A General Architecture for Social Intelligence in the Human Mind & Brain

Jeungmin Lee1,2, Jerald D. Kralik1* and Jaeseung Jeong1,2*

1Department of Bio and Brain Engineering, College of Engineering
2Program of Brain and Cognitive Engineering, College of Engineering
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, 34141, South Korea

Abstract

Substantial evidence has amassed for social cognition in the human mind and brain, but surprisingly little theoretical work exists that organizes the information in a complete, coherent framework. To address this deficiency, we recently developed a general framework of social intelligence. Here we describe multiple further developments. In particular, we develop metacognitive control over the entire process of problem-solving based on social information, including management of multiple simultaneous problems, and active outcome monitoring. We better delineate the critical influence of affect/emotion, and we distinguish structure vs process throughout the model more explicitly, including specifying the inputs and outputs of each main process. Ultimately, a comprehensive social cognitive architecture will be needed for the common model of cognition.

Keywords: Social cognition; decision-making; theory of mind; social neuroscience; multiagent systems; artificial social intelligence

* Corresponding author. Tel.: +82-42-350-4397; fax: +82-42-864-5318
E-mail address: *jerald.kralik@gmail.com; jsjeong@kaist.ac.kr

© 2018 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the 9th Annual International Conference on Biologically Inspired Cognitive Architectures.

Keywords: Social cognition; decision-making; theory of mind; social neuroscience; multiagent systems; artificial social intelligence

* Corresponding author. Tel.: +82-42-350-4397; fax: +82-42-864-5318
E-mail address: *jerald.kralik@gmail.com; jsjeong@kaist.ac.kr
1. Introduction

Social cognition is both highly complex and crucial for humans and artificial systems navigating our everyday multiagent world, and it must therefore be a key component of a common model of cognition [6, 7]. Although there are a number of social cognitive architectures, to our knowledge, none is sufficiently developed to handle the richness of even our current understanding of human high-level social cognition, such as our extensive mental models of others [3]. To address this deficit, we recently began developing a social cognitive architecture that allowed us to generate nine predictions that we have since tested empirically [8]. Here, we present a newer version of the model with much more detail elaborated. In particular, we have developed metacognitive control over the entire process of problem-solving based on social information, including management of multiple simultaneous problems, and active outcome monitoring. We delineate the critical influence of affect/emotion, and we distinguish structure vs process more explicitly, including specifying the inputs and outputs of each main process.

In what follows, we first describe the general paradigm used for the model, including the input events, agents and their goals. We then describe the current model, showing how a given social event (e.g., cheating, murder, assisting someone) is processed and acted upon. The goal of our project is to obtain a complete account of social cognition from a top-down perspective, by first providing a comprehensive architecture to then use to build a working computational model that we are also currently constructing.

The general architecture of the model — i.e., all components (main processes, memory stores, metacognitive managers) — is based on extensive evidence in social psychology and neuroscience, and thus often referenced by a standard compendium in the field [3]. Hence, they are candidate structures to include in a common model of cognition. More detailed specifications are also generally supported, though also contain our own proposals.

2. General Paradigm

We first describe the overall paradigm [8]. In general, the model is initiated by an event that occurs with a target agent. When a central problem-solving agent learns of this event, she must process it and respond based on its relevance to her and others. Minimally this requires updating her own knowledge, and maximally also taking some action to attempt to resolve a problem produced by the event. For example, if the target were caught cheating on an exam, the central agent may wish to ‘right this wrong’ both to her, others, and perhaps even on principle [1-3, 8]. We focus on three main actions the central agent could take: directly confronting the target agent, indirectly confronting them via telling others (a form of gossip) or doing nothing. Regarding telling others, a third main agent in our model is the receiver, representing those who do not directly experience or hear about the event except only through the central agent—if she chooses to communicate it to them.

We next describe the basic paradigm components.

2.1. Agents and Minds

It is almost impossible not to interact with other individuals either directly (e.g., meeting in person, talking via phone or text messages) or indirectly (e.g., hearsay about acquaintances, from others or social media). In a multi-agent world, agents take turns playing different roles when they engage in a series of social interactions. For example, an agent who received news about someone’s action may disseminate the information to others the very next ‘moment’ (i.e., from sink to source). We use four types of roles that agents may play in social interactions, represented by three means of communication: direct communication, indirect communication, and doing nothing.

The Central Problem-Solving Agent (Ac) plays the focal role in our model of social interaction (Fig. 1). She receives social information input either via direct observation or through indirect transmission from a source and follows a series of internal processes to decide an action to obtain the most desirable expected outcome.
The source (AS) is the agent who passes on the social information to AC via communication. By disseminating the information, AS enables AC to know about events that happen outside of her own experience—a powerful ability indeed. AS, in fact, was in the role of AC in the previous state. The identity of AS is an important factor for AC to determine whether to trust the information in the first place: e.g., from a random internet forum or a prestigious newspaper; a convicted felon or a well-respected priest; friend or foe. Even then, credibility remains critical: e.g., whether reliable, lied in the past, etc.

The target (AT) is the main actor (or subject) of the social information and can be categorized by their relationship to AC in terms of closeness (e.g., family vs. strangers), status (e.g., friends vs. professors), and other group memberships (e.g., classmates, coworkers, and colleagues) [1, 2, 8]. An action of AT leads to the occurrence of an event, causing a change of state for AC, with the information regarding this change gathered and evaluated by AC. The identity and credibility of AT (as with AS) also influence AC’s internal process of deciding the veracity of the information—the information may not be further processed if AC decides it is unlikely that the event is even true. Moreover, previous knowledge such as background or personal traits of AT may influence AC’s perspective/prejudice (and therefore evaluation) of the event.

A receiver (AR) comes into play if AC chooses to disseminate the information (i.e., gossip) instead of directly contacting AT or doing nothing [1, 2, 8]. Once AC finalizes her decision and disseminates the information by means of gossip, AR becomes the new AC; the former AC becomes the new AS in the subsequent state. Like AT, AR can also be categorized by his/her relationship to AC.

Models of minds Characteristics of AS, AT, and AR influence AC’s decision-making; and to do so in a meaningful way, AC must have accurate knowledge about them: i.e., models of the minds of the other agents [3, 4, 8]. To build accurate and precise models of others’ minds, constant monitoring and updating of social information is necessary. The models of minds consist of factors including personal history, cultural/societal background, personality traits, beliefs, and their relationship with other agents in the system. Moreover, each agent has their own goals both short- and long-term. Because the social world is complex and fluid, as every agent is attempting to achieve their own goals, it is challenging to say the least for AC to maintain an accurate mental representation of others; gathering as much reliable information as possible, therefore, is a critical ongoing task for AC. A good model of others’ minds allows AC to make better predictions about possible subsequent states based on her own actions.

Goals Goals of individual agents in the system vary according to their unique, independent minds: e.g., placing priority on economic gains (i.e., nonsocial goals), with others placing higher value on friendship and social bonds (i.e., social goals). Either way, the ultimate goal of every agent is assumed to be maximizing the expected outcome [3, 4]. In our social model, we conceptualize value maximization (or outcome optimization) via a slightly different tack: with equilibrium points.

In some cases, the goal may be simply to maximize outcome (more is better); however, in many others, especially

[Figure 1: The central agent receives information about a social event of the target agent (both the specific event and context), directly, or indirectly from an information source.]
in the social domain, it is not. Consider cases of fairness and equity. If someone commits a beneficial act to another, assisting them in some way, it typically is the case that the person assisted owes the assistor a comparable amount—there is an equity balance among people that generally must be maintained and thus restored [1-3, 8]. For example, AC finds out that her colleague Sarah (AT in our model) always comes to work about half an hour early every day to clean up the office. Because everyone benefits from this, it would eventually seem unfair and feel uncomfortable not to offer Sarah something in return for her diligent work. AC’s goal, then, would be to pay off this implicit ‘debt’ and return to the state of equilibrium where she does not owe anything to Sarah. To accomplish this goal, AC may pay it directly to Sarah (e.g., buying her a cup of coffee to thank her) or indirectly via others (e.g., spreading nice words about Sarah, raising her reputation as a good person).

When the equilibrium is disrupted in any content domain (with multiple ones in the social realm, like fairness or harm), AC must not only look to rectify the imbalance, but also update her understanding (and that of all others) of current events, leading to a complicated process of social analysis and decision-making that we describe below.

2.2. Events and Other Social Information Components

An event currently consists of three main components: target, content, and valence [8]. Target of the event indicates the identity of AT, the subject of the information. Content is the social domain of the event, with our model currently having eight pillars of social knowledge: (1-5) five well-established dimensions of morality (care/harm, fairness/cheating, loyalty/betrayal, authority/subversion, sanctity/degradation) [5], (6) altruism/selfishness, (7) competition (positive and negative versions), and (8) general social affairs. Valence is the directionality of the event, whether positive or negative. The degree or severity is derived from a combination of the content and valence, determined during internal processing, which produces the degree of equilibrium shift (potentially weighted by additional factors, such as target identity, described below).

Other important components add richness to the information, which we generally call context, including (a) the more direct context of AT’s action, such as killing someone vs doing so in self-defense, and (b) more subtle situational or nonverbal cues that also provide further explanation about the event, e.g., home or workplace, cheated on quiz vs final exam, or AS’s (i.e., the source) non-verbal behavior (e.g., facial expression, voice tone, and body gestures displayed by AS during communication).

AC therefore receives both event and context information when a social event occurs, via (1) direct observation or (2) dissemination by AS (see Fig. 1).

3. General Architecture of Social Intelligence

Once information about AT — i.e., the event and context — is received as input stimuli, AC immediately carries out a series of internal processes to decide the best action to reach her goal(s). Before describing these, we highlight three general model components: (1) processes that actually process the information by transforming inputs to new outputs, (2) managers that control and organize the processes, and (3) knowledge storage structures (or knowledge) that contain necessary information (i.e., memory) for the processes to function. These general model components are indicated in Figure 2 by different box shapes: single-lined rectangles for processes, double-lined rectangles for managers, and rounded rectangles for knowledge.

3.1. Sensation

Social information about an event and context is received by AC initially as a stream of sensory stimuli via various sensory system receptors: e.g., if text, visual; if spoken, auditory [3]. There are two main tasks conducted here: receiving the information and then parsing it. Thus, for example, if spoken, sound waves into units of spoken words, separated at pauses. The parsed input stimuli, however, are not given any meaning at this earliest stage of processing.
3.2. Perception

In this stage, the parsed stimuli are given meaning [3]. To accomplish this task, knowledge of concepts and meanings is required (Fig. 2). Processes get access to the knowledge via an accessor. Because the brain contains a massive amount of knowledge, it would be extremely hard (and inefficient) to find the exact data that one needs without a well-organized inventory list. In that regard, a knowledge accessor is like a book’s index, enabling the pinpointing of knowledge of interest. Hence, at the end of perception, the sensory stimuli become meaningful. For example, the input stream “Tom confronted his colleague aggressively.” becomes basically understood, yet lacks deep, systematic understanding, especially in terms of emotional valence (i.e., was this a good or bad thing?) and its degree (e.g., how much did this deviate from the ‘norm’?), which occurs in the next stage.

![Figure 2: The complete model of the social problem-solving process regarding information transmission. There are three general model components: processes (single-lined boxes), managers (double-lined boxes), and knowledge (rounded rectangles). Knowledge accessor is KA.](image)

3.3. Initial Cognitive Process

After perception, the stimuli with superficial meanings are sent to initial cognitive processing. Here, the initial understanding is enriched with more meaning; in particular, a deeper sense of the social domain at issue, and the corresponding affective value. As described, the degree of the value shift caused by the event (i.e., Ar’s action) will vary according to its domain. Because Ac’s goal is to reestablish the equilibrium state, the current change must be determined. To do this, the model accesses Ac’s ‘affect knowledge’ through an accessor (as with concept knowledge). The values returned are then used to tag (or label) the event information (e.g., Tom punched his colleague) with its domain (e.g., harm) and an affect value (e.g., -100 units of harm) [3]. Thus, the outputs are (1) domain of the event to be further processed, (2) affect value of that domain, (3) the identity of Ar, (4) the identity of As, and (5) contextual circumstances. They are then sent forward for further analysis and problem-solving.

3.4. Shift Detector (i.e., Possible Problem Detector)

The shift detector, or internal alarm system, detects a possible problem — in this case, loss of equilibrium due to the event caused by Ar’s action — and initiates a problem-solving process [3]. It takes the input from the initial cognitive process and determines if there is a value shift. Because the assessment is based solely on the initial affect value, it does not know if the shift is real or not; hence it is a ‘possible’ problem detector. Further investigation of the shift value is conducted in the following stages. If the affect value of the input event has some directionality (i.e., the...
event is valenced either positively or negatively), the shift detector passes the input variables from the previous stage and a “shift alert” to notify the next manager (i.e., problem-solving controller) of a possible problem.

3.5. Problem-Solving Controller

Once the detector makes the preliminary assessment of a possible problem, the system needs to verify the shifted value to make sure that there is, in fact, a real problem to solve. There are six processes (from establishing a problem to executing an action and preparing for an outcome) that are organized by the problem-solving controller. Before detailing them, we clarify the role of the controller. The problem-solving controller initiates each underlying process by providing the necessary inputs, and then takes the outputs and uses them to obtain the necessary inputs for the subsequent process, etc. It is the key metacognitive process orchestrating the entire system [3]. In every case, however, the underlying process is only called if the system ‘cares’—i.e., if there is sufficient affect about it: e.g., does the agent care about verifying the truth of the information source. In other words, affect produces a gate or switch for more detailed processing [3]. This procedure is carried out by accessing the affect knowledge (see Fig. 2). If the affect value (e.g., for questioning source veracity) is above a threshold, the gate is effectively opened, and the controller initiates the given process. We next describe the series of processes controlled by the metacognitive controller.

3.5.1. Establishing the Problem

There are three steps to establishing the problem, each requiring their own process: (1) questioning the veracity of the event based on AC’s knowledge of AT and AS, (2) updating social knowledge based on the information, and finally (3) defining the problem(s) at hand that AC needs to solve. We describe each in turn.

(1) Questioning the Veracity

AC first needs to confirm that the information is trustworthy and not fake [2, 8]. Reliability of information depends on two main factors: (1) knowledge of AS (is the source trustworthy; someone I know well; have they lied to me before; could the source get access to the information); and (2) AT (e.g., would this behavior be expected of them). To check the veracity, AC first needs to reference her social knowledge (i.e., the models of the others’ minds) (Fig. 2). Then the two variables (the veracity of AS and likelihood of AT’s action being true based on previous history) are combined to produce the information veracity. Context is also used to increase accuracy. (Note that if AC does not have prior knowledge of AS and AT, she must use heuristics, relying on context and her general knowledge about an event and sources similar to the ones she is currently facing.) The veracity value is then sent back to the controller, which then compares it to the internal threshold of trust. If the veracity value does not exceed the threshold, no further processing will occur, since the original event is not believed.

Input for this process: AT and AS, identity, context
Output to the controller: Veracity value of the information

(2) Updating Social Information

Once the veracity of the information is confirmed, the controller initiates the next process to update the social information, including (a) AC’s social knowledge, and (b) scaling the current event based on extenuating circumstances. The first step occurs because the ‘models of minds’ that AC maintains are now broken due to AT’s action and the shifted value resulted from the event. The second step provides nuance to event understanding, especially based on prior social knowledge. Hence, to repair this damage and determine the accurate value of the shift, the controller provides the domain of the event (e.g., punching/harm), its corresponding initial affect value (obtained from affect knowledge), identity of AT and AS, contextual factors, and the information veracity estimated from the previous stage.

The first step, then, requires accessing the social knowledge (models of minds) (via an accessor) for relevant updating. The most obvious place is AC’s internal model of AT that contains various components that make up the person “AT” inside AC’s mind. These components include personal traits (e.g., how trustworthy, moral, or socially-oriented), personal background (e.g., cultural, societal, political stance), and AT’s beliefs inferred by AC according to her past history related to AT [2-4, 8]. Thus, if AT were caught cheating, the trustworthy value would be reduced. Given that AS, say Sam, is also aware of AT’s cheating (providing the information to AC), AC must also update her model of Sam’s beliefs. (AC may also update Sam’s relevant personal traits: e.g., Sam, the informer, really cares about
The other factors that require adjustment are the social accounts/contracts—debts that an agent owes or is owed by others. In other words, the values of the social relationships between agents (including AC herself and AT) are tracked and stored in social accounts. And because AT took an action that caused some value shift in the system, the values that AT owes or is owed need to be modified. For example, again say AC’s classmate Sam (AS) informed AC that Timothy (AT) cheated on the final exam and got an A+ (which he does not deserve) for his final grade. Since he damaged the equity (or fairness) of the system by intentionally taking the unfair benefit, AT now owes that amount of value he received from cheating to everybody else who worked hard and followed the proper rules of conduct. Moreover, because Sam provided AC the valuable information, she now owes Sam for it. Hence, social accounting can be considered the memory for the value shifts in the overall MAS world, maintaining this assessment across time until the relevant equilibria are restored.

After social knowledge is updated, the second updating step is conducted: scaling the current event based on extenuating circumstances. In brief, AC can provide a better estimate of the equilibrium shift value by scaling the initial generic value from affect knowledge based on the particular instance. Scaling by prior information about AT and context are most critical. For example, if AT were normally an extremely trustworthy person, AC might assume there is some justification for the behavior, scaling the egregiousness down to some degree. For contextual factors, for example, cheating on the final exam may be considered more serious than cheating on a pop quiz (contextual factor: final exam or quiz).

The last factor is the information veracity estimated from the previous process. Although the controller has concluded that the information is reliable (at least to exceed the threshold), the degree of confidence for trusting may vary from one event to another. And because the degree of confidence (i.e., uncertainty) would never be one hundred percent, AC always must realize that the update could be wrong. Hence, the updated model is flagged with that uncertainty, to be reevaluated if the information (used for the update) turns out to be false [4].

At the end of this process, then, AC’s models of the others’ minds are updated, together with the uncertainty flags, and the shift value (i.e., the value of deviation from the equilibrium) is properly scaled. In sum:

Input: event domain and its affect value, identities of AT and AS, contextual factors, and information veracity
Output/ results: the updated models of minds with uncertainty flags, and an updated shift value

(3) Defining the Problem(s)

The final process for problem establishment is to define the current problems. The initial event has actually led to three problems: (1) the disequilibrium due to the value shift, (2) uncertainty in the updated model (flagged with confidence/uncertainty degree about the information actually being true), and (3) broken ‘models of minds’ of the other agents who are unaware of AT’s behavior—in our case the receiver AR. A problem set is then generated, with the agents, content domain, value shift, and uncertainty flag associated with each.

Input: The updated models of minds with uncertainty flags and estimated shift value
Output (to the controller): A set of problems to solve

3.5.2. Generate an Action Set

Once the problems are defined and the ingredients to continue problem-solving are prepared, the controller activates the process that generates an action set associated with each problem (currently hand-coded, but ultimately combined with learning). The process needs to access problem knowledge: i.e., information stored about known problems, with their relevant actions that AC can potentially take to resolve them (Fig. 2). Using the problem knowledge obtained via an accessor, the process organizes AC’s possible actions according to each problem, combines them, and places them into one complete action set. The action set is then sent back to the controller for valuation.

Input for this process: A problem set
Output to the controller: An action set

3.5.3. Valuation

In this process, the possible actions that AC can take are evaluated based on expected benefits and costs [3, 4]. That is, expected outcome of every consequence known to follow each action has to be estimated based on AC’s knowledge. The expected consequence of taking an action includes three main components: first, the expected subsequent state,
such as convincing the target agent $A_T$ to ‘come clean’ by confessing to an improper act of cheating, or finding $A_T$ upset if they react defensively, potentially causing $A_C$ future harm (such as via reputation); and thus, second, the possible future problem the subsequent state relates to (e.g., threat/personal harm); and finally, some positive (i.e., benefit) or negative (i.e., cost) value, used to ‘shift’ the state toward (or away from) the equilibrium point of the given related problem (e.g., cost as -100 units of potential future harm) (for more details about the actions and their consequences, see [8], especially Valuation on p. 4). Hence, importantly, action consequences may pertain to the current problem at hand or to others, all of which must be considered by $A_C$ for an accurate assessment of possible consequences.

To provide the complete valuation for each action, three steps are required. First, to obtain the initial valuation components (subsequent state, related problem, benefit or cost), $A_C$ needs to access her knowledge of possible consequences of each action (Fig. 2). However, even if $A_C$ has a vast amount of consequence knowledge, it is more like a generalized remedy (or rule of thumb) rather than a case-specific solution, and therefore requires further adjustment according to the current event particulars.

Hence, in the second step, to provide a more accurate and nuanced estimation of expected outcome, $A_C$ needs to access the social knowledge (i.e., the models of others’ minds) so that possible benefit and cost values can be scaled based on the actual, current agents (i.e., $A_T$, $A_S$, and $A_R$) and the particular relevant trait (e.g., trustworthiness or sensitivity) (Fig. 2). This second step is essentially a social filter.

Finally, each scaled benefit or cost value must be transformed in two ways: first by comparing it to the equilibrium value to obtain the effective amount of benefit or cost relative to the baseline equilibrium point; and second, to transform each value to a common scale [4]. All values for each action are then combined.

Input for this process: An action set
Output: The overall valuation for each possible action, as well as the list of consequences for each action, including subsequent state, corresponding problem, and expected outcome values

3.5.4. Action Selection

Once the controller receives the outputs from the valuation process, it activates action selection. Here, the expected values of each possible action are compared, and the action that will generate the highest overall benefit to $A_C$ will be selected: i.e., the one that enables the largest movement toward the equilibria across all problems affected by the action [4].

Input for this process: A list of expected outcomes for every action
Output: Selected action with the highest expected outcome

3.5.5. Action Execution

The controller initiates the action execution process. Process modules involved in action production (and execution) are activated (Fig. 2).

Input for this process: The selected action
Output to the controller: Action execution

3.5.6. Prepare for Outcome

$A_C$ next needs to monitor the actual outcome that will result after taking an action; this monitoring role is played by another process manager called the state monitor (see Fig. 2) [3]. For the monitor to accomplish this task, it first needs to know what to look for—in other words, in the real-world, many types of outcomes are subtle and would be missed if not for an active process of outcome assessment. Moreover, and importantly, any given action taken can lead to an array of potential effects, and thus all expected future problems must be monitored. Thus, the problem controller must obtain the information received from both valuation and action selection to produce an updated problem set that includes both the original set of problems, and additional ones that are potential consequences of the action taken. The updated problem set is then provided to the state monitor.

To conduct this more active process of subsequent state and outcome monitoring and assessment, the monitor provides top-down modulation to other processes (especially $A_C$’s sensation, perception, and initial cognitive processes) to bias them toward relevant stimuli related to the problem set.
**Input for this process:** An updated problem set

**Output:** Top-down modulation

### 3.6. Outcome and Learning

The state monitor then examines the outcome events. Because the outcome is, in fact, a new event (caused by the action of an agent in the system), the information will be transmitted to Ac just like other social information: received by Ac either by direct observation or indirectly via a source, together with context. And the monitor will expedite and optimize this process by calling attention to the stimuli related to the consequences of the selected action. Moreover, the outcome as a new event may (and likely will) generate a value shift, although the shift is meant to bring the state back to equilibrium, not to deviate from it. In this way, problems are either being solved or created, with the system designed to continue, with the overall goal of bringing all problems back to their equilibrium points (i.e., achieving all goals).

More specifically, and for learning, the shift detector (i.e., possible problem detector) will recognize the shifted value and activate the controller, initiating the problem-solving process. The procedure will continue until it reaches the point where the social knowledge (i.e., Ac’s models of minds) is updated and the value shift is scaled. At that point, the model is poised to compare the expected outcome (stored in the monitor) and the actual outcome (being treated as the new shifted value). The controller sends the actual outcome to the monitor for the comparison (see yellow arrow pointing up toward the state monitor in Fig. 2). The monitor then takes (1) the expected outcome calculated from the previous state and (2) the actual outcome represented as the value shift in the current state and sends them to the outcome learning process (Fig. 2). The learning process compares the two values, producing the learning error [4].

Multiple components are updated with the error (i.e., learn): e.g., the expected benefit and cost values stored in consequence knowledge, the models of minds, and the uncertainty flags, so that the previous state updates can be confirmed or rejected. In addition, other knowledge such as concept, affect, and problem knowledge will all be updated based on Ac’s experience. The newly updated knowledge (i.e., learning) will allow Ac to make better decision in the future with more accurate expected value calculations. Over time, the entire procedure can become more automated to the point of habitual processing and routines [3, 4].

### 4. Discussion

Prior to building a computational model of social cognition, we believe one must have a well-developed theoretical level sociocognitive architecture, which we started previously and have substantially developed here [8]. The current model has elaborated multiple key aspects, including metacognitive control over the entire system, management of multiple simultaneous problems, delineation of the fundamental role of affect, processes vs knowledge structures, inputs and outputs of each main process, and active outcome monitoring.

A computational model can now be developed, which we are currently doing. Limitations of the presented model include its focus on only the target, content, and valence of given events (versus, e.g., more contextual details), and its overall brittleness, without having learning more deeply embedded. All models will need to balance hardcoded information, which we know the human mind/brain contains, with the flexibility necessary to navigate the real, everyday world. But one step at a time. Future work will also integrate the social model into more general ones, such as the common model of cognition [6, 7].

### References


