Use of the Specification and Description Language SDL and SDLPS to Define and Simulate Complex Systems. Case Study: Spread of COVID-19 in Ecuador.

1Pamela Gómez Martínez, 2Paúl Freire Jaramillo, 3Ramiro Delgado Rodríguez, 4José Luis Marcillo Parra, 5Roberto López López

1Universidad de las Fuerzas Armadas ESPE, Ecuador
Email: pcgomez1@espe.edu.ec
ORCID: 0000-0003-2448-3257
2Universidad de las Fuerzas Armadas ESPE, Ecuador
Email: pafreire3@espe.edu.ec
ORCID: 0009-0009-6330-2122
3Universidad de las Fuerzas Armadas ESPE, Ecuador
Email: rndelgado@espe.edu.ec
ORCID: 0000-0003-2549-1486
4Universidad de las Fuerzas Armadas ESPE, Ecuador
Email: jlmarcillo@espe.edu.ec
ORCID: 0009-0001-8578-8321
5Universidad Regional Autónoma de los Andes (UNIANDES), Ecuador
Email: capostgrado@uniandes.edu.ec
ORCID: 0000-0003-0712-7496

Abstract. The Specification and Description Language (SDL) is a formal object oriented language for distributed systems which is widely used for the simulation of complex systems such as propagation of epidemics like COVID-19. SDLPS is a simulation tool that is used for testing complex systems implemented with SDL. Cellular Automaton is a mathematical and computational model applied to dynamic systems that evolve in discrete steps; and Intelligent Agents that are entities capable of perceiving an environment, process or perceptions and respond in a rational manner. The objective of this investigation was simulate the COVID-19 behavior in Ecuador based in SDL language, SDLPS tools and SEIR models using the real data published by the Ministry of Public Health of Ecuador from March 2020 to December 2021. The model for Ecuador was based on the model developed by Dr. Pau Fonseca i Casas and configured to responds to the specific variables of Ecuador. The main conclusion was that SDL and SDLPS are very important tolls to develop and simulate complex systems like propagation of epidemics. It was necessary to validate this investigation with others case studies.

Keywords: Specification and Description Language, SDLPS, COVID-19 Ecuador

1 Introduction

In the 1970s, appeared the need to create a language that would be able to fulfill two specific purposes:

- Accurately specify the functional characteristics of a system.
- Precisely, describe the functional characteristics provided by the existing systems.

This is how the Specification and Description Language (SDL) emerged, which is widely used in the area of telecommunications or the simulation of complex systems such as the propagation of epidemics; it's supported by numerous tools and its development started from the model of finite state machine used to design electromagnetic exchanges. On the other hand, SDLPS is a simulation tool that is used for testing complex systems implemented with this language, especially complex systems.

1.1 Specification and Description Language (SDL)

It's a formal object-oriented language for distributed systems, defined by the International Telecommunication Union – Telecommunication Standardization Sector (UIT-T) [1][2] that includes the description of the behavior, structure and data of a system. The language enables the specification and design of complex interactive applications, managed by events, in real time, that involve many simultaneous activities using discrete signals for communication. [3]

The definition of the model is based on five components:

1. Structure: system, blocks, processes and process hierarchy.
2. Behavior: defined through the different processes.
3. Data: based on abstract data types (ADT).
4. Communication: signals, with established parameters and channels.
5. Inheritance: describe the relationships between the elements of the model.

The language has 4 levels (i) System, (ii) Blocks, (iii) Processes and (iv) Procedure (See Fig. 1)

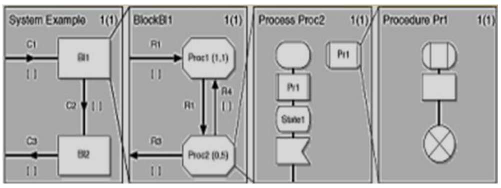

Fig.1. SDL levels. Retrieved from: https://www-eio.upc.es/~pau/cms/cmsfiles/Papers/2008%20-%20WSC%20(poster).pdf

The representation of the SDL formalization of the model is done using XML which simplifies the management of the language, structures and their transformation and manipulation in the infrastructure. [4].

1.2 SDLPS Tool

SDLPS is a distributed simulator that allows the definition of models using SDL language that facilitates their simulation without the need to implement the model, simplifying the validation and verification processes. SDLPS is designed to capture events and process them. This representation of the model is not only useful for communication purposes, but also simplifies the validation process. As stated by [5] "The verification of the computational model ensures that the computer programming and implementation of the conceptual model are correct. SDLPS is a distributed simulator that allows the definition of models using SDL language that facilitates their simulation without the need to implement the model, simplifying the validation and verification processes. SDLPS is designed to capture events and process them. This representation of the model is not only useful for communication purposes, but also simplifies the validation process. As stated by [5] "Computer model verification ensures that the computer programming and implementation of the conceptual model are correct. The main factor affecting verification is whether a simulation language or a higher-level programming language such as FORTRAN, C, or C++ is used, will generally result in having fewer errors than if a higher-level programming language of general use is used."

SDLPS is implemented in C++ and allows the distributed execution of different SDL blocks or processes on different machines. Each block uses a port and several input and output channels with which it communicates with the other blocks in the model [1]. Each block is assigned a machine with a specific IP and ports that receive or send signals from its origin or destination (See Fig. 2).

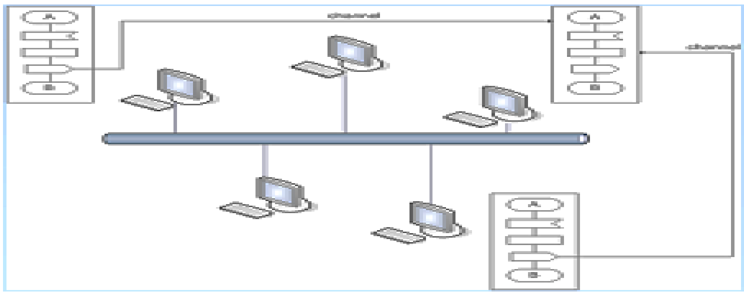

Fig. 2 Example of SDPLS Architecture. Retrieved from: https://www-eio.upc.es/~pau/cms/cmsfiles/Papers/2008%20-%20WSC%20(poster).pdf

When a process block receives a signal, the execution of the model initiates. Since the code embedded in the SDL blocks depends on the model, it must be compiled once the model is defined. This compilation generates an SDLCode.dll file that contains the methods and data necessary for the execution of the model; and, it is the one that is replicated in each machine.

1.3 COVID-19

A pandemic is an event that marks the history of mankind and generates great changes to commercial, educational, personal and family activities. Coronaviruses are a large family of viruses that can be transmitted from animals to humans; they can produce clinical pictures that go from the common cold to pulmonary diseases that develop into very serious clinical pictures. Significant examples are the severe acute respiratory syndrome virus (SARS-CoV) and the coronavirus cause of the respiratory syndrome in Middle East (MERS-CoV). The new coronavirus called SARS-CoV-2 or COVID-19 which was a type of virus first recognized in December 2019 in Wu-han City, Hubei Province, China.

An essential process in a pandemic is the study and prognosis of the number of infected and dead cases that could be presented, as well as the strategies and actions to be taken by governments to minimize its effect and control the disease. In this scenario, data collection, data analysis of themselves, formulation of a model and its simulation are very important for decision making.

On 11/03/2020 the WHO declared COVID-19 as a global pandemic; in Ecuador, through ministerial agreement No 00126-2020 released on 11/03/2020, a National Health Emergency State was declared.

2 Materials and Methods

Within complex systems, many of the problems can be solved using the SDL language because in most of them it is necessary to establish the characteristics of the system, the limits of the scope and the boundaries. For example, in a complex electronic model such as the behavior of a wave in certain situations, the SDL language needs the parameters governing the physics in order to obtain an accurate result. On the contrary, when analyzing the spread of COVID-19 in terms of confirmed cases of infection and deaths, as well as the measures applied, other aspects must be taken into account, such as non-pharmaceutical measures established to stop the spread. This is why each model using the SDL language is unique [6][7].

2.1 Cellular Automaton

It is a mathematical and computational model applied to dynamic systems that evolve in discrete steps. It is suitable for modeling natural systems that can be described as a massive collection of simple objects that interact locally with each other. [8]

2.2 Intelligent Agents

They are entities capable of perceiving their environment, process such perceptions and respond or act in their environment in a rational manner, in other words, in a correct way and tending to maximize an expected result.

2.3 Statistical Research Methodology

Operations Research consists of the use of mathematical and statistical models and algorithms to carry out a process of decision making, taking into account the lack of resources and determining how to optimize a defined objective such as profit maximization or cost minimization. [9] The statistical method implements a sequence of procedures for handling qualitative and quantitative research data and has the following stages: collection, counting, presentation, synthesis, and analysis. [10][11]

2.4 SDL Modeling Methodology

There are two versions of the SDL. The rigid version and the one oriented to agile development. The rigid version is appropriate for development teams and bigger projects that are not susceptible to changes during the process. The agile-oriented version develops the product incrementally and considers the frequency of the execution of the activities for security assurance, it is recommended for web or web-based application development. [12]

2.5 SEIR Models

One of the applications of the SDL language is to determine the behavior of pandemics in a period of time considering certain parameters, these models applied to epidemiology are called SEIR that consider the following parameters:

- S: susceptible population
- E: exposed population
- I: infected population
- R: eliminated population

The functions of the SEIR models are well known and allow the analysis of several scenarios; among them, what happens when no containment measures are applied, waiting for the entire population to become infected and the end of the pandemic. In order to analyze this case, two fundamental parameters were not available: the time necessary for the curve to evolve and the height that the curve may reach, which is a very important parameter since the curve may eventually collapse. In the case of COVID-19, these parameters were not exactly known at the beginning, so several simulations were required (See Fig. 3).

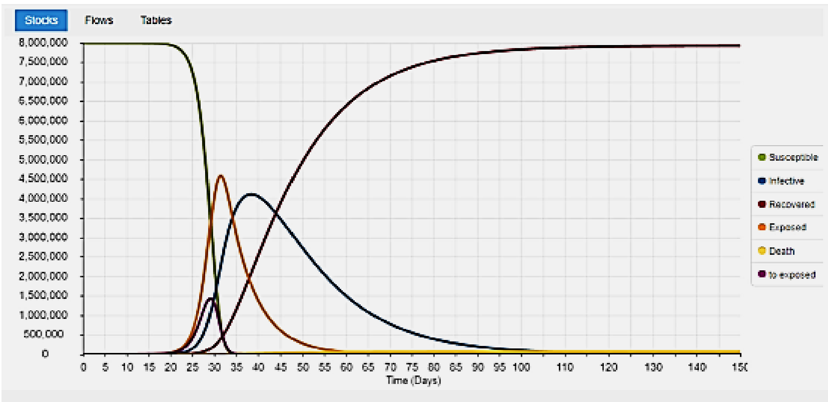

Fig. 3. Preliminary values for the System Dynamics SEIRD model. Retrieved from: https://doi.org/10.3390/app10155162

In a pandemic, it is not desirable to saturate the health system, so measures such as confinement are applied and this causes the curve to have a totally different behavior (see Fig 4).

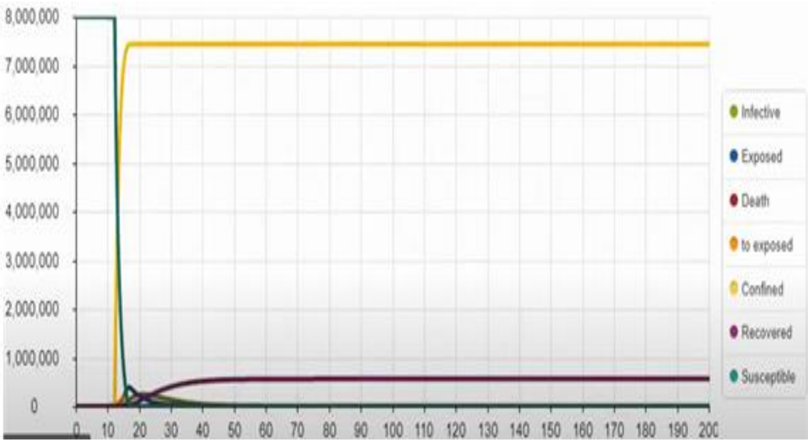

Fig. 4. Preliminary values for the System Dynamics SEIRD model. Retrieved from: https://doi.org/10.3390/app10155162

To obtain the curves in Fig 4, very complex variables to measure were considered, which required the application of a complex system supported by the SDL language and the SDLPS tool to model and perform the simulations with a detailed parameterization of the pandemic situation and a cellular automata approach, analyzing the effect of the containment measures to contain the spread of SARSCoV-2. A graphical notation was used throughout the research process to keep the interdisciplinary team informed in order to avoid confusion. [13][14]

2.6 SEIRD-CA-SDL model for COVID-19 containment scenarios.

Based on a validated SEIRD model for COVID19 infection, the model was extended to include a cellular automaton that simplified the definition of health zones affected by the pandemic. For the conceptualization, the SDL language and an extensible formal language were used to extend the model to cellular automatons; then the model specification was coded and created with a more powerful approach using Cellular Automaton and a Multi Agent Simulation (MAS) structure [15][16].

3 Results

3.1 Identification of Requirements

a) Obtain the data for the model.
b) Implement the SDL-PAND model (Pandemic Simulation to test the efficacy of the containment strategy through cellular automatons and intelligent agents using formal languages).
c) Configure variables (confirmed cases, containment actions to mitigate pandemic spread).

3.2 Requirement Analysis

Regarding requirement (a), the data were considered reliable because they come from the public health research center belonging to the Ministry of Public Health of Ecuador. In the case of any adjustment to the model, requirement (c) allows this to be done since the model is designed for any change.

3.3 Functional Requirements

- Reliable data from web-based sources of the Ministry of Public Health of Ecuador were used.

- The model generates a statistical graph showing the increase in infections during the months of the pandemic.
- The model generates a statistical graph showing the total number of confirmed cases by date since the beginning of the pandemic.
- The model presents in real time the spread of COVID-19.

3.4 Data Analysis

The data analysis focused on the variable "new cases", since this value is related to the number of hospitalizations, the vaccination level of the population, the age pyramid and the infrastructure of the country's health system.

3.5 Data Collection

Table 1. Averages per quarter from March 2020 to December 2021

	Mar 2020 a May 2020	Jun 2020 a Ago 2020	Sep 2020 a Nov 2020	Dic 2020 a Feb 2021	Mar 2021 a May 2021	Jun 2021 a Ago 2021	Sep 2021 a Nov 2021	Dic 2021
Casos día	424,98	811,62	965,65	1038,55	1 520	823	276	727
Fallecidos día	36,5	34,76	27,12	22,6	44	187	11	13,9
Estables día	233,11	701,86	592,33	648,56	1133,34	615,82	9,37	
UCI día	119,01	314,52	361,18	438,04	571,78	483,67	156,46	184,42
Hospitaliza-dos día	352.12	1016.38	953.51	1086.6	1705.12	1099.49	165.83	184,42
Recuperados día	42,39	895,13	918,15	867,71	1.383	747		

Table 2. Average percentage growth per quarter from March 2020 to December 2021

	Mar 2020 a May 2020	Jun 2020 a Ago 2020	Sep 2020 a Nov 2020	Dic 2020 a Feb 2021	Mar 2021 a May 2021	Jun 2021 a Ago 2021	Sep 2021 a Nov 2021	Dic 2021
Total casos	11,93%	1,17%	0.58%	0.44%	0.43%	0.18%	0.05%	0.14%
Total dosis	0%	0%	0%	87%	33%	17%	2%	2%
Total fallecidos	10,85%	0.73%	0.57%	0.22%	0.33%	0.97%	0.03%	0.04%
Total recuperados	30,47%	10.72%	0.76%	0.42%	0.45%	0.19%		

3.6 Model Construction

To build the SDL of the model, the following activities were developed (See Fig. 5)

1. Design of the SDL diagrams using Microsoft Visio 2013.
2. Generation of the XML file. For this, the Visio diagram system is loaded in the SDLPS program and a file with an extension .sdlps is obtained.
3. The XML file is loaded in the R program to perform the simulation of the model.

Fig. 5. Diagram of the model

3.7 Structure of the Model

The structure is an adaptation of the model developed by Dr. Pau Fonseca i Casas to ensure compatibility with the pandemic data in Ecuador. One database with confirmed cases per date of the pandemic was used.

Table 3. Case Diary Model

	A	B
1	Group.1	Casos Diarios
15	2020-03-13	23
16	2020-03-14	5
17	2020-03-15	9
18	2020-03-16	74
19	2020-03-17	0
20	2020-03-18	57
21	2020-03-19	92
22	2020-03-20	166
23	2020-03-21	106
24	2020-03-22	257
25	2020-03-23	192
26	2020-03-24	101
27	2020-03-25	129
28	2020-03-26	192
29	2020-03-27	224
30	2020-03-28	208
31	2020-03-29	89
32	2020-03-30	42
33	2020-03-31	336
34	2020-04-01	456
35	2020-04-02	405
36	2020-04-03	205

The main diagram of the System and its variables in the Microsoft Visio 2013 program is shown in Fig.6.

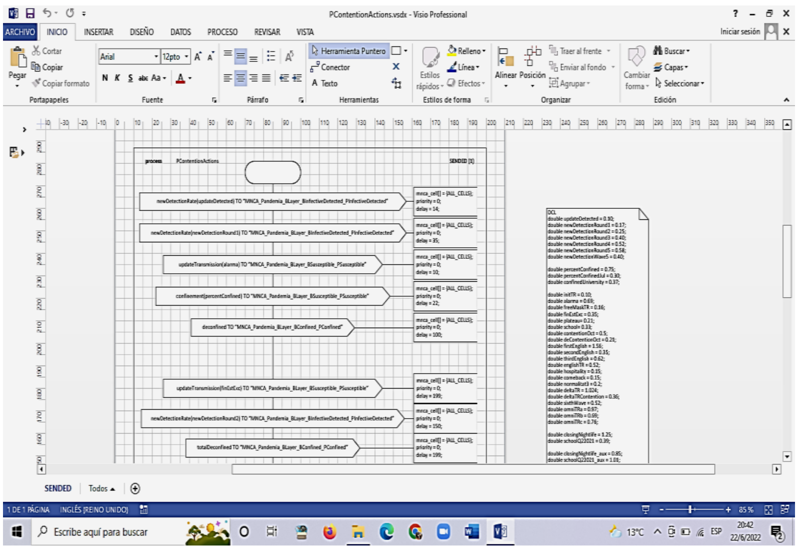

Fig. 6. Diagram of the system and its variables in Visio 2013.

3.8 Variables

The variables considered are those that allow validation of the containment measures taken to mitigate the pandemic (See Fig. 7)

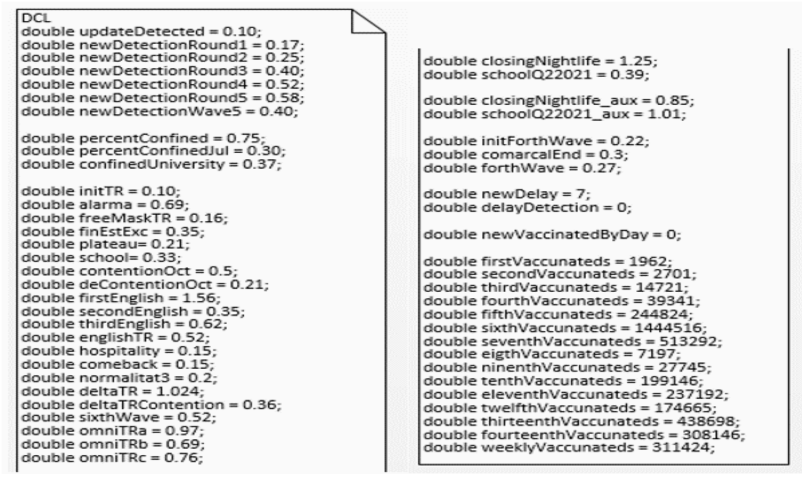

Fig. 7. System Variables

3.9 Model Execution Steps (See Fig. 8)

- Load the model (Home >> Open)

Select the .sdlps file of the model.

- Creation, compilation and link of the Code (Code -> Create, Compile and Link DLL)

The C code corresponding to the sdlps file is generated, compiled together with the external code and the link in a single .dll library.

- Simulation where the execution times are set (Simulation >> Configuration >> by time), the time is set at 1000.
- Initialize the simulation (Execution >> Run)

Initializes the model, executes the processes defined in the SDL from the "Start" state until the change to another state defined in the process.

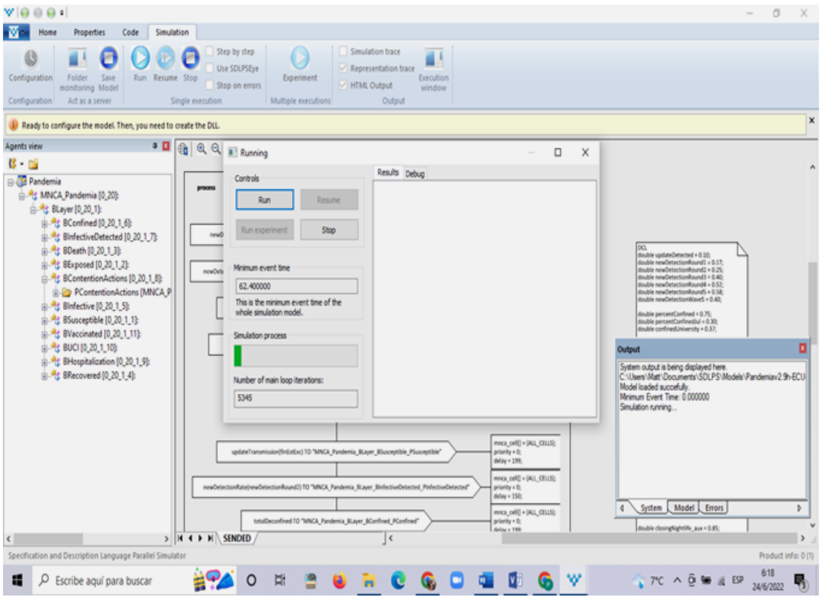

Fig. 8. Execution of the model in SDPLS

Once the execution of the model in SDPLS is finished, an .xml file is generated and is executed in the RStudio program (See Fig. 9).

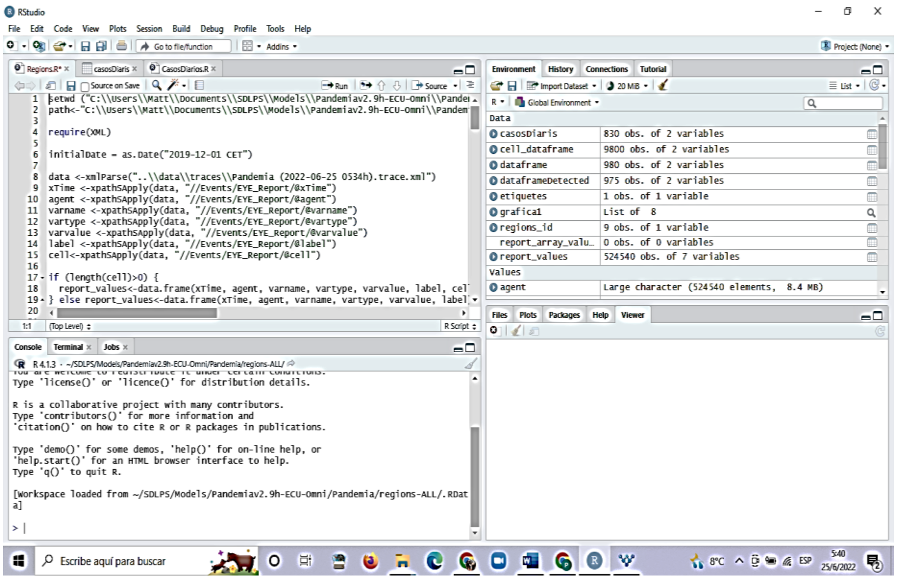

Fig. 9. Execution in RStudio program

Then, it can be visualized the curve that shows the results of the model execution (See Fig. 10)

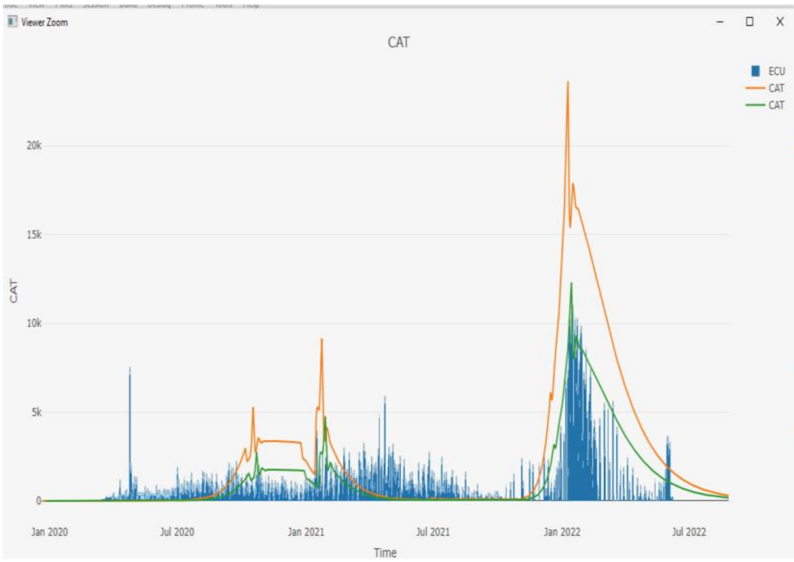

Fig. 10. Simulation curves of the pandemic in Ecuador

The blue lines represent the real cases of pandemic infection taking into account the non-pharmaceutical actions taken by the government as containment measures and are adjusted to the green curve in the simulation of the pandemic behavior.

4 Conclusions

- The SDL language proved to be useful for modeling complex natural systems such as the COVID-19 pandemic, since it allowed to establish its characteristics, analyze them and propose a mathematical model. Additionally, to evaluate the effects of different containment measures through the analysis of the curves obtained in the simulation.
- The SDL-PAND model simulated the behavior of the pandemic in Ecuador, validating the effectiveness of the containment strategies through cellular automatons and intelligent agents using formal languages from the data collected and presented by the Ministry of Health.

Bibliographic References

[1]E. Sherratt, I. Ober, E. Gaudin, P. Fonseca I Casas, F. Kristoffersen, SDL - the IoT language, 2015. https://doi.org/10.1007/978-3-319-24912-4_3.
[2]UIT-T. Lenguaje de especificación y descripción: descripción general de SDL-2010; UIT-T: Ginebra, Suiza, 2011.
[3]Campbell, F.; Archer, B.; Laurenson-Schafer, H.; Jinnai, Y.; Konings, F.; Batra, N.; Pavlin, B.; Vandemaele, K.; van Kerkhove, M.D.; Jombart, T.; et al. Increased transmissibility and global spread of SARS-CoV-2 variants of concern as at June 2021. Eurosurveillance 2021, 26, 2100509.
[4]P. Fonseca, I. Casas, Using Specification and Description Language to define and implement discrete simulation models, in: Proc. 2010 Summer Simul. Multiconference, SummerSim 2010, Ottawa, 2010: pp. 419–426
[5]Sargent, R.G. Verification and validation of simulation models. In Proceedings of the 2007 Winter Simulation Conference, Baltimore, MD, USA, 5–8 December 2010;pp 124-137.
[6]Fonseca i Casas, P. SDL, un lenguaje gráfico útil para describir modelos de simulación social. En Actas del Taller CEUR Actas, Bellaterra, España, 1 de enero de 2009; Volumen 442.
[7]P. Fonseca i Casas, Simple SEIRD model with containment mesures, InsighMaker. (2020).https://insightmaker.com/insight/192085/COVID-19-spread-with-containment-measures.
[8]Fonseca i Casas, P. Hacia una Representación del Autómata Celular Utilizando Lenguaje de Especificación y Descripción. En Análisis y Modelado de Sistemas. lenguajes, métodos y herramientas para la industria 4.0; Fonseca i Casas, P., Sancho, M.-R., Sherratt, E., Eds.; Springer Nature: Dordrecht, Suiza, 2019; págs. 163–179. ISBN 9783030306908.
[9]Stark, R.; Kind, S.; Neumeyer, S. Innovations in digital modelling for next generation manufacturing system design. CIRP Ann. 2017, 66, 169–172.
[10]I. Fonseca, P. Casas. Towards a Representation of Cellular Automaton Using Specification and Description Language. In System Analysis and Modeling. Languages, Methods, and Tools for Industry 4.0; Fonseca i Casas, P., Sancho, M.-R., Sherratt, E., Eds.; Springer Nature: Dordrecht, Switzerland, 2019; pp. 163–179. ISBN 9783030306908.
[11]Documentation - SDLPS. (s/f). Sdlps.com. Recuperado el 12 de diciembre de 2022, de https://sdlps.com/projects/documentation/3.
[12]Fonseca i Casas, P. Hacia una Representación del Autómata Celular Utilizando Lenguaje de Especificación y Descripción. En Análisis y Modelado de Sistemas. lenguajes, métodos y herramientas para la industria 4.0; Fonseca i Casas, P., Sancho, M.-R., Sherratt, E., Eds.; Springer Nature: Dordrecht, Suiza, 2019; págs. 163–179. ISBN 9783030306908.
[13]UIT-T. Lenguaje de especificación y descripción: descripción general de SDL-2010; UIT-T: Ginebra, Suiza, 2011.
[14]Fonseca i Casas, P. Uso del lenguaje de especificación y descripción para formalizar sistemas multiagente. aplicación Artefacto Intel. 2014, 28, 504–531
[15]P. Fonseca i Casas, V. García i Carrasco, J. Garcia i Subirana, SEIRD COVID-19 Formal Characterization and Model Comparison Validation, Appl. Sci. 10 (2020) 5162. https://doi.org/10.3390/app10155162.
[16]P. Fonseca i Casas, Simple SEIRD model with containment mesures, InsighMaker. (2020). https://insightmaker.com/insight/192085/COVID-19-spread-with-containment-measures