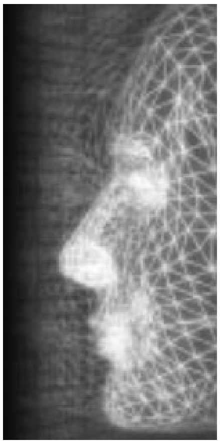
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2010; **21**: 573–587
Published online 27 September 2010 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/cav.363

Fuzzy cognitive maps for the simulation of individual adaptive behaviors



By C. Buche, P. Chevaillier, A. Nédélec, M. Parenthoën and J. Tisseau*

This paper focuses on the simulation of behavior for autonomous entities in virtual environments. The behavior of these entities must determine their responses not only to external stimuli, but also with regard to internal states. We propose to describe such behavior using fuzzy cognitive maps (FCMs), whereby these internal states might be explicitly represented. This paper presents the use of FCMs as a tool to specify and control the behavior of individual agents. First, we describe how FCMs can be used to model behavior. We then present a learning algorithm allowing the adaptation of FCMs through observation. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: behavioral simulation; cognitive map; agent modeling; learning by imitation

Introduction

To date, the majority of work on virtual reality has been related to the multi-sensory immersion of human users within virtual worlds. These increasingly realistic universes will continue to lack credibility as long as they are not populated with autonomous entities, i.e., having the ability to perceive and act in the environment according to their decision-making capacities. This overall autonomy is based on three forms: (1) sensorimotor autonomy, where each entity is equipped with sensors and effectors enabling it to be informed about and act upon its environment, (2) autonomy of execution, where the execution controller of each entity is independent from the controllers of other entities, and (3) autonomy of decision, where each entity makes decisions in accordance with its own experience of the environment (past experience, intentions, emotional state, perceptions, etc.).

For simulation purposes, decision-making by autonomous entities is defined according not only to external stimuli, but also to internal states. In this paper, we show that such behaviors can be described using fuzzy cognitive maps (FCMs) in which these internal states will be explicitly represented. The strengths of FCMs lie in the fact that they can be used to graphically represent specific behavior in the form of a semantic graph and that their evaluation at run-time is fast enough to meet the requirements of real-time simulations, as do connectionist architectures.

FCMs are the outcome of research by psychologists. In 1948, Tolman introduced the key concept of "cognitive maps" to describe complex topological memorizing behavior in rats.[1] In the seventies, Axelrod described "cognitive maps" as directed, inter-connected, bilevel-valued graphs, and used them in decision theory applied to the political-economics field.[2] In 1986, Kosko extended the graphs of Axelrod to the fuzzy mode which thus became FCM.[3] In 1994, Dickerson and Kosko proposed the use of FCMs to obtain overall virtual world modeling.[4] Recently, FCMs have been successfully used to describe and model complex dynamic systems,[5] both for medical diagnosis[6] and in decision-making.[7]

In all these studies, FCMs have been used to control a global system. Here, we propose to decentralize FCMs onto each agent, in order to model the agents' decisions within a virtual world. This paper proposes the use of FCMs as a tool to model the reactive and adaptive behavior of agents improvising in free interaction.

The paper is organized as follows. First, we highlight the fact that FCMs are particularly well adapted to specifying and controlling agents' decisions. We present the uses of FCMs in reactive behavior, illustrating these uses with a real-life example involving different types of agent: a shepherd, dogs, and a herd of sheep. Second, we describe the ability provided for agents to adapt their representation of other actors' behavior using FCMs, which leads to their predictions becoming more significant. This means that we give an agent the ability to learn through

*Correspondence to: C. Buche, UEB - ENIB - LISyC, CERV - 25 rue Claude Chappe, F-29280 Plouzané, France. E-mail: buche@enib.fr

imitation. The agent is then able to modify its own behavior to mimic an observed behavior by either another actor or an avatar controlled by a human operator. By observing the imitated model, the agent must adapt its representation of the model behavior. The mechanism used to control the imitating behavior model is independent of learning. Thus, imitated models can be driven by any decision-making mechanism. We apply this algorithm to the example given above (a sheepdog herding sheep). The learning mechanism allows the dog to adapt an FCM prey prototype to a given sheep in real time.

Reactive Behavior with Fuzzy Cognitive Maps

FCM Presentation

FCMs are influence graphs (see Figure 1). Nodes are named by concepts forming the set of concepts $\mathcal{C} = \{C_1, \ldots, C_n\}$. Edges (C_i, C_j) are oriented and represent the causal links between concepts (how concept C_i causes concept C_j). Edges are elements of the set $\mathcal{A} = \{(C_i, C_j)_{ij}\} \subset \mathcal{C} \times \mathcal{C}$. The edges' weights are associated with a link value matrix: $L_{ij} \in M_n(\mathbb{R})$. If $(C_i, C_j) \notin \mathcal{A}$ then $L_{ij} = 0$, else the excitation link (and inhibition link respectively) from concept C_i to concept C_j gives $L_{ij} > 0$ ($L_{ij} < 0$ respectively).

FCM concept activations take their value from an activation degree set $\mathcal{V} = \{0, 1\}$ or $\{-1, 0, 1\}$, or an interval. At moment $t \in \mathbb{N}$, each concept C_i is associated with two types of activations: an internal activation degree $a_i(t) \in \mathcal{V}$ and an external forced activation value $f_{a_i}(t) \in \mathbb{R}$. $a(0) = \mathbf{0}$, where $\mathbf{0}$ is the $\mathbb{R}^n$ null vector.

FCMs are dynamic systems. The dynamic obeys a recurring relationship involving link matrix products with internal activation vectors, and fuzzy logical operators between this result and external forced activation vectors.

Formalization of the FCM Dynamic

Until the end of this paper, δ indicates one of the numbers 0 or 1, and $\mathcal{V}$ one of the sets $\{0, 1\}$, $\{-1, 0, 1\}$, or $[-\delta, 1]$. Given $n \in \mathbb{R}^*$, $t_0 \in \mathbb{N}$, $\rho \in \mathbb{R}^*_+$, and $a_0 \in \mathbb{R}$.

An FCM $\mathcal{F}$ is a sextuplet $< \mathcal{C}, \mathcal{A}, L, A, f_a, \mathcal{R} >$ where:

1. $\mathcal{C} = \{C_1, \cdots, C_n\}$ is the set of n concepts forming the nodes of a graph.
2. $\mathcal{A} \subset \mathcal{C} \times \mathcal{C}$ is the set of edges (C_i, C_j) directed from C_i to C_j.
3. $L : \left| \begin{array}{lcl} \mathcal{C} \times \mathcal{C} & \to & \mathbb{R} \\ (C_i, C_j) & \mapsto & L_{ij} \end{array} \right.$ is a function $\mathcal{C} \times \mathcal{C}$ to $\mathbb{R}$ associating a weight L_{ij} to a pair of concepts (C_i, C_j), with $L_{ij} = 0$ if $(C_i,\ C_j) \notin \mathcal{A}$, or with L_{ij} equal to the weight of the edge directed from C_i to C_j if $(C_i,\ C_j) \in \mathcal{A}$. $L(\mathcal{C} \times \mathcal{C}) = (L_{ij}) \in \mathbb{R}^{n \times n}$ is a matrix of $\mathcal{M}_n(\mathbb{R})$. It is the link matrix of the map $\mathcal{F}$ that, to simplify, we note L unless indicated otherwise.
4. $A : \left| \begin{array}{lcl} \mathcal{C} & \to & \mathcal{V}^{\mathbb{N}} \\ C_i & \mapsto & a_i \end{array} \right.$ is a function that maps each concept C_i to the sequence of its activation degrees such as for $t \in \mathbb{N}$, $a_i(t) \in \mathcal{V}$ is its activation degree at the moment t. We note $a(t) = [(a_i(t))_{i \in [\![1,n]\!]}]^{\mathrm{T}}$ the vector of activations at the moment t.
5. $f_a \in (\mathbb{R}^n)^{\mathbb{N}}$ is a sequence of vectors of forced activations such as for $i \in [\![1, n]\!]$ and $t \geq t_0$, $f_{a_i}(t)$ is the forced activation of concept C_i at the moment t.
6. $(\mathcal{R})$ is a recurring relationship on $t \geq t_0$ between $a_i(t+1)$, $a_i(t)$ and $f_{a_i}(t)$ for $i \in [\![1, n]\!]$ indicating the dynamics of the map $\mathcal{F}$.

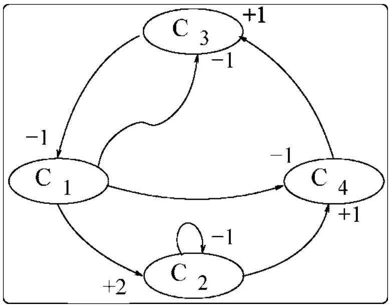


Figure 1. An FCM as an influence graph. The FCM seen opposite is made up of four concepts and has seven edges. At a moment t, each concept C_i has a degree of activation $a_i(t)$. The activation $a(t) \in \mathcal{V}^4$ and the links $L \in \mathcal{M}_4(\mathbb{R})$ are:

$$a(t) = \begin{pmatrix} a_1(t) \\ a_2(t) \\ a_3(t) \\ a_4(t) \end{pmatrix} \quad L = \begin{pmatrix} 0 & +2 & -1 & -1 \\ 0 & -1 & 0 & +1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 \end{pmatrix}$$

. One zero in the links matrix $L_{ij} = 0$ indicates the absence of edges from concept C_i to concept C_j, and a diagonal element $L_{ii} \neq 0$ corresponds to an edge from concept C_i to itself.

$$(\mathcal{R}) : \forall i \in [\![1, n]\!],\ \forall t \geq t_0,$$

$$\begin{cases} a_i(t_0) = 0 \\ a_i(t+1) = \sigma\left[g_i\left(f_{a_i}(t), \sum_{j \in [\![1,n]\!]} L_{ji} a_j(t)\right)\right] \end{cases} \tag{1}$$

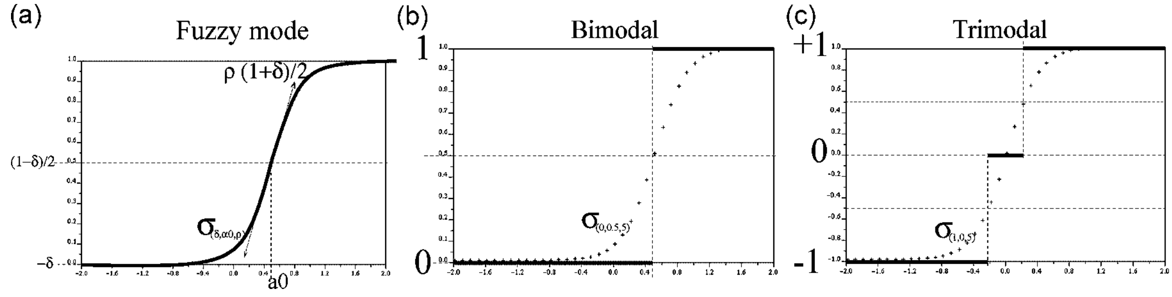


Figure 2. Cognitive maps' standardizing functions.

where $g_i : \mathbb{R}^2 \to \mathbb{R}$ is an operator combining two variables, for example:

$$g_i(x, y) = \min(x, y), \text{ or } \max(x, y), \text{ or } \alpha_i x + \beta_i y, \ \ldots$$

and where $\sigma : \mathbb{R} \to \mathcal{V}$ is a function from $\mathbb{R}$ to the set of activation degrees $\mathcal{V}$ normalizing the activations as follows (see Figure 2):

(a) If continuous mode (called fuzzy mode), $\mathcal{V} = [-\delta, 1]$, σ is the sigmoid function $\sigma_{(\delta,a_0,\rho)} : a \mapsto \frac{1+\delta}{1+e^{-\rho(a-a_0)}} - \delta$ centered in $(a_0, \frac{1-\delta}{2})$, with a slope $\rho \cdot \frac{1+\delta}{2}$ in a_0 and with limits at $\pm\infty$ respectively 1 and $-\delta$. The larger ρ is, the less linear the transformation will be. In practice, $\delta \in \{0, 1\}$: 0 stands for a bivalued discrete logic or a fuzzy logic with values in $[0, 1]$, while 1 corresponds to trivalued discrete logic or a fuzzy logic with values in $[-1, 1]$.

(b) If bimodal mode, $\mathcal{V} = \{0, 1\}$, $\sigma : a \mapsto$ $\left| \begin{array}{l} 0 \text{ if } \sigma_{(0,0.5,\rho)}(a) \le 0.5 \\ 1 \text{ if } \sigma_{(0,0.5,\rho)}(a) > 0.5 \end{array} \right.$.

(c) If ternary mode, $\mathcal{V} = \{-1, 0, 1\}$, $\sigma : a \mapsto$ $\left| \begin{array}{rl} -1 \text{ if } & \sigma_{(1,0,\rho)}(a) \le -0.5 \\ 0 \text{ if } & -0.5 < \sigma_{(1,0,\rho)}(a) \le 0.5 \\ 1 \text{ if } & \sigma_{(1,0,\rho)}(a) > 0.5 \end{array} \right.$.

The asymptotic behavior ($t \to +\infty$) of FCMs with constant externally forced activation vector sequence may be a fixed point, a limit cycle, or even a strange attractor if it is sufficiently complex.[3]

FCM for which Behavior?

It is difficult to describe the entire behavior of a complex system with a precise mathematical model.[8] It is therefore more useful to represent it graphically, showing the causal relationships between the concepts involved. Therefore, FCM can avoid many of the knowledge-extraction problems which are usually present in rule-based systems.[9]

FCMs are capable of forward chaining only, i.e., they can answer the question "What would happen if...?", but not "Why...?", due to the nonlinear nature of FCM dynamics. FCMs help predict the system's evolution (behavioral simulation) and can be equipped with Hebbian learning capabilities, as proposed by Dickerson and Kosko.[4] The fundamental difference between an FCM and a neural network (NN) is that, in FCM, all nodes have strong semantics defined by the concept's model, whereas internal nodes in NN graphs have weak semantics which are only defined by mathematical relationships. As regards the learning capacity, during the training phase, activation vectors must be given for all concepts of FCMs, whereas for NNs, activation vectors are only required for the peripheral neurons.

Constructing FCM

FCMs use symbolic representations which are able to incorporate experts' knowledge.[10–12] Human experts have drawn up FCMs for planning or making decisions in the field of international relations and political developments,[13] to model intruder detection systems[14] and demonstrate the impact of drug addiction[15].

First of all, the experts determine the concepts that best describe the system. They know which factors are crucial for modeling the system (characteristic, state, variable, etc.) and provide a concept for each one. Next, they identify those elements of the system which influence other elements and, for the corresponding concepts, determine the positive or negative effect of one concept on the others. Finally, not all causes affect a factor with the same intensity. Some of them have a greater, and others a lesser effect. The experts must assign effect intensities. In order to simplify matters, they might separate the relationships between factors into groups, for example high intensity

DOI: 10.1002/cav

($L_{ij} = 6$), moderate intensity ($L_{ij} = 3$), and low intensity ($L_{ij} = 1$).

Within the framework of studies to design an autopilot for racing yachts, we proposed to model the cognitive activity of action selection using FCMs.[16] In such FCMs, concepts are action strategies or affordances. Originally introduced by Gibson in ecological psychology, an affordance is defined as an "action possibility" latent in the environment, objectively measurable and independent of the individual's ability to recognize it, but always in relation to the actor.[17] In collaboration with an ergonomist, an expert lists a set of required affordances related to the activity, which is to be modeled. The affordance approach needs a model which explain how an individual selects one affordance out of a few; this where we use FCMs. Rather than considering activation levels or the internal inhibition of action graphs based on releases,[18] we again worked from the notions of attractors and repulsors in external environments, as suggested by research on affordance selection.[19–21] From this point of view, an affordance is not necessary an actual invariant of the environment, but it is a hypothesis made by the agent based on its immediate environment, which is associated with an action strategy. One part of the expert's knowledge is translated into inhibition/excitation relations between affordances: for instance, obstacles could inhibit pathways, gateways and landmarks, while gateways inhibit each other. This gives the link matrix (L_{ij}). The other part of the expert's knowledge concerns the affordance perception value. As proposed in experimental psychology,[22] for each affordance concept the expert proposes a formula for this perception value, which we use as the external activation f_{a_i} of the affordance concept. The FCM dynamics occur and activations a_i converge towards its attractor. The selected affordance i is the greatest $a_i(t)$ while $a(t)$ follows the path of the attractor (most often a fixed point or a limit cycle). Such a virtual agent acts according to the expert description and then increases its credibility.[23]

We have also used FCM to model emotional states. In collaboration with psychologists, we have described fuzzy emotional map (FEM) model, first presented in Reference [24]. Each emotion is modeled as an FCM. In this model we defined sensitive concepts (emotion input and state of mind), one output concept to determine an emotional intensity and four internal concepts to represent perception, feeling, sensitivity, and the construction of emotional output. The only features that require modification between different types of FEM are the influence weights between concepts of the map. Each weight is defined by a particular personality trait (e.g., anxiety or relaxation), and is used according to a specific kind of emotion.

FCM for Modeling Reactive Agents' Decision-making

Principle

FCMs can specify and control the behavior of autonomous entities (agents). Agents have sensors and effectors, and make independent decisions with respect to their behavior. FCMs working in relation with these agents have perceptive, motor, and internal concepts. The agents' decision-making is replaced by FCM dynamics in this way:

- agents' sensors define FCM perceptive concept activations through fuzzification[†]
- defuzzification[‡] of FCM motor concept activations provides agents' effectors.

Fuzzification and defuzzification are obtained using the principles of fuzzy logics,[25,26] where a specific concept is represented by a fuzzy subset, and its degree of activation represents the degree to which it belongs to this subset[27] (calculated using the membership function of the fuzzy set).

As an example, we aim to model an agent capable of perceiving its distance from an enemy. The agent will decide whether or not to escape from the situation, depending on this perceived distance. The closer the enemy is to the agent, the more frightened it will be, and *vice-versa*. The more frightened the agent, the more quickly it will try to flee. We model this escape behavior using the FCM in Figure 3a. This FCM has four concepts and three links: "enemy close," "enemy far," "fear," and "escape," with stimulating links (+1) from "enemy close" to "fear" and from "fear" to "escape", and an inhibiting link (−1) from "enemy far" to "fear". We chose fuzzy mode ($\mathcal{V} = [0, 1]$, $\delta = 0$, $\rho = 5$, $a_0 = 0.5$), not forced ($f_a = 0$). The sensitive concepts "enemy close" and "enemy far" are activated by the fuzzication of the sensor for the distance to the enemy (Figure 3c) while the defuzzification of "escape" gives this agent an escape speed (Figure 3d).

Sensation must be distinguished from perception, in that sensation results from the sensors alone, whereas

[†] Fuzzification consists in converting external FCM values to FCM concept activations. fuzzification is a function from $\mathbb{R}^n$ to $\mathcal{V}$.

[‡] Defuzzification consists in converting FCM concept activation to FCM external values. Defuzzification is a function from $\mathcal{V}$ to $\mathbb{R}^n$.

DOI: 10.1002/cav

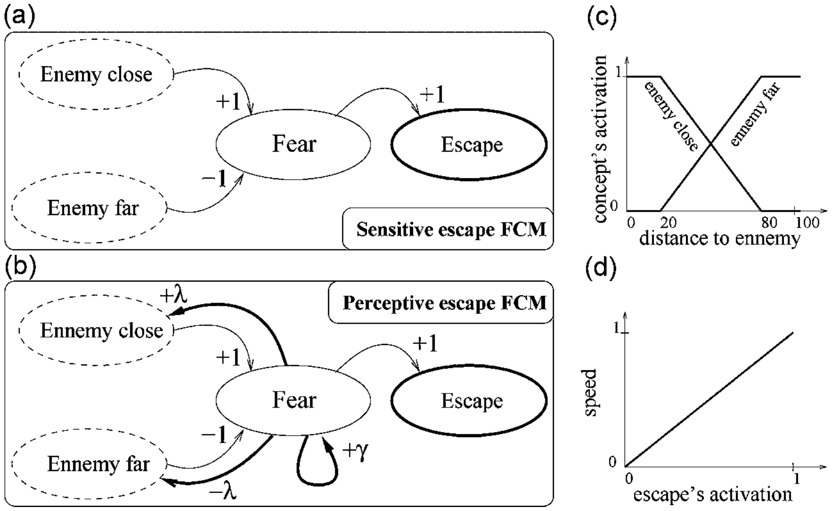


Figure 3. FCM for an agent's escape behavior. The sensitive concepts surrounded by dashes are activated by the fuzzification of sensors. The motor concepts in thick black lines activate the effector by defuzzification. (a) The concept C_1 ="enemy close" excites C_3 ="fear" whereas C_2 ="enemy far" inhibits it and "fear" excites C_4 ="escape". A purely sensitive FCM is hereby defined. (b) The FCM is perceptive: "fear" can be self-maintained (memory) and even influence feelings (perception). (c) The fuzzification of the distance to an enemy gives two sensitive concepts: "enemy close" and "enemy far." (d) The defuzzification of "escape" governs the speed of escape in a linear manner.

perception is the sensation influenced by an internal state. FCMs make it possible to model perception, thanks to the links between central concepts and sensitive concepts. For example, let us add three links to the previous escape FCM (Figure 3b). An initial self-stimulation of "fear" (link from "fear" to "fear" with ($\gamma \geq 0$)) simulates the effect of "stress": the more afraid the agent is, the more afraid it will feel. A second stimulation ($\lambda \geq 0$) goes from "fear" to "enemy close" while a final inhibitor ($-\lambda \leq 0$) from "fear" to "enemy far" simulates the phenomenon of being "fearful", i.e., when the agent is afraid, it tends to perceive its enemy as being closer than it actually is. The agent becomes perceptive according to its degree of fearfulness λ and stress γ (see Figure 4).

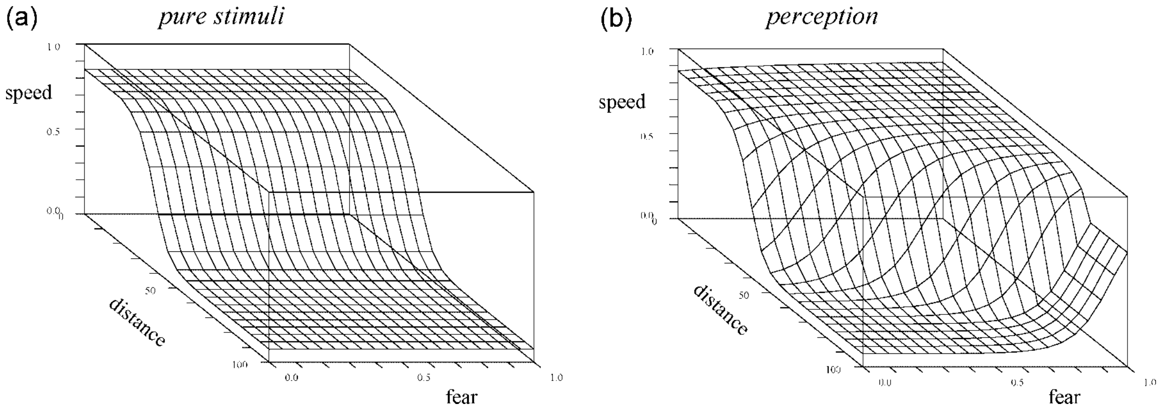


Figure 4. Escape speed decided by Figure 3b FCM. The perception of the distance to an enemy can be influenced by fear: depending on both the proximity of an enemy and fear, the dynamics of the FCM decide upon a speed obtained here with its 3rd cycle. (a) $\lambda = \gamma = 0$, the agent is purely sensitive and its perception of the distance to the enemy does not depend on fear. (b) $\lambda = \gamma = 0.6$, the agent is perceptive: its perception of the distance to an enemy is modified by fear.

DOI: 10.1002/cav

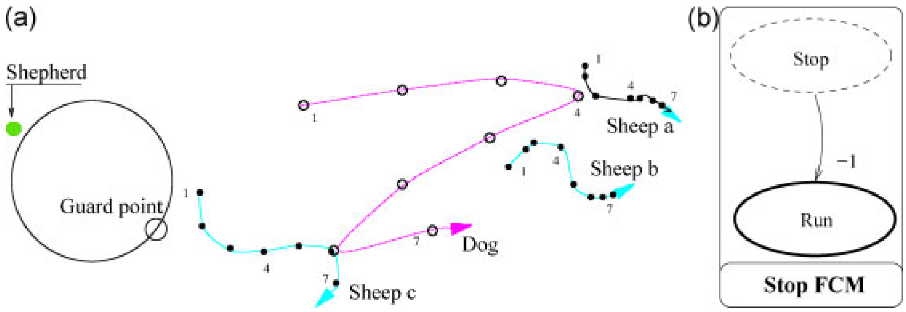


Figure 5. Sheepdog and sheep playing roles given by their FCMs. (a) The shepherd is motionless. A circular zone represents the area where the sheepdog must gather the herd. A guard point is located in the zone, diagonally opposite the shepherd. The sheepdog must return to this point when all the sheep are inside the designated area. Initially, the behavior of a dog without FCM is to run after the sheep which then quickly disperse outside of the herding area. (b) The elementary FCM carries out the actions of a dog obeying the shepherd's order to "stay," by inhibiting the desire to run.

Application

This section illustrates the usage of FCMs in simulating the behavior of believable agents. In this example, FCMs characterize believable agent roles in interactive fictions through a story taking place in a mountain pasture. "*Once upon a time there was a shepherd, his dog and his herd of sheep* ..." This example has already been used as a metaphor for complex collective behaviors within a group of mobile robots (RoboShepherd[28]), as an example of a watchdog robot for real geese (Sheepdog Robot[29]), and as an example of improvisation scenes (Black Sheep[30]).

The shepherd moves around in the pasture and can talk to his dog and give it information. He wants to round up his sheep in a given area. In this simulation, the shepherd is an avatar for a human actor that makes all his decisions. Thus, no FCM is attached to this actor. By default, he remains seated.

Each sheep can distinguish an enemy (a dog or a human) from another sheep and from edible grass. It can evaluate the distance and the relative direction (left or right) from an agent in its field of vision. It is able to identify the closest enemy. It can turn left or right and run without exceeding a certain maximum speed. It has an energy reserve that it regenerates by eating and exhausts by running. By default, it moves in a straight line and ends up wearing itself out. We want the sheep to eat grass (random response), to be afraid of dogs and humans when they are too close, and, in keeping with their gregarious nature, to stay close to other sheep. So, we chose a main FCM containing sensory concepts (enemy close, enemy far, high energy, low energy), motor concepts (eat, socialize, flee, run), and internal concepts (satisfaction, fear). This FCM calculates moving speed through defuzzification of the concept "run," and the direction of movement by defuzzification of the three concepts "eat," "socialize," and "flee." Each activation corresponds to a weighting on the relative direction to be followed: to go towards the grass, join another sheep, or to flee from an enemy respectively.

The dog is able to identify humans, sheep, the specific herding area within the pasture and the guard point. It distinguishes its shepherd from other humans and knows how to spot the sheep that is the farthest away from the area among a group of sheep. It knows how to turn to the left and to the right and run up to a maximum speed. Its behavior consists in running after the sheep, which quickly scatters them (see Figure 5a).

First, the shepherd wants the dog to obey the order "stay", which will lead the sheep to socialize. This is done by giving the dog a sensory FCM of the shepherd's message, which inhibits the dog's desire to run (see Figure 5b). The dog's behavior is driven by the FCM and the dog keeps still when asked to do so (message "stop"). The dog has an FCM based on the concepts associated with the herding area, for example whether a sheep is either inside or outside the area. These concepts also make it possible for the dog to bring a sheep back (Figure 6c–e) and keep the herd in the area by staying at the guard point, in other words, on the perimeter of the herding area and opposite the shepherd.

It is remarkable to observe the virtual sheepdog's path in this simulation forms an S shape (Figure 6c), which is a strategy that can be observed for real sheepdogs rounding up sheep.

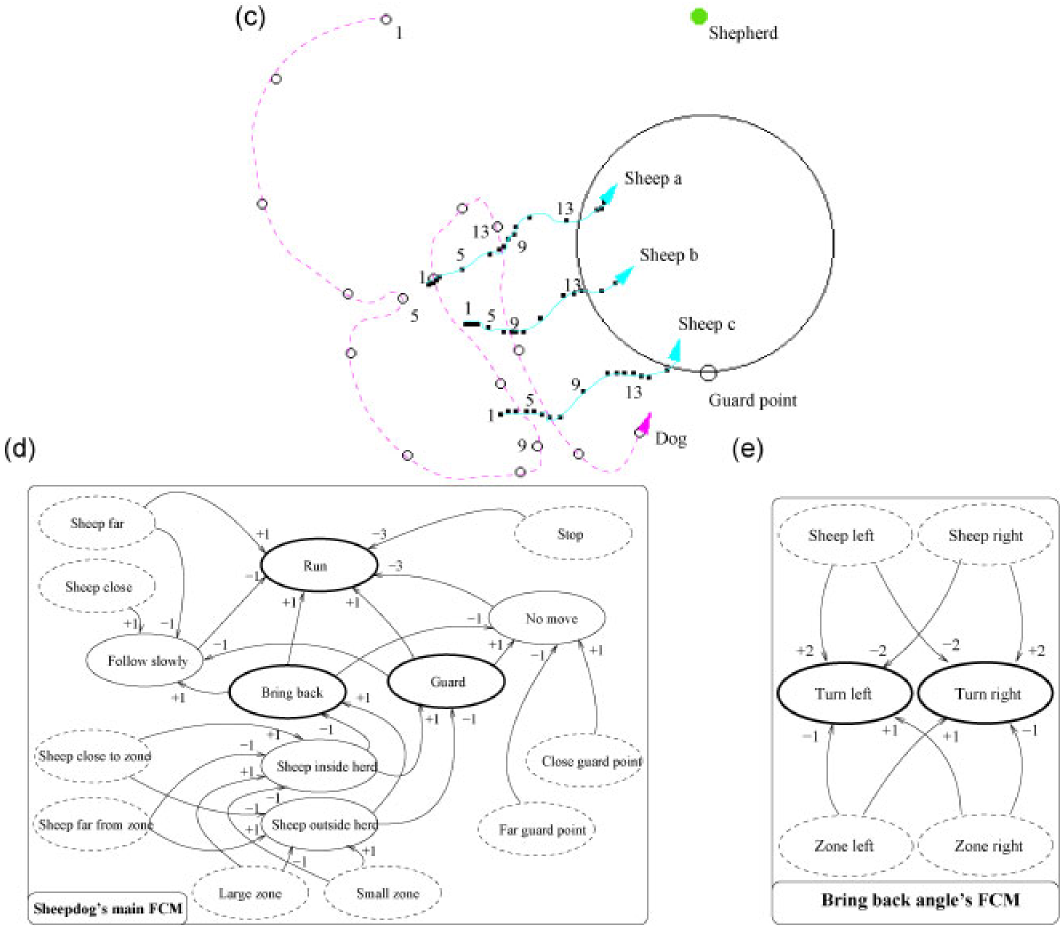


Figure 6. Sheepdog and sheep carrying out the roles given by their respective FCMs. For this simulation, the sheep's desire to gather is inhibited. (c) The film of the sheepdog bringing back three sheep was paused. The FCMs for the sheep and for the dog are represented in (d) and (e) respectively. (d) The dog's main FCM represents the role of bringing a sheep back to the area and maintaining a position in relation to the shepherd when all the sheep are in the desired zone. (e) The FCM decides the angle of incidence towards the sheep to bring it back into the zone: to go towards the sheep, but to approach it from the opposite direction.

Adaptive Behavior with Fuzzy Cognitive Maps

To obtain believable behavioral simulations, agents of the same type must have slightly different behaviors, and these individual behaviors must evolve over time. The actual behavior of a given agent is the result of its adaptation to the situations it has encountered. As each agent has its own past, evolution induces individual variability amongst agents' behaviors. When interacting with other agents, one agent has to adapt its behavior according to the way the behavior of its protagonists develops. For example, in a prey–predator interaction, the co-evolution of the two species has been observed in many cases and is known as the "Red Queen Effect."[31]

Imitation from Prototypic Behavior

The idea is to provide the ability for the agent to adapt its representation of other actors' behavior. This learning is done using the comparison between the simulation model and the observation of reality.[28] We propose a learning based on imitation[32,33] of observed behaviors to modify predefined prototypes.

Four main types of approach to learning by imitation can be distinguished: logical, connectionist, probabilistic, and prototypical approaches.

DOI: 10.1002/cav

1. *Logical approach*: Learning consists in generating a set of rules based on logic[34,35] and the sensorimotor observation describes the example (see XSTL logic[36]). This approach is difficult to adapt to our perceptual modeling behavior based on FCM, requiring that the weighting of the edges' be linked with such sensorimotor rules.
2. *Connectionist approach*: The actions are correlated with sensation by an adaptive NN, possibly inhibited by a mechanism of exception recognition (see: architecture PerArc[37]); modeling a NN provides a statistically satisfactory, but not semantically explicit, behavior instead of FCMs.
3. *Probabilistic approach*: Many internal variables are used,[38] but since they do not model the emotions and do not reflect our concept of perception, these internal variables do not change the variables through sensory feedback effects.
4. *Prototypical approach*: Learning from prototypes creates an animation by finding primitives generating the imitated movement.[39,40] An FCM is an explanatory model suited to behavior specification. Thus, an expert will be able to develop a library of prototypic behaviors. This library represents the agent's behavioral culture.[40] For example an animal's library is made up of the prototypic behavior of both predator and prey. Therefore, our agents have a library of prototypical behaviors, but unlike Mataric or Voyles, our primitives are at the decision of the movements, not within the movements themselves.

Principle

We consider that an agent has sensors allowing it to perceive its environment, as well as the effectors it uses to perform action. Any given agent also has a library of prototypic behaviors specified by FCMs.

In parallel to the virtual world, an agent also has an imaginary world, in which it can simulate its own behavior as well as the behavior of other actors. This imaginary world corresponds to an approximate representation of the environment from the agent's point of view, along with the representation of other actors' behavior. Agents use prototypic behaviors in order to simulate other actors' behavior. They imagine their own behavior by simulating their own decisional mechanisms and imagine the behavior of the other actors using prototypic FCMs. They can use their imaginary worlds to choose one strategy amongst several possibilities, not through logical reasoning but rather by behavioral simulation. Thus, they will be able to predict evolutions within the environment.

Learning

In this section, we present a method for adapting prototypic behavior through imitation in real-time. Agents observe their environment (i.e., other agents), which allows them to simulate the behavior of other entities in their imaginary worlds with prototypic FCMs. The idea is to provide a more relevant simulation by adapting prototypic FCMs through imitation. The modification of prototypic FCMs reduces the difference between predictions in the imaginary worlds and reality.[28] We assumed that agents have sensors to deduce information relating to prototypic FCMs. This means estimating sensor and effector values that will fuzzify sensor values, and comparing the result of defuzzified motor concept activations with the model's effector values.

The learning mechanism consists in retrieving the simulation results in the imaginary world, comparing them to what happened in the virtual world, and thereby adapting the prototypic FCM. To be consistent with the knowledge-based modeling of the behaviors, leading the designer to explicit both concept and links of the FCM, the learning solely consists in adapting the weights of the causal links between concepts of the prototypic FCM. Therefore, the learning algorithm does not modify either the structure of the FCM's influence graph, or the fuzzification of the sensors, or the defuzzification of the motor concepts. This modification in the causal connections between concepts could be controlled by the expert. In particular, the expert could verify the FCM's structure, impose signs for some links, and define some link interval values. This is what we call "meta-knowledge about learning."

Why not Modify the FCM Structure?

FCMs have the ability to visually represent behavioral expertise by means of a semantic graph. The concepts, the causal links between them, and these links' signs are assigned semantic descriptions. In this case, learning does not alter the structure of the influence graphs, so that the behavioral coherence as seen by a human observer may be preserved.[41–43] Nor does it alter the fuzzification of the sensors or the defuzzification of the motor concepts which remain unique for each agent.

DOI: 10.1002/cav

Algorithm

Kosko[44] proposed two different Hebbian learning methods.[45] One is based on the correlations between activations[26] and the other on a correlation of their variations (differential Hebbian learning).[4] Differential learning modifies only the links associated with correlated variations in the concepts' activations, while non-differential correlation learning runs the risk of inappropriately modifying all the links. Kosko's differential learning is based on the knowledge of a limit cycle which takes all concepts into consideration, and which is provided by an expert. However, we cannot use such a limit cycle because only estimated model sensors and effectors can be observed, and the FCM which generated them is unavailable. In addition, Kosko's differential learning makes the assumption that external activations are constant; however, the virtual world is a dynamic system and external activations evolve over time. It is therefore necessary to adapt Kosko's differential Hebbian learning to simulate realistic behaviors in a dynamic virtual environment.

The adaptation algorithm that we propose is a four-stage iterative cycle (see Figure 7):

1. *Model estimation*: The agent estimates model-sensors and model-effectors through observation. We make the assumption that these features are available.
2. *Simulating prototypic behavior*: Sensors are fuzzified into external perceptive concept activations. The FCM's dynamics are calculated, and image-effectors are obtained, by activating inner motor-concept defuzzification.
3. *Calculating reconsiderations*: Image-effectors and model-effectors are compared, generating a set of desired pseudo-activations. These pseudo-activations are obtained by going back up along the graph from motor concepts towards perceptive concepts, without modifying links and by using meta-knowledge about learning.
4. *Updating causal links*: FCM causal links are updated by applying discrete differential Hebbian learning to the sequence corresponding to the transition from FCM activations to desired pseudo-activations.

In the following sections we shall examine these four stages in more detail.

Observation

During the first stage, the agent measures features of the actor-model which are required for model-sensor and model-effector estimations. For example, the agent "dog"

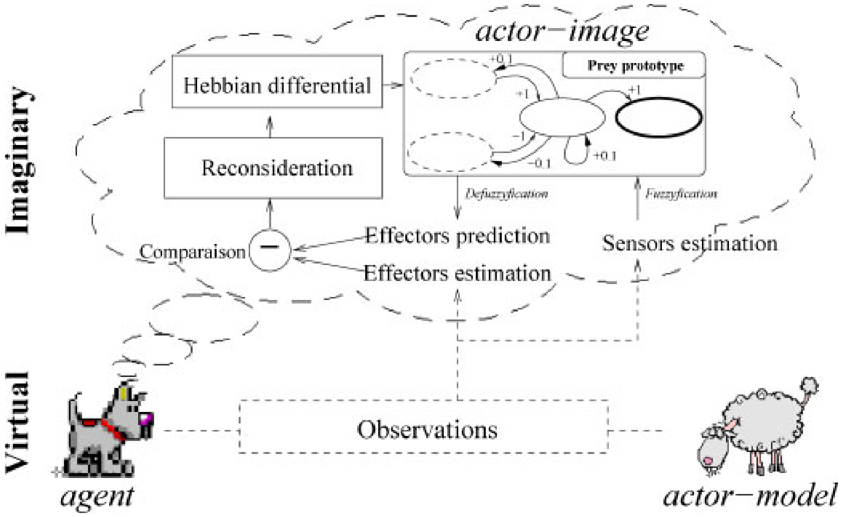


Figure 7. The agent, the actor-model, and the actor-image. The "dog" agent and the "sheep" actor-model evolve within the virtual world. The dog possesses is own imaginary world in which it can simulate prototypic behavior from a library of behaviors, containing the "prey." In its imaginary world, the dog tries to imitate the sheep by an actor-image using the prey prototype. The imitation is conducted in real-time according to the events occurring in the virtual world, while comparing the observed effectors of the sheep in the virtual world with the predicted effectors by the prey prototype in the imaginary simulated world (as estimated by the sheep's sensors). If necessary, discrete reconsideration takes place at the level of the prey's internal activations in order to reduce this difference. The prey prototype is then updated via differential Hebbian learning.

DOI: 10.1002/cav

estimates the distance between the sheep and a predator by the difference in position at that instant t, and it estimates sheep's speed by the difference in the sheep's positions at the instants t and $t-1$.

Prediction

The second stage simply corresponds to the classic use of FCMs for controlling a virtual actor and determines image-actor FCM activations at $t+\delta t \approx t$ in the imaginary world, according to model-sensor estimation and FCM dynamics with N iterations:

$$a(t+\tfrac{I}{N}\delta t) = S\left(G(f(t), L^{\mathrm{T}} \cdot a(t+\tfrac{I-1}{N}\delta t))\right) \quad \text{for } I = 1, \ldots, N\,;\ \ \delta t \ll 1 \tag{2}$$

N equals the length of the longest acyclic path added to the length of the longest cycle in the influence graph in order to make sure that sensor information is spread to all nodes. n is the FCM concept number, $f=(f_i)_{i\llbracket 1,n\rrbracket}$ external activations coming from sensor fuzzification, $a=(a_i)_{i\llbracket 1,n\rrbracket}$ internal activations, $L=(L_{ij})_{(i,j)\llbracket 1,n\rrbracket^2}$ link matrix, $G:(\mathbb{R}^2)^n \to \mathbb{R}^n$ a comparison operator and S a standardization function transforming each coordinate by the sigmoid function: $\sigma_{(\delta,a_0,\rho)}(x) = \frac{1+\delta}{1+\mathrm{e}^{-\rho(x-a_0)}} - \delta$, with parameters $(\delta,\rho,a_0) \in \{0,1\} \times \mathbb{R}_*^+ \times \mathbb{R}$. FCM motor concept defuzzification at $t+\delta t \approx t$ provides image-effectors. To clarify, we note a the resulting inner activations $a(t+\delta t)$ in the following paragraphs.

Reconsideration

The third stage recursively generates sets of pseudo-activations $(P_i)_{i\in\llbracket 1,n\rrbracket}$ representing the orientation for FCM dynamics. This is done by moving back up the influence graph from motor concepts towards perceptive concepts, proposing pseudo-activation values according to meta-knowledge about learning, and bringing image-effectors closer to the estimates of model-effectors. We did not use the gradient backpropagation method[46] because FCMs are cyclical processes and their topology is not organized in layers (recurrent links). Furthermore, the gradient backpropagation method does not hold graph semantics and we wanted to be able to apply specific meta-knowledge to specific nodes. We shall now look more closely at the recursive process.

Initialization $m=0$. Entering into the FCM from effectors. A set I_0 represents indices of concepts defuzzified onto image-effectors. For each $i \in I_0$, we apply learning meta-knowledge. Two potential pseudo-activations $p_i^{\pm} = \sigma(a_0 \pm \frac{2\alpha_i}{\rho})$ simulate an active/inactive concept C_i, $\alpha_i \geq 1$ representing choice radicality. Including the a_i value, there are three possible pseudo-activations $p_i = a_i,\ \ p_i^+$ or p_i^- for each C_i. The $3^{\mathrm{Card}\, I_0}$ combinations are defuzzified and compared to model-effector estimates. The best combination $(p_i^{0,\{\}})_{i\in I_0}$ is retained (the 0 deals with defuzzification and the {} is a set of future labels). $\forall i \in I_0,\ \ P_i = \{p_i^{0,\{\}}\}$. The other pseudo-activation sets $(P_i)_{i\in(\llbracket 1,n\rrbracket \setminus I_0)}$ are empty. We propose this discrete reconsideration rather than a gradient-scaled calculation.[47,48] A discrete choice like this facilitates the agent's decision-making process.

Progression from m to $m+1$. Let $I_m \subset \llbracket 1,n\rrbracket$ be the index set of concepts whose desired pseudo-activation set is not empty. For $i \in I_m$, note a_i (respectively f_i) internal (respectively external) activation of concept C_i, $P_i = \{p_i^{k_1,\{\cdots\}}, \ldots, p_i^{k_L,\{\cdots\}}\}$ its desired pseudo-activation set whose cardinal equals L and $J \subset \llbracket 1,n\rrbracket$ the index set of concepts which are causes for the concept C_i (i.e.:$L_{ji} \neq 0$) and such that the edge from C_j to C_i has not been studied: $\forall \lambda \in \llbracket 1,L\rrbracket,\ \ j \neq k_\lambda$. We will calculate pseudo-activations P_j for $j \in J$ as follows:

- For each $j \in J$, we apply learning meta-knowledge: two potential pseudo-activations, p_j^+ and p_j^-, are calculated using the formula (3) so that their influence on a_i causes a clear choice between an active C_i or an inactive C_i. This accounts for external activations, with $\alpha \geq 1$ representing the choice radicality:

$$p_j^{\pm} = \left(a_0 \pm \frac{2\alpha_j}{\rho} - f_i - \sum_{l\neq j} L_{li} a_l \right) \Big/ L_{ji} \tag{3}$$

  Note that a_0 and ρ are FCM sigmoid function parameters (see Figure 2); α is a learning algorithm parameter.
- Then we randomly select a $\lambda \in \llbracket 1,L\rrbracket$. This gives $p_i^{k_\lambda,\{\cdots\}} \in P_i$ and we choose between the possible $3^{\mathrm{Card}\, J}$ combinations $p_j^i = a_j,\ \ p_j^+,$ or p_j^- for $j \in J$, the one $p_j^{i,\{\cdots,k_\lambda\}}$ which gives a C_i activation $\sigma\left(G_i(f_i, \sum_j L_{ji} p_j^i)\right)$ the nearest to $p_i^{k_\lambda,\{\cdots\}}$.
- Thus we obtain a new set of concept indices with a pseudo-activation set which is not empty: $I_{m+1} = I_m \cup J$ with $P_j = P_j \cup \{p_j^{i,\{\cdots,k_\lambda\}}\}$for $j \in J$.

Termination. If for each $i \in I_m$, the corresponding J set is empty, that means every edge belonging to the paths arriving in $(C_i)_{i\in I_0}$ has been studied.

We use a discrete method by proposing three pseudo-activations. The discrete method chosen will allow us on one hand to limit the calculations and on the other, to represent a radical choice. Our argument is that to learn,

DOI: 10.1002/cav

the proposed modifications need to correspond to radical choices rather than minor alterations.

Update

The fourth and final stage modifies the matrix of the FCM's links in such a way that its dynamics are directed towards resulting behavior similar to that of the actor-model. We use discrete Hebbian learning to pass from internal activations a to questionings p calculated during the previous stage. Unlike Kosko who used a limit cycle and a learning rate decreasing towards zero over time to ensure convergence (see Reference[4]: p. 186), we only make them learn the passing from internal activations a to reconsideration p in order to modify the links without creating cycles while maintaining a constant learning rate $r(t) = R$. A cycle would teach not only the passing from a to p, but also from p to a, which is inappropriate. Our learning rate is constant so that the agent will conserve its adaptive nature. Theoretically, there is nothing to prevent the learning rate from being modified over time. This can be achieved by making it follow a series decreasing towards zero, with an infinitely decreasing associated series, as, for example, $r(t > t_0) = \frac{R}{t-t_0}$. This would ensure that the weights of the FCM's edges would converge, but adaptability would decrease over time.

Formally, noting $\mathcal{A} \subset [\![1, n]\!]^2$ the edge set of the FCM, $\beta \in]0; 1+\delta[$ a sensitivity level and $s : \mathbb{R} \to \{-1, 0, 1\}$ the discrete function $s(x) = -1,\ 0$ or 1 if $x \leq -\beta,\ -\beta < x < \beta$ or $x \geq \beta$ respectively, the learning algorithm corresponds to the following equations:

$$\begin{aligned}
&\forall(i, j) \in \mathcal{A},\ \text{if } \exists k \in [\![0, n]\!],\ p_j^{k,\{\cdots,i,\cdots\}} \in P_j,\\
&\text{with this } k\ :\\
&\begin{cases}
\Delta_i = s(p_i^{j,\{\cdots\}} - a_i),\ \Delta_j = s(p_j^{k,\{\cdots,i,\cdots\}} - a_j)\\[2ex]
L_{ij(t+1)} = \left| \begin{array}{ll} L_{ij(t)} + R(\Delta_i \Delta_j - L_{ij(t)}), & \text{if } \Delta_i \neq 0 \\ L_{ij(t)} & \text{, if } \Delta_i = 0 \end{array} \right.
\end{cases} \qquad (4)\\
&\text{otherwise } \mathcal{A}_{ij} \notin \{\text{path to effectors}\} :\ L_{ij(t+1)} = L_{ij(t)}
\end{aligned}$$

It must be noted that we preserve coherence in the modification of links as specified in the initial prototype provided by the expert. Link emergence, link suppression, or modification of a link's sign are therefore forbidden. Furthermore, some links can be maintained in terminals $\mathcal{B}_{ij} = [L_{ij}^{\min}, L_{ij}^{\max}]$ so that the modified behavior might remain consistent with the expert's initial description: if $L_{ij}(t+1) < L_{ij}^{\min}$ then $L_{ij}(t+1) = L_{ij}^{\min}$ and if $L_{ij}(t+1) > L_{ij}^{\max}$ then $L_{ij}(t+1) = L_{ij}^{\max}$.

Complexity

The complexity of this algorithm is a polynomial function of the number n of FCM concepts, and even $\mathcal{O}(n)$. For an expert, a concept's causes are always very limited in number (seldom more than seven). Therefore, the number of edges arriving on each concept is increased by M ($M \approx 7$). Card$\,J \leq M$. $3^{\mathrm{Card}\,J}$ is thus increased in practice, whatever the number of concepts involved in the FCM. The same applies to the calculation of FCM dynamics with a complexity of $\mathcal{O}(n)$ whereas they could seem to be $\mathcal{O}(n^2)$, due to the great number of zeros in the link matrix. The number of not null links in a column is no more than M, whatever n might be. This algorithm can thus be implemented for use in real-time.

Application

Based on the sheepdog environment described above, we implemented three applications. First, the dog learns one way of rounding up sheep by imitating a human operator or another dog. In these cases, the prototypic FCM used is the dog's own FCM. Second, the dog's prey prototype adapts to a given sheep in real time. This application is described in this section. Third, fearful sheep learn how to be surrounded by other sheep. The sheep remain frightened but no longer flee when they come upon a dog. Immobilizing fearful links means that the sheep's behavior can be adapted whilst at the same time preserving a fearful "personality."

To simulate sheep behavior, the dog uses prototypic FCMs of prey from its behavioral library. The dog actually represents each sheep's behavior through prototypic "prey" FCMs in its imaginary world, with each sheep being associated with its own prototype. The dog can therefore simulate the sheep's behavior in order to make predictions and to test different strategies.

The prototypes will adapt to each sheep through imitation. One FCM controls the prototype's speed and another controls its direction. Comparisons between the result of the imaginary and the virtual worlds are used to adapt prototypic FCMs in real-time through learning. Figure 8 illustrates the modification through imitation of a prototype's speed and the representation of one sheep's speed using the imaginary world. We chose this set of learning periods to ensure that the process would converge.

In order to imitate, the dog first observes the sheep. It adapts the prototypic prey behavior allowing it to simulate the sheep's behavior in its imaginary world. By observing the sheep, it estimates the information necessary

DOI: 10.1002/cav

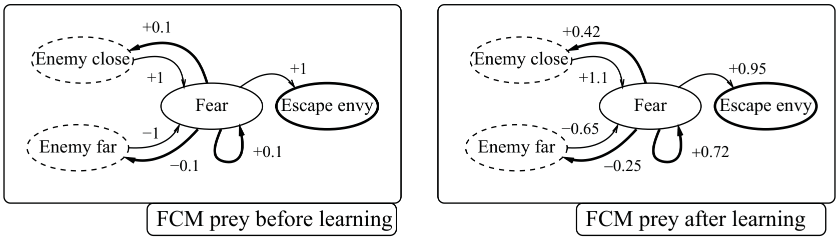


Figure 8. An FCM of perceptive prey from the library of prototypic FCMs which adapt themselves by learning.

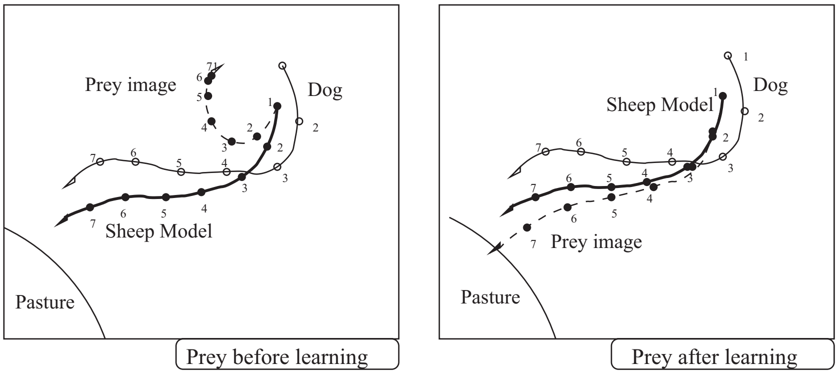


Figure 9. More pertinent predictions can be obtained from the imaginary world by using imitation learning.

for the fuzzification of the prototype (*Phase 1: observation*). Estimated sensor values are fuzzified in activating the concepts “Enemy close” and “Enemy far.” The prototype dynamic occurs and, by defuzzifying the activation of the motor effector “Escape envy”, we obtain the image effector (*Phase 2: prediction*). This corresponds to the dog’s representation of the prey’s speed. The image effector from the prototype is compared to an estimation of the sheep’s effectors and this comparison is used to calculate a set of pseudo-activations associating the desired modifications to the FCM’s links (*Phase 3: reconsideration*).

In Figure 9, we compared the simulation of the sheep’s behavior obtained from the prototype in the imaginary world (“Prey image”), with the sheep’s behavior in the virtual world (“Sheep Model”), both before and after learning, while the dog performs the same trajectory (“Dog”). The modeled sheep is controlled by the map in Figure 3b (with $\lambda = 0.5$ and $\gamma = 0.6$).

The human operator decides on the training period from start to finish. The dog’s acquired imitation experience illustrated in Figure 9 represents around 100 cycles, during which the dog approaches and moves away from the sheep twice. If learning were to become permanent, parameter λ in the “prey” prototype would be reduced to as little as 0.3 when the sheep remains at a distance from the dog (over 1000 cycles) but which quickly (below 10 cycles) goes back to a value of around 0.7 as soon as the dog begins to move towards the sheep. This proves the need for a constant learning rate: adaptability remains extremely responsive no matter what the duration of the learning.

We then went on to generate a herd of 30 sheep with different "personalities" (differences λ and γ in the sheep’s FCM). We assigned the dog a “prey” prototype for each sheep and requested the same number of parallel learning processes as there were sheep in the herd. We obtained significant predictions, with each prey prototype adapting quickly to each sheep (relative stability of the coefficient after 1500 cycles; the time required for the dog to approach each sheep at least once). However, a simultaneous learning technique would not be possible for larger herds, as when there are more than 300 sheep, the dog no longer has the time to learn in real-time.[§]

[§] Our models were implemented using oRis[49] language, and were made on a basic linux PC with a 400 MHz CPU and 128 MB of RAM.

DOI: 10.1002/cav

Conclusion

In order to be believable, autonomous behaviors must be driven, not only according to external stimuli, but also according to internal emotions such as fear, satisfaction, love, or hate. We described these behaviors using FCMs where these internal states are explicitly represented. Instead of the conventional use,[4] we *embedded FCMs* into each agent. This defines the decision-making period of their lifecycle. The agents implemented with FCMs are not only sensitive, but also perceptive; their behavior depends on their internals states retroacting on the sensors.

We described the use of FCMs as a tool for modeling the behavior of virtual actors improvising in free interaction. We highlight specific modeling features which can prove particularly advantageous, such as the flexibility concerning system design and control, the comprehensive structure and operation, adaptability to problem-specific prerequisites, and the capability for abstractive representation and fuzzy reasoning.

Our agents possess a behavioral library made up of prototypic FCMs. While acting in the virtual world, the prototypic FCMs allow the agent to simulate the behavior of other actors in its imaginary world. These FCMs simulate different strategies, allowing the agent to make predictions. We use FCMs to predict behavior, not by formal reasoning as it was realized for the reasoning on beliefs, the distributed decision and the organization of agents in interaction from the global standpoint,[50] conceptual graphs for human experts, but by *behavioral simulation*.

We presented a learning algorithm allowing the prototypic FCM to adapt through observation. Our algorithm changes the weights of FCM connections. It does not, however, modify the structure, the fuzzification of the sensors, or the defuzzification of the motor concepts. The applications depict a sheepdog using a prototypic FCM to predict the behavior of a herd of sheep.

The following points are the major limits of our proposal. Currently, the *prototype choice* (for internal simulations and for learning) is provided by the designer of the simulation. Moreover the *learning periods* are not chosen by the agent, they are imposed by the designer.

Transferring these competencies to the level of agents will increase their autonomy, while automating the entire process. Consequently, future research will aim to implement a process that selects a prototype in the library through observation of the model behavior to be imitated. Furthermore, the learning period will be selected automatically. We are also working on adapting the fuzzy transformations associated with fuzzification and defuzzification.

References

1. Tolman EC. Cognitive maps in rats and men. *Psychological Review* 1948; **55**(4): 189–208.
2. Axelrod R. *Structure of Decision*. Princeton University Press: USA, 1976.
3. Kosko B. Fuzzy cognitive maps. *International Journal of Man-Machine Studies* 1986; **24**: 65–75.
4. Dickerson JA, Kosko B. Virtual worlds as fuzzy cognitive maps. *Presence* 1994; **3**(2): 173–189.
5. Koulouriotis DE, Diakoulakis IE, Emiris DM, Antonidakis EN, Kaliakatsos IA. Efficiently modeling and controlling complex dynamic systems using evolutionary fuzzy cognitive maps. *International Journal of Computational Cognition* 2003; **1**(2): 41–65.
6. Stylios C, Georgopoulos V, Malandraki G, Chouliara S. Fuzzy cognitive map architectures for medical decision support systems. *Applied Soft Computing* 2008; **8**(3): 1243–1251.
7. Rodriguez-Repiso L, Setchi R, Salmeron J. Modelling IT projects success with fuzzy cognitive maps. *Expert Systems with Applications* 2007; **32**(2): 543–559.
8. Hafner V. Learning places in newly explored environments. In *SAB2000 Proceedings Supplement Book*, J. A. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S. W. Wilson (eds). Publication of the International Society for Adaptive Behavior, 2000.
9. Stylios C, Georgopoulos V, Groumpos P. The use of fuzzy cognitive maps in modeling systems. In *5th IEEE Mediterranean Conference on Control and Systems, Paphos*, 1997. paper 67 (CD-ROM).
10. Hagiwara M. Extended fuzzy cognitive maps. *IEEE International Conference on Fuzzy Systems*, 1992.
11. Stylios C, Groumpos P. Fuzzy cognitive map in modeling supervisory control systems. *Journal of Intelligent & Fuzzy Systems* 2000; **8**: 83–98.
12. Salmeron JL. Augmented fuzzy cognitive maps for modelling lms critical success factors. *Knowledge-Based Systems* 2009; **22**(4): 275–278.
13. Taber R. Knowledge processing with fuzzy cognitive maps. *Expert Systems with Applications* 1991; **2**: 83–87.
14. Siraj A, Bridgesk S, Vaughn R. Fuzzy cognitive maps for decision support in an intelligent intrusion detection system. *International Fuzzy Systems Association/North American Fuzzy Information Processing Society (IFSA/NAFIPS) Conference on Soft Computing*, 2001.
15. Calais G. Fuzzy cognitive maps theory: implications for interdisciplinary reading: National implications. *FOCUS On Colleges, Universities, and Schools*, 2, 2008.
16. Parenthoën M, Tisseau J, Morineau T. Believable decision for virtual actors. *IEEE International Conference on Systems, Man and Cybernetics (IEEE-SMC)*, Vol. 3, Tunisia, 2002, p. MP2R3.
17. Gibson JJ. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates: London, 1979.

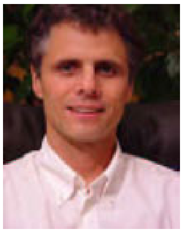
DOI: 10.1002/cav

18. Norman DA, Shallice T. Attention to action: willed and automatic control of behavior. *Consciousness and Self-regulation* 1986; **4**: 1–18.
19. Lewin K. *Principles of Topological Psychology*. McGraw-Hill: New York.
20. Reed ES. The intention of the use a specific affordance: a conceptual framework for psychology. In *Development in context, acting and thinking in specific environment*, Wozniak RH, Fisher KW (eds). Lawrence Erlbaum Associates: New York, 1993; 45–76.
21. Lahlou S. Engineering Psychology and Cognitive Ergonomics. In: Don Harris, *HCI (13), Lecture Notes in Computer Science*, 4562, Springer, 2007.
22. Stoffregen TA, Gorday Y-Y, Sheng KM, Flynn SB. Perceiving affordances for another person's actions. *Journal of Experimental Psychology: Human Perception and Performance* 1999; (25): 120–136.
23. Mateas M. An Oz-centric review of interactive drama and believable agents. *Technical Report CMU-CS-97-156*. Carnegie Mellon University: Pittsburgh, PA, 6, 1997.
24. Nédélec A, Follut D, Septseault C, Rozec G. Emotions, Personality and social interactions modelling in a multiagent environment. *Proceedings of CASA 2005*, Hong Kong (China), October 2005; 103–108.
25. Wenstop F. Deductive verbal models of organizations. *International Journal of Man-Machine Studies* 1976; **8**: 293–311.
26. Kosko B. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall: Engelwood Cliffs, 1992.
27. Sugeno M, Nishida M. Fuzzy control of a model car. *Fuzzy Sets and Systems* 1985; **16**: 103–113.
28. Schultz AC, Grefenstette JJ, Adams WL. Roboshepherd: learning a complex behavior. *Technical Report AIC-96-017*. Naval Research Laboratory: Washington, DC, 1996.
29. Vaughan R, Sumpter N, Henderson J, Frost A, Cameron S. Experiments in automatic flock control. *Journal of Robotics and Autonomous Systems* 2000; **31**: 109–117.
30. Klesen M, Szatkowski J, Lehmann N. The black sheep: interactive improvisation in a 3D virtual world.*I3*, 2000; 77–80.
31. Van Valen L. A new evolutionary law. *Evolutionary Theory* 1973; **1**(1): 1–30.
32. Meltzoff AN. Understanding the intentions of others: reenactement of intended acts by 18-month-old children. *Developmental Psychology* 1995; **31**: 838–850.
33. Gallese V. The inner sense of action: agency and motor representations. *Journal of Consciousness Studies* 2000; **7**(10): 23–40.
34. Isla DA. The virtual hippocampus : spatial common sense for synthetic creatures. *PhD Thesis*. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2001.
35. Yu Q, Terzopoulos D. A decision network framework for the behavioral animation of virtual humans. *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, 2007; 119–128.
36. Bimbo AD, Vicario E. Specification by example of virtual agents behavior. *IEEE Transactions on Visualization and Computer Graphics* 1995; **1**(4): 350–360.
37. Gaussier P, Moga S, Quoy M, Banquet JP. From perception-action loops to imitation process: a bottom-up approach of learning by imitation. *Applied Artificial Intelligence* 1998; **12**: 701–727.
38. Le Hy R, Arrigoni A, Bessiere P, Lebeltel O. Teaching Bayesian behaviours to video game characters. *Robotics and Autonomous Systems* 2004; **47**: 177–185.
39. Voyles R, Morrow J, Khosla P. Towards gesture-based programming: Shape from motion primordial learning of sensorimotor primitives. *Robotics and Autonomous Systems* 1997; **22**(3–4): 361–375.
40. Mataric MJ. Visuo-motor primitives as a basis for learning by imitation: linking perception to action and biology to robotics. In *Imitation in Animals and Artifacts*, Dautenhahn K, Nehaniv C (eds). MIT Press, 2002; 392–422.
41. Papageorgiou EI, Stylios C, Groumpos PP. Active hebbian learning algorithm to train fuzzy cognitive maps. *International Journal of Approximate Reasoning* 2004; **37**(3): 219–249.
42. Papageorgiou EI, Groumpos PP. A new hybrid method using evolutionary algorithms to train fuzzy cognitive maps. *Applied Soft Computing* 2005; **5**(4): 409–431.
43. Papageorgiou EI, Stylios C, Groumpos PP. Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links. *International Journal of Human-Computer Studies* 2006; **64**(8): 727–743.
44. Kosko B. Hidden patterns in combined and adaptative knowledge networks. *International Journal of Approximate Reasoning* 1988; **2**: 337–393.
45. Hebb DO. *The Organization of Behaviour*. John Wiley and Sons: New York, 1949.
46. Rumelhart D, Mcclelland J. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations (Parallel Distributed Processing)*. MIT Press Cambridge, MA, USA, August 1986.
47. Mozer M. A focused backpropagation algorithm for temporal pattern recognition. In *Backpropagation: Theory, Architechtures and Applications*, Chauvin Y, Rumelhart D (eds). Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1995; 137–169.
48. William R, Zipser D. Gradient-based learning algorithms for recurrent networks and their computational complexity. In *Backpropagation: Theory, Architechtures and Applications*, Chauvin Y, Rumelhart D (eds). Lawrence Erlbaum Associates, 1995; 433–486.
49. Chevaillier P, Harrouet F, Reignier P, Tisseau J. Virtual reality and multi-agent systems for manufacturing system interactive prototyping. *International Journal of Design and Innovation Research*, INRIE, Paris, January 2000.
50. Chaib-Draa B. Causal maps: theory, implementation, and practical applications in multiagent environments. *IEEE Transactions on Knowledge and Data Engineering* 2002; **14**(6): 1201–1217.

DOI: 10.1002/cav

Authors' biographies:

Cédric Buche (1979) is an assistant professor in computer science and works at the European Center for Virtual Reality (CERV). His research concerns the simulation of adaptive behaviors. He is the leader of the Fuzzy Cognitive Map (FCM) project.



Pierre Chevaillier (1960) has been an associate professor computer science at the Computer Science Laboratory for Complex Systems (LISyC), since 1996. His research interests lie in virtual reality, knowledge representation and multiagent systems. He has been the head of the CERV, since 2008.

Alexis Nédélec (1961) is an assistant professor in computer science and works at the CERV.

Marc Parenthoën (1969) is an assistant professor in computer science and works at the CERV.

Jacques Tisseau (1953) is a professor in computer science at the Brest National Engineering School (ENIB). His research concerns the autonomy of virtual entities. He is the founder of the CERV and the present ENIB director.

DOI: 10.1002/cav