# Dynamically Tracking the Real World in an Agent-Based Model

H. Van Dyke Parunak[1(✉)], S. Hugh Brooks[2], Sven Brueckner[3], and Ravi Gupta[3]

[1] Soar Technology, Green Court Suite 600, P.O. Box 3600 Ann Arbor, MI 48105, USA
van.parunak@soartech.com
[2] P.O. Box 25275 Washington, DC 20027, USA
shb32@georgetown.edu
[3] Axon Connected, LLC, 2322 Blue Stone Hills Dr. Suite 20, Harrisonburg, VA 22801, USA
{sven.brueckner, ravi.gupta}@axonconnected.com

**Abstract.** Computational Social Science (CSS) models are most commonly used to articulate theories and explore their implications. As they become more mature, they are also valuable in monitoring real-world situations. Such applications require models to be linked to dynamic real-world data in real time. This paper explores this distinction in a specific application that tracks crowd violence in an urban setting.

**Keywords:** Forecasting · Calibration · Apoptosis · Stigmergy

## 1 Introduction

On May 20, 2012, NATO held a summit in Chicago, IL. Protesters planned a demonstration. They registered with the authorities for a permit, which specified a route ending near the secure area where NATO delegates and heads of state would be meeting. The Cook's County Sheriff's office invited NEK Advanced Security Group to demonstrate the usefulness of social media in tracking unrest, and NEK invited us to demonstrate how agent-based modeling could help monitor the demonstration in real-time and give near-term forecasts of possible "hot spots" requiring additional police attention. In response, we constructed and demonstrated a prototype of CAVE (Crowd Analysis for Violence Estimation).

Crowd simulation is an important and fairly mature area of computational social science (CSS). We do not offer any theoretical advances over previous research, from which we borrow liberally. However, we do apply these techniques in a novel way. In a research setting, CSS models serve to articulate a theory in a precise way, and (calibrated with static input data) to test the theory against historical observations. CAVE must continuously update itself with real-world data to provide an ongoing estimate of the state of the world a short distance into the future. Our contribution is demonstrating practical techniques for tracking the real world with a computational model.

S.J. Alam and H. Van Dyke Parunak (Eds.): MABS 2013, LNAI 8235, pp. 3–16, 2014.
DOI: 10.1007/978-3-642-54783-6_1, 

Section 2 of this paper distinguishes three applications of CSS: theory articulation, static prediction, and real-time monitoring and forecasting. Section 3 briefly reviews the particular CSS model that we adapt, summarizes its structure and operation (described more fully in a separate ODD specification [11]), and explains how we interface it with the real world. Section 4 reports the behavior of CAVE during the NATO summit, and Sect. 5 concludes.

## 2 CSS for Theory, Prediction, and Monitoring

CSS models can be applied in several different ways. We distinguish three.

### 2.1 Theory Articulation

A CSS model provides an unambiguous expression of the interaction of various causal influences within and among actors in a social scenario. It is a detailed embodiment of claims about factors and interactions that in a previous era could only be outlined verbally. A number of different formalisms for such models have been demonstrated [3]. We focus on agent-based models, which represent individuals (or small groups of individuals) as software agents [14]. Valuable insights concerning social phenomena can be gleaned from interview protocols (e.g., [15, 16]), but the resulting theories are difficult to test. A computational model is not only more precise than a verbal theory, but it also allows testing of hypotheses by executing the model. Even without external data, it can demonstrate testable qualitative trends and emergent behaviors that are not obvious from a verbal statement. For example, in the study on which CAVE is based [7], the tendency of groups to form as a function of crowd size is markedly different with two populations of different sizes than with balanced populations, and the emergence of violence depends on the size of the overall crowd.

### 2.2 Static Prediction

Qualitative agreement between simulation and observation is good, but accurate quantitative predictions (e.g., [1]) are even better, since their results are more directly comparable with observations from the real world, and they can support decisions that depend on a quantitative trade-off between cost and benefit. The first benefit is seen in an implementation that ingests live data at one point in time, then compares model outputs with subsequent observations. The second is clear in “what-if” exercises, in which the user runs the model off-line, then examines its results to guide a decision.

### 2.3 Real-Time Monitoring

Like static prediction models, CAVE seeks to align itself with data from the real world. However, it runs on-line, not off-line. It continuously ingests observed data and adjusts its configuration to give the user a continuously updated short-horizon forecast of the system being modeled.

Our approach is motivated by a fundamental limitation of predicting complex nonlinear systems. The farther one seeks to project the dynamics of the system, the more random the projection becomes, resulting in a "prediction horizon" beyond which such a prediction is no better than random. We have demonstrated this horizon in simple agent-based models [12]. The limitation is fundamental in nature, not due to noise in the input data or shortcomings in model accuracy [17].

Abstractly, we can view the system as a vector differential equation,

$$\frac{d\vec{x}}{dt} = f(\vec{x})$$

When $f$ is nonlinear, long-range prediction is impossible. However, it is often useful to anticipate the system's behavior a short distance into the future. A common technique is to fit a convenient low-order form for $f$ to the system's trajectory in the recent past, and then extrapolate this fit into the future (Fig. 1, [8]). Iterating this process provides the user with a limited look-ahead into the system's future. The process is like walking through the woods on a moonless night. The traveler cannot see the other side of the forest, but her flashlight can show her the next few meters, and when she has covered that distance, it can show her the next few meters beyond that.

Realizing the program of Fig. 1 directly requires specifying the state space of the system explicitly, writing a set of differential equations that characterize it, and fitting an analytical function to recent observations. Agent-based modeling is attractive for social systems just because it is difficult to define the complete state space and express the system's behavior in terms of analytical functions. Thus it is difficult to use this technique to produce a fit. This paper shows how to approximate the strategy of Fig. 1 in an agent-based social simulation.

To motivate our approaches, let's look in more detail at local approximations to the system's state trajectory (Fig. 2). At time $t_1$, we fit a linear model $a$. At a subsequent time $t_2 > t_1$, we fit model $b$. These two models differ in two ways, each of which leads to errors. We can use observational data to correct both kinds of error.

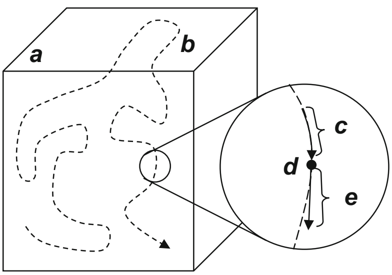


**Fig. 1.** Real-Time Monitoring of Complex Trajectories.—*a*: state space. *b*: system trajectory. *c*: recent observed system state. *d*: model update. *e*: short-range forecast

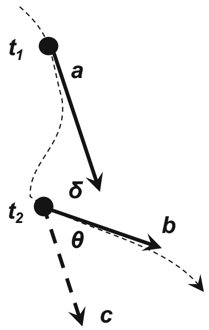


**Fig. 2.** Two adjustments in real-time monitoring.—Correcting system location ($\delta$) and model fit ($\theta$) at two time instances $t_1 < t_2$.

1. They differ in *direction*, which in this case corresponds to the internal structure and parameters of the model. The direction of the later fit $b$ differs from that of $a$ by $\theta$.
2. They differ in *origin*. Model $a$, an approximation, experiences an error $\delta$ with respect to the real system.

The simplest use of observational data is simply to restart the (original) model at the new, observed location, yielding model $c$. If the model parameters are not completely off the mark, the model still moves in the same general direction as the system.

In addition to reinitializing the model, we can also retune its parameters. When analytical approaches are not applicable, we use synthetic evolution. Figure 3 illustrates the polyagent approach [13], representing each real-world entity by a single persistent *avatar* and a swarm of *ghosts*. The avatar continuously inserts a stream of simple agents in a faster-than-real-time model of the environment, a short distance in the past, and evolves their behavioral parameters until they correspond to observed behavior, then lets them run into the future to generate a prediction. The ghosts are apoptotic: they die after a specified period, so the system does not become clogged with an increasing number of agents.

We have demonstrated this approach in combat modeling [10]. While effective, it requires detailed observations of each entity being modeled in order to tune the ghosts' behaviors. In CAVE, we have aggregate observations of crowd size and composition, but not individual observations. So we do not evolve agents' behavioral models, and do not maintain the multiple representations of the world at different epochs required by the polyagent model. The CAVE approach resembles vector $c$ in Fig. 2. Agent execution provides a short-range look-ahead into the future, while apoptosis (systematic removal of agents after a specified time limit) limits the depth of the look-ahead and allows us to reinitialize agents based on real-world data, shifting the origin (though not the parameters) of the agents.

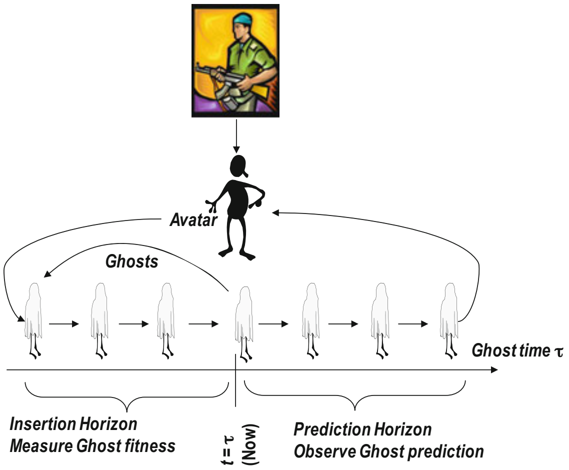


**Fig. 3.** Polyagent mechanism for dynamically learning agent behavioral parameters

# 3 The CAVE Model

CAVE draws on existing models of crowd psychology, using apoptosis and real-time data acquisition to adjust the model continuously.

## 3.1 Underlying CSS Theory

CAVE draws on two areas of research in crowd dynamics.

First, from the extensive literature on crowd psychology [21], we use the extended social identity model (ESIM) [7, 15, 16]. Unlike many other models, it is extensively supported by real-world evidence. While ESIM is not restricted to aggressiveness or violence, it has lent itself to several previous agent-based models of these behaviors [2, 7, 20] from which we draw inspiration. We adopt two conventions from [7].

1. Agents' aggressive behavior is driven largely by an internal state variable ([7]'s "aggression motivation") that in turn is influenced by events around them. In CAVE, this variable is called "Agitation."
2. Agents are not homogeneous, even within one side of a two-sided conflict, but differ in their degree of commitment to the cause.

Second, there is increasing anecdotal evidence that agitators in public events use network technology such as instant messaging and other social media in real-time to coordinate their activities [19], and that the contents of such media can be analyzed to track crowd sentiment [4].

An important feature of our work (like that of reference [7]) is that each agent is a simple rule-based entity without an elaborate model of individual cognition.

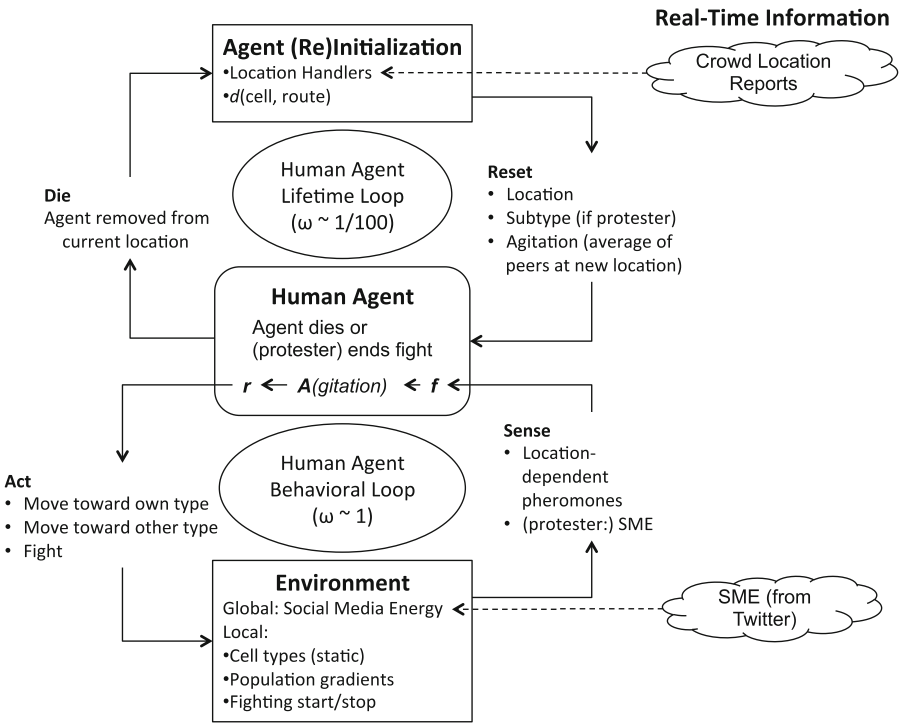


**Fig. 4.** Overall Information Flow in CAVE. The text discusses (in order) the environment (bottom of the figure), the human agent behavioral loop (bottom loop), the human agent lifetime loop (top loop), and real-time information (right side)

### 3.2 Model Structure

Figure 4 summarizes the overall information flow in CAVE. The following sections discuss the regions of the Figure. Further details about the implementation are provided in the standard ODD format [6] in a separate document [11].

**The Environment.**
The environment (bottom of Fig. 4) is a square lattice with cells 40 m on a side, representing downtown Chicago, derived from a GIS map.[1] We label each cell to indicate whether it contains a road, the approved protest route, the security zone within which the summit activities take place, and an extra-high security exclusion zone. The shading in Fig. 5 shows the cells corresponding to each of these categories. Agents are only created on roads. They can move off of roads, but their interactions are limited to roads, and in no case can they enter the Security or Exclusion zones.

[1] https://data.cityofchicago.org/browse?tags=shapefiles


**Fig. 5.** Coding of the CAVE environment

**Human Agents: Behavioral Loop.**
CAVE has two types of agents representing humans: protesters and police. These agents execute two loops. This section discusses the behavioral loop (at the bottom of Fig. 4), with a frequency $\omega$ of once per simulation step. The next section discusses the lifetime loop, which implements agent apoptosis.

Protesters are of three subtypes.

- *Leaders* aggressively seek to disrupt society, and energize their followers via social media. They can often be identified visually: they often wear bulky clothing to conceal hidden weapons, and also sometimes organize "black blocks," wearing black clothing and ski masks and moving cohesively to advertise their unified strength. Black blocks are known for engaging in violence and inciting clashes with the police.
- *Followers* accept the leader's agenda, but are not leaders.
- *Pacifists* are following the protesters out of curiosity more than ideology.

Police are of two types.

- *Patrol officers* are the usual cadre of an urban police force.
- *Riot police* have special training in dealing with unruly crowds, as well as specialized equipment such as riot shields and heavier padded armor.

The user initializes the total number of protesters and police, and the subtypes are allocated according to fixed proportions that are model parameters.

At the beginning of a run, the agents are distributed randomly on the roads in the environment that are outside the Security and Exclusion zones. Each road cell has a probability of receiving an agent that depends on how far the cell is from the protest route. The initialization function for protester agents concentrates them on the protest route, that for leaders lets them wander farther than pacifists, and that for police keeps them near the protest route but on it (so that they do not block traffic).

Each agent's behavior is determined by its level of Agitation, a variable that is defined by its drivers and its consequences. The *drivers* of an agent's Agitation are the presence of fighting in its cell, and (in the case of protesters) the level of Social Media Energy (SME) attested by tweets from the leaders. Increases in each of these lead to an increase in agitation. In the bottom loop of Fig. 4, an agent's input *f*(unction) translates the environmental state that it senses into a level of *A*(gitation), which is then translated via a *r*(ule) into one of three actions. The *consequences* of increased Agitation are that the agent first moves toward other agents of its own type (Protester or Police) for protection, then moves toward agents of the opposite type (in preparation for confrontation), and then engages in a fight.

Agents execute in random order, without replacement within a given simulation step. A single step corresponds to one min of real-world time; the actual elapsed time is much less, and depends on the speed of the processor.

Agents interact, not directly, but through a shared environment in which they are localized. The environment is not passive, but executes some processes that support the agent coordination. This pattern of coordination is called "stigmergy," a biological term that recalls the use of chemical markers (pheromones) by social insects [5, 9]. Agents interact only when they are on roads.

Each time an agent executes, it deposits digital pheromones on its cell in the environment, indicating its type, its presence, its level of Agitation, and whether it is engaged in Fight behavior. Similarly, agents sense a fight in the neighborhood by monitoring Fight pheromone, and move toward other agents of a specified type by climbing the gradient of the presence pheromone associated with agents of that type.

The environment supports pheromone-based interaction by evaporating all pheromones exponentially, thus removing obsolete information from the system. In AI terms, it provides basic truth maintenance (maintaining the consistency of a database), a task that is NP-complete in symbolic representations, with time complexity $O(1)$.

**Human Agents: Lifetime Loop.**

Apoptotic agents are central to CAVE's real-time updating. When an agent is created, it is assigned a lifetime is assigned from a uniform distribution on [50, 150]. When its lifetime is over, the agent is reinitialized to another location, based on real-time observations of the distribution of protesters and police. The top loop of Fig. 4 summarizes this life-cycle, whose frequency $\omega$ is on the order of 1/100.

Apoptotic agents address two challenges facing an agent model that seeks to be aligned with the real world.

1. How does the model adjust its *internal state* to stay aligned with the real world (Fig. 2)?
2. How do we manage the relation between the simulation's *internal clock* (which depends on processor speed) and the dynamics of the real world?

The reassignment of agents to new locations at the end of their life addresses the first challenge, of state alignment. Each time CAVE receives an observation of a concentration of protesters or police, it instantiates a special agent (a "location handler") at the location of the observation. If the current population of agents at a location is greater than the location handler desires (that is, greater than the observation), it inhibits the assignment of reinitialized agents to that location, and apoptosis eventually reduces the population to the observed level. If the current population is too low, the location handler attracts reinitialized agents to its location. Thus, with a half-life of 100 min (the mean of the lifetime assignment distribution), population levels in the model adjust to match observed levels.

Apoptosis also mitigates the problem of varying execution speed. The mean agent lifetime is 100 min. Because lifetimes are randomized in [50, 100], agents are reborn at different times. After a few hundred steps, the average agent has been active for about 53 steps (Fig. 6), and the strength of the estimated violence reflects a lookahead about this distance into the future. With a modern computer, a single simulation step takes only a few milliseconds, so the view on the display is looking roughly 53 min into the future. Agent apoptosis keeps them from running indefinitely into the future and formulating an unjustified long-range forecast.

**Data Sources.**
CAVE is continuously updated with two real-world data sources, shown on the right-hand side of Fig. 4: an estimate of *SME* from Twitter feeds (modulating the behavioral loop), and estimates of crowd density from human observers (modulating the lifetime loop).

Leaders use social media such as Twitter to communicate with their followers. The effectiveness of this communication mechanism depends on their Twitter handles and relevant Twitter hashtags being known, so police can monitor their tweets. CAVE processes this stream of tweets through a simple natural language processor that computes the frequency of profanity and other indications of unrest. The higher the frequency of such traffic, the higher our estimate of *SME*.

Observers on the ground enter local observations of crowd density to CAVE via a web or smartphone interface. Figure 7 shows the web interface, and Fig. 8 the smartphone interface. The smartphone's geolocation capability provides the location of the observation automatically, allowing police and other observers to update location estimates easily from the ground, and its display of violence estimates provides them with immediate awareness of likely trouble locations.

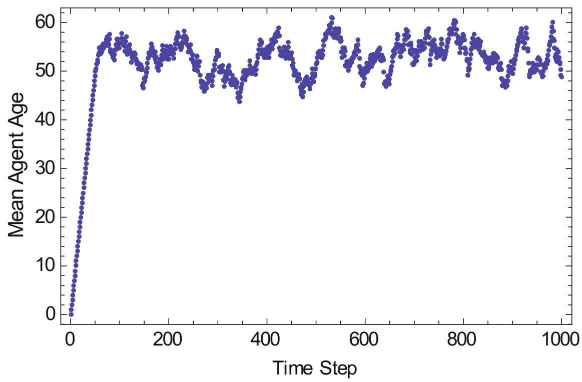


**Fig. 6.** Mean agent age = average lookahead

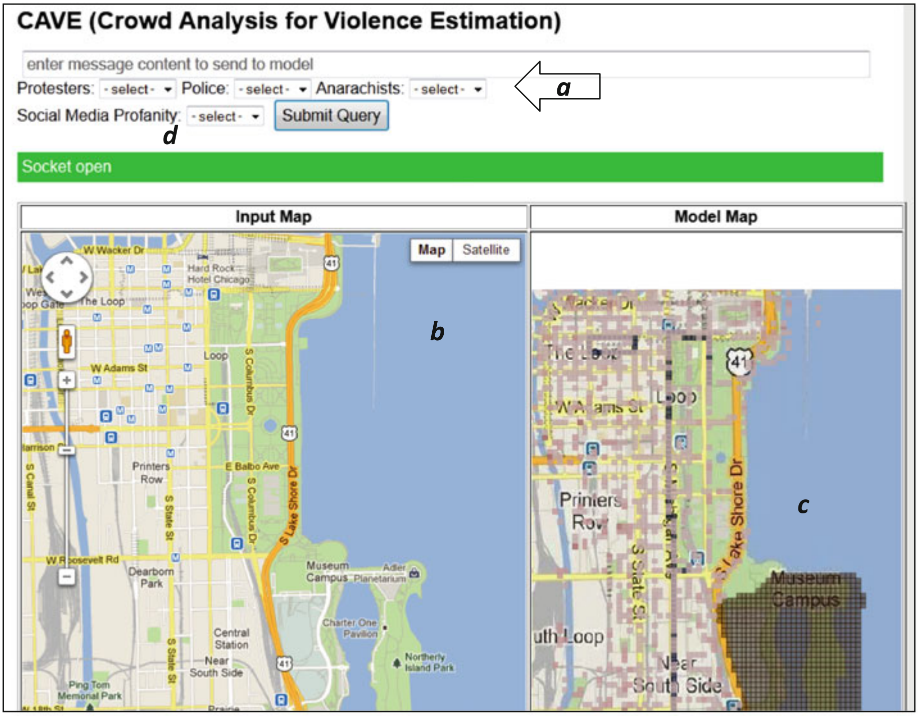


**Fig. 7.** CAVE interface. The operator enters estimated number of people of each type observed at a location (*a*), clicks on the left-hand map to show the location of the observation (*b*), and observes regions of high likelihood of violence on the right-hand map (*c*). In the prototype, *SME* is entered through this same interface (*d*), though the framework supports a direct feed from a NLP analysis program.

In the May 20 demonstration, SME estimates were entered by hand, based on manual monitoring of the Twitter feed. The smartphone interface was not deployed to observers on the ground, so crowd estimates were entered through the web interface based on real-time police reports and several streaming video feeds of the event recorded by protestors and journalists among the crowd.

## 4 Experience with the Model

The CAVE prototype shows the feasibility of integrating real-time data with an agent-based crowd simulation. The Cook County Sheriff's Department commented on the contribution of NEK's tool suite, "The intelligence we received from NEK was relayed to various law enforcement entities, such as the FBI, during the NATO event. The agencies were very appreciative of the information and it helped to enhance all of the intelligence information" [18].


**Fig. 8.** Smartphone interface. The smartphone reports its location automatically.

Though the objective of our model is to integrate and present real-world information rather than to study crowd theory, its emergent behavior does provide interesting evidence for the impact of social media. Figure 9 shows distribution of the violence estimate for the same distribution of protesters and police, but in two different conditions. High *SME* (middle map) leads to numerous regions of elevated risk of violence, but with low *SME* (right map), only one location near the exclusion zone anticipates high violence.

The nature of our engagement with the Cook County Sheriff's Department did not permit detailed assessment of CAVE's accuracy in this prototype application. Such validation is possible in principle; the main obstacles are social and bureaucratic, not scientific. The purpose of the model is to give law enforcement personnel advance notice of geospatial locations where violence may break out. If the model is valid, one expects a higher than average correlation between outbreaks of violence and predicted violence, with outbreaks tending to occur at locations where the model predicts high violence, and at delays of 53 min or less after the prediction. Two details of implementing this program require attention: collecting the data, and quantifying the temporal dynamics.

Social and ethical considerations make it undesirable to stimulate riots in order to validate agent-based models. In some social settings, "war games" can be staged to evaluate predictive mechanisms (the approach we took in [10]), but the expense is

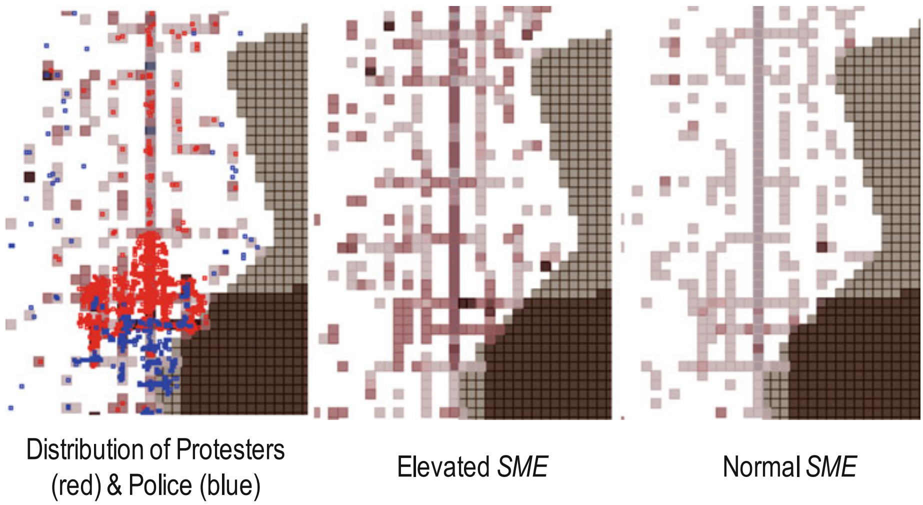


**Fig. 9.** Dependence of violence on *SME*

high, and questions remain about the fidelity of the reactions of actors who know that they are only playing a game. A more promising approach is to collect crowd observations and social media traffic during an actual event, then explore the correlation of CAVE's violence predictions with (say) the number of actual arrests for disorderly conduct as a function of place and time, after correcting for the number of officers available to conduct such arrests at each location in space-time. Government units will be reluctant to release such information for publication because of legal and privacy issues, but might conduct such an analysis in evaluating the technology for operational use.

The temporal dynamics are also problematic. In the current form of the model, the rates at which agitation builds up in the presence of violence or high SME, as well as the rate of its decay in the absence of stimulation and the threshold at which agitation turns into violence, are purely notional. They yield qualitatively coherent results. For the purpose of estimating areas at risk of violence anywhere within the 53 min look-ahead of the model (as opposed to the actual time at which violence breaks out), these results may be very useful. But the lead time of our predictions depends on the actual values of agitation growth and decay and the violence threshold. In fact, observation of the time delay between prediction and the distribution of actual violence with greatest spatial correlation with the prediction may enable us to obtain more realistic estimates of these critical parameters.

## 5 Conclusion

Computational Social Science models have reached a level of maturity that allows them to be used in practical applications. Many such applications, such as crowd monitoring, require the simulation to be continually updated on the basis of real-time information from the domain. CAVE demonstrates how a stigmergic agent-based simulation with apoptotic agents can achieve this objective.

Our objective in this paper has been to describe mechanisms for updating a model continuously with real-world data, not to make claims about its predictive accuracy. We have outlined one approach to such validation for future research, but note that it is fraught with ethical and legal challenges.

## References


1. Bert, F., Podesta, G., Rovere, S., North, M., Menendez, A., Laciana, C., Macal, C., Weber, E., Sydelko, P.: Agent-based modeling of land rental markets: comparison between simulated and observed prices in the Argentina Pampas. The Computational Social Science Society of the Americas (CSSSA 2011), Santa Fe, NM (2011)
2. Chao, W.M., Li, T.Y.: Simulating riot for virtual crowds with a social communication model. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part I. LNCS, vol. 6922, pp. 419–427. Springer, Heidelberg (2011)
3. Gilbert, N., Troitzsch, K.G.: Simulation for the Social Scientist. Open University Press, Buckingham (1999)
4. Gonzalez-Bailon, S., Borge-Holthoefer, J., Rivero, A., Moreno, Y.: The dynamics of protest recruitment through an online network. Sci. Rep. **1**(197) (2011). doi:10.1038/srep00197
5. Grassé, P.-P.: La Reconstruction du nid et les Coordinations Inter-Individuelles chez *Bellicositermes Natalensis et Cubitermes sp.* La théorie de la Stigmergie: Essai d'interprétation du Comportement des Termites Constructeurs. Insectes Sociaux **6**, 41–84 (1959)
6. Grimm, V., Berger, U., DeAngelis, D.L., Polhill, J.G., Giske, J., Railsback, S.F.: The ODD protocol: a review and first update. Ecol. Model. **221**(23), 2760–2768 (2010)
7. Jager, W., Popping, R., van de Sande, H.: Clustering and fighting in two-party crowds: simulating the approach-avoidance conflict. J. Artif. Soc. Soc. Simul. **4**(3) (2001)
8. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge University Press, Cambridge (1997)
9. Parunak, H.V.D.: 'Go to the Ant': engineering principles from natural agent systems. Ann. Oper. Res. **75**, 69–101 (1997)
10. Parunak, H.V.D.: Real-time agent characterization and prediction. International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'07), Industrial Track, pp. 1421–1428. ACM, Honolulu, Hawaii (2007)
11. Parunak, H.V.D.: ODD Protocol for CAVE. Soar Technology. http://www.abcresearch.org/papers/ODD4CAVE.pdf (2013)
12. Parunak, H.V.D., Belding, T.C., Brueckner, S.A.: Prediction horizons in agent models. In: Weyns, D., Brueckner, S., Demazeau, Y. (eds.) EEMMAS 2007. LNCS (LNAI), vol. 5049, pp. 88–102. Springer, Heidelberg (2008)
13. Parunak, H.V.D., Brueckner, S.: Concurrent modeling of alternative Worlds with polyagents. In: Proceedings of the 7th International Workshop on Multi-Agent-Based Simulation (MABS06, at AAMAS06), pp. 128–141. Springer (2006)
14. Parunak, H.V.D., Savit, R., Riolo, R.L.: Agent-based modeling vs. equation-based modeling: a case study and users' guide. In: Gilbert, N., Conte, R., Sichman, J.S. (eds.) MABS 1998. LNCS (LNAI), vol. 1534, pp. 10–25. Springer, Heidelberg (1998)

15. Reicher, S.D.: The St. Pauls' riot: an explanation of the limits of crowd action in terms of a social identity model. Eur. J. Soc. Psychol. **14**, 1–21 (1984)
16. Stott, C., Hutchison, P., Drury, J.: 'Hooligans' abroad? Inter-group dynamics, social identity and participation in collective 'disorder' at the 1998 World Cup Finals. Brit. J. Soc. Psychol. **40**, 359–384 (2001)
17. Strogatz, S.H.: Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering. Addison-Wesley, Reading (1994)
18. Towne, B.: Letter from Cook County Sheriff's Dept. to NEK. Letter to Porterfield, T., 23 May (2012)
19. Wasik, B.: #Riot: Self-Organized, Hyper-Networked Revolts—Coming to a City Near You. http://www.wired.com/magazine/2011/12/ff_riots/ (2011)
20. Wijermans, N., Jorna, R., Jager, W., van Vliet, T.: Modelling crowd dynamics: influence factors related to the probability of a riot. In: Proceedings of The Fourth European Social Simulation Association Conference (ESSA 2007) (2007)
21. Zeitz, K.M., Tan, H.M., Zeitz, C.J: Crowd behavior at mass gatherings: a literature review. Prehospital Disaster Med. **24**(1), 32–38 (2009)