A General Cognitive Architecture for Agent-Based Modeling in Artificial Societies

Peijun Ye, Shuai Wang, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—Artificial Society is an analytical foundation of various complex eco- and social systems. Such system is usually implemented via multiagent approach. However, there is no consensus on how to model the agent's decision-making process, since different application scenarios concentrate on different facets. This, to some extent, hinders model reuse and system integration. This paper proposes a general cognitive architecture that attempts to adapt all the aspects of agent's decision-making in artificial societies, so that different programs and software can be reorganized and integrated conveniently. To illustrate its implementation, two simulations—emergent evacuation and population evolution—are conducted. These tests clearly show that the proposed architecture is able to support different agent-based models. Problems that might be encountered, as well as possible strategies, are also proposed in the end.**

***Index Terms*—Agent-based modeling, artificial society, cognitive architecture (CA).**

I. Introduction

THE past decades have seen extensive applications of agent-based artificial society (ABAS), ranging from computational demography [1], [2], urban transportation research [3], [4], computational economics [5], [6], land use planning [7], [8], and contagious disease propagation [9], [10] to military simulation [11], [12]. Since proposed in the early 1990s, ABAS has gone beyond the social simulation and becomes a fundamental analytical tool for complex eco- and social systems [13]. By importing the computational models of actual population, environmental and personal behaviors, computer scientists, as well as specific domain researchers, have (re)discovered the wheels of social orders. They are endowed with the ability to focus on the heterogeneity of individuals with numerous possible combinations of characteristics, rather than the traditional statistical models where persons are homogenously calculated as numbers. Furthermore, as social scientists are always facing the dilemma that certain kinds of controlled experimentation are difficult, or sometimes impossible, to implement, ABAS provides a feasible approach for them. Thus, hypotheses regarding responses of individuals to specific policies or social events can be tested and validated at a much lower expense.

Probably, the idea of agent-based simulation may be traced back to von Neumann and Burks [14], which aims to simulate the interaction between individuals with autonomy and study the global characteristics emerged from bottom-up, but the formal terminology comes from artificial intelligence (AI) two decades later [15]. Currently, three distinct types of agent models have emerged in the field of ABAS. The first is called the reactive agent, which is mostly used in production rule systems [16], [17]. This type of agents usually contains a set of rules that give responses to environment and stimulus. Intelligence can be gained from the interactions with its peers and without considering or even understanding its surrounded environment. Although it seems relatively simple, reactive agent has some advantages. As Maes [18] summarized, it reduces the communication load and is suitable for data processing. The second type is called the deliberative agent [19]. It contains an internal mental state and generates its behavior plan through the interaction with the environment. In contrast with the reactive one, intelligence of the deliberative agent lies on its ability of planning, reasoning, learning, and striving for its desired states named goals. Obviously, deliberative agent can simulate the decision cycle of humans but is much more computational expensive. Therefore, a third type—the hybrid agent—attempts to integrate the advantages of the former two. In its decision cycle, the hybrid agent deliberatively achieves a goal-oriented behavior plan and decomposes the plan into several subplans. Such subplans associate with a set of rules that can be fired. Some reflexive actions are directly contained in the reactive layer.

Although both of the reactive and deliberative agents have been used in ABAS, the latter is more suitable for the field (a detailed discussion will be given later). To develop a deliberative agent, one has to deal with its computational model. The so-called agent cognitive architecture (CA) has attracted extensive studies for decades. Its objective is to understand the functions in human cognitive process and simulate human decision-making. Currently, most CAs are particularly for the robot control in AI, but they can be applied in ABAS

Manuscript received June 8, 2017; revised September 13, 2017; accepted November 19, 2017. Date of publication December 19, 2017; date of current version February 23, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61603381, Grant 61533019, and Grant 71232006 and in part by the China Scholarship Council under Grant 201704910284. *(Corresponding author: Peijun Ye.)*

P. Ye is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Parallel Data Research Center, Qingdao Academy of Intelligent Industries, Qingdao 266109, China (e-mail: peijun_ye@hotmail.com).

S. Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China.

F.-Y. Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the Military Computational Experiments and Parallel Systems Research Center, National University of Defense and Technology, Changsha 410073, China, and also with the Parallel Data Research Center, Qingdao Academy of Intelligent Industries, Qingdao 266109, China.

Digital Object Identifier 10.1109/TCSS.2017.2777602

simulation as well. One problem of these architectures is that they usually concern about special behaviors. This brings an obstacle to merge and integrate different systems so that the existing designs or programs can be reused. Thus, a general architecture is required to direct the modeling of ABAS, which elicits the motive of this paper. The contribution of this paper is twofold: 1) to propose a general agent CA that adapts all the aspects of agent behaviors in ABAS and 2) to summarize current problems in ABAS and propose future possible strategies.

The remaining part of this paper is organized as follows. In Section II, we explain what is the CA and why it is essential to construct ABAS. Section III will summarize some representative architectures. The focused aspects in ABAS modeling are analyzed in Section IV, based on which our general architecture is presented. To support its implementation, Section V presents two simulation experiments, emergent evacuation and population evolution, to verify that the proposed architecture can integrate different agent-based models (ABMs). Then, in Section VI, we show the current problems in ABAS and discuss their related possible strategies. Finally, this paper concludes in Section VII.

II. Why Is the Cognitive Architecture?

Originally, the research of CA, to a large extent, comes from AI, but some of them have also been applied in ABAS or social simulation. This field attempts to model the main factors participated in our thinking and decision and concentrates on the relationships among them. Generally, CA research is an interdisciplinary field, ranging from psychology, neurology, and philosophy to sociology, but in computer science particularly, CA mostly refers to the computational model simulating human's cognitive and behavioral characteristics. It is a most fundamental abstract framework with a deliberative agent. Basically, there are several important characteristics that make CA most appropriate for ABAS. They are elucidated in the following.

The ultimate goal of CA study, which dates back to 1950s, is to achieve human-level intelligence at the level of computational model. Such intelligence might be realized in four different patterns: systems that think like humans, systems that think rationally, systems that act like humans, and systems that act rationally [20]. Here, the "rationality" refers to achieving consistent and correct conclusions (given its available information) for arbitrary tasks. On the one hand, ABAS in most cases concentrates on the evolution of social systems for a period of time. The result emerges from massive individual rational behaviors, not the "low-level" reflexive actions, which are activated by specific stimuli. In this sense, the rationality of individual behaviors just conforms to the definition aforementioned. On the other hand, individual rational behaviors are correlated in the temporal dimension. For example, to achieve a goal, people usually decompose it into subgoals and take several steps sequentially. When the task is partially completed, they are inclined to continue even though their surroundings turn detrimental (of course, it depends on the personality and endurance). At this point, people can bear the negative impact and persist in conducting their original plans. Obviously, deliberative agent is more appropriate than reactive agent for such typical paradigm. And CA is the most suitable framework to model such plan execution. Furthermore, CA is able to simulate people's internal deliberation, concerning not only planning but also other functions, such as reasoning, emotion, and learning, and their connections. Since rational behaviors are the results of human decisions, they originally stem from human's deliberation in essence. Based on this perspective, the generation of rational behaviors is naturally modeled as the cycle—perception (or communication), thinking, and action—that is constantly repeated through the agent's whole "life." Such cycle has already been implemented in some CAs.

In application, CAs usually deal with relatively large intelligent agent systems that have many heterogeneous parts and subcomponents, which operate as a whole to solve multidomain problems and tasks. Typically, they are built to control artificial agents, which run both in virtual worlds and physical robots acting in the real world. Without this framework, the deliberative agent is not convenient to computationally maintain its internal status, let alone conducting effective thinking. It is also difficult for engineers to develop agent programs without an explicit CA guidance. Although different from the software architecture, CA can portray the decision-making logics as a reference for the agent software design.

III. Representative Cognitive Architectures

Generally, the CAs used in ABAS can be categorized into four types. The first is the production rule system [21]. It is actually a primitive and decentralized structure originated in 1970s. All knowledge of an agent is stored distributed in the form of "If…Then…" rules. In each decision-making cycle, the agent interprets its observed information from the environment as the preconditions of particular rules. One of the rules that match the interpretations is activated (also called fired). Then, the action related to its postcondition is executed, and the internal state is updated if needed. If there is more than one rule that can be fired, a conflict resolution mechanism is introduced to determine which to apply. Many algorithms are designed for different goals, such as optimization of time or computational resources [22]. If the interpretations match no rule, the decision-making process will stop. The action of a particular rule can be the firing of another rule and thus constructs a forward chaining and simulates the reasoning process. Usually, the knowledge base is divided according to the problem domain as several subbases, among which the one that maintains the data of current state or beliefs is called the working memory (or short-term memory). The production rule system is suitable for reactive agent and has low computational complexity. It is widely applied, specifically in large-scale simulations.

The second type of agent is based on the belief–desire–intention (BDI) architecture, a classic framework for deliberative agent. BDI is short for belief–desire–intention, which was originally founded on ideas expressed by philosophers [23]. In contrast with the production rule system, BDI deems that each agent has its own "mental state," which is the basis for its reasoning. Belief is the personal information about the world. It represents an agent's recognition of its environment, which may not be correct. Desires are the possible states of affairs that an agent might like to accomplish. They are the objectives that the agent pursues. Not all desires will be acted upon by the agent, but they only provide options that might impact its actions. An intention (also referred to as a plan) is a commitment to a particular course of actions for achieving a particular goal, which represents the state that an agent wants to achieve. During each iteration of a BDI system, the agent's beliefs, usually in the form of the first-order logic, are updated through its perceptions. Intentions generated by reasoning are pushed into a stack. For the top intension in the stack, all the plans with postconditions matching the intension and with preconditions satisfying the agent's beliefs are accessed as the possible actions by searching the plan library. The agent then selects the most relevant plan according to its internal state. The whole process is viewed as deliberation [24]. BDI has many extensions. One is the emotional BDI, which accounts for the influence of emotions in order to model human behavior properly [25], [26]. In this architecture, internal emotions are represented as abstract plans (called capabilities) and resources. Since the agent is exposed to a limited area of the environment, it only has partial information and may not be aware of all its resources and its own capabilities. Capabilities need to be matched against the agent's ideas of ability and opportunity to become specific plans. An emotional state manager is responsible for controlling the resources and the capabilities used in the information processing phases. All kinds of emotions decay with time in arbitrary rates. Another framework is beliefs–desires–obligations–intentions (BOID) [27]. In addition to the mental attributes of BDI, it accounts for the obligations, one of the social norms that models agent's sociality. The basic idea behind BOID is that a multiagent system needs to endow its agents with the deliberation about whether or not to follow social rules and contribute to collective interests. Such deliberation is typically achieved through argumentation of obligations—customs must comply for the social good [28], [29]. In decision-making, BOID is similar to BDI, only different in the intention generation where agents also account for internalized social obligations.

The third type is the normative model. Unlike the BOID where behaviors are determined purely by internal motivators, such as beliefs and desires, the normative model treats social norms as the external factors to the agent. Agents are influenced and governed in their reasoning by the norms cultivated in their surrounded "society." Briefly, the questions that how to computationally represent norms, how to model the influence to the agent, what conditions will cause an agent to adopt a new norm or violate a current norm have been addressed extensively among scholars [30]. To name a few, three of them are put forward. Deliberate normative agent, actually developed earlier than BOID, suggests a mental module to record the norms [31]. The reasoning cycle at its core is similar to the BDI but the norm generalization is a separate process, which starts with a recognition through observation or communication. Then, the norms are estimated according to the internal context, and the agent decides which to comply and which to ignore. Another well-known architecture is the EMIL-A agent, a research achievement of an EU-funded FP6 project that attempts to simulate the "two-way dynamics of norm innovation." EMIL-A tries to model both the top-down and bottom-up links in the norm formation and considers the learning, internalization, and using of the norms in decision-making [32], [33]. In each normative reasoning cycle, two kinds of normative information may be observed by the agent. They are the information related to a previously recorded norm, which will be assimilated for the update of the norm's activation frequency, and the new normative information, which will initialize a norm frame. A third architecture is the normative agent, which models the social aspect of norms as an explicit mental state [34]. At a particular time, a normative state contains a certain norms, such as obligations, permissions, and prohibitions, which an agent will refer to when constructing its plans. A classic reasoning cycle consists of two phrases, that is, the plan activation and norm declaration, which instantiates optional plan and norm candidates, and the successive selection and execution, which determines which candidate to perform. In summary, although the normative models concentrate specifically in the sociality that ABAS most concerns about, many of them have remained rather abstract.

The last type of the model, inspired by computational psychology and neurology, is more complicated than the previous three. To the best of our knowledge, it involves many human cognitive components and applied in multiple domains, ranging from robot control and pattern recognition to knowledge discovery. Here, we only give a glimpse on several well-known ones, specifically state, operator, and result (SOAR), connectionist learning with adaptive rule induction online (CLARION), and adaptive control of thought-rational (ACT-R). SOAR is short for "state, operator, and result." It treats agent's decision achievement as a goal-oriented search through problem spaces [35], [36]. Apart from the perception and action modules, the architecture primarily contains a working memory, a long-term memory, and a decision procedure. Each execution cycle of SOAR starts by adding inputs to the working memory to fire production rules. Each fired rule suggests an operator, and the following decision procedure selects one according to its knowledge information. If no operator candidates are proposed, or there is no such kind of knowledge for selection, SOAR will create an impasse and recursively try the possible operators at random until the goal state has been reached or all of the options are run out. The trajectory of the problem solution will be learned as its experience. CLARION stresses the representational differences and learning differences of the implicit and explicit knowledge [37], [38]. CLARION is composed of four subsystems: action-centered subsystem (procedural knowledge), nonaction-centered subsystem (declarative knowledge, both semantic and episodic), motivational subsystem (goal structure and drives), and meta-cognitive subsystem (reinforcement learning, goal setting, and filtering selection regulation). It integrates reactive routines, generic rules, learning, and decision-making to develop versatile agents. ACT-R is a well-known architecture and aims to understand how people organize knowledge and produce intelligent behavior [39], [40]. ACT-R has several modules, such as sensory modules for visual processing, motor modules for action, an intentional module for goals, each of which holds an associated buffer that is deemed as the agent's short-term memory (called "chunks," but different from those in SOAR). A long-term production memory records the past effects of declarative chunks. In each cycle, the agent selects the production with the highest utility and executes its actions.

It can be seen in the previous review that each type of structures has its advantages and disadvantages. At first, production rule systems have low computational complexity but cannot simulate the complex behavior. Applications using this type emerge in the early stage of social simulation and only stay at a "game" level. One example is the famous SugarScape [41]. At the opposite extreme, psychological and neurological architectures are much more complicated than others, which require expensive computations. Since ABAS usually studies massive agents, this type is not as popular as the other three limited by the computational resources. Moreover, psychological and neurological architectures have very little focus on social aspects, albeit they can reasonably model human decision-making. This may not be a surprise, as they mainly come from AI or neuroscience and tend to reconstruct brain's working process.

IV. General Framework for Agent Modeling

Though the BDI and normative models are prevalent and take some cognitive aspects into account, they are lack of systemic consideration. Basically, two categories of problems are most concerned by ABAS. One is the short-term systemic dynamics. Agent in this scenario usually has to make decisions based on his own cognition in a limited time. Thus, he may not be always right or optimal. The other is the long-term systemic evolution. If existed, it also seeks potential equilibrium among multiple agents. Agent in such situation plays much more rationally, since he usually has enough time to deliberate and decide. Thus, he mostly chooses his optimal strategy in each cycle. From these classic missions of ABAS, this section proposes a general architecture and then elucidates each of its aspects. Reasoning logics in decision-making process based on these modules are also explained in the following discussions.

The overall structure of an agent is shown in Fig. 1. The modules in solid boxes are the data that store different types of knowledge or information. The boxes surrounded by dashed lines represent procedures that probably concerning specific intelligent algorithms. Each arrow marks the operation to the pointed module. Note that in the implementation of particular cases, some of these modules can be omitted if the problem only involves a part of its aspects, but for the universality, we have explicitly put them here.

A. Perception and Actuation

Perception and actuation may be the most basic components that an agent has. From robotics to software-defined agents, they are the fundamental units that the agent entities can complete their "observe and act" cycles. Each time, the agent acquires his local environment through its sensors. Such information is recorded as pieces of facts (probably in the form of formal logics) and sent to update his memory. Here, the perception refers to the observation of the environment rather than other agents (which will be put in the interaction module later), just like the local amount of sugar it sees in SugarScape.

B. Learning

Learning is a process that an agent converts his received information (from perception and/or interaction) into his knowledge. Its primary output is to adjust the long-term memory. Note that this adjustment is only based on local observation or communication and thus is possibly incorrect. Social norms, which can be viewed as a specific type of long-term memory, are also obtained and updated through learning. The agent's learning style is influenced by emotion and personality. For example, in the Baysian learning, the agent will calculate his opponents' previous strategy frequencies and choose his best response that may be very different from his historical measures. If the agent has an easy-to-change personality, this shift seems natural; but if it is "stubborn," the transition may consider the past to a certain extent.

C. Working Memory and Long-Term Memory

These two parts contain the main beliefs of the agent and form the foundation of his reasoning. Beliefs in the memories are the internal view that the agent has about the world. They are not required to correspond with reality. Rather they could be outdated or distorted, but they are deemed absolutely correct by the agent. Working memory can be switched to the long-term memory when it is not relevant to the current problem any longer. Similarly, when the agent encounters a new problem, it will search the long-term memory and read the concerned beliefs into his working memory. According to the time and importance, beliefs in the two memories also impact the agent's attention.

D. Norm, Emotion and Personality, and Property State

The three modules are the reflection of the agent's internal states. Norm refers to a collection of constraints that imposed significant impacts on one's behavior to adapt to cultural or expectations of the whole society or an organization. These constraints are neither in the legislative level nor obligatory. But an agent will probably receive a punishment such as being isolated by others if it does not comply with them. For example, shaking hands to express a friendly attitude after sports is a widely used social norm. Almost every athlete adopts this behavior. Otherwise, he will incur drastic critics and successive detriments in his future career. Norm can be acquired by directly teaching from others as well as learning from the interactions. Emotion and personality are another two factors influencing one's decision. Emotion refers to temporary feelings characterized by intense mental activity and a high degree of pleasure or displeasure, while

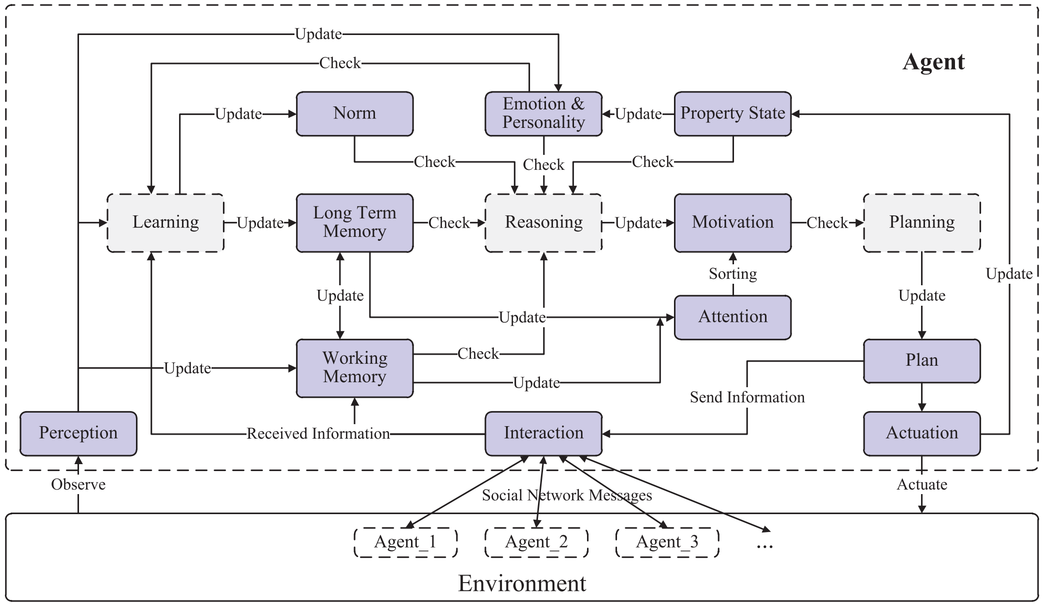


Fig. 1. General agent CA.

personality means the individual style or pattern of behavior. The two aspects are distinct from each other in that emotion plays a more influential role in decisions with limited time and uncertainty, whereas personality determines the long-term strategies. In this sense, personality is more "rational" than emotion. Since ABAS not only focuses on the final equilibrium in the long run but also is usually applied to investigate the dynamics of a particular social phenomenon in a short period of time, which may probably involve agents' temporary actions, emotion is incorporated to consider its effects. As the previously mentioned Baysian learning example, emotion and personality may affect the learning process. The basic emotion contains anger, disgust, fear, happiness, sadness, and surprise. In addition, personality is represented in various forms, such as dominance, influence, conscientiousness, and steadiness. Emotion can be dynamically updated through the agent's perception and physical state. Personality is not as erratic as the emotion. However, it is the personality that greatly determines the heterogeneity of individuals. The module property state includes one's physical conditions and social characteristics. In contrast with the norm and personality which lie on the psychological and cognitive levels, they can be viewed as a low level of state the agent has. Property state is determined by the actuation and the actual situation of environment. It also impacts the agent's reasoning and emotion in turn. A fatigue physical condition may give the agent a stressful mood.

E. Reasoning

Reasoning is perhaps the most central part of the decision-making. As can be seen in Fig. 1, it is a procedure based on multiple inputs. Beliefs in the working memory and long-term memory are the foundation of reasoning, while the norm, emotion and personality, and property state impose constraints on this process. Many decision algorithms or mechanisms can be adopted in the reasoning module. One of the most representative instances may be the utility maximization algorithm, which is very popular in economic studies. For a specific problem, the agent gives each solution candidate a utility value according to the current percept, the knowledge stored in long-term memory, and constraints from other modules. If a candidate does not satisfy some "hard" constraints such as violating a compulsive norm or cannot be conducted by the restriction of current physical conditions, its utility will set to be zero, and thus, this solution will be excluded from the agent's considerations. Finally, the candidate with the highest utility will be selected as the determination. It needs to be pointed out that the procedure explained before is a high-level reasoning, and its output is not concrete actions but motivations, which will be discussed in the following.

F. Motivation and Attention

Motivation includes the purposes that an agent pursues in various aspects. It can be seen as a concrete form of desires. In the daily life, one may concentrate on multiple problems of different fields, as he is endowed with multiple social roles. He may expect to complete a project as well as he can, and also look forward to keeping a compatible relationship with the customers. For each field, one pursuit (or some near optimal alternatives) is maintained in the motivation module until it is fully achieved. The motivation can be dynamically adjusted, as the reasoning result may change in different cycles. Attention reflects an agent's focus or the degree of the importance of the problems he tackles. In our model, attention is updated by the beliefs in memories according to their time (such as a deadline) and significance. Attention acts on the motivations derived from the reasoning procedure and arranges the motivations in a specific sequence, usually by attaching each a utility of importance. The motivation with high utility will be arranged at the top position and will be conducted in priority with its plans.

G. Planning

After the motivation generated and sorted, each motivation will be realized through a series of activities or actions named a plan. A motivation can be further decomposed into submotivations, and their corresponding activities are linked as an activity chain. A plan is constructed by dynamically computing or by searching the preliminarily established plan library. Similar to other procedures, planning can use many classic algorithms. For instance, an agent in traffic simulation can use dynamic programming to calculate his travel route according to his known congestion information.

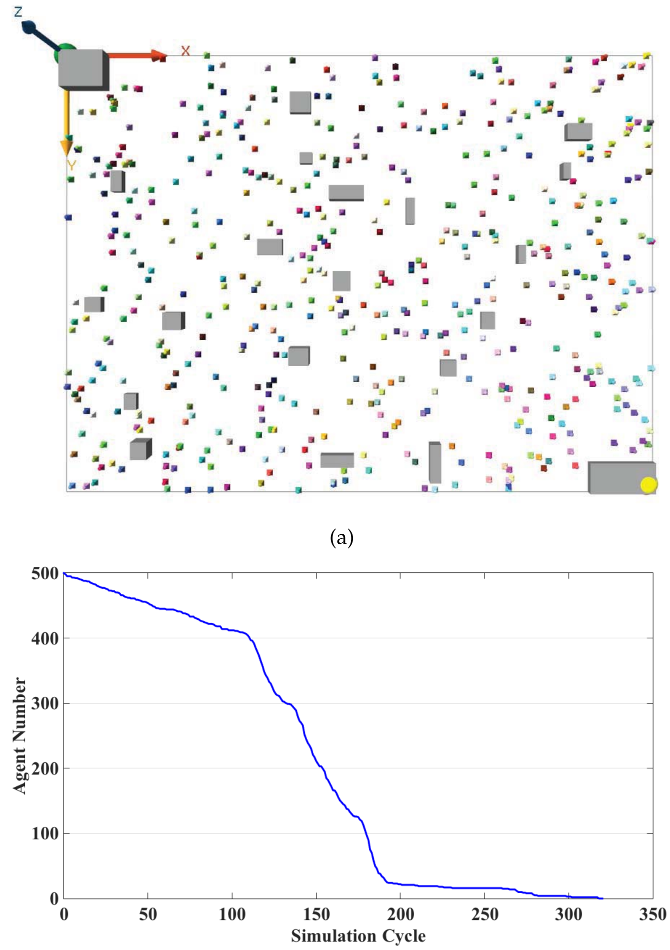
H. Interaction

Interaction refers to the communication and information exchange with other agents. Since ABAS mostly studies group dynamics and systemic behaviors, it is a very important facet that deserves extensive research. This is because the interaction among heterogeneous agents reflects the sociality, and complex social phenomena also emerge from the interactions. Usually, the interaction is implemented by passing messages among different agents. The received messages from other agents (actively or passively) are sent to working memory and learning procedure. And the information that the agent would like to "tell" or "show" his "friends" is dispatched through the interaction module. Protocols developed by the Foundation for Intelligent Physical Agents (FIPA), an international organization that is dedicated to promoting the interoperability of intelligent agents by openly developing specifications, are often adopted to guarantee efficient and regular communications.

In the ABAS simulation, each agent constantly repeats the decision-making process, which starts with observation and interaction. The received messages from other agents are unpacked in the interaction module and sent to the working memory and learning procedure. The perception from the environment is sent to these two components as well. Inputs of such two channels will update the working memory that concerns about current problems. The learning procedure, influenced by the emotional and personality, gives new knowledge to update the long-term memory and the social norms. After the module update, the agent will check his norm, emotion and personality, and current state to conduct reasoning based on his memories and generates multiple motivations. The motivations are sorted according to the attention, which means that the most concentrated one is arranged at the top of the motivation queue and will be processed in priority. Each motivation is used as the input of planning. The generated plans will send the information that the agent wants to express to others to the interaction module and will be executed via actuation. Finally, property state may be updated after the actuation.

V. Implementations

To illustrate the proposed architecture, this section gives two representative simulation scenarios—emergent evacuation and



(b)

Fig. 2. Evacuation experiment. (a) Environment. (b) Evacuation result.

population evolution. The former can be deemed as a "game," such as the famous SugarScape, while the latter tries to study a more practical social trend. These experiments aim to show how different ABMs can be organized and integrated by the architecture given in Fig. 1.

A. Emergent Evacuation

The objective of the evacuation experiment is to study the systemic short-term evolutionary dynamics by simulating behaviors of a heterogeneous crowd. Therefore, agent's computational model needs to consider the factors that affect personal short-term decisions, such as time constraints, emotions, fatigue, and so on. Specifically, perception, learning, working memory, long-term memory, emotion and personality, property state, reasoning, motivation, attention, planning, and actuation are implemented in the ABM. In the initialization, 500 agents are stochastically scattered in an area, which contains several buildings [Fig. 2(a)]. Every agent attempts to avoid buildings and evacuate to a safe point as fast as he can, but it depends on the impact of others and his own "physical" states. There are two safe points, located in the top-left corner and bottom-right corner. In each cycle, an agent percepts its surrounded environment by observing:

TABLE I
MODULES IN EVACUATION EXPERIMENT

Module	Implementation	Rule
Perception	Observe others' targets and buildings in his sight	$WM\_Vel \leftarrow Others'\ Vel$; $WM\_Tar \leftarrow Others'\ target\ points$; $nearby\_obstacles \leftarrow buildings\ nearby$
Learning	Calculate agent number for each destination	$tar1\_num \leftarrow WM\_Tar\ for\ target\ 1$; $tar2\_num \leftarrow WM\_Tar\ for\ target\ 2$; $impact \leftarrow \max\left(tar1\_num, tar2\_num\right)$
Working Memory	Current surrounded agents and buildings	-
Long-Term Memory	Memorized route already passed	-
Emotion & Personality	Agent's self-confidence	1(tired)-3(energetic)
Property State	Fatigue level	0(least confident)-9(most confident)
Reasoning	Determine destinations and building avoidance	$IF\ obDest! = currDest\ \&\&\ confidence < impact$ $THEN\ currDest \leftarrow obDest$; $IF\ nearby\_obstacles! = Null\ THEN\ avoid\ buildings$
Motivation	Achieving destination and avoid buildings	-
Attention	Sort the motivations	$avoid\ buildings > achieving\ destination$
Planning	Determine velocity according to fatigue level	$vel \leftarrow \{currVel.x * fatig, currVel.y * fatig\}$
Actuation	Conduct actions and update internal state	$IF\ fatig = 1\ THEN\ fatig \leftarrow 3\ ELSE\ fatig \leftarrow fatig - 1$ $LTMem \leftarrow LTMem + loc$

1) the velocity and target point of others within his sight range and 2) buildings if there are some. This information is stored in his working memory and sent to the learning module to calculate the number of agents that selects each destination. Then, the agent will perform a simple reasoning to determine which safe point he will choose or whether he will change his destination. The reasoning is influenced by three impacts: the learning results, his emotion and personality, and his property state, which represent social impact, subjective intention, and physical constraints. The emotion and personality includes a self-confidence level that shows his "persistence" or "pliableness." The property state here mainly refers to his fatigue level. It provides restrictions in the reasoning and subsequent planning. The results of reasoning—achieving the latest destination and avoiding buildings nearby—are his two motivations. Note that agent considers the building avoidance only when he observes some buildings in sight. And this motivation, sorted by the attention mechanism, is prior to the destination achievement. Then, the agent will determine the actions in the planning process according to the sequential motivations. The actions are finally executed via the actuation component and the property state as well as the long-term memory is updated. Implemented modules are summarized in Table I, and the final result (average of ten experiments) is shown in Fig. 2(b).

B. Population Evolution

Different from the emergent evacuation "game," population evolution experiment is much more realistic. The aim is to reproduce and further predict the population scale and its spatial distribution, so that we can investigate the systemic long-term dynamics or equilibrium. Agent in this test scenario is much more "rational." Specifically, his decision process is not suffered from time constraints. Thus, he always selects his optimal strategy in the decision cycle. The test scenario is Chinese population evolution. Fertility, mortality, and interprovincial migration are considered in our experiment. To illustrate the proposed architecture, perception, working memory, social norm, property state, reasoning, and actuation are implemented in the ABM. Our initial population is synthesized according to the 2000 national census data and located in each province [42] [Fig. 3(a)]. The scale factor is 10 000, which means actual 10 000 people are mapped as one agent in simulation. In each cycle, female agent who is between 20 and 50 and has no child will have a chance to get a child. The probability is calculated from the birth rate of that year. Each agent above 50 may die and be dropped out in simulation. Migration takes place among people under 50 years old. It mainly depends on three aspects: wage level, distance between current location and destination, and registration place. In agent's decision-making, he will compare the local average wage with that of each province. If the local wage is lower than others, he will get the impetus to go to the richer area. The greater the wage gaps, the stronger the impetus for emigration. In addition, if the agent is not registered in his current province, he will get extra impetus to go to his registration place. This can be viewed as a social norm (but not a legislation), since many people are familiar with their homes. The extra impetus is quantified by the distance between two provinces. The implemented modules are summarized in Table II, and the total population result (average of ten experiments) is shown in Fig. 3(b).

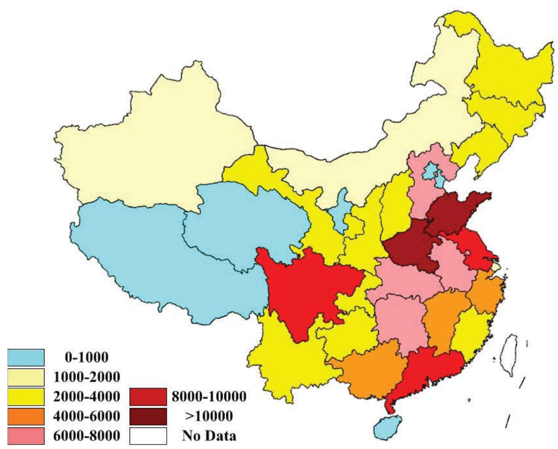
VI. CURRENT PROBLEMS AND POSSIBLE DIRECTIONS

Up to now, ABAS has been playing a primary role in the complex system study, and it is winning high expectations in various fields. Unlike the traditional mathematical method where crowds are analyzed through a unified formula, ABAS allows researchers to introduce individual heterogeneity. This flexibility, however, has brought several downsides as well. To put ABAS more practical, these problems are required to be solved carefully.

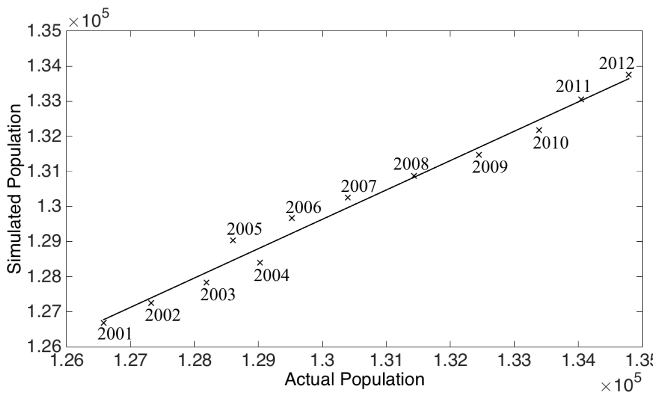
First, assumptions of ABMs are sometimes deemed arbitrary and disconnected from the literature [43]. This is because ABM aims to simulate the internal drive of human decisions. It attempts to illuminate the generation of individual behavior from the source of mentality and cognition.

TABLE II
MODULES IN POPULATION EVOLUTION EXPERIMENT

Module	Implementation	Rule
Perception	Collect wage levels of each province	$WM\_Wage \leftarrow Aver\_Wage$
Working Memory	Wage levels of each province	-
Social Norm	Registration impact factor: reg	0(not impact)-1(totally impact)
Property State	Current province, gender, age, registration province, etc.	-
Reasoning	Compute utilities and decides whether to have a child	$U[dest] \leftarrow wag\frac{salary[dest]-currSalary}{currSalary} + reg \cdot e^{-1/dist} + \gamma$
Actuation	Emigrate and create a new agent; die in a chance; update internal state	$IF\ migration\ THEN\ update\ current\ province$



(a)



(b)

Fig. 3. Population evolution experiment. (a) Initial distribution. (b) Total population result.

Usually, such factors seem obscure and bring many difficulties to the formal model construction. One of solutions for the dilemma may be the quantification of individual behavior via social experiments. This quantification may not be quite accurate but provides a relatively operational way to construct computational models. A well-known example comes from Berg *et al.* [44], who reported an investment game that became the prototypical trust game in the subsequent works. They conducted the experiment by endowing two players with \$10 each. In stage 1, the first mover decides how much money to pass to an anonymous second mover. All money passed are tripled. In stage 2, the second mover decides how much to return to the first mover. In the original experiment, out of 32 first movers, 30 sent positive amounts and only 2 sent 0, whereas out of 28 players who received amounts greater than \$1, 12 returned \$0 or \$1 and 12 returned more than their paired player sent them. So, the results clearly departed from the Nash equilibrium outcome that would be reached by perfectly rational and selfish players. This experiment has been replicated many times since then, showing that these results are quite robust. It is now widely accepted that trust and reciprocity are fundamental aspects of human social behavior, and the results can be used to achieve explicit computational models.

Second, ABM often exhibits too many degrees of freedom and is therefore nonfalsifiable [45]. Basically, this problem arises from the fact that ABM belongs to the category of microscopic models. In contrast with the analytic model, ABM incorporates many microscopic decision factors. And these factors may vary from person to person. This elicits the question that how to testify the model. To the best of our knowledge, the solution relies on both social experiments and collaborations among researchers from different areas. On the one hand, social experiments can determine whether a cognitive or psychological attribute has contribution to human decision-making, as the investment game cited before. They show solid evidence that the investigated factor is or is not included in the decision process. On the other hand, microscopic decision factors from different facets of individuals are studied by different disciplines. They can be integrated via collaborations of various scholars. Actually, high degree of freedom is one of the advantages that ABM has. Due to this flexibility, heterogeneity can be introduced with relatively fewer individual variables compared with analytic models. It avoids researchers to solve complicated equations and also enables us to study micro- and macroabnormal behaviors that may not appear in reality. Therefore, we merit a stronger prediction power but suffer from the high degree of freedom.

Third, ABM often lacks a sound empirical grounding and is often limited to some *ad hoc* calibration [46]. Similar to the second critique, this is a main deficiency that skeptics claim, and it is also caused by the microscopic property of ABM. The heterogeneity not only comes from the difference of personal decision factors but also the various levels of the same factor. However, those parameters are implicit to us and cannot be investigated in a large scale. This brings an obstacle to endow the model with a solid ground. To solve this problem, two directions may be promising. One is also the social experiments, which is already explained before. As the amount of money sent and received in the investment game, the values can be converted into normal ones and used as relative reward parameters. However, since extensive experiments are not feasible, the parameter values achieved in this way only represent the surveyed people and can be seen as prior knowledge. Thus, a more important approach to accurately calibrate the parameters is to use the overall statistical data. This requires seeking a relationship between the micromodel parameter and the macrostatistical distribution, so that the parameter value can be estimated through the overall features. As computational model is deterministic (random variables can be represented by their expectations), such kind of functions is not difficult to find.

Fourth, ABM is oftentimes poorly documented and hardly replicable [47]. This issue seems quite technological. Since ABAS attracts much more concentrations from different fields these days, a great amount of the literature has emerged. It is necessary to systematically collect the research achievements, especially classic models. Correspondingly, many algorithms are also developed to solve multidomain problems. Although most of them involve randomness to simulate stochastic behaviors, they are indeed replicable. What needs to be clarified here is that the replication does not refer to the accurate repetition of each metric, but rather their statistical approximation. Mostly, if an ABM can reproduce the trend of historical data, it should be deemed as reasonable. Thus, researchers are encouraged to open their algorithms as well as source codes, so that achievements can be replicated and systematically categorized.

Fifth, writing an ABM requires quite a lot of programming skills; code is often not reusable and projects are not incremental [48]. When implementing an ABM such as the proposed architecture in this paper, advanced programming skills are essential. It is because a good ABAS simulation model involves many modules and complicated communications. Moreover, projects and programs are usually isolated and cannot be reused. Based on the general guidelines and protocols published by FIPA, various multiagent platforms and agent-oriented programming language can be bridged. This facilitates the ABAS development by enabling researchers and engineers to concentrate on their agent function design without considering many trivial implementation details, but we can go further. Besides making the syntax of two programs compatible when they are integrated, it is more important to keep their semantics consistent. It requires a general architecture concerning multidomains and a common knowledge base to guarantee that a particular symbol in the two systems has identical meaning. The proposed architecture in this paper can play the role for the first stage of pursuit. If the existing agent programs or the new developed ones are (re)organized in a general architecture, they are more convenient to be integrated.

VII. Conclusion

ABAS has been playing a vital role in a complex social system analysis. It is applied in computational demography, transportation simulation, urban land use planning, computational economics, military computation, and many other domains. However, there is not a decision-making architecture concerning most facets of human complex behaviors, which brings an obstacle to merge and integrate different systems and programs. This paper proposes a general CA in ABAS and elucidates the agent's decision-making logics. Two simulations, emergent evacuation and population evolution, are presented to show that the proposed architecture is able to support different ABMs. Current problems of the agent-based modeling are summarized, and possible solutions are proposed in the end. Generally, social experiments, calibration via statistical distributions, common knowledge base, and collaborations among various fields are main factors that can put ABAS more practical in the future.

References

[1] P.-J. Ye, X. Wang, C. Chen, Y.-T. Lin, and F.-Y. Wang, "Hybrid agent modeling in population simulation: Current approaches and future directions," *J. Artif. Soc. Social Simul.*, vol. 19, no. 1, p. 12, 2016.

[2] A. Grow and J. V. Bavel, *Agent-Based Modelling in Population Studies: Concepts, Methods, and Applications*. Springer, 2017.

[3] F.-Y. Wang and S. Tang, "A framework for artificial transportation systems: From computer simulations to computational experiments," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Vienna, Austria, Sep. 2005, pp. 1130–1134.

[4] H. Zhao, S. Tang, and Y. Lv, "Generating artificial populations for traffic microsimulation," *IEEE Intell. Transp. Syst. Mag.*, vol. 1, no. 3, pp. 22–28, 2009.

[5] M. G. Richiardi, "Agent-based computational economics: A short introduction," *Knowl. Eng. Rev.*, vol. 27, no. S2, pp. 137–149, 2012.

[6] D. Wen, Y. Yuan, and X.-R. Li, "Artificial societies, computational experiments, and parallel systems: An investigation on a computational theory for complex socioeconomic systems," *IEEE Trans. Serv. Comput.*, vol. 6, no. 2, pp. 177–185, Apr./Jun. 2013.

[7] Y. Ou, S. Tang, and F.-Y. Wang, "Computational experiments for studying impacts of land use on traffic systems," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, St. Louis, MO, USA, 2010, pp. 1813–1818.

[8] F. Hosseinali, A. A. Alesheikh, and F. Nourian, "Assessing urban land-use development: Developing an agent-based model," *KSCE J. Civil Eng.*, vol. 19, no. 1, pp. 285–295, 2015.

[9] C. I. Siettos, C. Anastassopoulou, L. Russo, C. Grigoras, and E. Mylonakis, "Modeling the 2014 Ebola virus epidemic—Agent-based simulations, temporal analysis and future predictions for liberia and sierra leone," *PLoS Currents*, vol. 7, Mar. 2015, pp. 1–12. [Online]. Available: http://currents.plos.org/outbreaks/article/modeling-the-2014-ebola-virus-epidemic-agent-based-simulations-temporal-analysis-and-future-predictions-for-liberia-and-sierra-leone/pdf

[10] R. A. Nianogo and O. A. Arah, "Agent-based modeling of noncommunicable diseases: A systematic review," *Amer. J. Public Health*, vol. 105, no. 3, pp. e20–e31, 2015.

[11] G. Bichler and D. J. Birks. (2015). *Exposed: Agent-Based Simulation of Military Weapons Transfers*. [Online]. Available: https://www.researchgate.net/publication/277312662_Exposed_Agent-Based_Simulation_of_Military_Weapons_Transfers, doi: 10.13140/RG.2.1.1630.9927.

[12] W.-S. Yun, I.-C. Moon, and T.-E. Lee, "Agent-based simulation of time to decide: Military commands and time delays," *J. Artif. Soc. Social Simul.*, vol. 18, no. 4, p. 10, 2015.

[13] C. H. Builder and S. C. Bankes, "Artificial societies: A concept for basic research on the societal impacts of information technology," RAND Corp., Santa Monica, CA, USA, Tech. Rep. P7740, 1991. [Online]. Available: https://www.rand.org/pubs/papers/P7740.html

[14] J. von Neumann and A. W. Burks, *Theory of Self-Reproducing Automata*. Champaign, IL, USA: Univ. Illinois Press, 1966.

[15] M. Minsky, *The Society of Mind*. New York, NY, USA: Simon and Schuster, 1988.

[16] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. RA-2, no. 1, pp. 14–23, Mar. 1986.

[17] J. Ferber, "Simulating with reactive agents," in *Many Agent Simulation and Artificial Life*, E. Hillebrand and J. Stender, Eds. Amsterdam, The Netherlands: IOS Press, 1995, pp. 8–28.

[18] P. Maes, *Designing Autonomous Agents: Theory and Practice From Biology to Engineering and Back*. Cambridge, MA, USA: MIT Press, 1990.

[19] M. Scheutz and B. Brian Logan, "Affective vs. deliberative agent control," in *Proc. 8th Int. Conf. Artif. Life (ICAL)*, 2003, pp. 1–10.

[20] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.

[21] T. Ishida, *Parallel, Distributed and Multiagent Production Systems* (Lecture Notes in Artificial Intelligence), vol. 878. Berlin, Germany: Springer-Verlag, 1994. [Online]. Available: http://www.springer.com/cn/book/9783540586982

[22] J. Mcdermott and C. Forgy, "Production system conflict resolution strategies," Dept. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., 1976.
[23] M. E. Bratman, "Intention, plans and practical reason," Stanford Univ., Stanford, CA, USA, Tech. Rep. BF619.5.B73, 1987.
[24] R. H. Bordini, J. F. Hubner, and M. Wooldridge, *Programming Multi-Agent Systems in AgentSpeak Using Jason* (Wiley Series in Agent Technology). Hoboken, NJ, USA: Wiley, 2007.
[25] W. G. Kennedy, "Modelling human behaviour in agent-based models," in *Agent-Based Models of Geographical Systems*, A. J. Heppenstall, A. T. Crooks, L. M. See, and M. Batty, Eds. Dordrecht, The Netherlands: Springer, 2012, pp. 167–179.
[26] D. Pereira, E. Oliveira, and N. Moreira, "Formal modelling of emotions in BDI agents," in *Computational Logic in Multi-Agent Systems*, F. Sadri and K. Satoh, Eds. Berlin, Germany: Springer-Verlag, 2008, pp. 62–81.
[27] F. Dignum, D. Morley, E. A. Sonenberg, and L. Cavedon, "Towards socially sophisticated BDI agents," in *Proc. 4th Int. Conf. MultiAgent Syst.*, Jul. 2000, pp. 111–118.
[28] J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre, "Goal generation in the BOID architecture," *Cognit. Sci. Quart.*, vol. 2, nos. 3–4, pp. 428–447, 2002.
[29] M. Dastani and L. van der Torre, "Programming BOID-plan agents: Deliberating about conflicts among defeasible mental attitudes and plans," in *Proc. 3rd Int. Joint Conf. Auto. Agents Multiagent Syst.*, Washington, DC, USA, 2004, pp. 706–713.
[30] M. J. Kollingbaum, "Norm-governed practical reasoning agents," Ph.D. dissertation, Dept. Comput. Sci., Univ. Aberdeen, Aberdeen, Scotland, 2004.
[31] F. P. M. Dignum, C. Castelfranchi, C. Jonker, and J. Treur, " Deliberate normative agents: Principles and architecture," in *Proc. 6th Int. Workshop Intell. Agents VI, Agent Theor., Archit., Langu.*, Orlando FL, USA, 1999, pp. 364–378.
[32] G. Andrighetto, M. Campenni, R. Conte, and M. Paolucci, "On the immergence of norms: A normative agent architecture," in *Proc. AAAI Symp., Social Org. Aspects Intell.*, 2007, pp. 11–18.
[33] G. Andrighetto, R. Conte, P. Turrini, and M. Paolucci, "Emergence in the loop: Simulating the two way dynamics of norm innovation," in *Normative Multi-Agent Systems* (Dagstuhl Seminar Proceedings), G. Boella, L. van der Torre and H. Verhagen, Eds. Dec. 2007, pp. 1–29. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.169.1705&rep=rep1&type=pdf
[34] M. J. Kollingbaum and T. J. Norman, "Norm adoption in the NoA agent architecture," in *Proc. 2nd Int. Joint Conf. Auto. Agents Multi-Agent Syst.*, 2003, pp. 1038–1039.
[35] J. E. Laird, A. Newell, and P. S. Rosenbloom, "SOAR: An architecture for general intelligence," *Artif. Intell.*, vol. 33, no. 3, pp. 1–64, 1987.
[36] J. E. Laird, *The SOAR Cognitive Architecture*. Cambridge, MA, USA: MIT Press, 2012.
[37] R. Sun, "Memory systems within a cognitive architecture," *New Ideas Psychol.*, vol. 30, no. 2, pp. 227–240, 2012.
[38] R. Sun, T. Peterson, and E. Merrill, "A hybrid architecture for situated learning of reactive sequential decision making," *Appl. Intell.*, vol. 11, no. 1, pp. 109–127, 1999.
[39] J. R. Anderson and C. Lebiere, *The Atomic Components of Thought*. Mahwah, NJ, USA: Lawrence Erlbaum, 1998.
[40] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind," *Psychol. Rev.*, vol. 111, no. 4, pp. 1036–1060, 2004.
[41] J. M. Epstein and R. Axtell, *Growing Artificial Societies: Social Science From the Bottom Up*. Washington, DC, USA: Brookings Institution Press, 1996, p. 224.
[42] P. Ye, X. Hu, Y. Yuan, and F.-Y. Wang, "Population synthesis based on joint distribution inference without disaggregate samples," *J. Artif. Soc. Social Simul.*, vol. 20, no. 4, p. 16, 2017.
[43] M. Wickens, "How did we get to where we are now? Reflections on 50 years of macroeconomic and financial econometrics," *Manchester School*, vol. 83, no. S2, pp. 60–82, 2015.
[44] J. Berg, J. Dickhaut, and K. McCabe, "Trust, reciprocity, and social history," *Games Econ. Behavior*, vol. 10, no. 1, pp. 122–142, 1995.
[45] M. Richiardi. (2015). *The Future of Agent-Based Modeling. No 141, LABORatorio R. Revelli Working Papers Series, LABORatorio R. Revelli, Centre for Employment Studies*. [Online]. Available: http://EconPapers.repec.org/RePEc:cca:wplabo:141
[46] J. Grazzini and M. Richiardi, "Estimation of ergodic agent-based models by simulated minimum distance," *J. Econ. Dyn. Control*, vol. 51, pp. 148–165, Feb. 2015.
[47] R. Leombruni, M. G. Richiardi, N. J. Saam, and M. Sonnessa, "A common protocol for agent-based social simulation," *J. Artif. Soc. Social Simul.*, vol. 9, no. 1, p. 15, 2006.
[48] M. Richiardi and R. Leombruni, "Why are economists sceptical about agent-based simulations?" *Phys. A, Stat. Mech. Appl.*, vol. 355, no. 1, pp. 103–109, 2005.

Peijun Ye received the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2013.

He is currently an Assistant Researcher with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, and a Technical Engineer with the Qingdao Academy of Intelligent Industries, Qingdao, China. His current research interests include multiagent systems, social simulation, artificial intelligence, and intelligent transportation systems.

Dr. Ye serves as an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and the Secretary for the Social and Economic Computing ACM Chapter. He is also an invited Reviewer of the *Journal of Artificial Societies and Social Simulation*, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, *Science China*, the IEEE/CAA JOURNAL OF AUTOMATICA SINICA, and the IEEE International Conference on Intelligent Transportation Systems from 2011 to 2017.

Shuai Wang received the master's degree in control engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing.

His current research interests include social computing and parallel management.

Fei-Yue Wang (F'03) received the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined The University of Arizona, Tucson, AZ, USA, in 1990, where he became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Overseas Chinese Talents Program from the State Planning Council and the "100 Talent Program" from CAS. In 2002, he was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. From 2006 to 2010, he was the Vice President for Research, Education, and Academic Exchanges with the Institute of Automation, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory of Management and Control for Complex Systems, CAS. His current research interests include methods and applications for parallel systems, social computing, and knowledge automation.

Dr. Wang has been elected as a fellow of International Council on Systems Engineering, International Federation of Automatic Control, American Society of Mechanical Engineers (ASME), and American Association for the Advancement of Science. In 2007, he received the National Prize in Natural Sciences of China and was named the Outstanding Scientist by ACM for his research contributions in intelligent control and social computing. He received the IEEE Intelligent Transportation Systems (ITS) Outstanding Application and Research Awards in 2009, 2011, and 2015, and the IEEE SMC Norbert Wiener Award in 2014. Since 1997, he has served as the General or Program Chair of more than 20 IEEE, Institute for Operations Research and the Management Sciences, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, the Chinese Association for Science and Technology, USA, in 2005, and the American Zhu Kezhen Education Foundation from 2007 to 2008, and the Vice President of the ACM China Council from 2010 to 2011. Since 2008, he has been the Vice President and the Secretary General of the Chinese Association of Automation. He was the Founding Editor-in-Chief (EiC) of the *International Journal of Intelligent Control and Systems* from 1995 to 2000 and the *IEEE ITS Magazine* from 2006 to 2007, and an EiC of the IEEE INTELLIGENT SYSTEMS from 2009 to 2012 and the IEEE TRANSACTIONS ON ITS from 2009 to 2016. He is currently the Founding EiC of the IEEE/CAA JOURNAL OF AUTOMATICA SINICA and the *Chinese Journal of Command and Control*.