer hardware and software (Zuberek, 1980), process scheduling (Van der Aalst & Odijk, 1995), industrial process control (Oberheid & Söffker, 2008), system planning (Bello *et al.*, 2011), as well as food production system (Ganiyu *et al.*, 2019). Zuberek (1980) modelled two simple processor architectures and compared the resulting performance indices. Analysis of a finite labelled directed graph which represents the behaviour of Timed Petri nets developed for the two processors was done using Markov chains technique. Van der Aalst & Odijk, (1995) employed the Interval Timed Coloured Petri nets to model and analyse railway stations in order to evaluate both operation schedule and infrastructure of the station. Kovacs *et al.*, (2005) employed a timed stochastic coloured Petri net (TSCPN) to the air transport system to analyse the effect of the availability of taxiways on the capacity of the timing schedule and runway.

Also, Oberheid & Söffker, in 2008 further employed CPN to simulate arrival planning process in air traffic control, to establish a favourable sequence for aircraft to be led into the runway. Huang & Chung (2010) employed a TCPN model to successfully simulate an urban traffic light control system with five intersections. In 2011, Davidrajuh & Lin also developed a Petri Net model for the main operations of airport traffic with the GPenSIm tool that used to integrate the Petri Nets models with MATLAB toolboxes.

Xin (2015) established excellent insights into food processing using computer software where milk production process modelling was used as a case study. The developed model was simulated using ARENA software. Notably, Petri nets (simulation with CPN tools), Takagi-Sugeno fuzzy control approach (fuzzy logic toolkit for simulation), ARENA (mathematically oriented simulation software), snoopy (a Petri nets simulation software) and unified modelling language (a modelling tool) are useful for an abstract representation of a real-life scenario. Also, different simulation tools are used in various modelling domains. Network simulator-2 and network simulator-3 are notable simulation tools adopted in a wireless and mobile system. Odejobi and Owolarafe (2004) modelled and simulated a fuzzy logic-based process control system for gari fermentation plant. The simulation was done via computer software, MATLAB 6.0, running on Intel 2Ghz environment within the windows 2000 XP. Snoopy and CPN tools are also common simulation tools in the computer intelligence system domain. Petri Nets has comparative advantages over Snoopy in terms of additional features by using hierarchy to model and structure typical large model, using colours to model data and using the time to model durations.

Bello *et al.*, (2011) employed Petri net to model the Adire fabric production system to understand the process for the improved production process. Also, Ganiyu *et al.*, (2011) employed the Timed Coloured Petri Nets (TCPN) to model and simulate a multi-phase traffic light-controlled intersection for the T-type junction with an associated fixed signal timing plan to reduce traffic congestion and ensure improved safety. In 2015, Ganiyu *et al.*, used the TCPN modelling technique to proffer solution to the problem of break-continuity of the patient-health-care flow processes, to improve the flow process and thus save human lives. Olusanya *et al.*, (2017) employed the Hierarchical Coloured Timed Petri Net to model a Point of Sale (POS) cash deposit and Deposit Slip methods in Nigerian banks to allow for improvement in the delivering banking operations. Mandlik & Borkar, (2015) reported that the performance metrics to select and evaluate the best simulation software for food processing include code reusability, modelling flexibility, types of modelling structure (flat v/s hierarchical; nested v/s object-oriented), ease of use, dynamic business graphics, animation, graphic user interface, statistical capabilities, output reports, hardware and software requirements, graphical plots, customer support and documentation.

Petri net has been able to fulfil these attributes as demonstrated by Hussein in 2014 by employing the CPN tool to model a restaurant system, resulting in a reduced complexity and modelling time of the multi-transitions system. In 2016, Hubert, *et al.*, simulated a developed data-driven stochastic model of parts within the process of small-sized and medium-sized breweries to address high energy demand of the cooling system. The Hierarchical Timed Coloured Petri Nets (HTCPN) model by Ganiyu *et al.*, (2019) for a multi-process food (cassava flakes) production system. In this study, the first advantage of adopting the Petri net-based production approach is the precise and clear presentation of *iru* production in terms of events and conditions that leads to the formation of fermented condiment. The second advantage is the powerful modelling capability and mathematical formalism of the modelling technique. Third, the correctness and changes in system behaviours can be easily checked and analyzed (An *et al.*, 2018). Fourth, Petri nets could enable system designer and a system engineer to analyse the activities of a system at various stages of the design (Bello *et al.*, 2011). Fifth, the developed computational models could be redesigned to fit for future modifications through its associated modules for multi-process food manufacturing system (Ganiyu *et al.*, 2019). Lastly, the Petri net-based models are parameterizable in usage during the simulation and could be implemented using a standard functional meta-language (SML) programming language. In this study, the proposed models have comparative advantages over the situation without Petri nets. The Petri net-based approach gives baseline information which can cause better resources utilization at every stage of *iru* production. Especially, for better prediction of water usage at the dehulling and sieving stages using a processing machine. Likewise, production time and output (*iru*) per se could be predicted for any measured quantity of African locust beans. Practically, the developed computational models could serve as reference models for future process automation, further studying and improving *iru* production.

THE *IRU* PRODUCTION PROCESS

Iru Production Process (IPP) consists of several distinct processing stages that contribute to the production of fermented condiment (*iru*) from African locust bean (*iyere*). The noticeable unit operations that are well-defined in the traditional methods: sorting, washing, long-cooking, dehulling, sieving, short-cooking, draining and fermentation. Whereas in the mechanized process (motorised device) reported by Owolarafe *et al.* (2011) shown in Figure 2, the dehulling and sieving operations have been merged. Figure 3(a) and 3(b) illustrates the pure traditional method and the mechanized method respectively.

Figure 2: Snapshot of a motorised locust bean processing machine

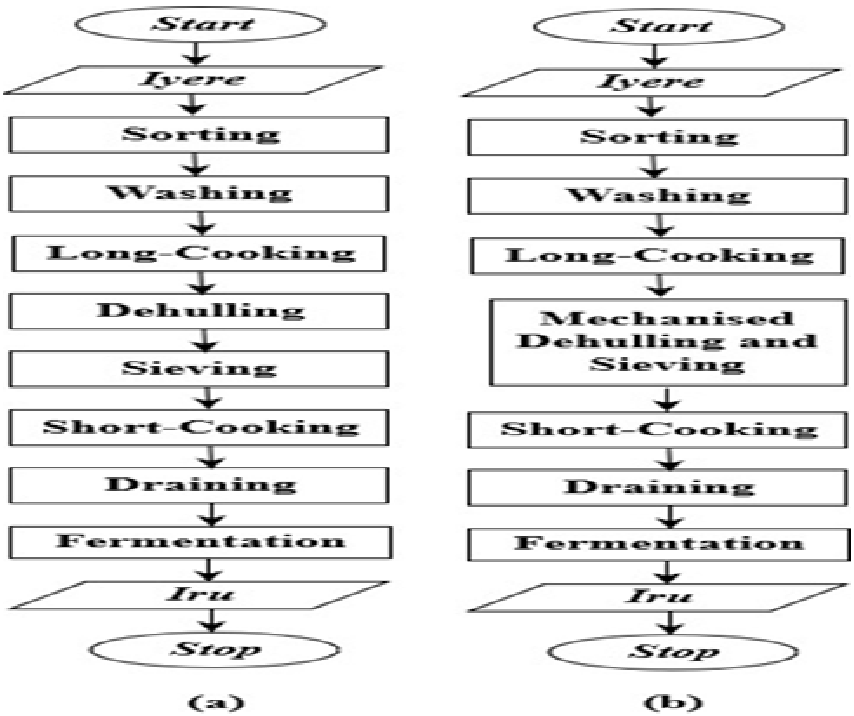


Figure 3: Process Flow Diagrams for the Traditional (a) and Mechanised (b) *iru* production processes,respectively.

The sorting operation is to remove unwanted particles such as woods, pebbles and other dirt from the bean. Next is the washing operation. The washing operation involves the use of clean water to get rid of fine sand that earlier mixed with the bean. The long cooking operation involves boiling of cleaned locust bean in water to soften the seed coat (hull) for its easy removal. This cooking operation takes 1 to 2 days. During the long-cooking process, wood ash is added to the *iyere* to hasten the boiling process. This is followed by the dehulling operation meant to remove the coat from the bean seed. The stage involves the use of pestle to pound the boiled seeds in a clean mortar. Another method is to tread on the cooked beans with either bare foot or rubber shoe by the processors, usually done along water streams. This stage generates a lot of unpleasant smell that takes a long time to evaporate. This stage consumes quite a lot of water. Then, the cleaned pounded seeds (or cleaned treaded seeds) are put in a bowl that contains some water at room temperature and a local sieve for sieving the shaft (coat) from the bean seed. The sieving operation completely removes the testa by rinsing it in water to obtain clean cotyledon. Cleaned dehulled seeds are then subjected to short cooking. The short-cooking operation involves putting clean cotyledons in a cooking pot with adequate water. For about one and half hours at steady heat supply to soften the cotyledons. The soft beans are drained with a local fine sieve and then spread in a light tray for a short time (between 60 and 120 s) to get rid of excess water. Next is the Fermentation stage which can produce either a mashed or grained product. Jute sacs are used to wrap the calabash bowl containing the soft cotyledons to provide the needed warmth for fermentation. This process retains the heat present in the beans to gradually ferment the seed. For a mashed product a softener, an extract from Hibiscus plant is mixed with the drained bean seed at this stage. The fermented bean (as shown in Figure 4) can turn out to be a grained product or mashed product. Each type is used for different types of assorted vegetable soup and stews (Musa, 1991; Akande *et al*., 2010).

Figure 4: Snapshot of fermented African locust bean

METHODOLOGY

System design reveals the detailed structure of the system developed. The design architecture used here involves the use of Petri Nets and the CPN tool. In this research, data were collected through a structured oral interview with five producers using the traditional method and four producers using the mechanized (motorized) method (shown in Figure 2) in Osun State, to elicit knowledge on the process of *iru* production in Nigeria. Each process in *iru* production was then modelled to cater for the different perspectives in various degree of abstraction. Computationally, tokens were used to denote dynamic resources, that is, *iyere* and quantity of water. Places were used to represent conditions or states of *iyere*, while transitions were used to represent activities within production. The design also caters for the abstract representation of a motorised locust bean processing machine using a place named as a "machine", while transitions are depicting beginning and end of each operation performed by the machine within the developed computational model for the MIPP. The machine (Figure 2) performs combined operations of dehulling and sieving. The following assumptions were made: (i) The resources can be represented with the use of the same symbol since the focus was only on the abstract description of the places and transitions, not physical properties and chemical properties of African locust beans, (ii) Heat generated is steady; optimum temperature for a cooking process is around 100 ^{0}C, and optimum temperature for fermentation is between 40 ^{0}C and 70 ^{0}C which have no significant effect on *iru* production, and (iii) Producers follow the entire production process(es) of *iru*. Figure 5 shows several decisions and likely repetition of actions/activities made by the processor(s) during the unit operation of *iru* production. Directed arcs were used to illustrate the changing conditions of *iyere* when transitions were enabled and fired. Enabling and firing of transitions denote when the conditions required were met, and activities were ready to take place or were being performed. In this paper, the computational model was designed using Hierarchical Timed Coloured Petri nets (HTCPN) formalism.

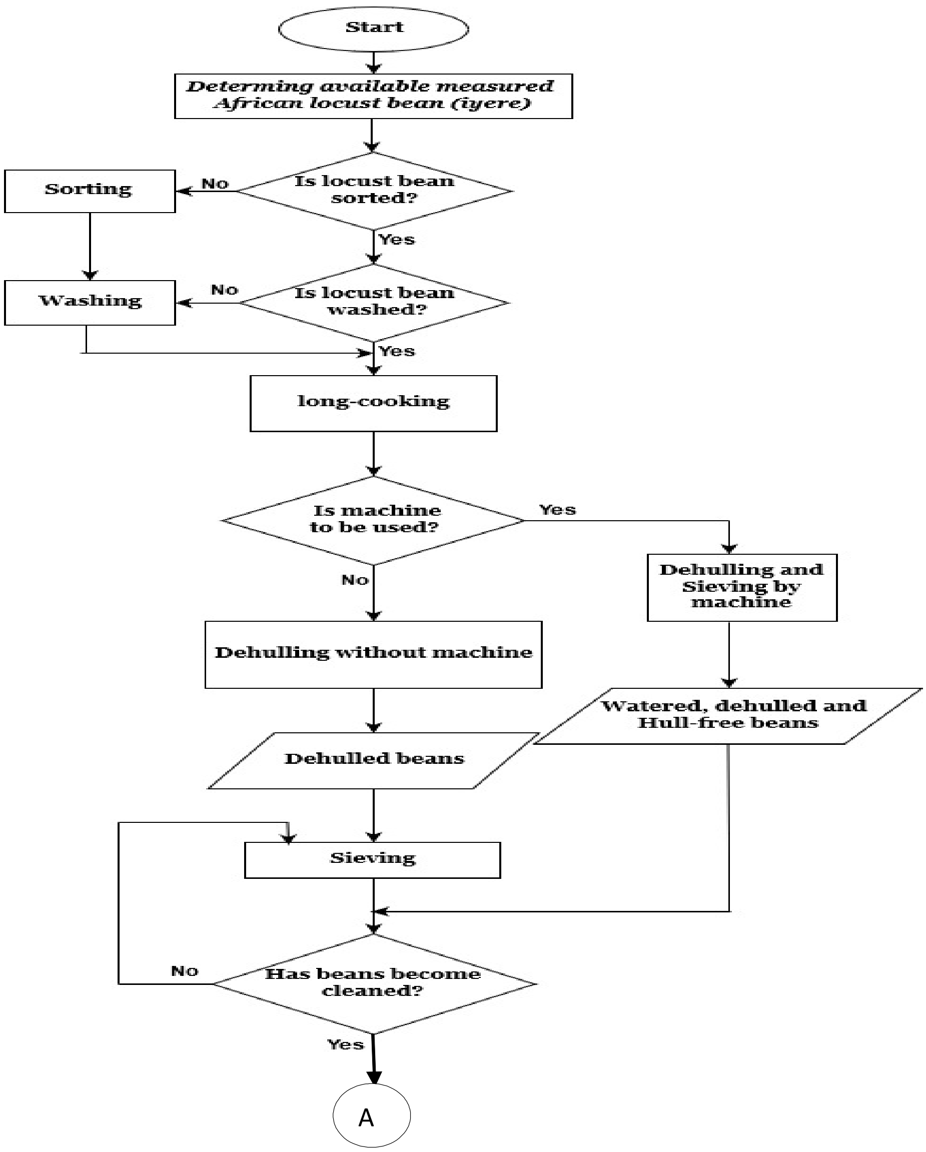
The hierarchical formalism allows the model developed to represent stages of the production processes as submodules. Timed formalism denotes the duration of each process, represented by timestamp. Coloured formalism stands for representation of different attributes of data without the use of the standard black token. Transitions represented activities within *Iru* Production Process. Table 1 has three columns: Activities, Pre-Condition and PostCondition. Activities column referred to the different stages of the production process. The Pre-Condition column specifies the state of the *iyere* before the particular activity, while Post Condition column gives the final state of *iyere* after the particular activity has been carried out on it. In designing the computational models, specific names were given to a number of places and transitions for modelling the state of *iyere* and events (actions) take place prior to the formation of *iru*, respectively.

System Modelling

Coloured Petri Nets (CPN) can describe industrial applications as the graphical representation with a well-defined semantics which allows formal analysis. Each token carries complex information which may describe the entire state of a process (Van der Aalst, 1994). CPN language makes it possible to organise a model as a set of modules, and it includes a time concept for representing the time taken to execute events in a modelled system (Jensen, 2013). CPN tool is an industrial-strength computer tool for constructing and analysing CPN models (Jensen & Kristensen, 2009). CPN tools is also an ideal tool to investigate the behaviour of the modelled system using simulation, to verify properties using state space methods and model checking and to conduct simulation-based performance analysis.

Table 1: Model Conditions Definitions in *Iru* Production Process

Activities	Pre-Condition	Post Condition
Loading	Resources needed to process *iru* is available	Resources put in production process
Sorting	Sorting of African locust bean (bean) commenced	Bean sorted
Washing	Sorted bean in production process	Bean washed
Long-cooking	Cleaned bean in production process	Bean partially-cooked
Dehulling	Cooked bean in production process	Bean dehulled
Sieving	Dehulled bean in production process	Bean sieved
Short-cooking	Pure sieved bean in production process	Bean fully-cooked
Draining	Short-cooked bean in production process	Bean drained
Fermentation	Pure drained bean in production process	Bean fermented
Release	Post processing activities commences	



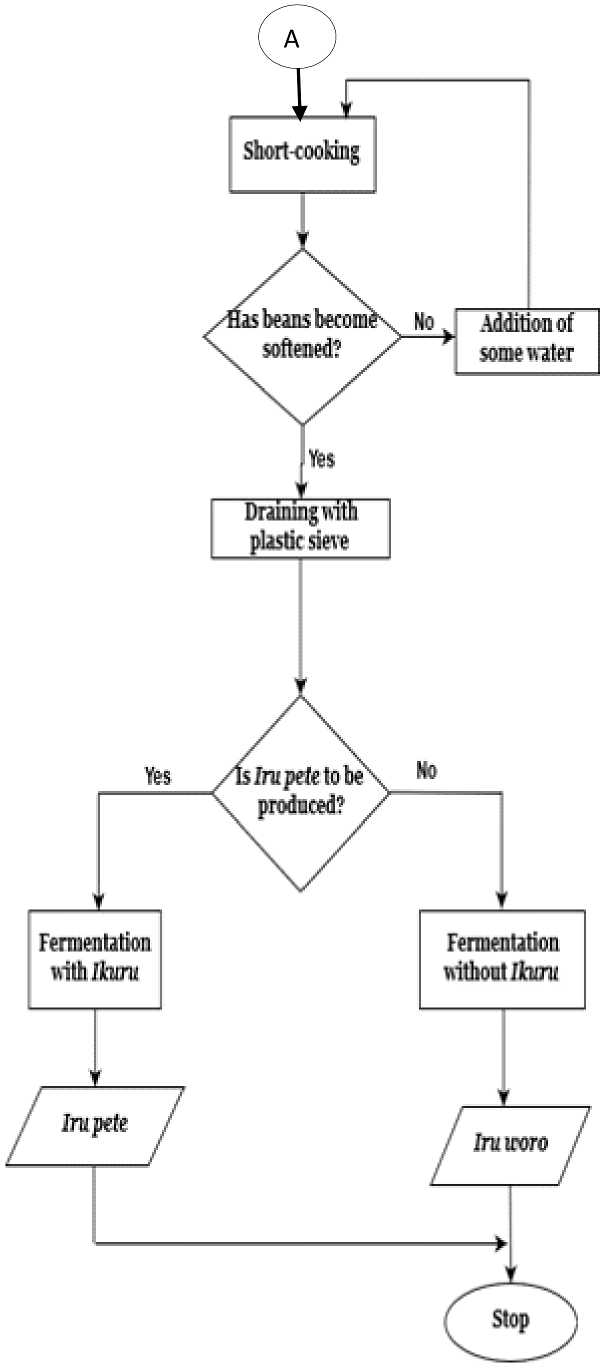


Figure 5: Flowchart for the *iru*production

The CPN language allows a model as a set of modules (Jensen, 1994) and the Hierarchical CPN allows modelling of a large model with several small CPNs. Such that in a hierarchical CPN, it is possible to relate a transition to a separate CPN, providing a more accurate and detailed description of the activities represented by the transitions. Hence, hierarchical CPN is analogous to the non-hierarchical CPN (Huang & Chung, 2010). In this paper, Coloured Petri Nets (CPN), Timed Coloured Petri Nets (TCPN), Hierarchical Timed Coloured Petri Nets (HTCPN) and mathematical formalisms were employed to model the traditional and the mechanized *iru* production processes.

CPN is 9 tuples: = ($\sum$, P, T, A, N, C, G, E, I) (Jensen, *et al.*, 2007) where:

is a finite set of non-empty types, called colour sets.

i. P is a finite set of places describing the change of states of the African locust beans during *iru* production processes.

ii. T is a finite set of transitions describing the processes involved in the production of *iru*.
iii. A is a finite set of arcs describing how processes are executed such that: P T = P A = T A = Ø.
iv. N is a node function. It is defined from A into PxTTxP.
v. C is a colour function. It is defined from P into .
vi. G is a guard function. It is defined from T into expressions such that:tT: [Type(G(t)) =B ∧ Type(Var(G(t)))].
vii. E is an arc expression function. It is defined from A into expressions such thata A : [Type(E(a)) = C(p) $_{MS}$ ∧ Type(Var(E(a)))] where p is the place of N(a).
viii. I is an initialization function. It is defined from P into closed expressions such that: p P: [Type(I(p)) = C(p) $_{MS}$].

Coloured Petri Net Model for IPP

(i) S = {African locust beans};
(ii) P = {Sorted beans, Washed beans, L.Cooked beans, Dehulled beans, Sieved beans, S.Cooked beans, Drained beans,..., Fermented beans};
(iii) T = {Sorting, Washing, Long-cooking, Dehulling, Sieving, Short-cooking, Draining,..., Fermentation};
(iv) A: Order followed to complete the task by the producer;
(v) N is a node function. It is defined from A into PxTTxP.

Timed Coloured Petri Net Model for IPP

Since CPN will describe the dynamic properties of the system in time and space, a global clock is required in a timed CPN model. This will attach a timestamp to each token. A timed non-hierarchical CPN has 3 – tuple TCPN<CPN, R, r_o> (Huang & Chung, 2010), where:

(i) CPN satisfies the coloured Petri nets definition.
(ii) R is a set of time values, also called time stamps. It is closed under + and include0.
(iii) r_o is an element of R called the start.

Hierarchical Timed Coloured Petri Net Model for IPP

The timed CPN (Jensen, 1981) allows the interpretation of the dynamic properties of the IPP system in time and space. The IPP procedure system is composed of eight phases, and each phase describes an individual step. Hence the IPP system can be illustrated by way of a hierarchy model. The presentation of hierarchical CPN (Jensen, 1994) allows the construction of a large system by combining a number of small CPNs. Therefore, the hierarchical CPN with time is employed to model the IPP in this study.

(a) Complete Module:

Hierarchical Coloured Petri Nets module can be expressed as a 4-tuple ΣM = (Σ, Tsub, Pport, PT), where:

(i) Σ = (S, P, T, A, N, C, G, E, I) is a non-linear hierarchical coloured Petri net;
(ii) Tsub⊆ T is a set of substitution transitions for Sorting, Washing, Long-cooking, Dehulling, Sieving, Mechanising, Draining, Short-cooking, Fermentation;
(iii) Pport⊆ P is a set of port places assign to continuous places for Sorted beans, washed beans, partially cooked beans, dehulled beans, sieved beans, fully cooked beans, drained bean and fermented beans;
(iv) PT: Pport → {IN, OUT, I/O} is a port type function that assigns a port type to each port place in IPPs.

(b) Submodule:

Hierarchical Coloured Petri net can also be expressed as a 4-tuple ΣH = (S, SM, PS, FS) (Huang & Chung, 2010), where:

(i) S is a finite set of modules. Each module is an interdependent Petri net module s= ((PS, TS, AS, CS, VS, WS, GS, ES, IS, TS sub, P S sub, PTS), it is required that (Ps1 U Ts1) n (Ps2 uTs2) = Ø for all S1, S2c S such that S1 ≠ S2;
(ii) SM: Tsub → S is a submodule function that assigns a submodule to each substitution transition. It is required that the module hierarchy is acyclic;
(iii) PS is a port-socket relation function that assigns a port-socket relation, PS(t) ⊆ Psock(t) x PSM (t) port to each substitution transition t, it is required that ST(P), C(P) = C(P I), and I (p1) for all (p, p1) ∈ PS (t) and all t ∈ Tsub in IPP;
(iv) FS⊆ 2P is a set of non-empty fusion sets of character time stamps such that C (p) - C (p1)

for all p, p1 ∈ Fs and fs ∈ FS.

Process Modules within the Computational Models

The module is a segment representing different production processes involved in *iru* production. The HTCPN modules are as depicted in Figures 6, 7 and 8. Figures 6 and 7 are top pages of computational models for the TIPP and MIPP, respectively, while Figure 8 is the sorting process. Each of the unit operations has this simulation pages. Top pages provide an overview of the entire processes of *iru* production. Each top page has eight and seven substitution transitions that depict eight and seven processes involved in the TIPP and MIPP, respectively. The modules associated with each top page revealed how the processing time, water quantity, states and conditions of *iyere* change across the processes as a token (with time stamp) moves from one place to another place. Also, each module within the computational models is characterised by the unique code segment. The code segment has input, output and action parts. Figure 9 shows a mechanised process module with attached code segment (as shown in Table 2). The input part of the code segment takes values of a measured quantity of *iyere* (variable, q), current production time (variable, cpt) and several machines used (variable, ma). The output part returns values of cumulative production time for the total number of production processes (variable, ncpt), quantity of *iyere* after the combined operations (variable, qt) and quantity of water consumed (variable, nqw). The action part has standard meta-language (SML) code which enables the developed models to perform each process (activity) of *iru* production. The rate of performing combined dehulling and sieving operations depends on the measured quantity of *iyere* (variable, q), production time used (variable, pt) and some machines used (variable, ma). Cumulative production time (ncpt) computed equal to the current production time (cpt) plus production time used (pt). Table 3 enumerated additional variables used in the code segment in the developed models.

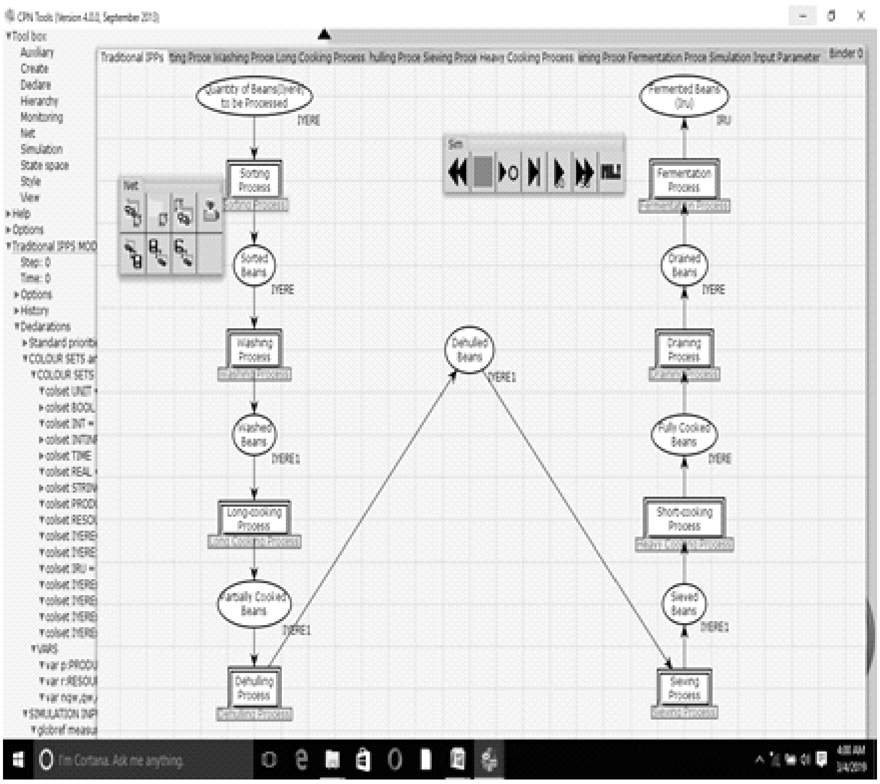


Figure 6: The Top Page of Computational Model forthe Traditional

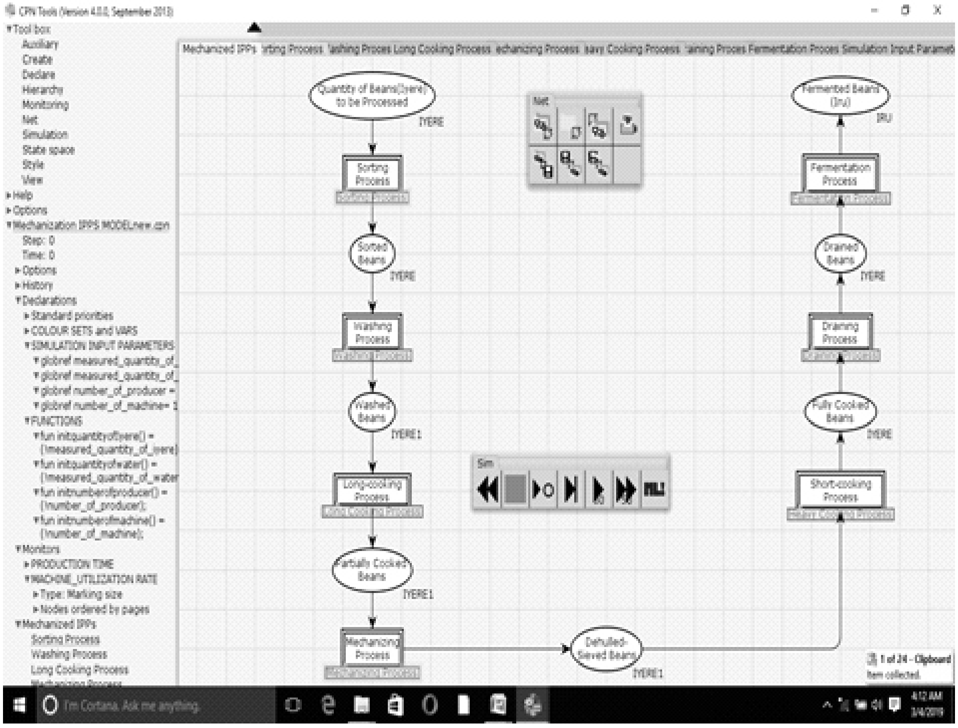


Figure 7: The Top Page of Computational Model for the Mechanized IPP

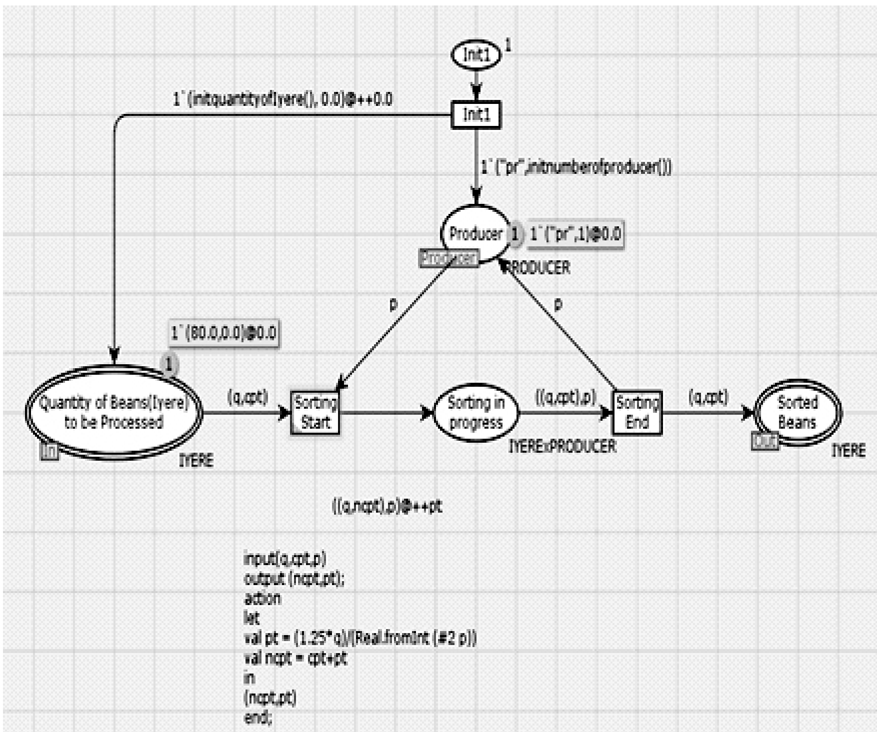


Figure 8: Sorting Process Module for the TIPP

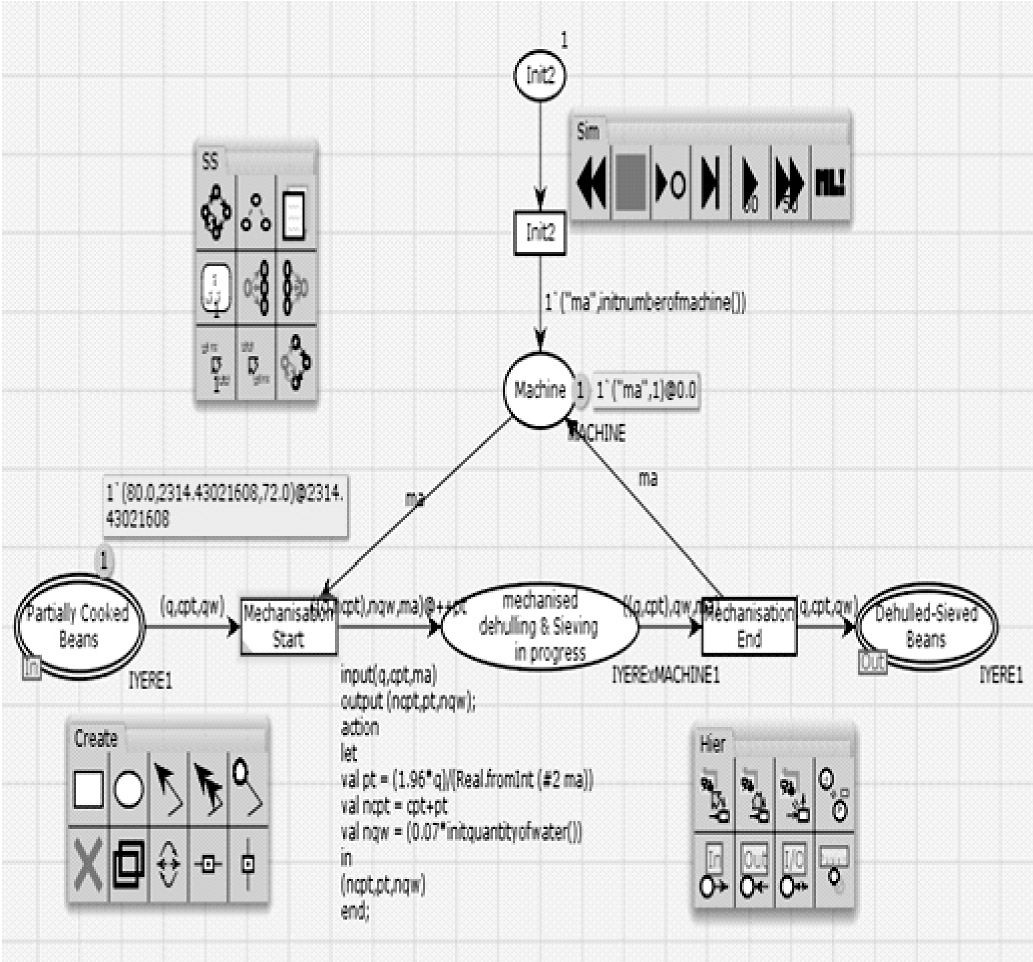


Figure 9: Mechanised Process Module for the MIPP

Table 2: Code Segment within the developed computational model

Mechanized Process Module
Input (q, cpt, ma) Output (ncpt, pt, nqw); action let valpt = (1.96*q) / (Real.fromint(#2 ma)) valncpt= cpn + pt valnqw= (0.07*initquantityof water (#100)) in (ncpt,pt,nqw) End;

place name in column two belongs to a specific place type. The noteworthy, place could belong to a continuous or discrete type as shown in Figure 10. Double-overlapped circles and rectangles stand for continuous places and continuous transitions, respectively. The continuous types are submodules with several tasks running within it. Stand-alone circles and rectangles stand for discrete places and discrete transitions, respectively. Continuous place means satisfying present conditions that lead to the execution of next actions connecting to another process module. Discrete place means satisfying present condition and perform only current activities within the specific module. Column four discusses the function of each place name. Every place-name ending with "in progress" (Figure 10) models the current state of locust beans at the present production process on the mid-way to the next production phase. Meanwhile, Table 7 shows different transition names, transition types and respective descriptions of their actions performed in the developed models. Each transition name in column two has two transition types. Each transition type could be a substitution or discrete as shown in column three. Substitution transition models execution of a present event that leads to the formation of new conditions in another process module. Discrete transition means to perform a specific activity within the processing module. Column four discusses the function of each transition name.

Table 4: Production Resources Usage Rate for the Developed TIPP Model

Process Name	Process Resources	Resources Process Time / Work Efficiency	Water Usage Rate Observed in the centre	Output to Input Ratio
Sorting	*Iyere* and Processor	1.25 min/kg	--	--
Washing	*Iyere*, Processor and Water	05 min/kg	Water (6.7 %) taken at the washing process = 0.067*initquantityofwater	--
Long-cooking	*Iyere*, Cooking pot and Water	1320-1440 min	Water (56.90 %) taken at long-cooking process = 0.5690*initquantityofwater	--
Dehulling	*Iyere*, Machine and Water	10 min/kg	Water (1.70 %) taken used with machine = 0.017*initquantityofwater	--
Sieving	*Iyere*, water and sieve	12.50 min/kg	Water (11.70 %) taken used with machine = 0.117*initquantityofwater	--
Short cooking	*Iyere*, Processor and Water	90-120 min	Water (23 %) taken at the short-cooking process = 0.23*initquantityofwater	--
Draining	*Iyere*, Processor and Sieve	2 min	--	--
Fermentation	*Iyere*, Processor, Calabar and Clothes	2520-2880 min	--	nq=0.80 *q

Table 3: Variables Used in the Developed Models

Variable (Data type)	Interpretations
initquantityof*iyere* (integer)	It describes the measured quantity of African locust beans
q (variable)	It describes the initial value of the measured quantity of *iyere*
cpt (variable)	It describes current production time
pt (variable)	It describes production time taken
initnumberproducer "Pr" (integer)	It represents a number of processors involved
qw (variable)	It describes the measured quantity of water available
nqw (variable)	It describes the measured quantity of waterused
ncpt (variable)	It denotes cumulative production time for the total number ofthe production process
initquantityofwater (integer)	It denotes the measured quantity of water
r (unit/const)	It represents additional resources (such as calabash and clothes)
ma (variable)	It denotes several machines used
initnumberofmachine (integer)	It denotes the number of machines available for usage
qt (variable)	It denotes the quantity of *iyere* left after the combined operations
	It denotes a new value of fermented African locust beans (*iru*).
nq	It describes the current input value of the measured quantity of *iyere*at the washing stage following MIPP
qn	

Simulation Results from the Computational Models

The computational models for the production process were simulated on DELL laptop computer with configurations of Intel (R) Pentium (R), CPU 3558U, 1.70 GHz, 4.00GB Memory Size and 64-bit windows10 operating system using Coloured Timed Petri Nets (CPN) tool and evaluated using a measured quantity of raw beans (per kilogram) with respect to the total production time (minute) as performance metrics.Knowledge elicited and information gathered concerning the type of production resources, some production resources, time taken in processing production resources and others were used in the developing the proposed models which were later subjected to simulations. Tables 4 and 5 provide baseline information and revealed work efficiency (WE) that benchmark every output value from stochastic processes of *iru* production. A processor spent 4.00 min to perform the sorting process over 3.20 kg. That is 4.00 min per 3.20 kg of locust beans at the sorting process by following the traditional method. Thus, 1 processor * 4.00 min/3.20 kg = 1.25 processor min/kg. At sieving stage, 3.20 kg took 40 min with work efficiency of 12.50 processor min/kg. That is, 1 processor * 40.00 min/3.20 kg = 12.50 processor min/kg and so on (Table 4). By following the mechanized method, 28 min per 4.80 kg of African locust beans at the sorting stage. Thus, 1 processor * 28.00 min/4.80 kg = 5.83 processor min/kg. At the washing stage, every measured quantity of 4.80 kg took 4.00 minutes with an additional time of 30.00 min, irrespective of input values of *iyere* being processed. With a processing machine,4.80 kg took 9.408 min, where 1 processor * 9.408 minutes / 4.80 kg= 1.96 min/kg and so on (Table 5).Researchers observed based on the information obtained from the centre and simulation results that the ratio of production of *iru* from iyere is 0.80: 1.00 and 0.79: 1.00 for TIPP and MIPP, respectively (Tables 4 and 5). The developed models were designed using hierarchical timed colouredPetri nets formalism. Hierarchical formalism means that structuring mechanism was applied. That is, the models developed was structured according to some production processes of *iru* and each process mainly represented as a module. Timed formalism denotes duration taken for each process which is denoted by timestamp. Coloured formalism stands for representation of different attributes of data without the use of the standard black token. Transitions represented activities within IPP. In designing, specific names were given to some transitions and places for the modelling of events (actions) take place before the formation of *iru* and the state of *iyere*, respectively. Table 6 shows the different place names, place types and respective descriptions of their functions. Each

Table 5: Production Resources Usage Rate for the Developed MIPP Model

Process Name	Process Resources	Resources Process Time / Work Efficiency	Water Usage Rate Observed in the centre	Output to Input Ratio
Sorting	*Iyere* and Processor	5.80 min/kg	--	--
Washing	*Iyere*, Processor and Water	(3.57(qn) min/kg)+30 min	Water (10 %) taken at washing process = 0.1*initquantityofwater	--
Long-cooking	*Iyere*, Cooking pot and Water	1320-1440 min	Water (60 %) taken at long-cooking process = 0.6*initquantityofwater	--
Combined Dehulling and Sieving	*Iyere*, Machine and Water	1.96 min/kg	Water (07 %) taken used with machine = 0.07*initquantityofwater	--
Short cooking	*Iyere*, Processor and Water	90-120 min	Water (23 %) taken at the short-cooking process = 0.23*initquantityofwater	--
Draining	*Iyere*, Processor and Sieve	2 min	--	--
Fermentation	*Iyere*,Processor, Calabar and Clothes	2520-2880 min	--	nq=0.79*q

Table 6: MajorPlaces Used in the Developed Models

S/No.	Place Name	Place Type	Description
1	Quantity of Beans (*iyere*) to be processed.	Continuous	Models quantity needed for the process of *iru* is available and measured in a kilogram.
2	Sorted beans	Continuous	Models measured and sorted beans putting into washing process within the stipulated time.
3	Washed beans	Continuous	Modes measured and sorted beans ready for the long-cooking process.
4	Partially cooked beans	Continuous	Models measured, sorted and partially cooked beans ready for the dehulling process.
5	Dehulled beans	Continuous	Models measured, sorted, washed, partially cooked and dehulled beans are ready forthe long-cooking process to be subjected to sieving process in *iru* production.
6	Sieved beans	Continuous	Models measured, sorted, partially cooked, dehulled and sieved beans ready and awaiting dehulling process.
7	Heavily cooked beans	Continuous	Models measured, sorted, washed, partially cooked, dehulled, sieved and short-cooking beans ready to proceed to draining process.
8	Drained beans	Continuous	Models fully drained beans ready for fermentation.
9	Fermented beans	Continuous	Models fully fermented African locust beans (*iru*).

Table 7: Major Transitions Used in the Developed Models

No.	Transition Name	Transition Type	Description
1	Sorting process	Substitution	Execution of this transition models activity involving the removal of unwanted particles from the beans.
2	Washing process	Substitution	Execution of this transition models getting rid of fine sands and having sand-free locust beans.
3	Long-cooking process	Substitution	Execution of this transition models softening of the beans during first-term cooking beans for a few hours.
4	Dehulling process	Substitution	Execution of this transition models removal of seed's coats (hull) ahead of sieving process.
5	Sieving process	Substitution	Execution of this transition models separation of dehulled beans from any unwanted particles.
6	Mechanizing process	Substitution	Execution of this transition models dehulling and sieving of locust beans using processing machine.
7	Short-cooking process	Substitution	Execution of this transition models further softening cooking of pure beans to speed-up fermentation.
8	Draining process	Substitution	Execution of this transition models activities towards achieving dry-pure locust beans.
9	Fermentation Process	Substitution	Execution of this transition models microbiological activities that leads to the formation of *iru*.

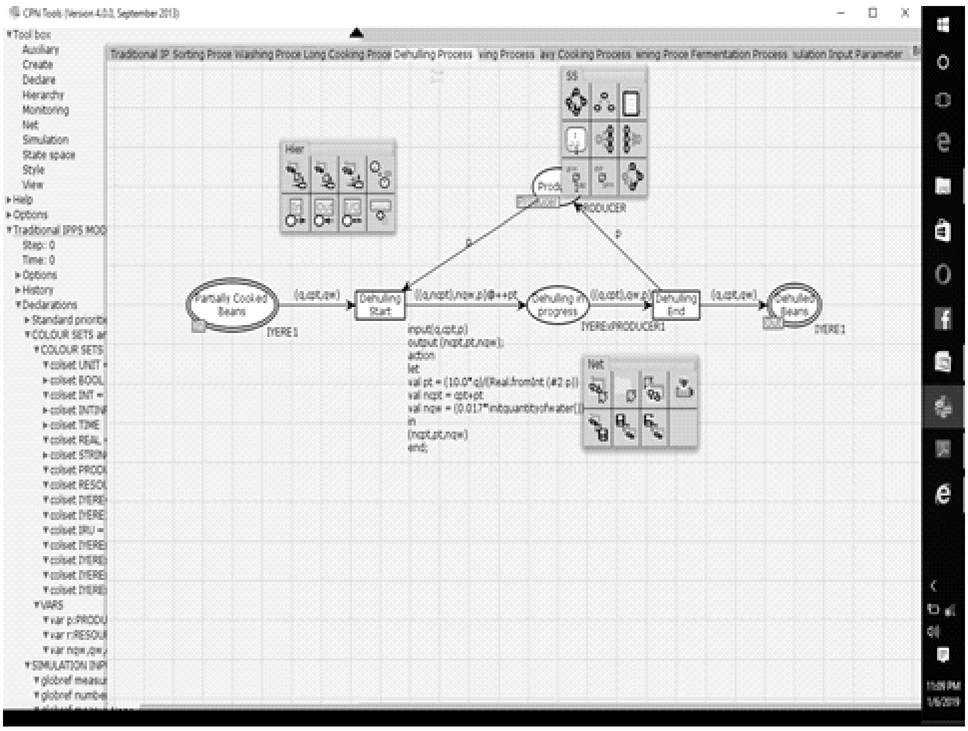


Figure 10: The Dehulling Process Module for Traditional IPP

The simulation results (as shown in Figure 11) reveals the production time against the quantity of raw beans for both mechanized and traditional process. 16, 32, 38 and 80 kg of *iyere* take 4644, 5094, 5559 and 6493 min of production time for TIPP; and 4540, 4728, 4910 and 5267 min of production time for MIPP, respectively. This result implies that for a higher quantity of raw beans, the mechanized process gives shorter production time. Also, the simulation results (as shown in Figure 12) showed the production ratio, that is, raw beans (input): fermented beans (output).The traditional process gave 16:12.8; 32:25.6; 48:38.4 and 80:64. Whereas, the production ratio for the mechanized process gave 16:12.64; 32:25.28; 48:37.92; and 80:63.2. This result implies not only that more product is obtained from the traditional process than the mechanized process but also reveals a higher production ratio TIPP than the MIPP. The result (as shown in Table 8) further reveals resource (water) usage rate in the production. 6.70, 56.67, 1.70, 11.70 and 23.00 % of the water required during washing, long-cooking, dehulling, sieving and short-cooking process of TIPP, respectively; and 10.00, 60.00, 7.00 and 23.00 % of the water required during washing, long cooking, mechanizing and short-cooking process of MIPP, respectively. This shows that the combined mechanized dehulling and sieving process of MIPP consume appreciable lower water than individual dehulling and sieving process in the TIPP. Hence, resources usage can be minimized in the mechanized process.

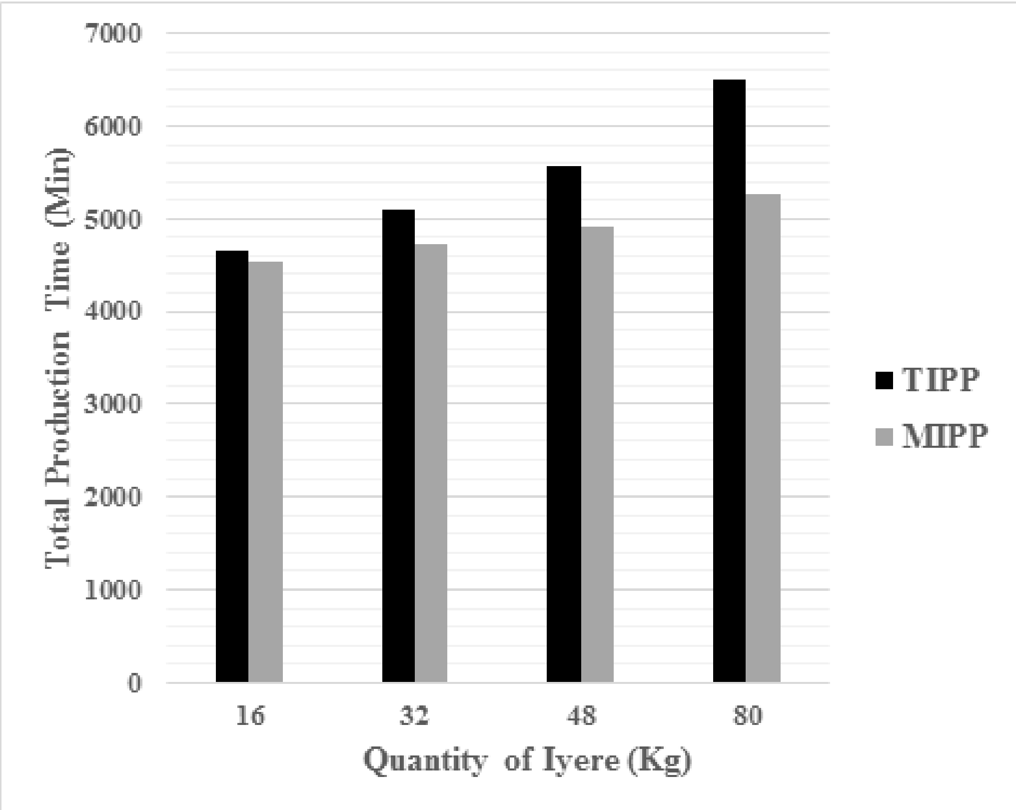


Figure 11: Comparing Production Time of TIPP and MIPP

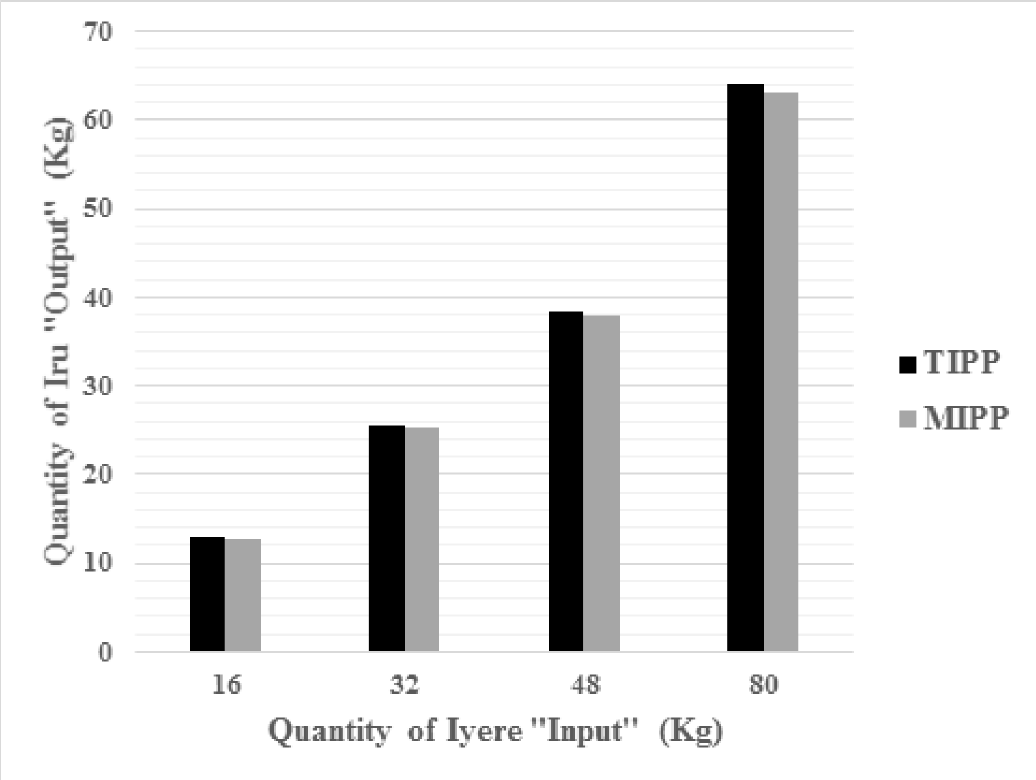


Figure 12: Comparing Production Ratio of TIPP and MIPP

Table 8: Resource (Water) Usage per Production Process

TIPP		MIPP	
Process	Water Usage Rate (%)	Process	Water Usage Rate (%)
Washing	6.70	Washing	10.00
Long-cooking	56.67	Long-cooking	60.00
Dehulling	1.70	Mechanised dehulling and sieving	7.00
Sieving	11.70		
Short-cooking	23.00	Short-cooking	23.00

Limitation, Conclusion and Future Work

The study has employed only the machine developed by Owolarafe *et al.*, 2011 which is the only once embraced for commercial production in the area under study. In this paper, we developed computational models for the two existing processes of fermented African Locust beans. The focus is on the processing time, production ratio and resource (water) usage rate using multiple production centres (as a case study) in Nigeria. Each module of the Hierarchical Timed Coloured Petri Nets model was developed based on collected data and knowledge elicited from the producers. The developed models were simulated using CPN tools and the Standard Meta Language to investigate different scenarios. The parametric model has revealed the while the mechanized process utilizes a lesser quantity of water and requires shorter production, the traditional production process yields a higher output of fermented African locust beans. When carefully followed in future automation as a reference model, the problem of high-water consumption, no precise production time, material wastage, great human resources and low profit could be ameliorated. Furthermore, future research may be tailored towards determination of a profitability per month using an applicable technique such as break-even point analysis.

ACKNOWLEDGEMENTS
The authors acknowledged visitation to *iru* processors and technologists at machine workshops, Department of Agricultural and Environmental Engineering, Obafemi Awolowo University in Osun State, Nigeria for their immense support in the collection of data, knowledge elicitation and allowing the access of their facilities for the study. The study was done at the department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria. During and after the study, we appreciated immeasurable contributions made by Dr. Rafiu Adesina Ganiyu and Mr. Taofeeq Alabi Badmus LAUTECH and FUOYE, respectively. Lastly, first author also appreciated the financial support from Nigeria Computer Society (NCS), Nigeria for the successful completion of the study.

REFERENCES

Adeniran, H. A. Farinde, E. O. and Obatolu, V. A. 2013. Effect of heat treatment and fermentation on anti-nutrients contents of lima bean (Phaseolus lunatus) during production of daddawa analoque. *Annual Research & Review in Biology,* 256-266.

Akande, F. Adejumo, O. Adamade, C. and Bodunde, J. 2010. Processing of Locust bean fruits: Challenges and prospects. *African Journal of Agricultural Research*, 5(17): 2268-2271.

An, Y. S. *et al*., 2018. Hierarchical coloured petri nets for modeling and analysis of transit signal priority control systems. *Appl. Sci*., 8(1): 141-147.

Bello, S. A. Sanni, M. L. and Odejobi, O. A. 2011. A Computational Modelling and Design of the Adire Making Process Using Modular Petri Net.*Journal of Institute of Mathematics and Computer Science*, 2(1): 51-58.

Campbell- Platt, G. 1980. African Locust Bean (Parkia Species) and its West African Fermented Food Product, Dawadawa. *Ecology of Food and Nutrition*, 9(2): 123-132.

Davidrajuh, R. and Lin, B. 2011. Exploring airport traffic capability using Petri net-based model. *Expert Systems with Applications*, 38(9): 10923-10931.

Dayan, P. 1994. Current Opinion in Neurobiology. Computational modeling, 4(2): 212- 217.

Farayola, C. O. Okpodu, V. and Oni, O. O. 2012. Economic Analysis of Locust Beans Processing and Marketing in Ilorin, Kwara State, Nigeria. *International Journal of Agricultural Research Innovation & Technology,* December, 2 (2): 36-43.

Ganiyu, R. A. Okediran, O. O. Muraina, B. O. and Badmus, T. 2019. Development of a Modularized Model for a MultiProcess Food Manufacturing System using Hierarchical Timed Coloured Petri Nets. *Annals. Computer Science Series Journal*, 17(1).

Ganiyu, R. A. Olabiyisi, S. O. Badmus, T. A. and Akingbade, O. Y. 2015. Development of a Time Coloured Petri Nets Model for Health Centre Patient Careflow Processes. *International Journal of Engineering and Computer Science,* 4(1): 9964-9961.

Ganiyu, R. A. *et al*. 2011. Modelling and Simulation of a Multi_Phase Traffic Light Controlled T-Type junction using Timed coloured Petri Nets. *American Journal of Scientific and Industrial Research*, 2(3): 428 - 437.

Girault, C. and Reisig, W. 1982. Application and Theory of Petri Nets. *Informatik-Fachberichte*, Volume 52.

Hack, M. 1976. Petri Net Language, Cambridge, USA: Massachusetts Institute of Technology.

Heiner, M. *et al*. 2012. Snoopy – A Unifying Petri Net Tool. Berlin, Heidelberg, Springer.

He, X. and Murata, T. 2005. High-level Petri nets—extensions, analysis, and applications. *Electrical engineering handbook*, 459–476.

Hill, L. Crosier, S. Smith, T. and Goodchild, M. 2001. A Content Standard for Computational Models. *D-Lib Magazine*, 7(6).

Huang, Y. and Chung, T. 2010. Modelling and Analysis of Traffic Light Control Systems Using Timed Coloured Petri nets. *InTech:Petri Nets Applications*, 567-587.

Hubert, S. Helmers, T. Groß, F. and Delgado, A. 2016. Data driven stochastic modelling and simulation of cooling demand within breweries. *J. Food Eng.*, 176, 97-109.

Hussein, L. 2014. Simulation of Food Restaurant using Coloured Petri Nets. *Journal of Engineering and Sustainable Development,* 18(4): 77-88.

Jensen, K. 1981. Coloured Petri Nets and the Invariant-Method. *Theoretical Computer Science*, 14(3): 317–336.

Jensen, K. 1993. An Introduction to the Theoretical Aspects of Coloured Petri Nets. A Decade of Concurrency, Reflections and Perspectives, Volume 803.

Jensen, K. 1994. Coloured Petri Nets. Basic Concepts, Analysis, Methods and Practical Use. II ed. Berlin: SpringerVerlag.

Jensen, K. 1997. A Brief Introduction to Coloured Petri Nets. Enschede, The Netherlands, TACAS 1997.

Jensen, K. 2013. Coloured Petri nets: Basic Concepts, Analysis Method and Practical Use. Book of Springer Science and Business Media, Volume 1.

Jensen, K. and Kristensen, L. 2009. Coloured Petri Nets: Modelling and Validation of Concurrent Systems. *Book of Springer Science and Business Media.*

Jensen, K. Kristensen, L. M. and Wells, L. 2007. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4): 213–254.

Jijo, K. J. and Ramesh Kumar, K. R. 2014. Process Automation for a Food Processing Plant. *International Journal of Science, Engineering and Technology Research (IJSETR),*3(6): 1744-1750.

Johanen, T. and Babuska, R. 2003. Multi-objective Identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 11(6): 847-860.

Kovacs, A. Nemeth, E. and Hangos, K. M. 2005. Modeling and optimization of runway traffic flow using coloured Petri nets. *Budapest, Hungary*, s.n., 881-886.

Li, Y. F. and Lau, C. C. 1989. Development of fuzzy algorithm for servo system. *IEEE Control System Magazine*, 65-71.

Mahalik, N. 2014. Advances in packaging methods, processes and systems. *Challenges*, 5(2): 374-389.

Mandlik, A. D. and Borkar, B. R. 2015. Simulation in Food Processing - A Review. *International Journal of Enhanced Research in Science Technology & Engineering*, 4(1): 151 - 155.

Morasca, S. Pezzè, M. and Trubian, M. 1991. Timed High-Level Nets. *The Journal of Real-Time Systems*, 3 (2), 165- 189.

Murata, T. 1989a. Petri Net: Properties, Analysis and Application. 77(4): 541-580.

Murata, T. 1989b. High Level Petrinet Extension, Application System, Florida: School of Computer Science.

Musa, H. 1991. Ginger and Locust bean tree. History, growth, uses and potentials. Kurrnin-Musa, Kaduna State, *'Tuk-Ham' symposium,* 1-16.

Oberheid, H. and Söffker, D. D. 2008. Cooperative arrival management in air traffic control – A coloured Petri net model of sequence planning. *Berlin, Heidelberg, Springer-Verlag*, 348-367.

Odejobi, O. A. and Owolarafe, O. K. 2004. Computer simulation of fuzzy control system for gari fermentation plant. *Journal of Food Engineering*, 68: 197-207.

Olaoye, J. 2016. Performance testing of Small-Scale Equipment for Depulping Locust Bean Seed. Transactions of theASABE, 59(3): 821-828.

Olusanya, O. Omidiora, E. Olabiyisi, S. and Ganiyu, A. 2017. Development of A Model for Bank Cash Deposit Transactions Using Hierarchical Timed Coloured Petri Nets. *International Journal of Engineering Research and Development,* 13(10): 25-32.

Owolarafe, O. K. *et al.* 2011. Development of a Locust Bean Processing Device. *Journal Food Science and Technology*, 19(3): 50-58.

Owolarafe, O. K. Adio, S. A. Olatunde, G. and Kolawole, O. 2013. Performance Evaluation of a Motorised Locust Bean

Processing Machine. *Journal of Agricultural Engineering and Technology*, 21: 21-27.

Owolarafe, O. K. Omobuwajo, T. O. Olatunde, G. A. and Owagbemi, O. O. 2010. Development of a locust bean depulping device. Ife Journal of *Technology*, 19(2): 69-73.

Rousu, J. *et al.* 2003. Novel computational tools in bakery process data analysis: a comparative study. *Journal of Food Engineering*, 57(1): 45-56.

Sarkar, P. Hasenack, B. and Nout, M. 2002. Diversity and functionality of Bacillus and related general isolated from spontaneously fermented soybeans (Indian Kinema) and locust beans (African Soumbala). *International Journal of Food Microbiology*, 7(3): 175-186.

Takagi, T. and Sugeno, M. 1985. Fuzzy identification systems and its applications to modelling and control. *IEEE Transactions on Systems Man and Cybernetics*, 5: 6-132.

Van der Aalst, W. M. P. and Odijk, M. A. 1995. Analysis of railway stations by means of interval timed coloured Petri nets. *Real-Time Systems*, 9:41-263.

Van der Aalst, W. M. P. 1994. Putting high-level Petri nets to work in industry. *Computers in industry*, 25(1): 45-54.

Van der Aalst, W. M. 1993. Interval timed coloured petrinets and their analysis. *In International conference on application and theory of petri nets. Springer, Berlin, Heldelberg*, 453-472.

Wang, H. and Qiu, D. 2003. Computing with words via Turing machines: A formal approach. *IEEE Transactions on Fuzzy Systems*, 742-753.

Webb, S. and Hoben, D. 1983. Modern Technology for an ancient art: Application: computer control of Whisky Production. *Elsevier Journal of Data processing*, 25(8), 31-33.

Xin, N. Z. 2015. Modelling of Milk Production Process using Simulation; *Bachelor ofIndustrialTechnology Management*, University of Malaysia, Pahang.

Zuberek, W. M. 1980. Timed Petri Nets and Preliminary Performance Evaluation. ACM, *'80 Proceedings of the 7th Annual Symposium on Computer Architecture*, 88-96.