A conceptual framework for the modelling and simulation of social systems

Gabriel Awad [a] & Hernán Alvarez [b]

[a] Facultad de Minas, Universidad Nacional de Colombia - Sede Medellín, Colombia. gawad@unal.edu.co
[b] Kalman research group. Facultad de Minas, Universidad Nacional de Colombia- Sede Medellín, Colombia. hdalvare@unal.edu.co

Abstract
This paper presents a conceptual framework for the modelling and simulation of properties, interactions and processes of social systems based on computational templates using discrete event system specification (DEVS) formalism and OMG Systems Modelling Language (SysML) diagrams. No antecedents of this combination were found in the literature, and so this is one of the contributions of this paper. Additionally, this article explains how the principles and rules of SysML can be applied to the analysis of social systems. An example of the proposed framework based on a basic Agent_Zero model is shown. The conceptual framework was built based on a critical literature review, and included new additional elements to create a complete but simple conceptual framework. The codes for the simulations were written in Python 3.

Keywords: conceptual framework; modelling and simulation; social systems; computational templates; SysML; DEVS.

Un marco conceptual para el modelado y la simulación de sistemas sociales

Resumen
Este artículo presenta un marco conceptual para el modelado y la simulación de las propiedades, las interacciones y los procesos de los sistemas sociales basado en plantillas computacionales usando el formalismo de especificaciones de sistemas para eventos discretos (DEVS) y el lenguaje de modelado de sistemas de OMG (SysML). No existen en la literatura antecedentes previos de esta combinación, y esto constituye una de las contribuciones de este artículo. Adicionalmente, este artículo explica cómo los principios y las reglas de SysML pueden ser utilizados para el análisis de sistemas sociales. Se presenta un ejemplo del marco conceptual propuesto basado en el modelo de Agent_Zero. El marco conceptual se construyó a partir de una revisión crítica de literatura, incluyendo nuevos elementos para un marco conceptual completo pero simple. Los códigos para las simulaciones se escribieron en Python 3.

Palabras clave: marco conceptual; modelado y simulación; sistemas sociales; plantillas computacionales; SysML, DEVS.

1. Introduction

Social reality, as a whole, has no disciplinary boundaries [1], however, each social science develops partial models that cover particular features of social systems. If it were possible to build models that were consistent with various theories at once, confidence in them could increase [2].

Traditional social science methods are being complemented and/or substituted by modelling and simulation (M&S) methods. As a consequence, scientific methodologies have moved from a "model-building era" to a "simulation era" [3].

M&S is a disintegrated field, where every discipline has its own style, and it has no transdisciplinary language. Thus, it is difficult to communicate to stakeholders both the structure and the behaviour of the models; to work collaboratively; to reuse others' ideas [4,5]; and to share best practices in methods, approaches and techniques with the scientific community [1]. Moreover, most simulations of social systems are context dependent; they are frequently

carried out without explicitly declaring the theoretical or mathematical models which support them, and little attention is given to methodological issues. Consequently, theoretical background is not formally stated, and assumptions are hidden or loosely defined [6].

As a result, many simulations of social systems are unique models [2], and even if the simulation looks great, its results could be irrelevant [6]. Hence, a new approach to M&S in social systems is needed, based on explicit computational models, where theoretical background and assumptions are stated in detail.

Models of human social behaviour are more complex than physical or engineered systems [2] because human beings are not physical particles [7]. Some substantial differences stem from this differentiation, for example, humans respond, rather than react, to the changes around them and, as a consequence, can have different responses to the same stimulus [4].

This study proposes a conceptual framework for Social Systems Modelling and Simulation, to model and simulate the properties, interactions, and processes of social systems using discrete event system specification (DEVS). A conceptual framework is defined in this paper as a set of concepts which serves as an analytical tool to organize ideas about a specific topic under study. The paper is structured as follows. The second section presents some of the previous works in the field. Then, the third section introduces social systems as a subject of study and explores the main features of modelling and simulation. The fourth section analyses systems specifications formalism for M&S, while the fifth section illustrates how DEVS and the OMG Systems Modelling Language (OMG SysML^{TM}) can be utilised to model social systems. Then, the sixth section develops a conceptual framework for the modelling and simulation of social systems, illustrating the proposal with an example. Finally, the conclusions and suggestions for future studies are put forward.

2. Some previous approaches

The need for the integration of theories and techniques indicates that the development of holistic approaches to M&S is pertinent. For social systems, there have been some attempts in this direction, such as computer simulation laboratory for social theories [8], SimPol [9], and SocLab [10].

Computer simulation laboratory for social theories is a tool that compares and combines social theories using agent-based models. It attempts to explain and forecast social changes. The model was designed to study terrorist insurgency in a developing country, with three kinds of agents: citizens, government fighters and insurgent fighters. The compared leadership theories are: Legitimacy theory, Coercion theory and Representative theory [8].

SimPol is a computational model of a polity (political system). It represents the political processes and structures of a generic society. Progressive versions of SimPol were built, going from simple to complex. It was developed using

object-oriented modelling (OOM), Unified Modelling Language (UML) and the methodology of Lakatos' research programs [9].

SocLab is a formal theoretical framework for the analysis, modelling and simulation of social systems of organized action. It considers individuals, resources, and rules for handling resources and purposes—both personal and organizational—. SocLab uses UML and algebraic structures to model the organization. It contains a meta-model with the main concepts and properties of social organizations which can be instantiated for real world situations. SocLab does not consider relationships between the organization and its environment, but it could be easily enhanced to include relationships which are not controlled by an organization's actor [10].

However, none of the approaches mentioned previously cover the entirety of the social sciences or allows for the full integration of different approaches to the M&S of social systems. Computer simulation laboratory for social theories covers neither the economic nor political aspects. SimPol is restricted to political systems. SocLab only applies to formal organizations. In order to overcome these limitations, this paper develops a conceptual framework based on computational models to model and simulate social systems, both formal and informal, from any social science approach.

3. Social systems, models, and simulations

Social systems are assemblies of social entities whose behaviours are influenced by other social entities and by social forces. Some examples of social systems are groups of friends, religious congregations, and political parties. Both the social system and its components are dynamic, i.e., they change with time.

Any social model represents the modeller's point of view. Due to changes in relationships, the same components can have multiple comparable or rival interpretations [5]. Knowledge of a social system requires the study of its composition, structure, its interactions with the environment, and its behaviour.

Composition refers to the set of components (ontological entities such as actors and things, and epistemological entities such as roles, primordial bodies, and constructed organizations) of a social system. The state of any social entity at a specific time dictates the set of properties that will be relevant at that time.

The structure, organization, or interior constitution of a system refer to the set of relationships connecting the set of components of a social system. A social system only exists when, and while, social relationships occur among its social entities. Two systems can have the same composition, i.e., their elements are the same, but a difference in structure causes their relationships to be different.

Structure includes both the state and the state transition mechanisms [4]. In order to characterize a social system, the initial distribution of control of resources among the actors should be stated. Inferring structure from behaviour is not always possible because different structures can produce the

same behaviour [4].

The environment of a social system is composed of a set of external entities (natural and artificial) that exerts a significant influence on the social system (input) or receives a significant influence from it (output). The environment conditions the behaviour of a social system.

The external behaviour of a system is its visible manifestation; it is the relationship between its input and output time histories [4]. Although there are plenty of interactions in a social system, M&S assigns importance to those related to a state change or a behavioural change. The behaviour of a social system is different to the set of behaviours of its social entities [11].

A model acts as a path that leads towards the understanding of a target question [7]. Models are neither theories nor the target. Many statements that are true about the model are not true regarding the theory or the target, this is because the model is a different entity; it represents the arrival at a target using a theory [4].

The essence of modelling is the setting up of relations between pairs of systems descriptions [4]. A formal model contains a detailed description of its elements, their relationships, and their changing mechanisms. Models are neither right nor wrong, but they can be useful or not in the handling of the research question [7].

Simulation is a particular type of modelling [7] that aims to generate an input/output behaviour based on rules, constraints, equations, and instructions. Even though all computer simulations can be represented by algorithms, a computer simulation only exists while the computer code is running. Thus, computer simulation has temporal existence only [12].

M&S is a cross-disciplinary tool which facilitates the dissemination of methods and theories between different disciplines [13]. M&S is method-driven, i.e., it is conditioned by the available tools and resources and outcome-oriented. In this way, models are built with the aim of reaching specific goals such as mimicking a behaviour or replicating data [14].

M&S models can be used as tools to complement or substitute analytic models [1]. M&S is useful when there is no consensus when specifying a problem, as is the case in the social sciences [5], where every social researcher chooses their own perspective, and has their own research questions.

4. System specification formalisms, a matter of time handling

System specification formalisms permit a local description of the dynamic behaviour of the social system. From these formalisms, it is possible to obtain the global dynamic behaviour of the social system, which is called the dynamic interpretation of the system formalism. System specification formalisms cover different kinds of modelling approaches such as continuous, discrete, and events modelling [4].

The three main paradigms that handle time and thus represent the dynamic behaviour of systems are the following: Differential Equations System Specification (DESS), Discrete Time System Specification (DTSS), and Discrete Events System Specification (DEVS). DESS (continuous) and DTSS (discrete) are both time step paradigms, while DEVS is an event step paradigm. DESS and DTSS have existed for at least three centuries and have been enhanced by computer capabilities, whereas DEVS, based on algorithms run on computers, has been used for just a few decades [4].

Events as time control tools are not a common approach in the M&S of social systems. Discrete event modelling represents human behaviour by switching between a number of relatively simple behaviours or states [4]. In this family of models, time advances from event to event, and does not use clocks nor calendars.

Discrete Events System Specification (DEVS) works by advancing time by means of discrete events. An event corresponds to a significant change in a variable (input or output) or a state. A change's required magnitude at which it is considered significant depends on the modeller's perspective and goals. A modeller sets up a yardstick that indicates the threshold of significance. Once an event is executed and the states' transitions are performed, the list of events is updated by adding and scheduling the new events, and cancelling and removing the old events [4].

DEVS is inherently bound by the capacities and restrictions of digital computers. In discrete time systems, at each time step, every component goes through an update of its state (state transition), regardless of whether its state truly changed. In real life, by contrast, only a tiny fraction of components actually changes at every time step. In discrete event systems, only the components affected by the event update their states at each event step. This implies that DEVS is computationally more efficient than DTSS because it only executes the required processes at each step, and when no events are scheduled, no components need to be examined [4].

Real life situations combine continuous, discrete, and event-time bases simultaneously. For example, in an automated transport control system, traffic monitoring cameras work on a continuous-time basis, while street lights operate on a discrete-time basis, and drivers' behaviour functions on a mixed discrete and event-time basis as each driver responds to both street light changes and the conduct of other drivers [4].

Real human behaviour occurs in leaps, where non-active moments alternate with active moments. Furthermore, nested systems can require different time scales for different levels [15], for instance, changes in macroscopic variables, such as unemployment, are slower than changes in microscopic variables, such as consumption of a good. Therefore, in order to capture this feature, the time base of the model must allow each variable to advance at its own pace [11].

The transformation of DESS and DTSS models into DEVS models is based on identifying when a significant variation in an event arises [4]. Evidence for equivalences between discrete and continuous time models can be found in the literature [16], as well as full discussions and evidence for the reasons and processes for the modelling and

simulation of DESS and DTSS using DEVS [4].

5. The toolbox

The two main tools used in the conceptual framework proposed here are Classic DEVS and SysML. Below, each is described as a starting point for framework proposal.

5.1. Classic DEVS

Classic DEVS, for an atomic model M, is structured according to Eq. (1) [4]:

$$M = \langle X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle \tag{1}$$

where X is the set of external events that can be input into the DEVS, Y is the set of output values from the DEVS, S is a sequence of sets of states, δ_{int} is the internal transition function that stipulates how timeouts transform the states, δ_{ext} is the external transition function that indicates how inputs modify the states, λ is the output function that gives the outcomes of each state, and ta is the time advance function (a set of positive reals within 0 and ∞).

For instance, if a nuclear family formed by two parents and two children is considered an atomic model, M would be the model of the nuclear family. X would be the set of events that are external to the nuclear family, but have some considerable influence on it, e.g., the nuclear family's income. Y would be the outcomes of the nuclear family's actions which affect its environment, e.g., the nuclear family's expenditures. S would be the potential states that the nuclear family could have, e.g., rich or poor, or, highly educated or without education. δ_{int} would be the effects of internal interactions on the nuclear family's properties. δ_{ext} would be the effects of external inputs on the nuclear family's states, e.g., the effects that income has on the nuclear family's welfare. λ would be the output function that generates the outcomes of each state, e.g., the nuclear family's expenditures as a consequence of the nuclear family's welfare. And ta would be the time advance function, e.g., an ordered set of the periods of time when events occur during the simulation.

Classic DEVS can be extended to permit the running of several models synchronously (Parallel DEVS), to handle a Discrete Event Specified Network, and to allow the network structure to be changed during the execution of the simulation (Dynamic Structure DEVS), and so on [4].

5.2. SysML as a language for modelling social systems

SysML, a general-purpose graphical modelling language [17], is neither a tool nor a methodology. Even though SysML's focus is engineering analysis, its principles and rules can be applied to the analysis of social systems.

SysML's main columns are behaviour, structure, system requirements and parametric relationships. The diagrams of SysML represent the model's elements, each of them is a different view of the model, and the full set of diagrams constitutes the model [17].

The behaviour of systems depends on the interactions between the components, their internal dynamics, and the intentions of the stakeholders. In SysML, the logical decomposition of the functions of a system is made using block diagrams, and the interactions of the blocks show their mutual influences. The same system can be represented by several diagrams, which differ in scope and in level of detail [16].

In the same way that not all the words and all the grammar structures of a natural language are used in every conversation, not all SysML diagrams are needed in every model. Goals drive usage, and the inclusion of a SysML diagram must rest on the purposes of the model.

6. Computational templates

Science has traditionally tried to understand the universe from a human viewpoint, i.e., anthropocentric epistemology. However, due to the advances in computer sciences, there are now non-human epistemic authorities, i.e., computational tools, which are faster, and more accurate than the human mind, and, as a result, an anthropocentric epistemology on its own is no longer suitable. The challenge is for humankind to comprehend and evaluate computer-based scientific methods that surpass their own capacities [12]. Heuristics used in both computer simulations and in model building can lay the foundations for a "new" scientific method [18].

A computational model is a self-sufficient piece of analysis. It is composed of the following six elements: initial justification, computational template, construction assumptions, correction set, interpretation, and output representation. While the computational template stays invariable, the other five elements adjust to the system which is being modelled, and to the approach being used [19].

A computational template is unrelated to the target system whereas a computational model has an interpretation that associates a formalism to a specific target system. Thus, computational templates can be common to different social disciplines, while the remaining elements summarise the characteristics of each discipline and context [20].

A theoretical template becomes a computational template when it is detached from its seminal theory and it is applied in other contexts and domains of knowledge. This procedure is common in engineering, where laws in diverse fields are based on the same mathematical structure. The key features for converting a theoretical template into a computational template are its tractability, its generality, and its versatility [13].

The construction process offers justifications for the M&S even before its outcomes are confirmed against data. Modellers include their paradigms, and quite often some of the stakeholders' paradigms, into the M&S assumptions, although some of these assumptions are then mitigated by the correction set [19].

The initial justification contains the purposes for developing the model and explains or substantiates the situation which is intended to be handled. The initial

justification can rely on the theoretical foundations, experiences, and intuitions of the modellers and/or the stakeholders [19]. Initial justification includes both the questions that the model should answer, and the justification of why it is important to answer them [6].

Construction assumptions are beliefs incorporated into a model in order to overcome uncertainties, some of these include: ontology, idealizations, abstractions, constraints, and approximations [21].

The correction set is target-dependent, so it tries to adjust the computational template to the specific situation, i.e., approximate the computational template to the context of the social system being modelled. The correction set is composed of relaxation of idealizations, e.g., changing perfect rationality to bounded rationality; relaxation of abstractions, e.g., adding government and international trade in an equilibrium model; relaxation of constraints, e.g., not allowing social entities to be in contact with close neighbours; refinement of approximations, e.g., reducing the time step from monthly to daily [19]; and changes in the ontology of the system, e.g., using an epidemiologic model in order to model and simulate the dissemination of an innovation. The latter can even give birth to a new construction process [21]. The same computational template will have different correction sets associated with it depending on the context of where it is being utilised [19].

Every model comes with an interpretation. The basic formalism and the interpretation are inseparable; their combination gives meaning to the M&S. Even though the same computational template can be used to model different systems, each application requires its own interpretation [19].

The interpretation grounds the computational template to the phenomenon being studied. A computational template helps to describe "how" a social system behaves, but not to decipher "why" the social system behaves in that way [19], and so the "why" must be supplied by the theory.

The output representation is the way outcomes of the M&S are communicated to stakeholders. It can include graphics, figures, and statistics [19]. When M&S refers to real world phenomena and there is real collected data available, it can be validated by comparing simulation outputs to real data. A careful selection of the output representation mode is crucial in M&S [21]. The output representation must be oriented to the needs of the stakeholders.

Computational templates are sets of equations, modelling techniques [20], or solvable computational methods which can be applied in many scientific disciplines. They both represent the target and facilitate calculations. Some examples of computational templates are the Poisson process, Lotka-Volterra equations, Laplace's equation [19], and the Ising model [13].

7. Proposal of a conceptual framework for the modelling and simulation of social systems

This paper proposes a new conceptual framework which generates a complete and rigorous description of the M&S applied to social systems. The framework allows both the communalities and the specificities to be clearly identified in any M&S of a social system, independent of the system specification used. The proposed framework combines the computational modelling approach with the SysML and DEVS paradigms to build a M&S of social systems. Fig. 1 shows the framework for the M&S of social systems.

The framework consists of four principal stages, each composed of small steps. These stages can be developed and presented, in succession or in parallel, depending on the modeller's judgement.

The principal stages are as follows:
- Build the computational model using verbal models, i.e., natural prose, and mathematical models, i.e., equations and logical propositions.
- Translate the computational model to SysML, i.e., make a graphical model of the computational model.
- Use the DEVS formalism to express the computational template, the construction assumptions, and the correction set.
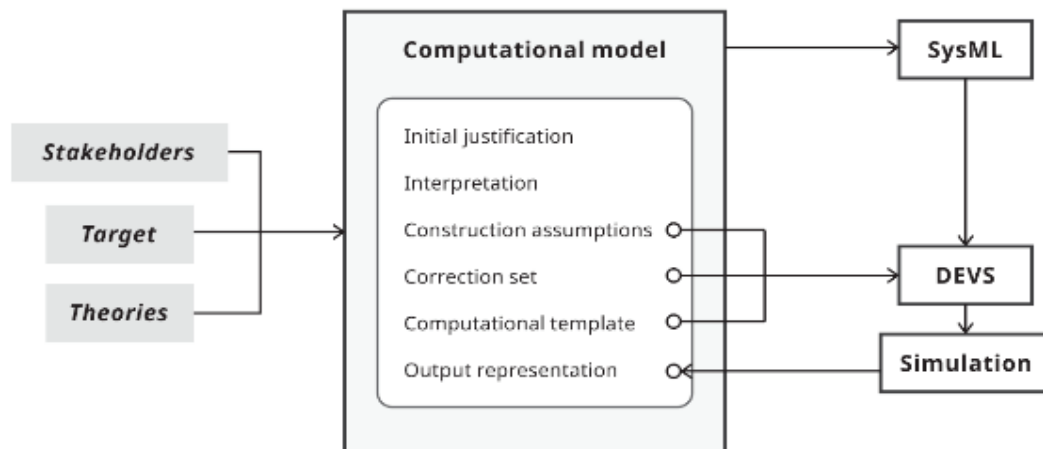- Finally, run the M&S to obtain the outcomes for the output representation.

Figure 1. Framework for M&S of social systems.
Source: The Authors.

The initial justification and interpretation comprise the purposes of the M&S, they then can be graphically represented using the SysML requirement diagrams and use case diagrams. The construction assumptions and correction set can be included in the SysML parametric diagrams. The computational template includes the structure and the behaviour of the M&S. Its graphical representation can be made in SysML by block definition diagrams, internal block diagrams, state machine diagrams, sequence diagrams, and activity diagrams. Finally, the output representation can be indicated by SysML package diagrams. DEVS formalism ignores the context of M&S, therefore, it does not include the initial justification or the interpretation. The computational template, the construction assumptions and the correction set are the components of the computational model used for the deployment of the DEVS formalism. The output representation is obtained once the model is executed, i.e., a simulation is run, from the sets of inputs, outputs, and states.

Next, the relationships between these six diagrams and DEVS will be presented for the simpler case (i.e., Classic DEVS). Eq. (1), laid out in Section 5, represents a formal definition of Classic DEVS where SysML sequence diagrams and activity diagrams contain X, a set of external events that can be input into the DEVS, and Y, a set of output values produced by the DEVS. X and Y correspond to flow ports and item flows. SysML block definition diagrams and internal block diagrams depict the objects, while state machine diagrams, sequence diagrams, and activity diagrams illustrate S, a sequence of the set of states.

SysML state machine diagrams, sequence diagrams, and activity diagrams are represented by δ_{int}, the internal transition function that stipulates how timeouts transform the states, and δ_{ext}, the external transition function that indicates how inputs modify the states. SysML state machine diagrams show λ, the output function that gives the outcomes of each state, and SysML sequence diagrams contain ta, the time advance function. The output representation, symbolised by package diagrams in SysML, shows the sets of inputs (X), outputs (Y), and states (S).

The framework described above relies on the computational model approach, SysML and DEVS paradigms. All of these are general-purpose elements; therefore, the framework has the potential to be applied in any kind of M&S situation.

8. A demonstrative example

The main aspects of the proposed conceptual framework will be shown using the Agent_Zero model as a reference [22]. This demonstrative example has the following four principal stages: computational model, SysML, DEVS, and running a simulation. This example only replicates the basic Agent_Zero model; it does not include any of its extensions.

The first stage consists of deploying the computational model. The initial justification arises from the following simple but astonishing fact: very often individuals behave differently in groups than when they are alone. In other words, sometimes people do things in public that they would never do when alone [22]. These behaviours are usually explained based on imitation hypotheses [1, 11], but they do not explain why the first-person acts, nor prove that other people acting later are really imitating the first actor. Thus, Epstein's model was built to answer the following questions: Why does the first-person act, and, are the following actors' imitators or are they acting "on their own"? [22].

Epstein proposes a basic model which considers that a person's conduct when they participate in a group depends on emotions, reason, and social influence. The interpretation of the Agent_Zero model suggests that personal behaviour in a group of people could be based on others' disposition to act, and not on the observable behaviours of individual members of the group [22]. In other words, what is disseminated are moods, not behaviours.

Fig. 2 shows the SysML diagram of requirements associated with the initial justification and the interpretation of the model. The main requirement consists of the M&S of the behaviour of Agent_Zero. This requirement must satisfy the social influence requirement and the disposition when alone requirement. The later requirement must satisfy the requirement corresponding to emotions and to reason.

The construction assumptions of the Agent_Zero model rely on three axioms. Firstly, behaviour is binary because a person either acts or does not act. Second, people's behaviour is not solely rational, their actions are the result of the interactions between emotions, reason, and social influence. Third, people do not imitate the behaviour of others. Each individual decides when and how to act based on their personal characteristics [22].

The individual disposition to act (D_i) is the result of emotional, cognitive, and network components, while the individual action threshold (τ_i) is a fixed non-negative value [22].

The affective component (V_{it}), and the deliberative component (P_{it}) are non-negative real numbers between zero and one, both extremes included. As people's feelings and thoughts change over time, the t subscript indicates these trajectories, either on a time or trial basis [22].

Different psychological models can be used for the emotional function (V_{it}), such as neural network models, temporal difference models, and the Rescorla-Wagner model. Agent_Zero uses a generalization of the Rescorla-Wagner model, which is based on the conditioned stimulus theory [22].

Fig. 3 depicts a SysML parametric diagram of the constraints of Agent_Zero, as was stated in the previous paragraphs. Behaviour must be a binary number, the individual threshold of action is a non-negative real number, and the affective and deliberative components are non-negative real numbers between zero and one, both extremes included.

The basic Agent_Zero model is a toy model, i.e., there is no real target associated with it, and it has no predictive power about a specific person. Its building ideas come from the neurosciences, but it lacks validation or adjustment against real data [22]. Therefore, this demonstrative example has no correction set. If there were data available, it could be used both to calibrate and to validate the model.
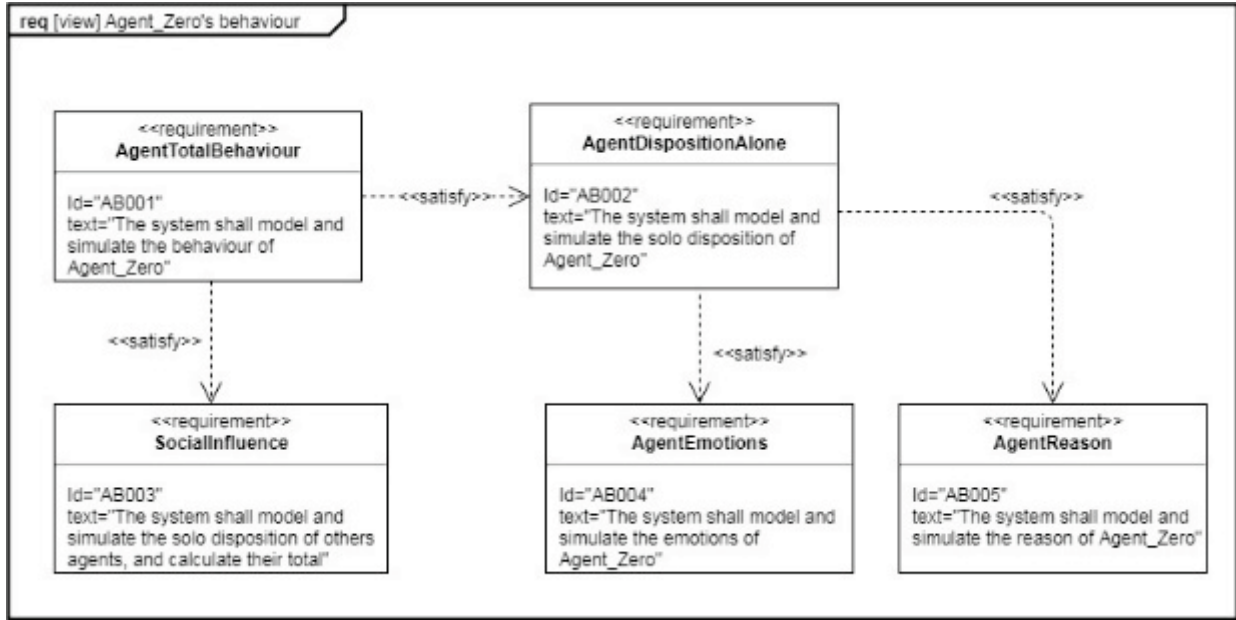
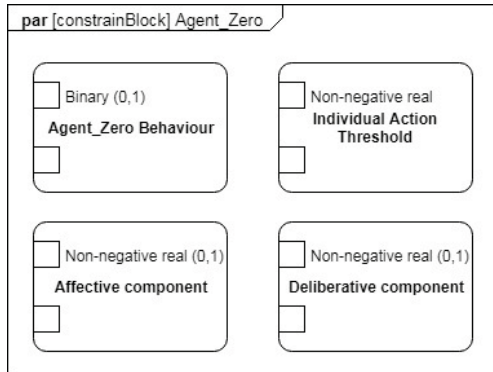Figure 2. Requirement diagram of Agent_Zero's behaviour.
Source: The Authors.

Figure 3. Parametric diagram of Agent_Zero.
Source: The Authors.

The computational template of Agent_Zero, based on its construction assumptions, has two elements. The first one is a decision rule, and the second one is an additive function of the affective, deliberative, and contagion components [22].

As stated in the construction assumptions, behaviour is binary. A person acts when their individual disposition (D_i) to act surpasses their individual action threshold (τ_i). Eq. (2) shows the decision rule for acting [22]:

$$A = \begin{cases} 0 & if \ \ D_i < \tau_i \\ 1 & if \ \ D_i \geq \tau_i \end{cases} \qquad (2)$$

where 0 means no action, while 1 implies action.

Fig. 4 shows the SysML state machine diagram of Agent_Zero. It only has two states: active and inactive. If the disposition is equal or greater than the threshold of action, Agent_Zero passes from inactive to active, and remains in this state while this condition is fulfilled. If the disposition falls below the threshold of action, Agent_Zero returns to the inactive state. The state machine starts at the initial time of simulation, i.e., t=0 for this demonstrative example, and ends when the goal of external stimulus (trials) is reached, i.e., trial =50 for this demonstrative example.

When a person is isolated, i.e., there are no other people nearby, their behaviour depends only on their emotions and their reasons. Eq. (3) shows the individual disposition when alone ($D_i^{solo}(t)$) [22]:

$$D_i^{solo}(t) = V_{it} + P_{it} \qquad (3)$$

Eq. (4) formally presents the Rescorla-Wagner model used in Agent_Zero [22]:

$$V_{t+1} = V_t + \alpha\beta(\lambda - V_t) \qquad (4)$$

where V_t is the value of learning at "time" t, α is the value of the conditioned stimulus, β is the value of the unconditioned stimulus, λ is the maximum possible value of V_t, i.e., maximum level of learning, and α, β, and λ are non-negative constants. α and β are also known as learning rates [22]. This equation implies that Agent_Zero has emotional memory; their heart never forgets.

It is worth noting that the Rescorla-Wagner model is a monotonically growing (i.e., never decreasing) function, and that, in this case, t means "trials" which confirms the use of discrete event systems specification (DEVS).

The evidentiary function (P_{it}) estimates a probability value based on "evidence", and Agent_Zero currently uses a random value, under the assumption of bounded rationality [22]. Thus, Agent_Zero has no rational memory. Their mind ignores all its previous states, and it behaves like a Markovian process.
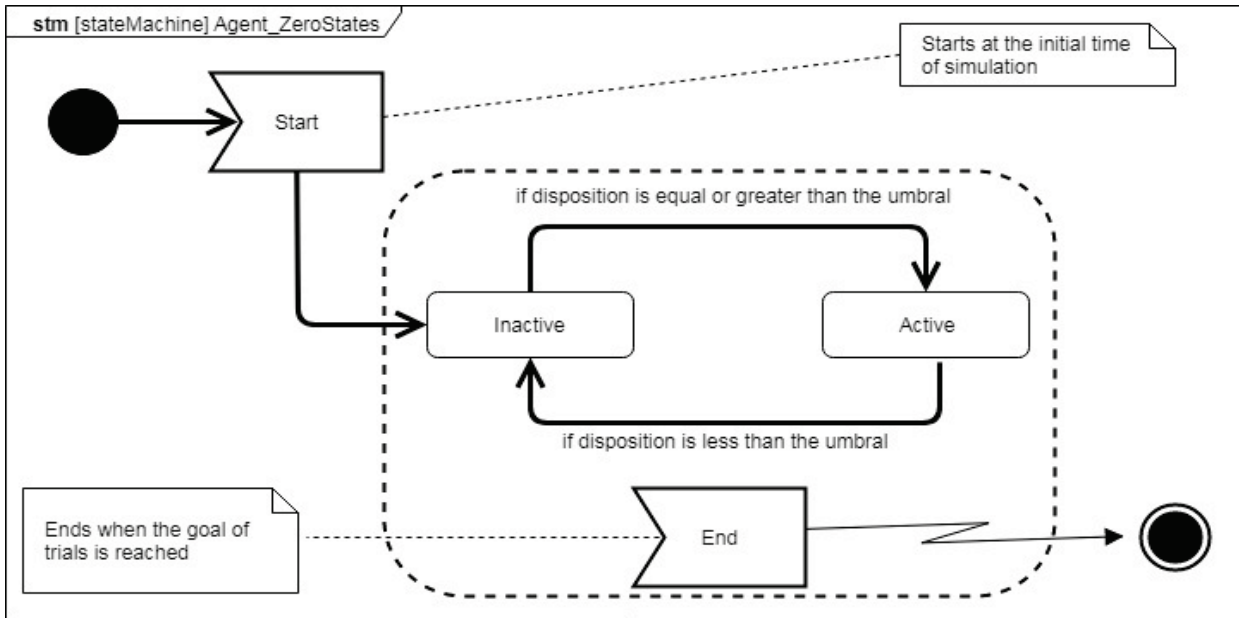
Figure 4. The states of Agent_Zero.
Source: The Authors.

The individual disposition of Agent_Zero, if they are alone, corresponds to a Classic DEVS as stated in Eq. (1), where X is the set of conditioned stimulus (α), Y is the set of values calculated using Eq. (4), and the sequence of states (S) matches the dichotomy active/inactive. There is no internal transition function (δ_{int}), because the update of values depends on the external stimulus only. The external transition function (δ_{ext}) is formed by the Eqs. (8), (9), and (10). The output function (λ) that gives the outcomes of each state is calculated using Eq. (4). The time advance function (ta) is the period when the stimulus occurs, and due to the lack of a correction set for this example, the time of each stimulus is generated using a uniform random generator function.

Thus, Eq. (1), deleting the internal transition function, simplifies to Eq. (5):

$$M = \langle X, Y, S, \delta_{ext}, \lambda, ta \rangle \qquad (5)$$

Once a person is in the company of a group of people, their individual disposition is influenced by the individual dispositions of each person in the group when alone. Eq. (6) formally expresses this situation [22]:

$$D_i^{tot}(t) = D_i^{solo}(t) + \Sigma_{i \neq j}\, \omega_{ji}\, D_j^{solo}(t) \qquad (6)$$

where $D_i^{tot}(t)$ is the individual disposition when in groups and ω_{ji} is the weighted solo dispositions of all other people. The existence of ω_{ji} is assumed to be a non-conscious value, and it allows different people to have different influences. Eq. (6) is a skeletal equation, which applies for distinct functions of V_{it} and P_{it} [22], i.e., it is a computational template in the sense proposed by Humphries [19], and Houkes and Zwart [20].

The individual disposition when in groups ($D_i^{tot}(t)$)

corresponds to a Parallel Classic DEVS as stated in Eq. (7), where many Agent_Zero-type actors are synchronously running the same Rescorla-Wagner model.

$$M= \langle X_M, Y_M, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta \rangle \qquad (7)$$

In this expression, X_M is the set of conditioned stimuli (α) and the outputs of the other Agent_Zero-type actors, Y_M is a set of individual dispositions of each Agent_Zero-type actor.

The sequence of states (S), the external transition function (δ_{ext}), the output function (λ), and the time advance function (ta) remain as described in Eq. (1). Again, there is no internal transition function (δ_{int}) and the confluent transition function that arbitrates collisions among states (δ_{con}) is not required because there is no potential time conflict among the parallel models, due to the absence of circular references. So, Eq. (7), deleting the internal transition and the confluent transition functions, simplifies to Eq. (8):

$$M= \langle X_M, Y_M, S, \delta_{ext}, \lambda, ta \rangle \qquad (8)$$

Commented codes using Python 3 that develop the ideas—corresponding to Eq. (8)—for running the simulations are available from the authors.

The threshold for action in the simulation, due to the lack of a correction set (i.e., no real data available), was arbitrarily set at 0.8 units.

Fig. 5 shows the individual disposition when alone (*Dsolo*), the total disposition (*Dtotal*), the state of Agent_Zero (action), and the threshold for action when Agent_Zero is in a group of twenty.

This example illustrates the four stages of the conceptual framework using a Parallel Classic DEVS as stated in Eq. (8).
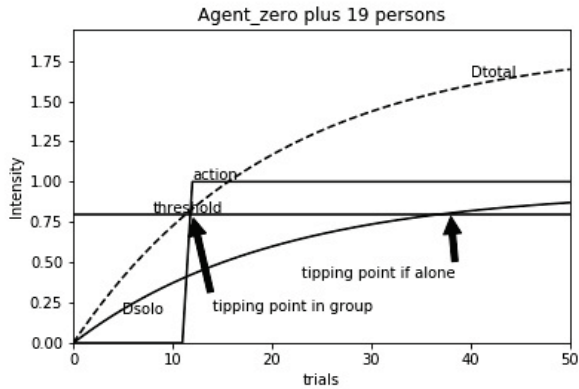
Figure 5. Agent_Zero 0 in a group of twenty.
Source: The Authors.

It shows the role of "the others" in the behaviour of Agent_Zero, and it illustrates that social influence, even without imitation, can affect the behaviour of Agent_Zero as proposed by Epstein [22].

The conceptual framework combines the computational modelling approach with SysML and DEVS paradigms. It consists of four principal stages, each composed of small steps. The principal stages build the computational model using verbal models, then, the computational model is translated to SysML. After that, the DEVS formalism is used to express the computational template, the construction assumptions, and the correction set. Finally, the M&S is run in order to obtain the outcomes for the output representation. These stages can be developed and presented in succession or in parallel, depending on the modeller's choice.

9. Conclusions and future work

This paper proposed a conceptual framework for social systems M&S using discrete event system specification (DEVS), allowing models to be built that are consistent with various social theories simultaneously, thereby increasing confidence in the models. Also, the conceptual framework supports the transition from a "model-building era" to a "simulation era" in the social sciences. It also can be employed for the M&S of any other system because it is developed from general purpose elements (computational models, SysML, and DEVS). Future work can apply the conceptual framework to other fields.

This kind of framework had not been proposed previously because the use of DEVS is not common among social scientists and a graphical modelling language that was simple enough to be used for scientists lacking a strong background in M&S did not exist.

The proposed conceptual framework promotes a detailed construction of M&S, clarifying the assumptions, postulates, and theorems extracted from social theories to make both the theoretical background and the gaps in theories visible. In this way, it provides all stakeholders a shared framework, and it helps to increase their mutual understanding. This framework approach eases communication between modellers and stakeholders, helping collaborative work, and enhances the replicability of M&S.

Despite its potential, social scientists do not widely use M&S. It would be advisable to promote its dissemination in the social sciences, and to build a repository of the structures of social models (i.e., computational templates) in order to facilitate their transferal within and between disciplines.

References

[1] Squazzoni, F., Agent-based computational sociology. John Wiley & Sons, 2012.

[2] Carley, K.M., Computational modeling for reasoning about the social behavior of humans, Computational & Mathematical Organization Theory, 15(1), pp. 47-59, 2009, DOI: 10.1007/s10588-008-9048-9.

[3] Hartmann, S., The world as a process, in: Modelling and simulation in the social sciences from the philosophy of science point of view, Springer, 1996, pp. 77-100.

[4] Zeigler, B.P., Praehofer, H. and Kim, T.G., Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems, 2nd ed., Academic press, San Diego, California, USA, 2000.

[5] Diallo, S.Y., Padilla, J.J., Gore, R., Herencia-Zapana, H. and Tolk, A., Toward a formalism of modeling and simulation using model theory, Complexity, 19(3), pp. 56-63, 2014, DOI: 10.1002/cplx.21478.

[6] Aumann, C.A., A methodology for developing simulation models of complex systems, Ecological Modelling, 202(3), pp. 385-396, 2007, DOI: 10.1016/j.ecolmodel.2006.11.005.

[7] Gilbert, N. and Troitzsch, K., Simulation for the social scientist. McGraw-Hill Education, UK, 2005.

[8] Whitmeyer, J.M. et al., A computer simulation laboratory for social theories, in: Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on, vol. 2, 2008, pp. 512-515, DOI: 10.1109/WIIAT.2008.169.

[9] Cioffi-Revilla, C., Simplicity and reality in computational modeling of politics, Computational & Mathematical Organization Theory, 15(1), pp. 26-46, 2009. DOI: 10.1007/s10588-008-9042-2.

[10] Sibertin-Blanc, C. et al., SocLab: a framework for the modeling, simulation and analysis of power in social organizations, Journal of Artificial Societies and Social Simulation, 16(4), pp. 1-24, 2013, DOI: 10.18564/jasss.2278.

[11] Helbing, D., Quantitative sociodynamics: stochastic methods and models of social interaction processes. Springer Science & Business Media, 2010.

[12] Humphreys, P., The philosophical novelty of computer simulation methods, Synthese, 169(3), pp. 615-626, 2009. DOI: 10.1007/s11229-008-9435-2.

[13] Knuuttila, T. and Loettgers, A., The productive tension: mechanisms vs. templates in modeling the phenomena, Representations, models, and simulations, 2012, pp. 2-24.

[14] Knuuttila, T. and Loettgers, A., modelling as indirect representation?. The Lotka-Volterra model revisited, The British Journal for the Philosophy of Science, 68(4), pp. 1007-1036, 2017. DOI: 10.1093/bjps/axv055.

[15] Troitzsch, K.G., Perspectives and challenges of agent-based simulation as a tool for economics and other social sciences, in: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1, 2009, pp. 35-42. DOI: 10.5555/1558013.1558016.

[16] Ljung, L. and Glad, T., Modeling of dynamic systems, 1994.

[17] The Object Management Group, OMG systems modeling language (OMG SysML) version 1.4. The Object Management Group, 2015.

[18] Humphreys, P., Computational science and scientific method, Minds and Machines, 5(4), pp. 499-512, 1995. DOI: 10.1007/BF00974980.

[19] Humphreys, P., Computational models, Philosophy of Science, 69(S3), pp. S1-S11, 2002. DOI: 10.1086/341763.

[20] Houkes, W. and Zwart, S.D., Transfer and templates in scientific modeling, 2012.

[21] Humphreys, P., Extending ourselves: computational science,

empiricism, and scientific method. Oxford University Press, New York, USA, 2004.

[22] Epstein, J.M., Agent_Zero: toward neurocognitive foundations for generative social science. Princeton University Press, Princeton, New Jersey, USA, 2013.

G. Awad, has a PhD. in Engineering from Universidad Nacional de Colombia, Campus Medellin. Currently, he is an associated professor at the Computing and Decision Sciences Department, Facultad de Minas, Universidad Nacional de Colombia – Sede Medellín. His research interests include modelling and simulation of social systems, blended-learning, and artificial intelligence.
ORCID: 0000-0001-8694-7315

H. Alvarez, has a PhD. in Control Systems Engineering from the Instituto de Automática at Universidad Nacional de San Juan, Argentina. Currently he holds a full professor tenure position at the Processes and Energy Department, Facultad de Minas, Universidad Nacional de Colombia – Sede Medellín. His research interests include chemical process modelling and control, artificial intelligence in automatic control, fuzzy inference systems and model predictive control.
ORCID: 0000-0002-2253-3583