Article
Journal Industrial Engineering
December 2022, Vol.6 No.17 8-12

Identification of the human-machine interaction process through the generation of a grammar based on automata theory, by means of a practical case

Identificación del proceso de interacción hombre-máquina a través de la generación de una gramática basada en teoría de autómatas, mediante un caso práctico

BENÍTEZ-QUECHA, Claribel†*, ALTAMIRANO-CABRERA, Marisol, MÉNDEZ-LÓPEZ, Minerva Donají and SANTIAGO-APARICIO, Nallely Elizabeth

Tecnológico Nacional de México, Campus Oaxaca, México.

ID 1st Author: *Claribel, Benítez-Quecha* / **ORC ID**: 0000-0001-6516-5760, **CVU CONACYT ID:** 657582

ID 1st Co-author: *Marisol, Altamirano-Cabrera* / **ORC ID:** 0000-0001-5800-9655, **CVU CONACYT ID:** 657390

ID 2nd Co-author: *Minerva Donají, Méndez-López* / **ORC ID:** 0000-0001-5336-1772

ID 3rd Co-author: *Nallely Elizabeth, Santiago-Aparicio* / **ORC ID:** 0000-0001-6142-9626

DOI: 10.35429/JIE.2022.17.6.8.12 Received July 15, 2022; Accepted December 30, 2022

Abstract

This project shows how the Automata Theory is a viable option for the process of defining the inputs and outputs that allow the design of a human-machine interface (HMI). In this case, it is applied to the design of a device that allows the "intelligent" control of a traffic light. For this purpose, the work is divided into an Introduction, a Theoretical Framework, a description of the Development, Results, and Conclusions. Objectives. General: To apply the automata theory to identify the input processes in a human-machine interface. Specific: Identification of the functionalities of the device, Formulation of Alphabet, Language, and Grammar, Creation of the automata. Methodology: The methodology followed is that of the Automata Theory, which defines the points mentioned in the specific objectives of this text. Contribution: To show how to apply Automata Theory in situations different from the development of compilers, as in this case to a process of implementation of the interface for human-machine interaction.

Grammar, Interface, Process

Resumen

En el presente proyecto se muestra la forma en cómo la Teoría de Autómatas es una opción viable para el proceso de definición de las entradas y salidas que permitan diseñar una interfaz hombre-máquina (HMI). En este caso se aplica al diseño de un dispositivo que permite controlar de forma ¨inteligente¨ un semáforo. Para tal efecto se divide el trabajo en una Introducción, un Marco Teórico, en la Identificación de Funcionalidades, Creación de la Gramática, Generación de los Autómatas, Resultados y Conclusiones. Objetivos. General. Aplicar la teoría de autómatas para identificar los procesos de entrada en una interfaz hombre-máquina. Específicos. Identificación de las funcionalidades del dispositivo, Formulación de: Alfabeto, Lenguaje y Gramática, Creación de los autómatas. Metodología. La metodología seguida es la de la Teoría de Autómatas, que define los puntos mencionados en la parte de objetivos específicos, de este texto. Contribución. Mostrar cómo aplicar Teoría de Autómatas, en situaciones diferentes al desarrollo de compiladores, como en este caso a un proceso de implementación de la interfaz para una interacción hombre-máquina.

Gramática, Interfaz, Proceso

Citation: BENÍTEZ-QUECHA, Claribel, ALTAMIRANO-CABRERA, Marisol, MÉNDEZ-LÓPEZ, Minerva Donají and SANTIAGO-APARICIO, Nallely Elizabeth. Identification of the human-machine interaction process through the generation of a grammar based on automata theory, by means of a practical case. Journal Industrial Engineering. 2022. 6-17:8-12.

* Author's Correspondence (E-mail: claribel.bq@oaxaca.tecnm.mx)
† Researcher contributing as first author.

Introduction

In the field of engineering, the human-machine interface (HMI) is a highly demanded process nowadays, since there are more and more devices that allow automating daily life processes through technological environments where a certain response is expected from a mechanism or device to an action provided by man.

However, these interfaces are not totally manipulated by software, but have a high programming content at the hardware level or electronic functions. Therefore, software development methodologies are not as useful.

For a human-machine interface the communication is done with programmable logic controllers (PLC) and input/output sensors that allow obtaining and displaying information based on which a response is generated through technological means. These systems can be used for a variety of technological events, from simple ones such as monitoring or tracking to intelligent ones, to make decisions based on information gathered by sensors depending on how they are implemented.

So, it is necessary to specify unambiguous, mechanical responses to commands that may come from another machine or from man.

It is here where the usefulness of the Automata Theory is considered, since it defines in a methodical way a series of actions that end in a response or action (known as acceptance state), through a sequence of valid events.

Process that can be easily configured in PLCs, such as arduinos, or other models of electronic controllers.

Theoretical framework

HMI is the abbreviation for Human Machine Interface. An HMI system is basically the interface that allows a person to interact with a machine, system or device. It is also identified as a control panel. These systems allow to obtain operational information to carry out a process. It transmits information, data and commands between man and machine, and vice versa.

The theory of Automata was developed to define the sequence of actions to follow for an interaction between devices that "speak" different languages.

It is the study of abstract computational devices, i.e., of "machines" (Hopcroft).

It allows defining an interface based on areas of knowledge such as mathematics, linguistics, machine theory, among others.

The software developed from this theory is a compiler, which validates a common language between different devices.

Identification of functionalities

In the field of Systems Engineering, one of the basic disciplines is Systems Analysis, since it is essential to understand what we want to automate, so the first point to develop is the establishment of the functionalities of the system to be created.

This system is an intelligent traffic light, which allows the operation of a pair of traffic lights that must detect the presence or absence of cars, to manage the passage of the same in an intelligent way, that is if a traffic light is on red light but there are no cars formed, it checks if there are cars waiting for the pass in the lane controlled by the traffic light that is red. If so, it should turn green the traffic light that is red to allow cars to pass and turn red the traffic light that is green, but has no cars waiting to pass.

The solution to be implemented is that the traffic light, through the sensors, identifies if there are cars in a certain lane, if there are not, then it allows the passage to the lane that has cars waiting, turning it to green light in case it is on red. And turning the empty lane to red light.

Three states of acceptance are identified in the operation of the traffic light:

Green light, yellow light and red light.

Journal Industrial Engineering
December 2022, Vol.6 No.17 8-12

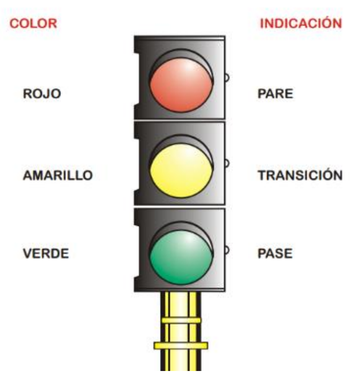

Figure 1 Acceptance states
Source: Own source

The control variable that allows detonating the defined states is the sensor, which has for this prototype, an approach value of the objects from 1 to 49cms.

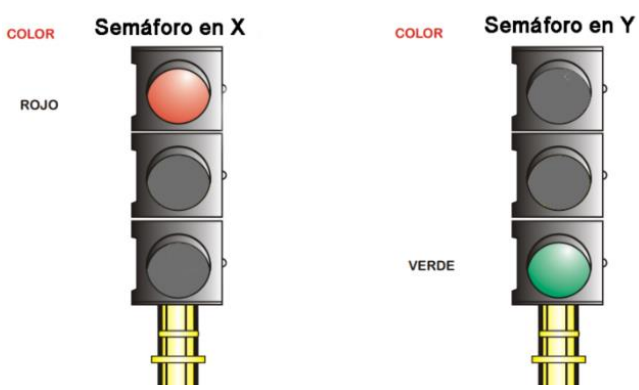

Figure 2 Identification of traffic lights
Source: Own source

The identified events (E1, E2, E3), which lead to the acceptance states are the following:

E1. The sensor checks if there are cars waiting in a lane with red light while in the other lane with green light there are no cars (Fig. 3).

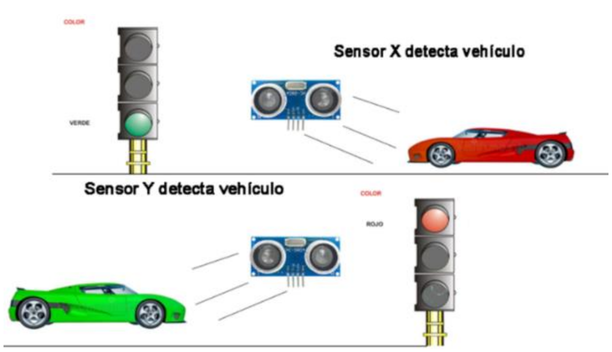

Figure 3 Event 1 (E1)
Source: Own source

E2. In the case of Event 1, the HMI modifies the lights in the traffic lights, the traffic light at X becomes green and the traffic light at Y becomes red (Fig. 4).

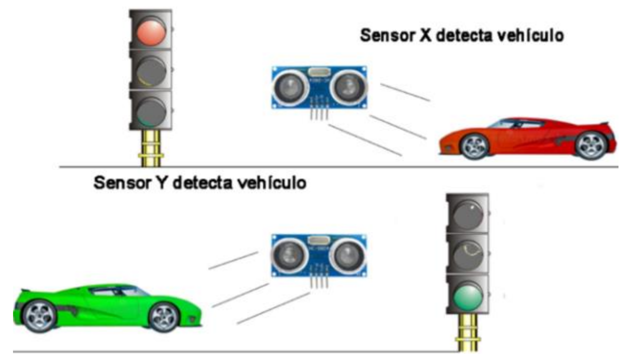

Figure 4 Event 2 (E2)
Source: Own source

E3.- Yellow light on at traffic light Y and red light on at traffic light X.

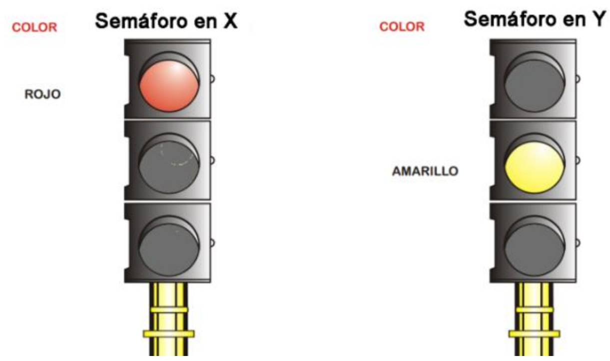

Figure 5 Event 3 (E3)
Source: Own source

From the above it is defined:

INPUTS.

Approximation of a car in X.

Approximation of a car in Y.

ACCEPTANCE STATUS (results of inputs).

Green light on from traffic light X.

Yellow light on from traffic light X.

Red light on from traffic light X.

Traffic light green light on Y.

Yellow light on for traffic light Y.

Red light on for traffic light Y.

ISSN 2523-0344

Creation of the grammar

General alphabet

$\sum_1$ = {Números enteros}

Language = {dx_1, dx_2, dy_1, dy_2}

dx_1 € dx_1 = [1cm - 49 cm] → Green light on from traffic light X and red light on from traffic light Y.

dy_1 € dy_1 = [1cm - 49 cm] → Green light on for traffic light Y and red light on for traffic light X.

dx_2 € dx_2 = [50cm - 99cm] → Yellow light on for traffic light X and red light on for traffic light Y.

dy_2 € dy_2 = [50cm – 99cm] → Yellow light on for traffic light Y and red light on for traffic light X.

Production rules

dx_1 → [1-4][0-9] | [1-9] (1)

dy_1 → [1-4][0-9] | [1-9] (2)

dx_2 → [5-9][0-9] (3)

dy_2 → [5-9][0-9] (4)

Generation of automata

Event 1 (E1)

Acceptance states = Green light on for traffic light X and red light on for traffic light Y.

Grammar

$G_1 = (\Sigma_1, N, S, P)$ (5)

N = {q0, q1, q2, q3}

S = q0

P = dx_1 → [1-4][0-9] | [1-9] (6)

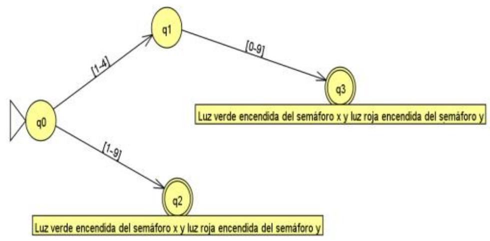

Figure 6 Event 1 automaton

Event 2 (E2)

Acceptance states = green light on for traffic light Y and red light on for traffic light X.

Grammar

$G_1 = (\Sigma_1, N, S, P)$

N = {q0, q1, q2, q3}

S = q0

P = dy_1 → [1-4][0-9] | [1-9] (7)

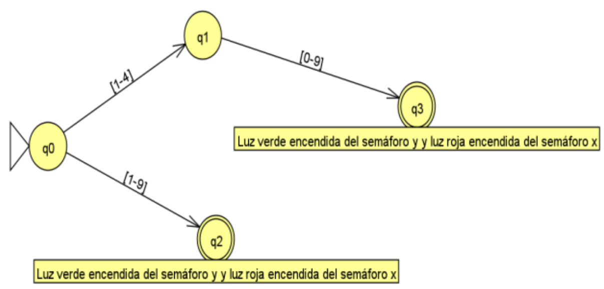

Figure 7 Event 2 automaton

Event 3 (E3)

Acceptance states = Yellow light on for traffic light Y and red light on for traffic light X.

Grammar

G1 = (Σ1, N, S, P)

N = {q0, q1, q2}

S = q0

P = dy2 → [5-9][0-9] (8)

ISSN 2523-0344

December 2022, Vol.6 No.17 8-12

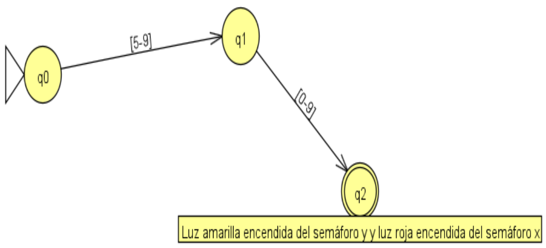

Figure 8 Automaton Event 3

The next step would be the coding itself. Which is done in the PLC (using Arduino) and sensors.

Results

As we can see, through the use of the Automata Theory, it was possible to specify precisely the events to be considered in an HMI interface.

Since there is no intensive use of software to program PLC's, but only basic code to generate an action through a given input, the relevant thing was to identify the sequence of events to obtain a specific response.

Therefore, it was not necessary to carry out all the actions of the analysis phase of the software development methodologies.

Acknowledgement

We thank the Tecnológico Nacional de México Campus Oaxaca, for their support for the realization of this work.

Conclusions

When creating interfaces for manipulation of technological devices, without the use of a computer, but through microcontrollers (PLC), the use of methodologies for software development is not so useful. Since they are simpler processes, where the number of actions is smaller than in an administrative type software.

Since what needs to be defined is a sequence of events that lead to an answer, the Automata Theory is a much more useful tool. The Automata Theory, the models associated with it, as well as the algorithms inherent to the compilers, can be very useful in a variety of software development, including, as we can see in this project, in software for programming devices in HMI interfaces, regardless of the source language or the target machine.

References

Aho, Alfred V. Compiladores. Editorial Pearson. 2011.

Habibi, Eddie. Pérez, Héctor. The High Performance HMI Handbook. Editorial PAS. 2014.

Hopcroft, John E. Motwani, Rajeev. Ullman, Jeffrey D. Teoría de Autómatas, Lenguajes y Computación. Editorial Pearson Addison-Wesley. 2007.

Fiset, Jean-Yves. Human Machine Interface Design for Process Control Applications. Editorial ISA (Instrumentation, Systems and Automation Society). 2009.

ISSN 2523-0344