CoFluences: Simulating the Spread of Social Influences via a Hybrid Agent-Based/Fuzzy Cognitive Maps Architecture

Philippe J. Giabbanelli*
Furman University
Computer Science Department
Greenville, SC, USA
giabbanelli@gmail.com

Michele Fattoruso
Northern Illinois University
Department of Computer Science
DeKalb, IL, USA
michele.fattoruso@gmail.com

Max L. Norman
Furman University
Computer Science Department
Greenville, SC, USA
max.norman@furman.edu

ABSTRACT

Social influences are key drivers of many human behaviors, and have been the focus of an abundance of discrete simulation models. In participatory modeling, the emphasis is on developing models in an intuitive and transparent manner. Fuzzy Cognitive Mapping (FCM) provides such an intuitive and transparent process, but it can only simulate the thinking of one entity rather than how entities influence each other. Hybrid architectures based on FCM and Agent Based Modeling (ABM) can bridge this gap, but current software implementing these architectures either restricted the models (e.g., limiting agent heterogeneity by requiring that they all follow the same rules) or required extensive coding (which participatory modeling avoids). In this paper, we contribute to software development by presenting `CoFluences`, and to the theory of modeling and simulation by better characterizing hybrid ABM/FCM architectures. `CoFluences` is the first software to develop and simulate hybrid ABM/FCM models in a participatory setting, and where agents can follow different rules. Although we take a User-Centered Design approach to develop `CoFluences`, a comprehensive usability study will be necessary to fully evaluate it in context. In addition, the growing interest in developing simulation software involving FCM will call for more standardization, and for a better understanding of how an FCM behaves in a hybrid simulation.

CCS CONCEPTS

• **Computing methodologies → Simulation environments**; *Agent / discrete models*; Modeling methodologies;

KEYWORDS

Cognitive Architecture; Mental Models; Simulation Software; Soft Computing

ACM Reference Format:
Philippe J. Giabbanelli, Michele Fattoruso, and Max L. Norman. 2019. CoFluences: Simulating the Spread of Social Influences via a Hybrid Agent-Based/Fuzzy Cognitive Maps Architecture. In *SIGSIM Principles of Advanced Discrete Simulation (SIGSIM-PADS '19), June 3–5, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3316480.3322887

*Corresponding author.

ACM ISBN 978-1-4503-6723-3/19/06…$15.00
https://doi.org/10.1145/3316480.3322887

1 INTRODUCTION

Human behaviors are partly shaped by how we expect others to behave (descriptive social norms) and how we perceive whether others will approve of a behavior (injunctive social norm). The influence of social norms on behavior has long been documented in psychology, sociology, and cognitive science. However, their application is not limited to such fields. For instance, social norms are important in public health: cross-sectional studies, experimental studies, and meta-analysis have consistently observed that social norms could provide effective levers to encourage individuals in adopting healthy behaviors [59, 61].

An abundance of simulation models has thus been devoted to social norms. In particular, discrete models are routinely employed: individuals are represented through discrete entities, and their behaviors are updated over discrete time units. Network models [4, 43] and Agent-Based Models (as exemplified by the work of Epstein [13], Axelrod [2], and others [3]) have both been particularly popular approaches to simulate social norms under often interchangeable keywords such as 'spread', 'cascade', 'contagion', or 'reaction-diffusion processes'. Agent-Based Models (ABMs) differ from network models by allowing for an explicit representation of the space, such that the influence of social norms on an agent's behavior may be mediated by local environmental drivers. However, many models portrayed as ABMs may simplify the environment to a single variable [60] thus blurring the line between ABMs and network models. In this paper, our focus is on heterogeneity in *decision-making* rather than on heterogeneity in the locations/movements of agents in space.

An advantage of using individual-level models is their ability to represent *relevant* differences among agents in a given application context. For instance, a simulation of eating behaviors is unlikely to include differences in the agents' eye colors or their political beliefs. Conversely, it would be typical to represent differences in how agents value the taste or healthiness of food items [37, 68] because such factors shape how the agents *form decisions.* A model may achieve heterogeneity in decision-making by giving the *same rules and variables* to all agents, and only assigning *different values* to variables that the output behavior is highly sensitive to. Behaviors have occasionally been described as chaotic, meaning that such minute differences in variables can be sufficient to set individuals to entirely different trajectories [58]. However, individuals may follow different rules, which explains why they make different decisions when facing the same evidence. Consider a socio-environmental system such as fishery management, in which the agents include recreational fishermen, managers of fishing clubs, or environmental experts. The same evidence may be presented to these agents (e.g., over-fishing and decreasing catch), but their different *mental models* will result in favoring different actions (e.g., bring in fish from elsewhere, temporarily ban fishing) [41]. Such contexts warrant models in which agents follow different rules, in addition to having different feature values.

There is no shortage of architectures or software approaches to create a population of agents equipped with different rules. For instance, there are hundreds of cognitive architectures elaborating how humans think [40], and several of them have been operationalized into modeling toolkits for agents such as the Belief-Desire-Intention (BDI) approach [8]. The modeling challenge is to have a *transparent* and *systematic* process to map the heterogeneous decision-making processes of humans onto heterogeneous rules for agents. Having a systematic process supports the comparison of models and the replication of a study, which is an important ongoing goal in modeling and simulation [64]. Transparency is important in *participatory modeling*, which involves those who are affected by decisions that stem from a model. A transparent process would allow participants to straightforwardly provide their mental models and see how they are used by a simulation, thus contributing to both the realism of the simulation and building trust in its outcomes. A fundamental assumption of a transparent process is that participants are actually aware of their mental models. While this assumption has historically been questioned, recent evidence suggests that participants are indeed able to externalize mental models for familiar situations [21]. Consequently, debates in participatory modeling have gradually shifted from *whether* participants can externalize mental models to *how* they may do it most accurately and transparently [36].

On the one hand, the methodology of Agent-Based Modeling *can* represent interactions between agents with different rules, but it does not provide a transparent and systematic process to obtain these rules by externalizing the mental models of participants. On the other hand, soft systems methodologies such as Fuzzy Cognitive Maps (FCMs) provide a transparent and systematic process to elicit the mental model of a participant or group, but do not provide a way for these models to interact. Metaphorically, the FCM methodology can provide the agents' brains, but brains cannot directly interact: they must be placed inside bodies. Several works have thus proposed hybrid models combining ABM and FCMs, for application contexts as diverse as simulating species in an ecosystem [26, 27], modeling participants in fisheries management [22] or healthy eating programs [23], or the spatial spread of insurgency [18, 55]. While there are widely used software for ABM (e.g., `NetLogo`, `Repast`, `AnyLogic`, `Mason`) and a variety of software supporting FCM (e.g., `MentalModeler` [30], `FCM Wizard` [48], `FCM Expert` [49]), there is a paucity of software combining ABM and FCM. The main two options are `EcoSim` [26], which is specific to ecosystem simulations (e.g., emphasis on reproduction and mutations of models), and our own prototype [22], which constrained agents to all have the *same* mental model and only differ in parameter values.

In this paper, we present `CoFluences`, the first simulation environment for the spread of social norms that creates heterogeneous rules in an Agent Based Model (ABM) using Fuzzy Cognitive Maps (FCM). Specifically, our primary contributions are two-fold:

- Previous hybrid simulation software creating ABM rules using an FCM required that all agents had the same FCM. We address the algorithmic, design, and implementation challenges of simulating agents with different FCMs.
- While FCMs are particularly useful in participatory modeling, the primary audience for previous ABM-FCM software was simulation experts rather than participants. We leveraged the usability lessons learned from participant-centric tools and our prototype to create a software that can be used in facilitated modeling sessions with participants.

The organization of this paper is as follows. In section 2, we provide a background on FCMs and how they have been combined with ABMs in previous simulation software. Section 3 starts the presentation of our software, by exposing the desired functionalities, mapping them onto a design mindful of usability by participants, and formally specifying the simulation using pseudocode. Building on this, we present our implementation in section 4 along with remarks on its verification. Lastly in section 6, we examine the strengths of our software and discuss possible solutions to its limitations, followed by our closing remarks.

2 BACKGROUND

Section 2.1 characterizes the intended audience for our software, and the specific modelling techniques that they tend to use. As simulation experts are often most familiar with Agent Based Modeling, we instead focus on the fundamentals of Fuzzy Cognitive Maps both intuitively (2.1) and formally (2.2) before combining them with agents (2.3).

2.1 Fuzzy Cognitive Maps for Participatory Modeling

The Participatory Modeling process supports participants to "use modeling to describe the problem, to identify, develop and test solutions, and to inform the decision-making and actions of the group" [67]. Approaches suitable for participatory modeling thus target non-computer scientists [6, 67], and must provide an accessible means to (i) design, such that participants can externalize their knowledge into formal, shared representations of reality; (ii) implement, without coding; and finally (iii) test questions of interests via 'what-if' scenarios. The idea that participants must take ownership of the design precludes a 'model coding' stage done solely by modelers, as may be found in facilitated simulation approaches [63]. The model's code thus has to be automatically generated from an intuitive design process, which explains why participatory modeling rarely involves sophisticated modeling approaches such as the Discrete Event System Specification (DEVS) or frameworks using abstract state machines (e.g., CoreASM [14]). Instead, participatory modeling involves qualitative techniques (e.g., rich pictures, causal loop diagrams) and (semi-)quantitative techniques (e.g., cellular automata, agent based models, system dynamics, fuzzy cognitive maps) [67]. Purely qualitative techniques have limited support for analysis as they cannot quantify effects. For example, analyzing a causal loop diagram can reveal central factors or unexpected groupings [19, 20], but it cannot compute the effect that an intervention may have on a specific target. Fuzzy Cognitive Maps (FCMs) stand apart among quantitative techniques by their relative simplicity.

They can be developed quickly, at low cost, with little data, and require a medium expertise of both stakeholders and modelers [67]. The relative simplicity to build FCMs is a common theme to several reviews and books on FCMs [1, 25, 35, 52, 53].

Intuitively, a Fuzzy Cognitive Map can be seen as an *augmented* causal loop diagram. Similarly to a causal loop diagram, an FCM represents factors as nodes and causes as labeled directed edges. The label specifies whether the target node increases ('+') or decreases ('-') when the source increases. An FCM goes beyond causal loop diagrams in two ways that are instrumental to producing quantitative outcomes. First, they *quantify* the values of nodes and edges. Node values are assigned based on a specific case: in Figure 1, an FCM models a subject whose baseline level of stress or propensity to obesity would set the node values. Edge values can be assigned in many ways by using machine learning [51], parsing documents [54, 56], or asking participants in a participatory setting [23, 55]. Second, given a case (i.e. node values) and causal structures (i.e. edge values), an FCM can compute what will *eventually happen.* This is performed by updating the node values in discrete steps, which do not map to physical time. Nonlinearities in the update process (section 2.2) prevent reverse causality, thus an FCM cannot be used to answer *why* questions (i.e. backward chaining) by starting from a desired end state and find what achieves it. Instead, FCMs can only start from a case and work out the implications (i.e. forward chaining), which answers *what-if* questions [9]. The main application of FCMs is thus to support participants in analyzing future scenarios [35] where the evidence base is qualitative, changing, and lacks a commonly accepted 'truth' [34].

There are many participatory approaches to build an FCM [33] whose commonality is to keep the process accessible. For instance, participants are rarely asked to directly provide a number as weight of an edge. Instead, they may be asked to use linguistic variables (e.g., 'very high', 'low') which are turned into numbers using Fuzzy Logic [23, 55], choose from a range (e.g., from '+' to '+++') which is implicitly numerical [28], or the simulation expert may assign weights based on the emphasis of participants [57]. Despite these differences in the building process, all *resulting* FCMs work in the same manner. Consequently, there exists many FCMs to capture the mental models of participants in medical decision-making [1], socio-ecological system management [29], of smart cities [10]. This resource is an important asset to develop agent-based models: rules for agents readily exist in a broad variety of social applications.

2.2 Mathematics of Fuzzy Cognitive Maps

An FCM is a nonlinear dynamical system akin to a neural network. Its structure is a fuzzy signed digraph with feedback, where nodes are fuzzy sets taking values in [0, 1] and edges are fuzzy rules [9]. Nodes are updated based on the value of neighboring nodes and causal edges between them (Equation 1). As updates are repeatedly applied, an FCM converges to "a fixed point, limit cycle, limit torus, or chaotic attractor" [9]. In the first case, we consider that the long-term behavior of the model is known. We thus stop updating an FCM when a desired subset S of output nodes changes from one update to the next by less than ϵ (Equation 2). This halting condition may not be met in the other cases, thus an additional halting condition is a maximum number of steps t_{max}.

Definition 2.1. A Fuzzy Cognitive Map $F^t = (V^t, E, f)$ at step t is composed of [41, 42]:

- A set V^t of n nodes representing concepts. The value of node i is represented by $V_i^t \in [0, 1]$ where 0 and 1 respectively indicate the absence and presence of the concept.
- A set E of edges. Their causal weights are represented by the adjacency matrix A where $A_{i,j}$ is the weight of the edge from i to j. When $A_{i,j}$ is positive then an increase in i causes an increase in j. Conversely, when $A_{i,j}$ is negative, an increase in i causes a decreases in j.
- A clipping function f also known as transfer function. It ensures that updated node values remain the range [0, 1].

The FCM F^{t+1} is updated from F^t by changing the node values:

$$V_i^{t+1} = f\left(V_i^t + \sum_{j=1, j\neq i} V_j^t \times A_{j,i}\right), \tag{1}$$

The update stops when the outputs stabilize, or an exceedingly high number of steps suggests the presence of a chaotic attractor.

$$\text{The FCM stops when: } \begin{cases} |V_i(t+1) - V_i(t)| \leq \epsilon, \forall i \in S \subseteq V, \text{ or} \\ t = t_{max} \end{cases} \tag{2}$$

Note that alternative definitions for an FCM may use a 4-tuple [48] instead of the 3-tuple in definition 2.1. The difference is often due to using separate notations for the concepts as nodes, and for their values. Here, the nodes and their values are conflated using V^t. Another variation is to provide Equation 1 as part of an FCM itself, in replacement of the clipping function f. Intuitively, this allows each FCM to use a different inference engine [27]. In rare cases, an FCM can be specified with as many as 6 tuples [65].

The clipping function f must be monotonic to preserve the order of nodes' values. The choice of f has important consequences on the dynamics and performance of an FCM. A discrete f can produce a finite number of states, such that the FCM converges either to a fixed point or a limit. A continuous f can also lead an FCM to a chaotic attractor [50, 66]. Typical options include a *hyperbolic tangent* [21, 31, 41, 44, 46], which is also commonly used as activation function of artificial neurons, and the sigmoid function, which performs best in simulation benchmarks [7] and produces a unique fixed point when using a small slope [39].

2.3 Simulating Hybrid ABM/FCM Models

While ABMs and FCMs have occasionally been studied together, one may have been used on the way to achieving the other as a final product [12], or one may serve as an abstraction of the other such as when each node of an FCM is simulated as an agent [44, 62]. In contrast, a *hybrid* model requires the *co-existence* of several modelling techniques. In a hybrid ABM/FCM model, the objective is to "quickly developing the system's rule in a participatory way from FCM while obtaining temporal and spatial explicitness from ABM" [22]. This objective can be achieved in two broad ways. First, several ABMs could be embedded within an FCM. For instance, concepts requiring spatial interactions could be simulated using an ABM, whose aggregate value are reflected in the FCM [22]. Second,

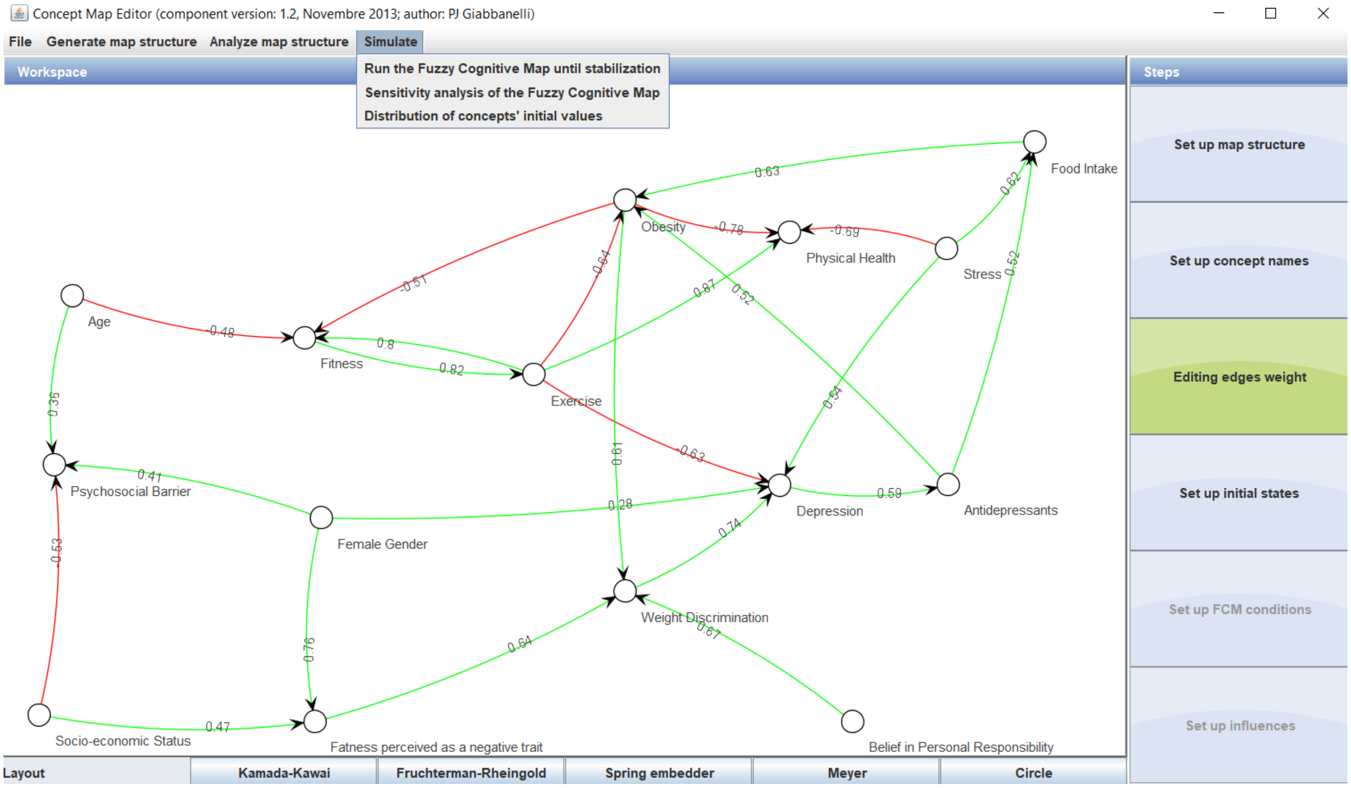

Figure 1: A Fuzzy Cognitive Map in the 2013 HYbridFuzzy-Agents Simulator (`HYFAS`) software [18].

we can embed an FCM within each agent to represent decision-making processes. We have applied this approach to health [23] and security scenarios [18, 55] while others used it to simulate ecosystems [27, 38] or the negotiation surrounding the price of goods [45]. This approach of designing the rules of agents using FCMs is the focus of this article.

A variety of software solutions have been employed where FCMs provide the mental models of agents. The model of Nachazel and the Multi-Agent based MObile Negotiation framework (`MAMON`) are both built in `NetLogo` [45, 47], although the former expressed an interest to "transfer the model from `NetLogo` to `AnyLogic`, which should improve performance" [47]. Although `NetLogo` is an accessible modelling platform, the design of a model requires coding, which participatory approaches avoid. The open-source `EcoSim` platform[1] writes models in C++, which contributes to efficiency as it can "manage several hundreds of thousands of such agents simultaneously into the world with reasonable computational requirements" [38]. The only solution that did not require coding was our HYbridFuzzy-Agents Simulator (`HYFAS`)[2] for Java 7. The paucity of solutions may be partly due to the relative novelty (and hence rarity) of hybrid ABM/FCM models in contrast to hybrids using ABM, System Dynamics, or Discrete Events. This is exemplified by the absence of FCMs among solutions considered by the 2018 panel on hybrid simulations [11].

[1] https://github.com/EcoSimIBM
[2] Available at https://osf.io/z5rf2/ under 'Previous Software'

`HYFAS` provided the backbone of several modelling studies [18, 22, 23]. It relied on a graphical user interface (Figures 1 and 2) to create the model structure and automatically translate it into Java code. Analytical capabilities were offered through several tools to measure the network structure of either the FCM (Figure 1) or the agents' connections (Figure 2-b). Similarly, network generators allowed users to quickly generate the basic structure of the FCM (which often starts as a star graph with one central concept driven by many others), or connect their agents using complex networks with small-world and/or scale-free properties.

3 SOFTWARE SPECIFICATION

3.1 Core Requirements

The overarching aim of `CoFluence` is to empower participants in creating hybrid models where the rules of interacting agents are driven by different fuzzy cognitive maps. This aim translates to three specific requirements, detailed as follows.

> Requirement #1: No coding is required from the users to create, run, and analyze a model.

Fuzzy Cognitive Mapping has achieved a successful track-record in participatory modeling (Section 2.1) in part because it does not require simulation expertise from participants. This removes barriers to participation in a study (when additional training is required),

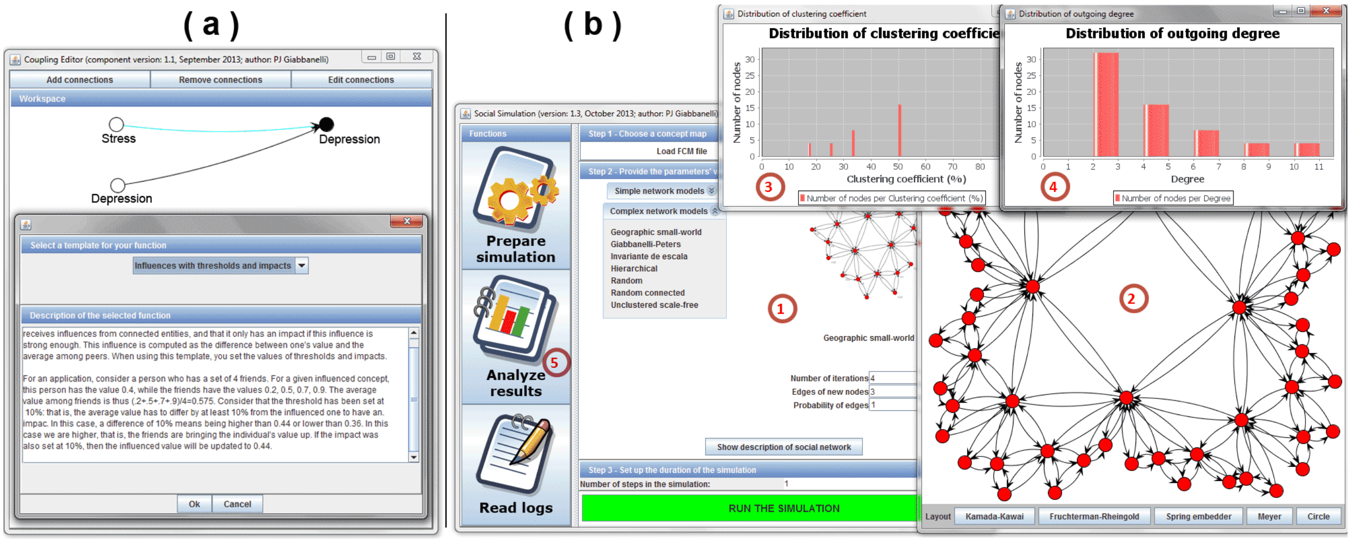

Figure 2: After creating their one FCM, users of HYFAS specify how concepts of an FCM are influenced (a-right) by concepts of connected FCMs (a-left). Then, they set up connections between agents (b-1), and optionally visualize (b-2) or analyze them (b-3; b-4) before investigating simulation results (b-5).

it preserves the transparency of the model building process, and it contributes to buy-in for simulation outcomes. While the introduction of another modelling paradigm (ABM) necessarily increases the complexity of model development, it is essential that it remains accessible for participants. A consequence of this requirement is that any *code has to be generated automatically* by the software, which may not be as efficient as if the code was written and optimized by professionals.

> Requirement #2: Usability requires an agile approach to model development and support to reuse common model components.

While usability is important for all users, participants may be more sensitive to it than simulation experts [16]. Our experiences with `HYFAS` revealed several usability challenges [18, 22, 23], leading to the requirement's emphasis on agile development and reuse. `HYFAS` used a six steps process to create an FCM, where each step had to be completed before the next. For example, the map structure and node names had to be finished before users could provide edge weights (Figure 1). In reality, participants do not necessarily follow such a step-by-step approach to model development. They may sketch some parts of the model, detail others, and revise later. Out of seven recently reviewed FCM simulation software [15], `MentalModeler` stands apart by its focus on helping participants to intuitively build and use FCMs, rather than providing advanced computational options such as machine learning. `MentalModeler` is a case in point of a flexible approach as users can go back and forth between setting edge values, creating new nodes, and so on[3].

[3]A walkthrough of `MentalModeler` is provided at https://www.youtube.com/watch?v=v1A_ZGO6fWk

In short, an agile approach is preferred instead of an overly rigid guidance.

Participants may also realize that they often use the same templates (or 'design patterns') to express how agents influence each other. In models of obesity, social norms on physical activity or food behaviors are often captured by computing the average behavior of connected agents and, if it is above or below a threshold, change the target agent's behavior accordingly [5, 37, 68]. Rather than providing a set, minimal list of building blocks for a model, we thus need a software architecture that flexibly accommodates the use of additional blocks. Examples include data flow software such as `Dataverse`, in which the many building blocks are organized in thematic collections and new ones can be created.

> Requirement #3: Agents may have different rules.

This requirement is more stringent than asking for agents to exhibit different *behaviors*, which may be obtained solely by having different feature values across agents (Section 1). This was the approach taken by `HYFAS`: all agents had the same FCM *structure* (i.e., nodes and edge weights) and only differed in the node values. For instance, obesity would always limit engagement in physical activity (captured by the edge weight), but not all agents were obese (captured by the node value). However, constraining all agents to follow the same rules severely limits heterogeneity, and cannot handle cases where agents of different *types* have different mental models. Participatory approaches are particularly used for problems involving multiple types of stakeholders, resulting in an FCM for each type (e.g., an FCM for elderly patients and another one for caregivers [57]). The simple requirement that different agents use different FCMs has profound consequences on the design of agent interactions (e.g., do we react to a prompt differently when it is expressed by a different type of agent?), and on setting-up the virtual populations (e.g., which agents get which FCM?).

3.2 Design Process

In Spring 2018, requirements from the previous subsection were mapped to user interactions by two independent groups, each composed of two students taking a course on modeling and simulation (cross-listed for graduate and undergraduate students). Each group proposed software mock-ups (Figure 3), and was then given access to the other group's work to critique it. The implementation was guided by a mock-up combining the strengths of both proposals.

Current designs for usability tend to minimize the number of clicks to perform key operations and avoid distracting visuals such as pop-up windows [16, 32]. These principles underlie the design of `CoFluences`. Using a Property panel for each node of the FCM (Figures 3 and 4) achieves a minimum number of clicks, does not require pop-up windows, and supports an agile approach to model design per requirement #2. Operations such as deciding that an FCM node is an output are accomplished with a minimum of two clicks by selecting the node and checking a box. Users can choose to be more detailed about the design of some nodes (e.g., assigning values and flagging for connection with other FCMs) than others (e.g., just assigning names) as they gradually develop the model. In contrast, `HYFAS` needed more clicks (due to pop-up windows) and enforced model development in six consecutive steps.

As agents can have multiple rules (requirement #3), there can be multiple FCM files within a single hybrid model. To emphasize that a model is *composed* of multiple FCMs, the design of `CoFluences` uses a project workspace similarly to simulation software such as `AnyLogic`. All files are listed in a left panel per category (e.g., FCMs, log files) and each open file is available in the central (main) space as a tab (Figure 4) with a dedicated icon to differentiate individual FCMs (Figure 3) from parts such as connecting FCMs (Figure 5).

The design of connections between FCMs is partly inspired by `HYFAS` (Figure 2-a). In short, the user specifies for each node of an FCM whether it will *influence* or *be influenced* by the FCMs of connected agents. Such nodes become available when creating connections, and their effect involves a transformation function. Consider the example depicted in Figure 6: an agent cannot directly drive the level of diabetes of another agent. Instead, diabetes in an agent can contribute to 'disease awareness' in another agent.

Consequently, diabetes is an *influencing* factor, and disease awareness is an *influenced* factor. Such influences can be a one-to-one matching, such as when our diet is directly influenced by the diet of peers. Alternatively, several influences can come together, as is the case when 'trust in local medicine' is driven both by whether other agents trust and use local medicine.

Requirements #1 and #2 strongly shaped our approach to the design of connections. To ensure that users do not have to code, we provide a set of transformation templates arranged in categories, similarly to the building blocks of of `AnyLogic` or `Dataverse`. The main space is divided into three parts, such that templates chosen by the user are positioned in the middle to graphically emphasize that they 'filter' the effect of influencing factors (left) onto influenced factors (right). In line with the typical use of color codes, issues are flagged by coloring a component in bright red. Note that this is the first place where we can automatically identify, and thus display, issues in the model developed by a user. An FCM reflects the knowledge provided by participants, which may be *improved* (e.g., by additional evidence or trained facilitators), but cannot be deemed *wrong* in a mathematical sense. In contrast, different transformation functions can be mathematically inconsistent, which the user needs to see and address before a simulation can proceed.

3.3 Simulation Pseudocode

To specify our simulation, we define the agents and how their interactions are handled by transformations between subsets of FCM nodes. We borrow the definitions of agents (3.1) and subsets of FCM nodes (3.2) from Giabbanelli *et al* [22]. In contrast to previous studies that only provided examples of transformations [22, 55], we formally specify (3.3) and categorize (3.4) the transformations to specify whether they are compatible (3.5).

Definition 3.1. At time step τ of the simulation, a population $\mathbb{A}$ consists of agents, where each one has a Fuzzy Cognitive Map $FCM_a(t), a \in \mathbb{A}$, and a set of agents with which it interacts, $I_a(\tau)$.

Definition 3.2. There are two *categories of FCM nodes*: influenc*ed* and/or influenc*ing*. We represent them by two (possibly overlapping) subsets FCM^{ed} and FCM^{ing} respectively.

Definition 3.3. A *transformation i* is a variadic function $f_i(ed_i, ing_{1,i}, \ldots, ing_{n,i}, p_{1,i}, \ldots, p_{m,i})$ where $ed_i \in FCM^{ed}$ is one influenc*ed* factor, $ing_{1,i}, \ldots, ing_{n,i}$ are influenc*ing* factors, and $p_{1,i}, \ldots, p_{m,i}$ are function parameters.

Definition 3.4. The type $type(f_i)$ of a transformation is either *relative* when its output depends on the influenced factor ed_i, or *absolute* when it does not depend on ed_i.

Definition 3.5. The transformations in a set $\mathbb{F}$ are *compatible* if, $\forall f_i, f_j \in \mathbb{F}, ed_i = ed_j \implies (type(f_i) = relative) \wedge (type(f_j) = relative)$.

We now briefly provide the intuition behind these definitions. Definition 3.3 states that a transformation can affect only *one* influenced concept, based on *one or more* influencing concepts. Consider the following three examples of such a transformation. First, we could replace the influenced concept by the average of an influencing concept among peers. Second, we could increase the influenced concept by 10% whenever it is lower than the average of an influencing concept among peers. Third, we could decrease the influenced concept by 5% whenever it is larger than the maximum for two influencing concepts among peers. The first case takes no parameter, whereas the other two cases take one parameter (set to 10% and 5% respectively). The first case is absolute, because the influenced concept is set to the group's average regardless of its current value. The other two cases are relative, as the next value is based on an increase or decrease of the current one. The two relative cases are compatible: the next value of an influenced concept c would be $c + 0.1c - 0.05c$. However, the first case is incompatible with the others because its result is not fully specified: would it consist of replacing the value by the group's average and then decreasing it by 10%, or first decreasing the value by 10% and then replacing it with the group's average? While this issue can be solved by assigning priorities to transformations, it would allow for superfluous

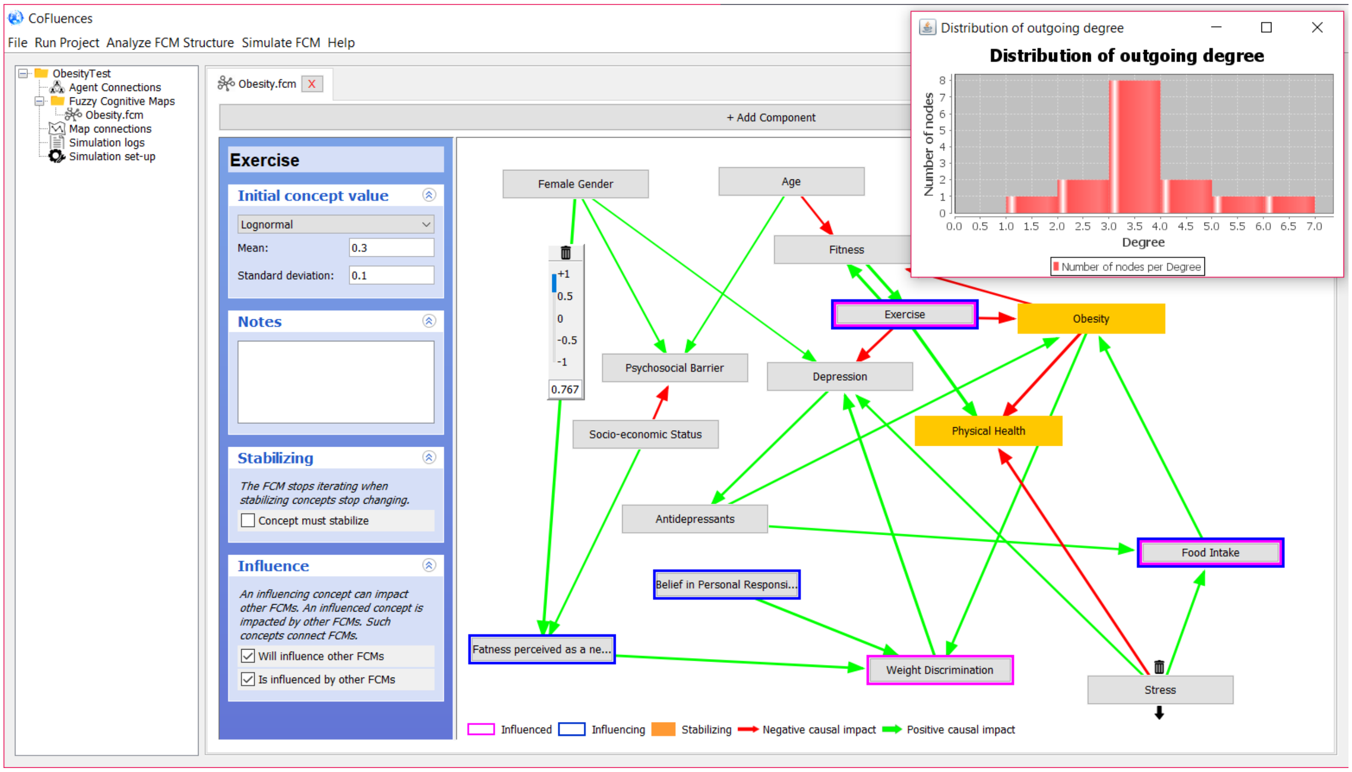

Figure 3: CoFluences showing the same FCM as in Figure 1. Each concept node has a property panel to set its distribution of values across the agents, enter notes, decide whether it's part of set S of outputs (Eq. 2), and if it is influencing or influenced by the FCMs of connected agents. Edge weights are set using a sliding scale (here showing 0.767). When an FCM file is open, the top menu gives access to analytical tools such as the model's outdegree distribution (top-right inset).

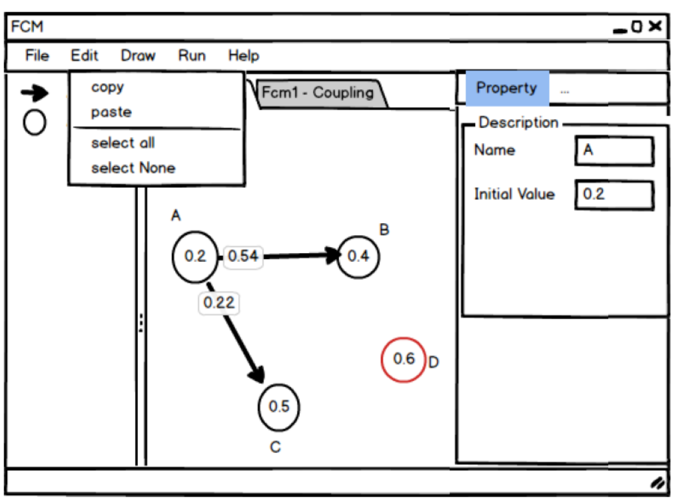

Figure 4: Sample of the mock-up produced in Balsamiq.

computations: there is no point in changing a variable by 10% if its value is then replaced anyway. In Figure 5, the top two transformations are not compatible and the influenced factor is shown in red: the topmost replaces the value by the one used in a randomly selected neighbor, whereas the other can give it a relative increase of 10% under certain conditions. In sum, compatibility ensures that users either apply a series of relative changes (whose order does not matter), or replace the value.

The simulation specified in Algorithm 1 takes turns between applying compatible transformations between agents, and updating their FCMs (Eq. 1) until halting conditions are met (Eq. 2). This captures how agents pass on social norms, change their views of the world accordingly, and continue to shape social norms.

There are two subtleties to Algorithm 1. First, we distinguish the agents' discrete *time* steps (τ) from the FCMs' use of ticks (t) that do not correspond to physical time. A time step of the simulation increments τ, whereas t always starts at 0 (initial mental model for this time step τ) and increases until the final mental model for this time step is obtained (thus forming the initial mental model for the next time step $\tau + 1$). Second, the buffering of transformations (line 9) is necessary *both* for the synchronicity of the updates, and because the same influenced node may be modified many times in the presence of multiple relative transformations.

4 SOFTWARE IMPLEMENTATION

The implementation took place from mid-spring to mid-fall 2018, followed by verification and finally the preparation of educational

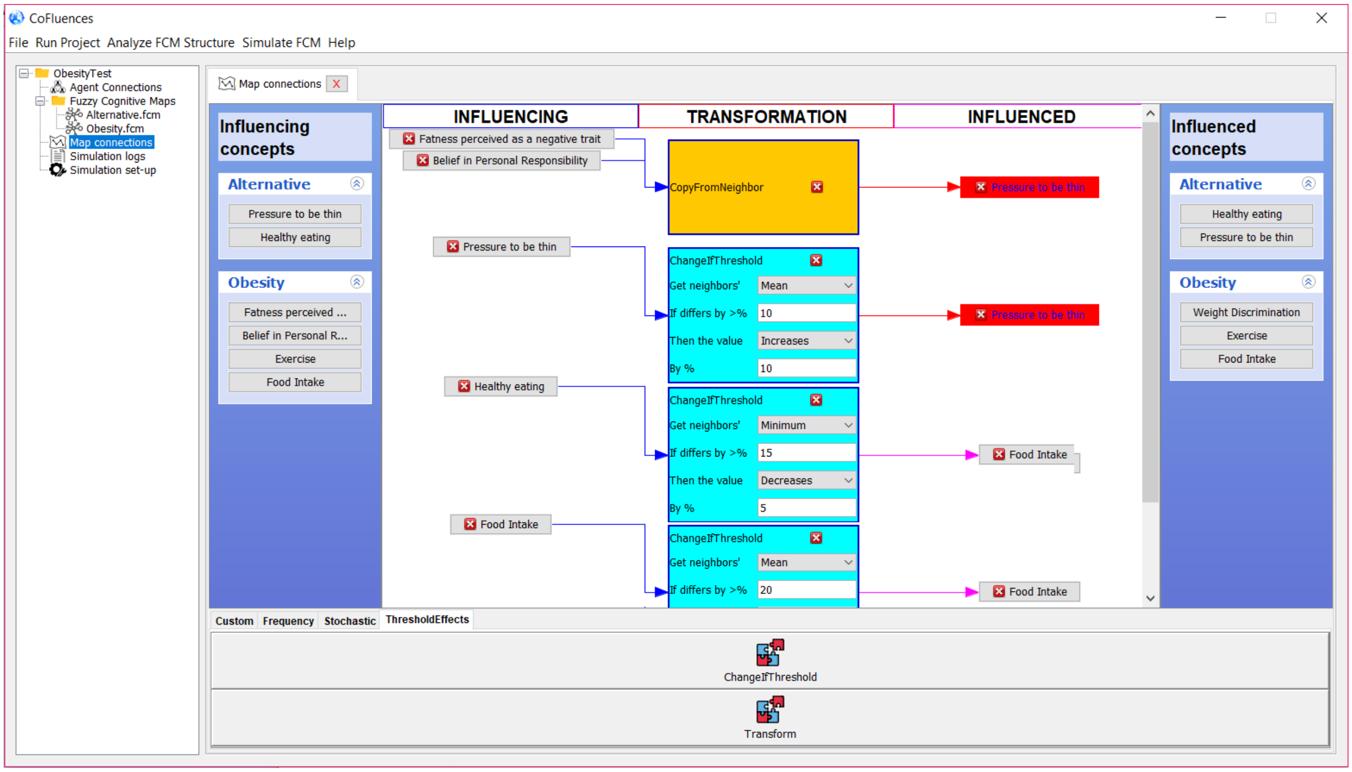

Figure 5: Connecting FCMs in CoFluences consists of choosing transformation templates from categories (bottom tabs) to link nodes that have been flagged as influencing and influenced factors (Figure 4; bottom). Errors are shown in red when design choices are incompatible, such as using an *absolute* and a *relative* template to drive the same influenced concept.

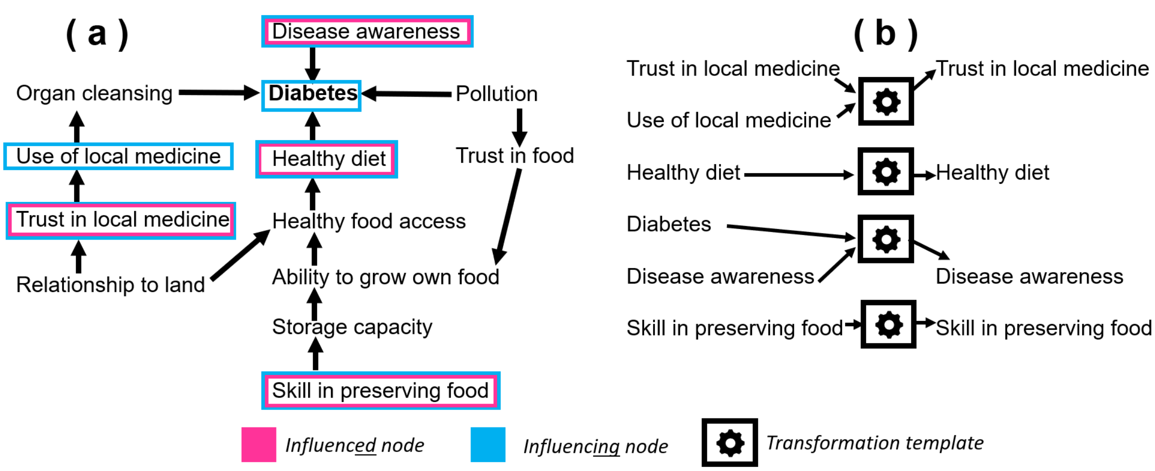

Figure 6: To connect FCMs, we first need to identify nodes that influence or are influenced by others (a; Figure 3). Then, we express how the value of an influenced node depends on the value of influencing node(s) (b; Figure 5). The simplified FCM is adapted from a large model of aboriginal perspectives on diabetes [24].

material. The software was developed in Java 10 using NetBeans[4]. Two libraries are extensively used. Jung (final release 2.0.1) provides data structures, algorithms, and visualizations for graphs. We use it both for the FCM and the agent population. That is, agents and FCM concepts are both nodes, while social ties between agents and causal FCM connections are both edges. JFreeChart (version 1.0.14) provides 2D charts to visualize analytical results. We

[4]The source code and NetBeans project are available at https://osf.io/z5rf2/

Algorithm 1 Updates the simulation for one time step

Require: Compatible transformations
1: *//Agents interact (synchronous time step)*
2: **for** $i \in \mathbb{A}$ **do**
3:   **for** $j \in I_a(\tau)$ **do**
4:     *//for each neighbor j influencing an agent i*
5:     **for** $f(ed, ing_1, \ldots, ing_n, p_1, \ldots, p_m) \in \mathbb{F}$ **do**
6:       **if** $ed \in FCM_i^{ed} \wedge ing_1, \ldots, ing_n \in FCM_j^{ing}$ **then**
7:         *//If there is a transformation f between their FCMs,*
8:         *//then applies and buffers it.*
9:         $V^1_{ed} = f(ed, ing_1, \ldots, ing_n, p_1, \ldots, p_m), V_{ed} \in FCM_i$
10: *//Agents update their mental models in parallel*
11: **for** $i \in \mathbb{A}$ **do**
12:   $t \longleftarrow 0$ *//We have the FCM's initial state*
13:   *//Loads the bufferized changes to the mental models*
14:   $V^0_x \longleftarrow V_x(1), \forall x \in FCM_i^{ed}$
15:   **repeat**
16:     Apply Eq. 1 on $FCM_i(t)$
17:     $t \longleftarrow t + 1$ *//One update of the FCM's states is done*
18:   **until** Eq. 2 is satisfied
19:   *//The final worldview here will start the next time step*
20:   $V^0_x \longleftarrow V^t_x, \forall x \in FCM_i$
21: *//The agents have finished one step of the simulation*
22: $\tau \longleftarrow \tau + 1$

use it to display the results from simulating or analyzing the structure of a single FCM (Figure 3; top-right inset). We also leveraged previously developed and verified code. The update of a single FCM until convergence is managed by code shared with applications such as `HYFAS` [18] and Computational Allocation of Participants in Trials (CAPT) [21][5]. The creation of social ties between agents (Figure 7) draws from a large set of previously developed network generators (e.g., small-world and/or scale-free) [17].

A defining feature of `CoFluences` is its reliance on transformation templates (Figures 5-6) to combine FCMs and ABM. Although our intention is to provide participants with sufficient templates to suit their needs, there is always the possibility of a unique situation or developers wishing to experiment with alternatives at least for theoretical purposes. Consequently, the implementation needs to support the addition of new templates. The worst scenario would be to re-compile the whole application only to make new templates. Instead, our implementation has a pluggable architecture: developers can add new classes (to the 'plugins' folder), and `CoFluences` will compile them at runtime. Consequently, adding a template is achieved by adding a Java file into a folder. Running the application will compile this file into a class file[6], and add it to the list of templates that participants can see. Similarly, creating a new category of templates is achieved simply by making a new folder.

[5]To support reproducibility, the code for both applications can be freely accessed without registration at https://osf.io/z5rf2/ and http://www.crutzen.net/capt/ respectively. In addition, the two peer-reviewed publications of the methodology and software are also open access.

[6]Because of the need to compile templates, `CoFluences` cannot be run with a Java JRE. Instead, it requires the Java JDK. As participants are not assumed to be familiar with these notions, or setting up system variables such as the Path, we provided a wrapper ('CoFluences.exe') to ensure that the application starts with the installed JDK.

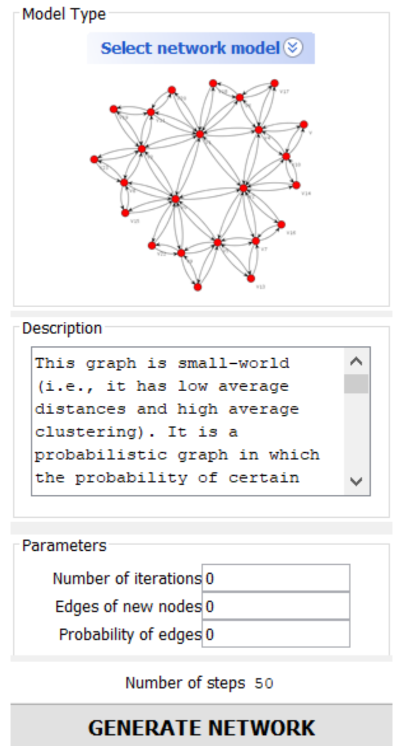

Figure 7: Social ties can be created using several network generators, where each one includes a description with explanations of its properties and links to references.

Mathematically (Def. 3.3) and for the participants, a transformation template is a function taking a list of influencing concepts, a set of parameters, and one influenced concept. The simulation algorithm is responsible for identifying which transformation(s) can be applied between a given agent and its neighbors (Alg. 1, lines 5-6). The implementation must thus bridge the gap between the high-level user specification of the model (which does not specify any agent) and the simulation (which must update specific agents). Consequently, executing a transformation template requires four arguments: the agent being influenced, the agent population (to retrieve the neighboring agents), the name of the FCM concept being influenced, and the name of the influencing concepts. Since the execution method is written within the transformation's class, parameter values are retrieved at simulation time from the graphical components (e.g., drop-down menu, text fields) rather than given as arguments. Executing a template typically starts by retrieving the value of the influenced concept, and of the influencing concepts in the neighbors. All of the transformation templates extend a 'TransformComponent' which provides such retrieval functions. There are two benefits to this technical choice. First, developers wishing to create their own transformation templates can use such functions to lessen their task. Second, a central gateway into retrieving values simplifies the process of monitoring access, which can be used to improve the simulation's efficiency (e.g. by caching).

STEP	A:Healthy eating	A:Obesity	A:Eating disorders	B:Food intake	B:Obesity	B:Stress
0	0.3	0.2	0.1	0.5	0.6	0.8
1	0.25	0.22	0.13	0.45	0.50	0.8
2	0.2	0.23	0.15	0.43	0.49	0.8
3	0.2	0.23	0.16	0.43	0.49	0.8

Table 1: The population log consists of each concept of each FCM, averaged across all of the agents using it at each time step.

STEP	0 - Healthy eating	0 - Obesity	0 - Eating disorders	1 - Healthy eating	1 - Obesity	1 - Eating disorders	2 - Food intake	2 - Obesity	2 - Stress
0	0.2	0.4	0.2	0.4	0	0	0.5	0.6	0.8
1	0.1	0.44	0.26	0.4	0	0	0.45	0.50	0.8
2	0.04	0.46	0.3	0.4	0	0	0.43	0.49	0.8
3	0.04	0.46	0.32	0.4	0	0	0.43	0.49	0.8

Table 2: The individual log shows the value of each FCM concept within each agent (0, 1, 2). In this example, the transformation templates are between agents with FCMs A and B (Table 1). Assuming that agents 0 and 2 respectively use FCMs A and B, *and* share a social tie, then they influence each other. In contrast, agent 1 is not changing either as it is not connected to other agents, or only connected to agent 0 and there is no transformation function defined between agents having both FCM A.

A simulation produces three log files: the simulation setup (e.g. network generator used), population averages over time (Table 1), and each agent over time (Table 2). Note that *time* refers to the ABM (τ) rather than the FCM (t) as we do not export the massive amount of data generated when each FCM is iterated until stabilization. While population averages would typically be used to report final results, tracking each one of the agent allows to verify the results at a more granular level.

Given that the code to generate the social ties or iterate an FCM has been previously verified, the focus of our verification was on the transformation functions. We generated small population of agents with different FCMs, and checked that transformation functions were correctly performed, for instance by randomly copying values from connected neighbors. Upon verifying the software, we produced two educational videos. One serves to explain the mathematical underpinnings of an FCM and its combination with ABM in our software[7], while the other provides an introduction to knowledge representation using FCMs[8].

5 DISCUSSION

The discrete modeling technique of Fuzzy Cognitive Mapping has a successful track-record in participatory modeling to externalize the mental model of individuals, in applications ranging from socio-environmental systems to health [1, 25, 35, 52, 53]. However, two fuzzy cognitive maps cannot directly interact: our thoughts do not directly influence others, so we need to translate these thoughts into observable actions. Agent-based modeling is a commonly used discrete modeling technique in which agents can interact with each other. A challenge is to develop an agent-based model with the level of transparency and intuitiveness required by participatory modeling. In other words, we need a solution to elicit the mental models of participants, *and* embed these mental models into interacting agents. A hybrid approach based on fuzzy cognitive map and agent-based modeling was previously proposed [18, 22, 23], thus contributing to the literature on cognitive architectures [40] and their operationalization in agent-based modeling [8]. However, the only two software aimed at creating such models did not aim to both support participatory modeling and represent agents with different fuzzy cognitive maps. `EcoSim` [26] focused on ecosystems (thus emphasizing reproduction and mutation of models) rather than human systems and required models to be written in C++ [27, 38]. `HYFAS` could create models of human systems without programming, but required all agents to use the same map, which cannot represent settings where multiple stakeholders have different mental models [22]. In this paper, we present the design and implementation of `CoFluences`, the first simulation software that can design hybrid models without coding and allow agents to follow different rules.

`CoFluences` takes a User-Centered Design (UCD) approach to map key software requirements (e.g., agile approach to model development, emphasis on 'building blocks') onto a specific interface and interactions. `CoFluences` also contributes to the nascent theory of hybrid ABM/FCM models by detailing how agents influence each other through *transformation templates*. In particular, we introduce the concept of *compatible* transformations to automatically verify whether the influences envisioned by the user can be simulated or should be re-conceptualized.

There are several follow-up studies of interest regarding software development. As we recently discussed, the *usability* of large FCM-based software has to be evaluated in detail with participants [54]. A usability study would have at least a six months horizon, as it involves (i) identifying key tasks and performance metrics, (ii) enrolling a representative sample of participants into the study, (iii) analyzing recordings to compute the metrics when participants perform the task, and finally (iv) assessing what design elements drive the performance metrics. While `CoFluences` is based on current design principles and lessons learned from other FCM-based software, a usability study is necessary to examine how it supports participatory modeling specifically.

[7] https://www.youtube.com/watch?v=laXOJWUDYZY
[8] https://www.youtube.com/watch?v=D-2Q2IHclo4

Another follow-up is the matter of *standardization* across FCM-based software. There are at least seven software to create fuzzy cognitive mapping [15], several Python[9] and R[10] libraries, some software relying on an FCM for analysis [21, 54], and software for hybrid simulations involving FCM [22, 27, 38]. Despite the explosion of software development in fuzzy cognitive mapping, there is no standard file format, which particularly hampers efforts to re-use previously developed FCMs or even to compare the performances of different simulation software on the same model. Standardization is thus essential for the maturity of the field.

A third interesting line of research consists of either broadening what can be represented by hybrid ABM/FCM models, or understanding the consequences of limitations inherent to this modeling approach. In particular, hybrid ABM/FCM models work in a time-stepped manner. Consequently, the model requires that social interactions between an individual and peers take place at the same time and with the same speed. However, heterogeneity in propagation speeds over different links does occur in the real-world, and may affect the convergence of the FCMs.

Convergence is also at the heart of several theoretical questions on the development of hybrid ABM/FCM models. Running an FCM *once* is computationally inexpensive, thus it is straightforward to detect when an FCM does not stabilize: it simply reaches the maximum number of iterations. A stochastic use of an FCM to represent a population of agents significantly raises the computational costs of detecting non-stabilizing cases. Indeed, given the probability distribution associated with each FCM concept in a population, we would have to instantiate an FCM for every combination of values, and assess if it stabilizes. The computational burden could be lessened using Design of Experiments (DoE) techniques to identifying which concepts truly need to be tested at different values [41]. New situations may emerge in hybrid ABM/FCM models. Could the external inputs received from connected agents turn a stable FCM into an unstable one? Conversely, could an unstable FCM become stable when receiving inputs from connected agents? Characterizing situations in which an FCM gains or loses stability would help to automatically verify the design of hybrid models.

Altering the design of a model to avoid non-stabilizing FCMs would improve the computational efficiency of a simulation, since the FCMs would take fewer iterations. But is the impact of non-stabilizing FCMs only a matter of simulation performance, or also a matter of correctness in the results? Instead of using the *final* mental model of an agent to affect another, a non-stabilizing FCM would lead to using the *transient* mental model of an agent. Intuitively, an agent would act before it has finished 'thinking'. Future research may thus characterize how the (expected) presence of non-stabilizing FCMs impacts the confidence margin in the simulation outcomes.

6 CONCLUSION

We presented `CoFluences`, the first software that supports participants in developing hybrid ABM/FCM models where agents can follow different rules. Our software follows design principles that favor usability in participatory modeling, and proposes technical innovations such as checking for compatibility in interactions between agents. We outlined several research questions pertaining to either software development (e.g., standardization, usability testing) or the theory of modeling and simulation (e.g., criteria and consequences for the stability of FCMs in a hybrid setting). Addressing these questions will provide a fertile agenda for fuzzy cognitive mapping in the coming years.

CONTRIBUTIONS

PJG supervised the project, wrote the initial version of `CoFluences` and of the manuscript. MF oversaw the re-design of the software, and re-wrote it extensively. MLN prepared educational videos and contributed to the writing.

ACKNOWLEDGMENTS

We are indebted to Michael Vick and his team at Furman University for producing a series of educational videos in support of `CoFluences`. In particular, we thank Hans Stoetzer for presenting a portion of these videos. The design of `CoFluences` benefited from the work of Joseph Lee, Christopher Orsolini, and Austin Sommerlot at Northern Illinois University.

REFERENCES

[1] Abdollah Amirkhani et al. 2017. A review of fuzzy cognitive maps in medicine: Taxonomy, methods, and applications. *Computer methods and programs in biomedicine* 142 (2017), 129–145.
[2] Robert Axelrod. 1997. Advancing the art of simulation in the social sciences. In *Simulating social phenomena*. Springer, 21–40.
[3] Christopher L Barrett et al. 2008. EpiSimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks. In *High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for*. IEEE, 1–12.
[4] Joshua Becker, Devon Brackbill, and Damon Centola. 2017. Network dynamics of social influence in the wisdom of crowds. *Proceedings of the national academy of sciences* (2017), 201615978.
[5] Rahmatollah Beheshti, Mehdi Jalalpour, and Thomas A Glass. 2017. Comparing methods of targeting obesity interventions in populations: An agent-based simulation. *SSM-population health* 3 (2017), 211–218.
[6] Pierre Bommel et al. 2016. Cormas: an agent-based simulation platform for coupling human decisions with computerized dynamics. In *Simulation and Gaming in the Network Society*. Springer, 387–410.
[7] Salvador Bueno and Jose L Salmeron. 2009. Benchmarking main activation functions in fuzzy cognitive maps. *Expert Systems with Applications* 36, 3 (2009), 5221–5229.
[8] Philippe Caillou et al. 2017. A Simple-to-use BDI architecture for Agent-based Modeling and Simulation. In *Advances in Social Simulation 2015*. Springer, 15–28.
[9] Julie A Dickerson and Bart Kosko. 1994. Virtual worlds as fuzzy cognitive maps. *Presence: Teleoperators & Virtual Environments* 3, 2 (1994), 173–189.
[10] Sara D'Onofrio, Elpiniki Papageorgiou, and Edy Portmann. 2019. Using Fuzzy Cognitive Maps to Arouse Learning Processes in Cities. In *Designing Cognitive Cities*. Springer, 107–130.
[11] Tillal Eldabi et al. 2018. Hybrid simulation challenges and opportunities: a life-cycle approach. *Proceedings of the Winter Simulation Conference* (2018), 1500–1514.
[12] Sondoss Elsawah et al. 2015. A methodology for eliciting, representing, and analysing stakeholder knowledge for decision making on complex socio-ecological systems: From cognitive maps to agent-based models. *Journal of environmental management* 151 (2015), 500–516.
[13] Joshua M Epstein. 2001. Learning to be thoughtless: Social norms and individual computation. *Computational economics* 18, 1 (2001), 9–24.
[14] Roozbeh Farahbod and Uwe Glässer. 2011. The CoreASM modeling framework. *Software: Practice and Experience* 41, 2 (2011), 167–178.
[15] Gerardo Felix et al. 2017. A review on methods and software for fuzzy cognitive maps. *Artificial Intelligence Review* (2017), 1–31.
[16] PJ Giabbanelli, RA Flarsheim, CX Vesuvala, and L Drasic. 2016. Developing technology to support policymakers in taking a systems science approach to obesity and well-being. *Obesity Reviews* 17 (2016), 194–195.

[9] See https://github.com/payamaminpour/PyFCM/wiki and https://osf.io/qyujt/
[10] https://cran.r-project.org/web/packages/fcm

[17] Philippe J. Giabbanelli. 2010. Impact of complex network properties on routing in backbone networks. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*. IEEE, 389–393.

[18] Philippe J Giabbanelli. 2014. Modelling the spatial and social dynamics of insurgency. *Security Informatics* 3, 1 (2014), 2.

[19] Philippe J Giabbanelli. 2018. Analyzing the Complexity of Behavioural Factors Influencing Weight in Adults. In *Advanced Data Analytics in Health*. Springer, 163–181.

[20] Philippe J Giabbanelli and Magda Baniukiewicz. 2018. Navigating complex systems for policymaking using simple software tools. In *Advanced Data Analytics in Health*. Springer, 21–40.

[21] Philippe J Giabbanelli and Rik Crutzen. 2014. Creating groups with similar expected behavioural response in randomized controlled trials: a fuzzy cognitive map approach. *BMC medical research methodology* 14, 1 (2014), 130.

[22] Philippe J Giabbanelli, Steven A Gray, and Payam Aminpour. 2017. Combining fuzzy cognitive maps with agent-based modeling: Frameworks and pitfalls of a powerful hybrid modeling approach to understand human-environment interactions. *Environmental Modelling & Software* 95 (2017), 320–325.

[23] Philippe J Giabbanelli, Piper J Jackson, and Diane T Finegood. 2014. Modelling the joint effect of social determinants and peers on obesity among canadian adults. In *Theories and simulations of complex social systems*. Springer, 145–160.

[24] Brian G Giles et al. 2007. Integrating conventional science and aboriginal perspectives on diabetes using fuzzy cognitive maps. *Social science & medicine* 64, 3 (2007), 562–576.

[25] Michael Glykas. 2010. *Fuzzy cognitive maps: Advances in theory, methodologies, tools and applications*. Vol. 247. Springer.

[26] Robin Gras et al. 2011. EcoSim: an individual-based platform for studying evolution.. In *ECAL*. 284–285.

[27] Robin Gras, Didier Devaurs, Adrianna Wozniak, and Adam Aspinall. 2009. An individual-based evolving predator-prey ecosystem simulation using a fuzzy cognitive map as the behavior model. *Artificial life* 15, 4 (2009), 423–463.

[28] Steven Gray, Johanna Hilsberg, Andrew McFall, and Robert Arlinghaus. 2015. The structure and function of angler mental models about fish population ecology: the influence of specialization and target species. *Journal of Outdoor Recreation and Tourism* 12 (2015), 1–13.

[29] Steven A Gray et al. 2015. Using fuzzy cognitive mapping as a participatory approach to analyze change, preferred states, and perceived resilience of social-ecological systems. *Ecology and Society* 20, 2 (2015).

[30] Steven A Gray, Stefan Gray, Linda J Cox, and Sarah Henly-Shepard. 2013. Mental modeler: a fuzzy-logic cognitive mapping modeling tool for adaptive environmental management. In *System sciences (hicss), 2013 46th Hawaii international conference on*. IEEE, 965–973.

[31] Peter P. Groumpos and Chrysostomos D. Stylios. 2000. Modelling supervisory control systems using fuzzy cognitive maps. *Chaos, Solitons & Fractals* 11, 1âĂŞ3 (2000), 329 – 336.

[32] Vishrant K Gupta, Philippe J Giabbanelli, and Andrew A Tawfik. 2018. An Online Environment to Compare Students' and Expert Solutions to Ill-Structured Problems. In *International Conference on Learning and Collaboration Technologies*. Springer, 286–307.

[33] Patrick T Hester and Kevin MacG Adams. 2017. Complex Systems Modeling. In *Systemic Decision Making*. Springer, 101–125.

[34] Antonie J Jetter. 2006. Fuzzy cognitive maps for engineering and technology management: What works in practice? *PICMET* (2006).

[35] Antonie J Jetter and Kasper Kok. 2014. Fuzzy Cognitive Maps for futures studies–A methodological assessment of concepts and methods. *Futures* 61 (2014), 45–57.

[36] Rebecca Jordan et al. 2018. Twelve questions for the participatory modeling community. *Earth's Future* 6, 8 (2018), 1046–1057.

[37] Amin Khademi et al. 2018. An Agent-Based Model of Healthy Eating with Applications to Hypertension. In *Advanced Data Analytics in Health*. Springer, 43–58.

[38] Marwa Khater, Elham Salehi, and Robin Gras. 2012. The emergence of new genes in ecosim and its effect on fitness. In *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 52–61.

[39] Christopher JK Knight, David JB Lloyd, and Alexandra S Penn. 2014. Linear and sigmoidal fuzzy cognitive maps: an analysis of fixed points. *Applied Soft Computing* 15 (2014), 193–202.

[40] Iuliia Kotseruba and John K Tsotsos. 2018. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review* (2018), 1–78.

[41] Eric A Lavin et al. 2018. Should we simulate mental models to assess whether they agree?. In *Proceedings of the Annual Simulation Symposium*. Society for Computer Simulation International, 6.

[42] Eric A Lavin and Philippe J Giabbanelli. 2017. Analyzing and simplifying model uncertainty in fuzzy cognitive maps. In *Simulation Conference (WSC), 2017 Winter*. IEEE, 1868–1879.

[43] David Lazer et al. 2009. Life in the network: the coming age of computational social science. *Science (New York, NY)* 323, 5915 (2009), 721.

[44] Kun Chang Lee et al. 2013. An agent-based fuzzy cognitive map approach to the strategic marketing planning for industrial firms. *Industrial Marketing Management* 42, 4 (2013), 552–563.

[45] Kun Chang Lee, Habin Lee, and Namho Lee. 2012. Agent based mobile negotiation for personalized pricing of last minute theatre tickets. *Expert systems with applications* 39, 10 (2012), 9255–9263.

[46] Vijay K. Mago et al. 2013. Analyzing the impact of social factors on homelessness: a Fuzzy Cognitive Map approach. *BMC Medical Informatics and Decision Making* 13, 1 (2013), 94.

[47] Tomá Nacházel. 2015. Optimization of decision-making in artificial life model based on fuzzy cognitive maps. In *Intelligent Environments (IE), 2015 International Conference on*. IEEE, 136–139.

[48] Gonzalo Nápoles et al. 2015. A computational tool for simulation and learning of Fuzzy Cognitive Maps. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*. IEEE, 1–8.

[49] Gonzalo Nápoles, Maikel Leon Espinosa, Isel Grau, and Koen Vanhoof. 2018. FCM Expert: Software Tool for Scenario Analysis and Pattern Classification Based on Fuzzy Cognitive Maps. *International Journal on Artificial Intelligence Tools* 27, 07 (2018), 1860010.

[50] Gonzalo Nápoles, Elpiniki Papageorgiou, Rafael Bello, and Koen Vanhoof. 2017. Learning and convergence of fuzzy cognitive maps used in pattern recognition. *Neural Processing Letters* 45, 2 (2017), 431–444.

[51] Elpiniki I Papageorgiou. 2012. Learning algorithms for fuzzy cognitive maps–a review study. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 2 (2012), 150–163.

[52] Elpiniki I Papageorgiou. 2013. *Fuzzy cognitive maps for applied sciences and engineering: from fundamentals to extensions and learning algorithms*. Vol. 54. Springer Science & Business Media.

[53] Elpiniki I Papageorgiou and Jose L Salmeron. 2013. A review of fuzzy cognitive maps research during the last decade. *IEEE Transactions on Fuzzy Systems* 21, 1 (2013), 66–79.

[54] Venkata Sai Pillutla and Philippe J. Giabbanelli. 2019. Iterative generation of insight from text collections through mutually reinforcing visualizations and fuzzy cognitive maps. *Applied Soft Computing* 76 (2019), 459 – 472.

[55] Simon F Pratt et al. 2012. Rebel with many causes: A computational model of insurgency. In *Intelligence and Security Informatics (ISI), 2012 IEEE International Conference on*. IEEE, 90–95.

[56] Simon F Pratt et al. 2013. Detecting unfolding crises with visual analytics and conceptual maps emerging phenomena and big data. In *Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on*. IEEE, 200–205.

[57] Noshad Rahimi, Antonie J Jetter, Charles M Weber, and Katherine Wild. 2018. Soft Data Analytics with Fuzzy Cognitive Maps: Modeling Health Technology Adoption by Elderly Women. In *Advanced Data Analytics in Health*. Springer, 59–74.

[58] Ken Resnicow and Roger Vaughan. 2006. A chaotic view of behavior change: a quantum leap for health promotion. *International Journal of Behavioral Nutrition and Physical Activity* 3, 1 (2006), 25.

[59] Amanda Rivis and Paschal Sheeran. 2003. Descriptive norms as an additional predictor in the theory of planned behaviour: A meta-analysis. *Current Psychology* 22, 3 (2003), 218–233.

[60] David A Shoham et al. 2015. Modeling social norms and social influence in obesity. *Current epidemiology reports* 2, 1 (2015), 71–79.

[61] F Marijn Stok et al. 2014. Don't tell me what I should do, but what others do: The influence of descriptive and injunctive peer norms on fruit consumption in adolescents. *British journal of health psychology* 19, 1 (2014), 52–64.

[62] Maja Stula, Darko Stipanicev, and Ljiljana Bodrozic. 2010. Intelligent modeling with agent-based fuzzy cognitive map. *International journal of intelligent systems* 25, 10 (2010), 981–1004.

[63] Antuela A. Tako and Kathy Kotiadis. 2018. Participative Simulation (PARTISUM): a facilitated simulation approach for stakeholder engagement. *Proceedings of the Winter Simulation conference* (2018), 192–206.

[64] Simon JE Taylor et al. 2018. Crisis, What Crisis: Does Reproducibility in Modeling & Simulation Really Matter? *Proceedings of the Winter Simulation Conference* (2018).

[65] Jacques Tisseau et al. 2005. Comportements perceptifs d'acteurs virtuels autonomes. *Technique et Science Informatiques* 24 (2005), 1259–1293.

[66] Athanasios K Tsadiras. 2008. Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. *Information Sciences* 178, 20 (2008), 3880–3894.

[67] Alexey Voinov et al. 2018. Tools and methods in participatory modeling: Selecting the right tool for the job. *Environmental Modelling & Software* 109 (2018), 232–255.

[68] Donglan Zhang et al. 2014. Impact of different policies on unhealthy dietary behaviors in an urban adult population: an agent-based simulation model. *American journal of public health* 104, 7 (2014), 1217–1222.