Revista ARACÊ
ISSN: 2358-2472

SYSTEM CONTROL FOR A CAKE MACHINE

https://doi.org/10.56238/arev6n4-334

Submitted on: 20/11/2024 **Publication date:** 20/12/2024

Luiz Henrique Sousa da Silveira[1], Uri Santos de Souza[2], Gabriela Fidelis de Oliveira[3], Paulo Victor Vasconcelos Marajó[4] and José Cavalcante Lacerda Junior[5]

ABSTRACT
This paper studies the application of Petri Nets and Finite Automata for the optimization of systems in a cake machine. Petri Nets are employed to model and visualize the dynamic interactions between the various components of the system, providing a robust and detailed graphical representation of workflows and processes. On the other hand, Finite Automata are used to capture and represent the discrete states and transitions of the system, allowing an accurate and rigorous analysis of expected and possible behaviors. The integration of these two approaches provides a comprehensive methodology for the operational optimization of the cake machine. Petri Nets allow you to identify and analyze critical points and bottlenecks in the system, while Finite Automata facilitate the clear definition of states and the management of transitions between them. This combination enables a detailed analysis of systemic interactions, allowing the implementation of structured and efficient improvements that result in optimized performance. In addition, the joint application of these techniques provides a flexible and adaptable platform for the simulation and validation of different operational scenarios, ensuring that the implemented changes are effective and sustainable. In summary, this paper demonstrates how the junction between Petri Nets and Finite Automata can be exploited to achieve significant optimization in the operation of cake machines, highlighting the relevance and applicability of these tools in the engineering of complex systems.

Keywords: Petri Networks, Finite Automata, System Optimization, Dynamic Modeling, Systems Engineering.

[1] Graduating from the Control and Automation Engineering Course at the Federal Institute of Education, Science and Technology Campus Manaus Industrial District
E-mail: luizhenrique.silveira07@gmail.com
[2] Graduating from the Control and Automation Engineering Course at the Federal Institute of Education, Science and Technology Campus Manaus Industrial District
E-mail: uri.souza@hotmail.com
[3] Undergraduate student of the Control and Automation Engineering Course at the Federal Institute of Education, Science and Technology Campus Manaus Industrial District
E-mail: gbrielaffidelis@gmail.com
[4] Graduating from the Control and Automation Engineering Course at the Federal Institute of Education, Science and Technology Campus Manaus Industrial District
E-mail: pauloxd.vasconcelos@gmail.com
[5] Dr. in Environmental Sciences and Sustainability in the Amazon. Professor at the Federal Institute of Education, Science and Technology Campus Manaus Industrial District
E-mail: jose.cavalcante@ifam.edu.br

Revista ARACÊ
ISSN: 2358-2472

INTRODUCTION

The food industry is experiencing a remarkable transformation with the arrival of automation and cutting-edge technologies. In particular, in the confectionery sector, automated machines play a fundamental role in the standardization and efficiency of production processes. This text aims to investigate the improvement of the operational process in equipment intended for the manufacture of cakes, a crucial sector of the food industry.

With the increased demand for high-quality cakes and customized varieties, it becomes essential to improve the effectiveness of the operational process. This quest focuses on the delicate balance between preserving artisanal quality and maximizing resources. In this scenario, we will analyze the tactics, technologies, and best practices that can be applied to improve the production of cakes in quantity, while preserving the sensory and gustatory integrity that consumers are looking for.

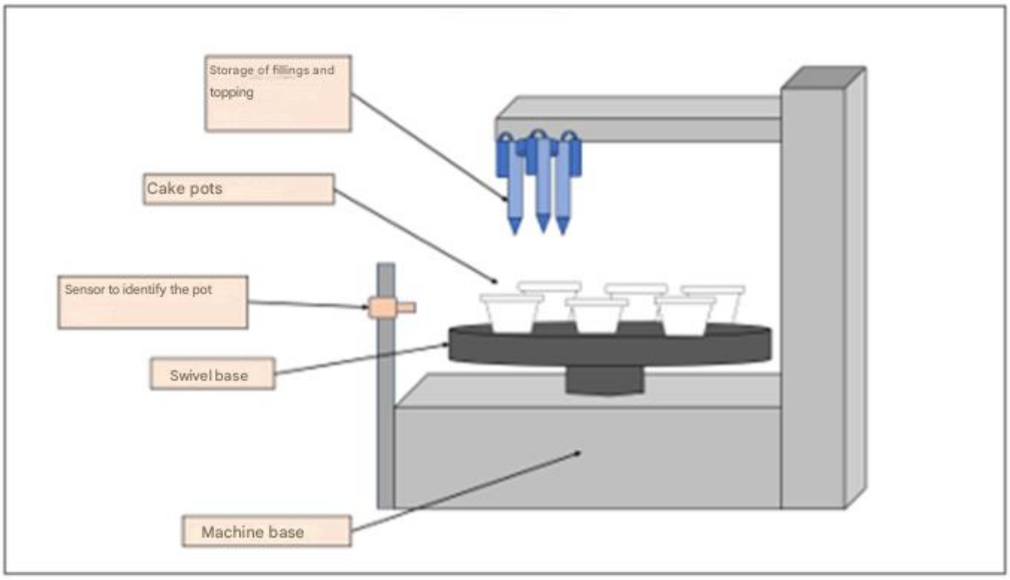
In this article, we will discuss the various options for enhancing a process, ranging from meticulous ingredient choice to intelligent physical automation of the process, emphasizing case studies, emerging trends, and future prospects. By doing so, our goal is not only to improve production efficiency, but also to elevate the experience offered by the user-machine interaction.

PROBLEM DEFINITION AND PROCESS OPTIMIZATION PROPOSAL

In this automated cake production system, a stock of ready-made cake batter is used, stored to be used as needed. The operator feeds the process by adding the dough to the pots, which are then placed on a turntable of the machine. The process consists of six distinct steps: initially, the feeding of the process with the pots; then the activation of the system; the customer chooses the filling option and whether they want topping; the addition of the filling is done according to the customer's choice; the addition or not of coverage is also carried out according to the customer's preference; Finally, the customer can pick up their cake. As we can see in Figure 1.

Revista ARACÊ
ISSN: 2358-2472

Figure 1: Cake-making process



Source: Authors, 2024

The purposes to be achieved through process optimization can be synthesized to reduce the total operating time, save materials and improve the distribution of fillings and frosting throughout the cake making procedure. These goals aim not only to increase the efficiency of the process, but also to contribute to the overall effectiveness of production, resulting in significant benefits in terms of time savings, waste reduction, and improvement of the quality of the final product to be delivered to the customer.

SYSTEM MODELING

Computational paradigms such as Petri Nets and Finite Automata have played a crucial role in modeling and optimizing complex systems, offering powerful tools to analyze and improve the performance of processes in several areas. Petri nets, introduced by Carl Adam Petri in the 1960s, graphically represent the dynamics of competing systems, providing a visual and formal view for understanding interactions between components. Recent research, such as that of (Murata *et al*., 2016), highlights the application of Petri nets in the modeling of distributed systems, demonstrating its effectiveness in the analysis of parallelism and concurrency.

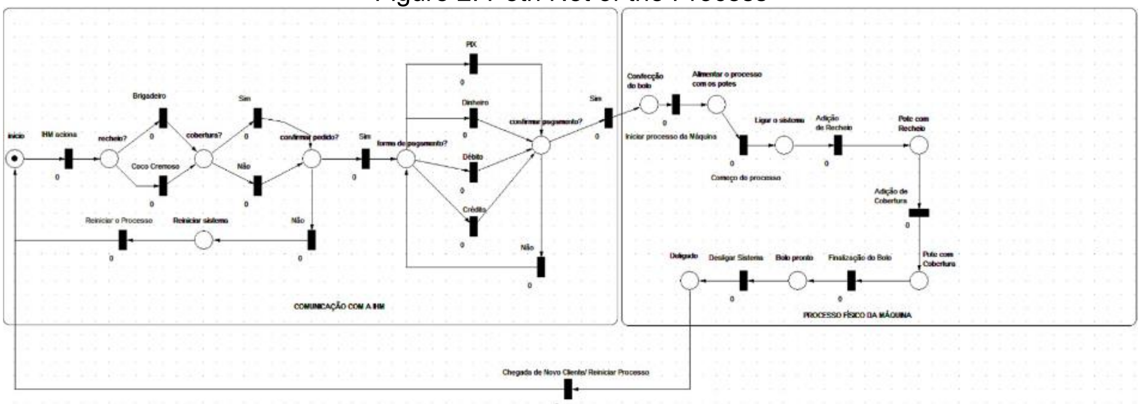
In turn, Finite Automata emerge as a fundamental abstraction for the representation of sequential behaviors in computational systems. Studies such as that of (Hopcroft *et al*., 2006) highlight the versatility of Finite Automata in modeling formal languages, making them valuable instruments for the optimization of processes with defined control logics. In this

Revista ARACÊ
ISSN: 2358-2472

article, we will explore the contributions of these models, discussing relevant case studies and highlighting the relevance of Petri Networks and Finite Automata in the optimization of modern systems.

The program used to perform the modeling and representation of the Petri net of the process in question was Visual Object Net ++, through which it was possible to create a controlled environment where it was possible to visualize all the steps of the process, in detail.

Figure 2: Petri Net of the Process



Source: Authors, 2024

Through the Petri net, we can observe the steps of the process, including its initial branches and derivatives, visually demonstrating how the machine would act in real time. Thus, the article intends to demonstrate solutions that seek to facilitate the management of the system, making it efficient and automated, in the process of ordering and making cakes. It ranges from choosing fillings and toppings to confirming the order, payment method and finalizing the cake. The system aims to optimize workflow, minimize errors, and provide a more effective customer service experience.

METHODOLOGY

CHARACTERISTICS OF THE OPERATION OF THE PROCESS

The system modeled by the Petri net presented is intended for order management and cake making. It comprises different states (places) and transitions that represent the evolution of the process, from the beginning to the delivery of the cake. The interaction between places and transitions is guided by arcs, reflecting the conditions and events that trigger changes in the system. Compared to traditional approaches or manual systems for cake order management, modeling using a Petri net offers significant benefits.

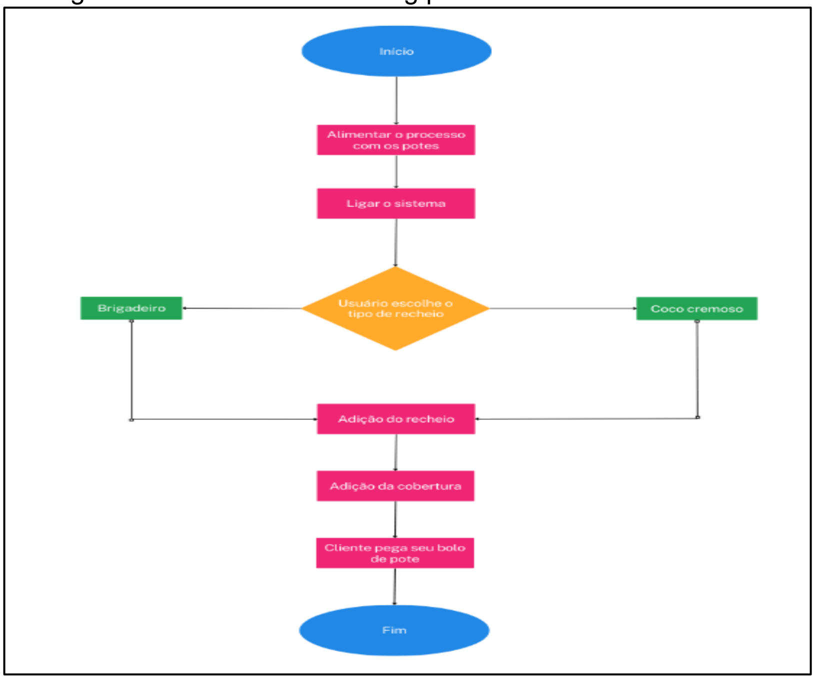
Revista ARACÊ
ISSN: 2358-2472

Compared to linear or sequential systems, by using Petri nets it becomes possible to better represent the concurrency and the simultaneity of events, providing a more comprehensive view of the system's behavior. Additionally, modeling offers a clear visual representation of the process, making it easy to understand and analyze.

The key differentiator of this solution is its ability to model complexities inherent to the cake-making process, considering variables such as choices of fillings, toppings, and payment methods. The flexibility of the Petri net allows it to easily adapt to changing process requirements, making it a scalable and dynamic choice. In addition, the intuitive visual representation of the Petri net makes it easy to identify bottlenecks and continuously optimize the process.

CAKE MACHINE SYSTEM DEVELOPMENT

Figure 3: Outline of the working process of the cake machine



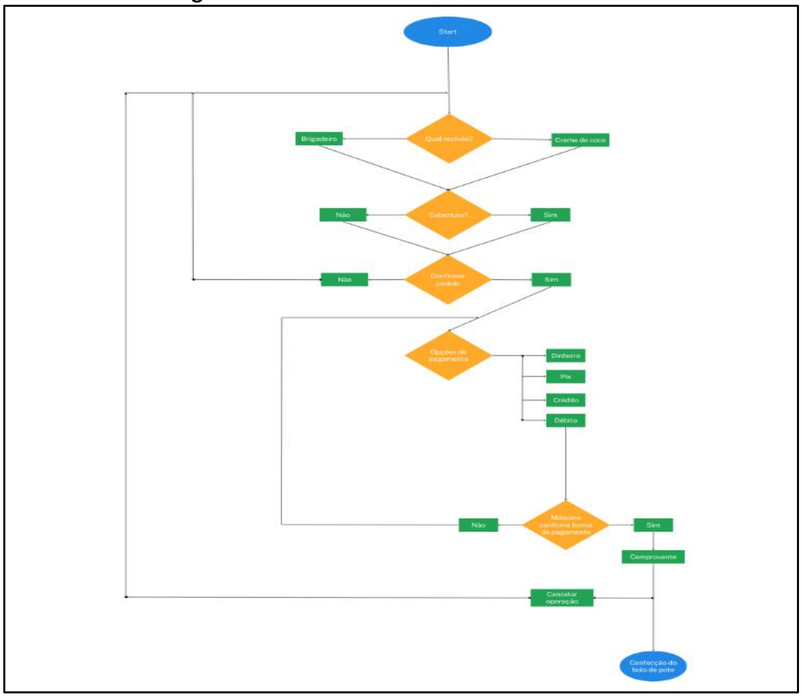
Source: Authors, 2024

As can be seen in the figure above, we have a representation of how the entire process of making the cakes will be executed, from the feeding process to the preparation and later its receipt. We use the ready-made cake dough, which will be stored and according to the need to feed the process, the operator places the dough inside the pot and places it on the machine's turntable. The process makes 1 cake every 2min, the process is divided into 6 steps, which consist of feeding the process with the jars, turning on the

Revista ARACÊ
ISSN: 2358-2472

system, the customer chooses his filling option and if he wants frosting, then the filling is added according to the customer's choice, then the icing is added or not according to the customer's choice, Finally, the customer can pick up his cake.

Figure 4. Cake machine HMI schematic



Source: Image by the author

The case structure is essential for the efficiency of a system because it allows the organization and categorization of information in a logical and accessible way. This facilitates the analysis and resolution of complex problems in a systematic way, ensuring faster and more accurate decisions based on pre-defined criteria for each specific situation. Observing the image represented in figure 4, we can obtain in more detail the case structure of the developed system, with its operating conditions, thus demonstrating in a more visual and clear way the route obtained by the system when completing each of the operations or steps during the initial process at the end.

STRUCTURES OF THE PETRI NET CAKE MAKING SYSTEM:

The formal elaboration of the Petri Net model for the processing system of the pot cake making machine was meticulously constructed by applying several specific parameters. These parameters were carefully selected in order to segment the process, making it possible to obtain the corresponding operating diagram. These parameters are defined as follows:

Revista ARACÊ
ISSN: 2358-2472

Table 1: Definition of structures and parameters of the petri net

Structures	Parameters
Set of Places (P)	P={"Start","Filling?","Frosting?","Confirm order?","restart system","Payment method?","Confirm payment?","Cake making","feed process","turn on system","pot with filling","pot with frosting","cake ready","off"}
Transition Set (T)	T={"IHM activates", "brigadeiro", "coconut", "yes", "no", "pix", "money", "debit", "credit", "no", "yes", "start", "start", "add filling", "add topping", "finish cake", "turn off system", "new customer", "restart system"}
Arch Set	A={("start","HMI triggers"),("HMI triggers","Filling?"),("Filling?","brigadeiro"),("Filling?","coconut"),...}
Initial Marks	M("start") = 1
Incidence Function	I : (P×T)∪(T×P)→N I("start", "HMI triggers") = 1 I("HMI activates", "Filling?") = 1

INTERPRETATIONS OF THE PETRI NET

Marks in Places: Marks in places represent the current state of the system.

Transitions: Transitions indicate events or actions that can occur when the entrance places have enough tags.

Arches: Arches connect places and transitions, indicating the influence of one on the other.

EXECUTIONS IN THE PETRI NET

For the effective implementation of the system, it was decided to outline a comprehensive structure of cases, thus covering the various stages inherent to the process in question. These steps were meticulously defined, aiming at a complete and efficient coverage of development. In this way, specific categories were established that make up the structure, encompassing crucial elements for the fluid and successful execution of the proposed system. Being defined as:

Chart 2: Definition of the steps that can occur in each case.

Cases	Steps
Success Story	- Initial mark in "beginning". - "HMI triggers" transition is triggered. - Brand is moved to "Stuffing?". - Continue the flow until it reaches the "cake ready" state.
Failure Case	- Attempt to perform an action outside of the standard flow. - Matching transition is not triggered. - Leads to a state of blocking.
Restart Case	- "System restart" transition is triggered. - Brings the system back to its initial state. - Allows you to simulate the behavior of the system in a visual and understandable way.

ANALYSIS AND CLASSIFICATION OF THE DEVELOPED PETRI NET:

The structure of the Petri Net developed is composed of places, transitions and arcs, offering the ability to model the behavior of the system in question. Regarding the properties of positions (places), it is relevant to consider aspects such as the presence or absence of markings, and a place can be marked (containing a mark) or not marked. Initially, it is highlighted that the place "beginning" is marked. As for the temporality of the markings, there is no explicit information that indicates whether they are temporary or permanent. This characteristic will depend on the specific rules of the modeled system, which were not detailed in the description.

In the context of transitions, it is crucial to understand whether they are enabled or not. A transition is considered enabled when all entry places are marked. As an example, the "HMI triggers" transition is enabled only when the "start" place is checked. Additionally, when analyzing the properties of the transitions, it is observed that, based on the description provided, the transitions seem to follow a deterministic pattern. This means that for a given configuration, there is only one possible transition.

In the context of global properties, brand conservation is mentioned as a common feature in Petri nets. However, in the description there is specific information about how the Petri Net occurs, which is characterized as a directed, ordinary, pure, state machine circuit. This particular property of the network, known to be conservative, reinforces that the total sum of marks everywhere remains constant during execution, remaining secure and limited to k=1. In addition, the Petri net presents confusion-type conflicts, an aspect inherent to its nature of conservation. Regarding additional observations, it is emphasized that the Petri Net model presented is relatively simple and can be expanded or refined as needed, according to the specific details of the system being modeled.

PETRI NET LOGICS FOR THE CAKE MAKING SYSTEM:

In figure 2, we can visualize a Petri net, designed specifically to solve the problem in question and have a basis for how the process of making and interacting the cake machine with the user. The structure of the Petri net presented is composed of several points, each representing a specific state of the system. Initially, we have point p1, which corresponds to the "start" state. Then, the network goes to points p2 and p3, representing the choices about the cake's "Filling?" and "Topping?" respectively. The process progresses to p4, where the user must "Confirm order?". In case of need to restart the system, point p5 is triggered. The choice of "Payment method?" is represented by p6, followed by the transitions to t8, t9, t10 and t11, which correspond to the payment options via pix, cash, debit or credit, respectively. The process proceeds to p7, where the user must "Confirm

Revista ARACÊ
ISSN: 2358-2472

payment?". Point p8 marks the "Cake Making" phase, fed by the process in p9 and started with the system being connected to p10. The intermediate states include the presence of the pots with filling (p11) and frosting (p12), culminating in the finished cake in p13. Upon completion, the system is shut down at p14.

Transitions t14 to t17 indicate the flow of adding filling and frosting, leading to the cake being finished at t18. The system is turned off at t19, returning to the p1 starting point. Point p20, representing the arrival of a new customer, is connected to p1, while p5 and t21 make it possible to reboot the system.

This structure is visualized through the arcs that connect the different points and transitions. For example, the arc from p1 to t1 indicates that the HMI triggers the transition t1 when the system is in the "start" state. Similarly, subsequent arcs reflect the directed relationships between places and transitions, defining how the states of the system are influenced by transitions and vice versa. This representation in the Petri net offers a clear view of the dynamics of the system, evidencing the interactions between the different elements involved.

STRUCTURES OF THE FINITE AUTOMATA CAKE MAKING SYSTEM:

To model the system using a discrete automaton, we opted for a deterministic finite automaton (DFA) or a non-deterministic automaton (NFA), depending on the complexity of the system. The automaton chosen for the development of the project was the non-deterministic finite (NFA), and its structure was defined as:

Table 3: Structure and definitions for finite automata.

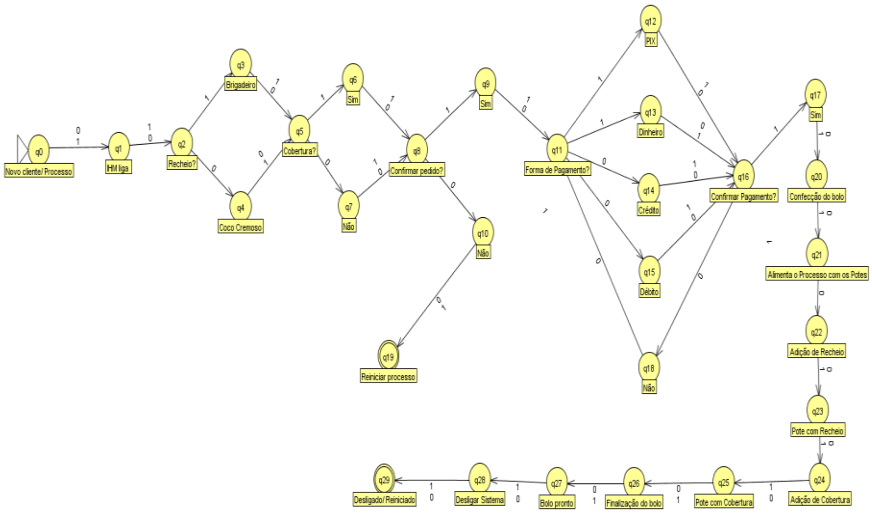
State Set (Q)	Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, Q20, Q21, Q22, Q23, Q24, Q25, Q26, Q27, Q28, Q29
Input Alphabet (Σ)	0, 1
Set of Initial States	q0 (New Customer/Process)
Final States Set	q19 (Restart Process), q29 (Off/Restarted)

The automaton models the flow of the system from the initial state ("start") to the desired final states ("cake ready" and "off"). It reflects the dynamics of the cake-making process, covering the steps of choosing fillings, toppings, order confirmation, choosing the payment method, and finalizing the cake. For its simulations, the JFLAP program was chosen, because its use offers advantages due to its intuitive graphical interface and features that facilitate the construction, visualization and simulation of automata.

Revista ARACÊ
ISSN: 2358-2472

JFLAP simplifies the AFD design process by allowing users to easily model, edit, and analyze automaton transitions and states. In addition, the software provides validation and simulation functionalities, helping to detect errors and understand AFD's behavior. These characteristics make JFLAP an effective tool for teaching and learning computational theory, contributing to a practical and visual understanding of concepts related to deterministic finite automata.

Figure 5. Automaton Scheme (AFN)



Source: Image by the author

Figure 4 represents the model obtained through the JFLAP program, after its preparation and simulation process.

ANALYSIS OF THE LANGUAGE INTERPRETED BY THE AUTOMATON

NFA, or Non-Deterministic Finite Automaton, is a theoretical model used in computation theory to represent computational systems and formal language recognition processes. Unlike deterministic finite automata (DFAs), in which the transition from one state to another is deterministic, in an NFA, a state can have multiple possible transitions to different inputs or to the same input. In the developed project, a possible sequence could be "HMI activates, yes, brigadeiro, confirm order, pix, cake ready, off", this sequence of events would take the system through the relevant states of the automaton. For a better example of the cases, some situations with different strings were presented.

Revista ARACÊ
ISSN: 2358-2472

Table 4: Examples of valid and rejected combinations of the system in NFA.

Strings	Steps
Accepted String	HMI drives (1.0), filling? (1), brigadeiro (1.0), topping? (1), yes(1,0), Confirm Order(1), pix(1), Confirm Payment? (1),... Cake Ready(1.0), Off
String Rejeitada	new customer (1.0), Recheio? (1), brigadeiro (1.0), topping? (1), yes(1,0), Confirm Order(1), pix(1), Confirm Payment? (0)

This sequence passes through the relevant states validly, resulting in the final "Off/Restarted" state. For this to be valid, it is necessary by restriction of this element, so the seventh and eleventh elements of the string must be 1, in order to make the cake. If the seventh element is 0, the string can have 8 elements, for every time element 11 is 0, the string must have two more elements in the string. In addition, the string must have a minimum of 21 elements to end the process. The rejected String is invalid, even though the sequence follows the correct process flow, however, it does not have enough string elements to end the process. The system cannot be shut down by a new customer without going through the appropriate stages.

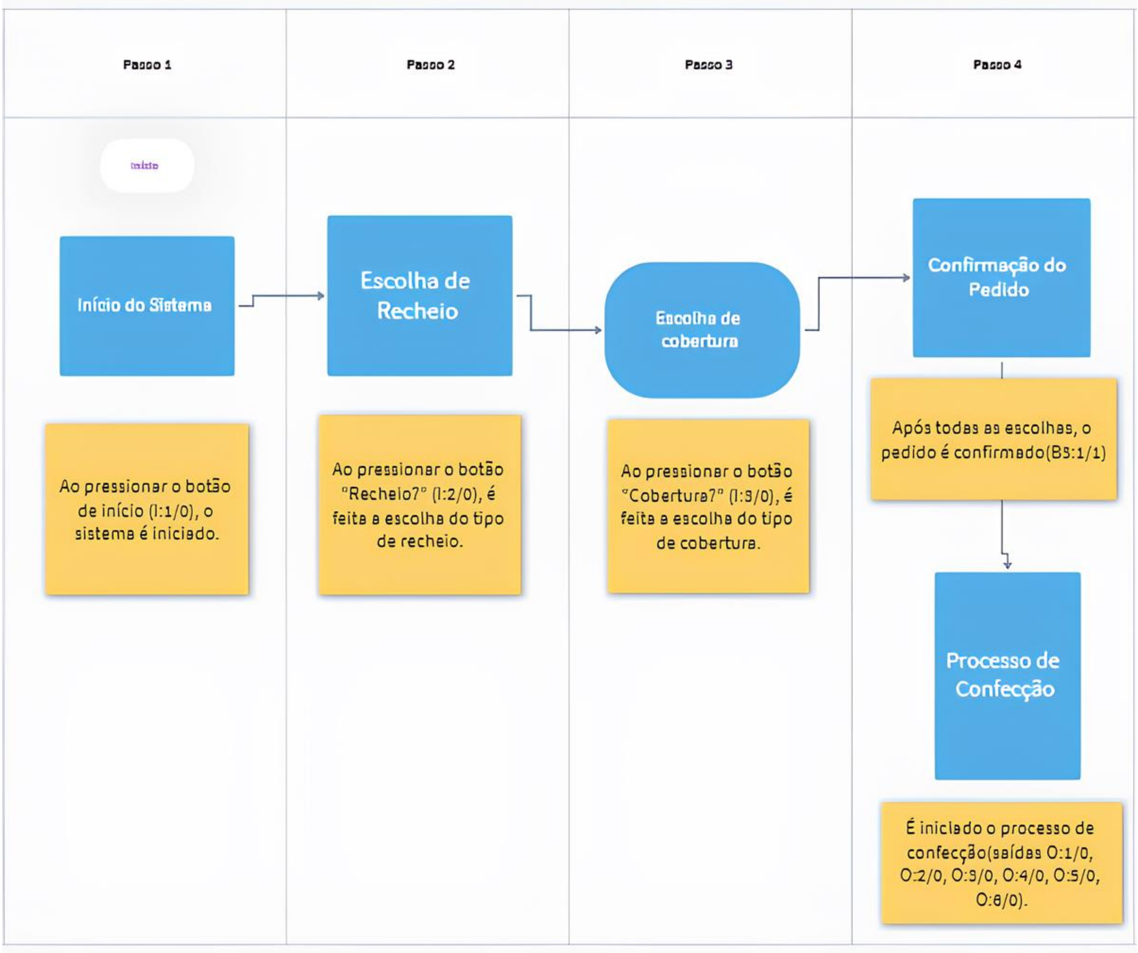
LADDER LANGUAGE STRUCTURES FOR THE CAKE MACHINE SYSTEM

The Ladder programming language (LADDER Logic) plays a crucial role in programmable logic controllers (PLCs) used for industrial automation. However, when trying to translate Petri Net or Automata models directly into Ladder, one is faced with the complexity arising from the different conceptual approaches between these methodologies. In this context, we seek to develop a simplified Ladder representation that encapsulates some key elements from Petri modeling.

The direct translation between these models may encounter limitations, given that the Ladder is oriented to the control logic, while the Petri Nets are more appropriate for process modeling. The correlation between both will therefore be conceptual, and the actual implementation will depend on the specific control logic and instructions available in the Ladder environment used. Simplified Ladder Control Logic for the proposed system:

ISSN: 2358-2472

Figure 6. System structure flow chart.



Source: Image by the author

The buttons for choosing filling and topping are examples of manual inputs, which can be sensors, buttons, or inputs from other devices in the real environment. The logic presented here is simplified and does not contemplate all the specific conditions or details of the system.

PROPOSALS FOR IMPROVEMENTS TO THE SYSTEM

One proposal to significantly improve the system in question involves a detailed and expansive analysis of order management, with a special focus on the payment process. The modifications suggested below seek to make the modeling more comprehensive and efficient, providing a clearer and more detailed view of the workflow:

1. Inclusion of a Detailed Payment Subprocess: Enhance the representation of the payment process by adding specific places and transitions. This would include differentiating between payment options, such as credit card, debit card, or pix. This

approach will allow for a more accurate and granular understanding of the payment flow.

2. Introducing Multiple Payment States: Expand the modeling by creating additional places that represent different states of the payment process. Examples include "Waiting for Card Confirmation" or "Waiting for Payment by Pix". This inclusion will provide a more comprehensive view of the status of each transaction, making it easier to identify potential points for improvement.
3. Adding Visual Feedback to Diagram: Improving diagram clarity by including visual feedback to indicate specific system states. The suggestion is to use colors to highlight successful transitions, wait states, or end states. This approach will provide a quick and intuitive understanding of the status of the system.
4. Identification of Possible Bottlenecks: perform a thorough analysis of the modeling in search of possible bottlenecks in the process. Additionally, it provides information about runtimes or other relevant parameters that may indicate specific areas for improvement. This approach aims to optimize the efficiency of the system.
5. Consideration of Competition: if the system involves the simultaneous handling of several orders, it is recommended to include elements that represent competition. This could include, for example, making several cakes at the same time. This consideration is crucial for systems that handle multiple simultaneous operations. The implementation of these modifications in Petri Net modeling will depend on the specific details of the system in question and the associated functional requirements. The importance of adapting suggestions according to the complexity and objectives of the system being modeled is emphasized, ensuring a personalized and effective approach.

FINAL CONSIDERATIONS

After the implementation of the Petri net system and finite automata, a significant improvement in the bakery's operational performance was observed. Previously, the total production time for a pot cake was approximately 8 minutes, taking into account the transport and pouring of the cream and filling, and the user interaction machine. With the new strategies implemented, the operating time was reduced to 5 minutes and 40 seconds, as a result of the efficiency improvement in managing the machine's ingredient addition processes and in the process of acquiring the customer's order, resulting in remarkable

Revista ARACÊ
ISSN: 2358-2472

efficiency, which directly impacted the machine's kw/h consumption and the unnecessary use of materials.

IMPROVED EFFICIENCY CALCULATION

Machine Power = 1500W

Price Kw/h = 0,80c

Before

8min = 0.13h

$$0{,}13h\ x\ 1500\text{W} = 195W/Cicle$$

Duty Cycle = 195W

$$195\frac{\text{W}}{\text{C}}\ x\ 80\text{cents KW}/h = 156cents$$

After

5:40 min = 0.09h

$$0{,}13h\ x\ 1500\text{W} = 135W/Cicle$$

Duty Cycle = 135W

$$135\frac{\text{W}}{\text{C}}\ x\ 80\text{cents KW}/h = 108cents$$

This optimization had a direct impact on daily production, allowing for the manufacture of more cakes in a shorter period of time. Before, the process took advantage of a period of 16 hours a day, totaling approximately 4 hours of retained production. Now, a greater quantity of cakes can be produced in a reduced time of 3 hours, freeing up an additional 4 hours of production time, reducing the operating cost generated by downtime. In addition, the customer's interaction with the machine has been improved, providing a more fluid and efficient experience. These results highlight the effectiveness of the implementation of the Petri net system and finite automata in optimizing the production processes in a cake making and selling machine, evidencing significant gains in terms of time and operational efficiency.

Before the system application:

$$30\ days \times \frac{16h}{dia} = 480\ by\ month$$

After the system is applied:

$$30\ days \times 20h/day = 600h/month$$

ACKNOWLEDGMENT

This article is the result of the Research and Development (R&D) project aranouá, which is funded by Samsung, using resources from the Information Technology Law for the Western Amazon (Federal Law No. 8,387/1991), and its disclosure is in accordance with the provisions of article 39 of Decree No. 10,521/2020.

Revista ARACÊ
ISSN: 2358-2472

REFERENCES

1. Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). Automata theory, languages, and computation (International Edition). Boston: Addison-Wesley. ISBN 978-0321455369.

2. Sena, L. de A., Miyagawa, M., & Silva Júnior, W. S. da. (2013). Sistema de venda automática utilizando pagamento eletrônico através de dispositivos móveis. In Congresso Nacional de Iniciação Científica - CONIC-SEMESP, 1, São Paulo. Anais (v. 1). São Paulo: CONIC-Semesp. Disponível em: https://conic-semesp.org.br/anais/files/2013/trabalho-1000015410.pdf. Acesso em: 01 julho 2024.

3. Franco, T. A. (2007). Projeto para mantenabilidade de máquinas automáticas (Tese de doutorado, Escola Politécnica, Universidade de São Paulo, São Paulo). Disponível em: https://www.teses.usp.br/teses/disponiveis/3/3151/tde-09032023-084553/publico/TiagoAlmeidaFrancoOrig.pdf. Acesso em: 07 julho 2024.

4. Barros, T. de C., et al. (2000). Supervisão de sistemas de produção baseada em redes de Petri. Relatório técnico. [S.l.]. Disponível em: https://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/3461. Acesso em: 15 julho 2024.

5. Scheffler, J. B., & Silva, M. L. V. da. (2021). Sistema de automação industrial para dosagem e mistura para sólidos granulados. [S.l.: s.n.]. Disponível em: https://www.publicacoeseventos.unijui.edu.br/index.php/salaoconhecimento/article/view/20555. Acesso em: 17 julho 2024.