

A Context-Based Approach to Detecting Miscreant Behavior and Collusion in Open Multiagent Systems

Larry Whitsel and Roy Turner

School of Computing and Information Science
University of Maine
Orono, ME 04469-5752 USA
{lwhitsel,rmt}@cs.umaine.edu

Abstract. Most multiagent systems (MAS) either assume cooperation on the part of the agents or assume that the agents are completely self-interested, for example, in the case of bidding and other market-based approaches. However, an interesting class of MAS is one that is fundamentally cooperative, yet open, and in which one or more of the agents may be self-interested. In such systems, there is the potential for agents to misbehave, i.e., to be *miscreants*. Detecting this is tricky and context-dependent. Even more difficult is the problem of detecting collusion between agents.

In this paper, we report on a project that is beginning to address this problem using a context-based approach. Features of the MAS' situation are used by a subset of the agents to identify it as an instance of one or more known contexts. Knowledge the agent(s) have about those contexts can then be used to directly detect miscreant behavior or collusion or to select the appropriate technique for the context with which to do so. The work is based on context-mediated behavior (CoMB), and it develops a new form of collusion detection called society-level analysis of motives (SLAM).

Most work on multiagent systems (MAS) has either assumed that the agents are all cooperative (e.g., in cooperative distributed problem solving approaches [1]) or all self-interested (e.g., in contracting and bidding approaches [7]). An interesting case, however, is a MAS where the fundamental intent is for it to be cooperative, but which may include self-interested agents. This class of MAS corresponds to many real-world systems, for example open client-server networks such as the Web, Wi-Fi networks, and so forth. In this kind of system, it is important to recognize when agents are at odds, either intentionally or unintentionally, with the goals of the system—when they are, as we term them, *miscreants*.

Detecting miscreant agents is difficult. It essentially is a problem of trust: Do we trust a particular agent (1) to be able to abide by the rules of the society, as well as the spirit of those rules (*capability*), and (2) to be willing to do so (*trustworthiness*)?

This paper appears in the Proceedings of the Seventh International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'11), which is available at www.springer.com . © 2011 Springer-Verlag.
--

A very important kind of miscreant behavior is *collusion*: multiple agents either acting in unison or according to a secret agreement that in either case violates the intention of the society. It is essentially unsanctioned coalitions among agents.

Collusion detection is difficult due to the variety of ways in which it can manifest. For example, explicit collusion often occurs with the fact of the collusion itself being hidden, for example, by a priori agreements, by using covert communication channels, or by masking any communication over shared channels. The agents involved may act differently under different circumstances, and the time course of their collusive behavior can vary, sometimes to intentionally mask the collusion. Implicit collusion can occur without any communication between the agents involved at all, and it is ephemeral: agents that collude in one situation may compete in another.

Even though in general there is great variation in miscreant behavior and collusion, in many domains there will be patterns that, if identified, can help detect the behavior. A simple example of this is the case in which two agents are known to have explicitly colluded in the past. When an agent observes these two agents entering the MAS, then this defines a context in which it should suspect collusion to be likely. There are other, society-level features as well that define contexts where miscreant behavior or collusion is likely. For example, in a MAS where there is some competition between agents for rewards (e.g., some types of contracting systems), there is adequate motive for self-interested agents to cheat, and implicit collusion should be watched for in which two agents “gang up” on a leader.

To capitalize on these patterns of features, we are developing an approach to handling miscreant agent behavior and collusion based on our prior work in context-mediated behavior (CoMB) for intelligent agent control [9, 8]. This approach explicitly represents contexts that are important from the standpoint of detecting miscreant behavior. These are represented as knowledge structures called *contextual schemas* (c-schemas), which in addition to describing contexts, also contain prescriptive knowledge about how to behave while in the context. One or more trusted agents in the MAS monitor the situation, constantly attempting to diagnose it as an instance of one or more contexts it knows. When this is possible, then the c-schemas representing the contexts can directly provide hypotheses about possible miscreant behavior and collusion as well as suggestions for techniques to detect or confirm such behavior.

1 Detecting Miscreant Behavior and Collusion

Previous work on detecting miscreant behavior generally treats the problem as one of making social trust decisions. Work has typically focused on three approaches, based on either: socio-cognitive trust utility (e.g., [2]), primarily cast in terms of game theory; reputation-based approaches (e.g., ([6]); or machine-learning approaches (e.g., [3]). Each of these approaches has its own strengths and weaknesses that determine for which situations it is appropriate. Elsewhere,

we report on simulation experiments that compare the techniques in our test domain [10] under different circumstances. While no single technique is appropriate for all situations, the group of techniques can be thought of as a toolbox from which a MAS can, once the context is known, select the appropriate response.

Detecting collusion is difficult, and it varies based on whether the collusion is implicit or explicit. Implicit collusion is difficult to detect because it can arise without any formal agreement between the agents or even without any communication (in-channel or out-of-channel) between the agents at all. For example, two agents may independently realize that a third is close to a victory in a game or close to winning a bid and begin to tacitly cooperate to bring the agent back into parity or to defeat it.

Detecting explicit collusion may be as simple as intercepting communication between the colluding agents, but even this is likely to be difficult. Communication can be encoded, with even the fact of the encoding be masked, or the communication may be covert (out-of-channel) communication. For example, an agent might communicate with another by a particular agreed-upon pattern of movements (for physical agents) that look innocent to others, or by taking actions to modify some system-level parameter, such as paging rate, I/O rate, or the availability of some resource [5] (for software agents).

2 Identifying Important Situational/Societal Features

In our approach, agents use features of the situation as well as society-level features to determine what the context is, which then allows the agent to identify or predict miscreant behavior or collusion. Here, we briefly discuss some of these important features.

Some features of the overall situation—for example, the potential existence of rewards for cheating—can predict the presence of miscreant agents. Features of the agents participating in the MAS are also important. Obviously, if all the agents are designed by the MAS’ designers, then trust can increase, but the MAS may be able to reason about the likelihood of cheating even if this is not the case. For example, if the MAS knows the reasoning abilities of an agent, it may in some cases be able to predict its responses to the ongoing situation, and so predict whether or not it will behave. Even when the reasoning processes may not be so transparent, the MAS may be able to reason about an agent’s likely behavior based on what it knows in general about the class of agent or about the particular agent, e.g., from past interactions or other sources of reputation data.

Features of the society can also give clues about the presence collusion. We use the term *society-level analysis of motive* (SLAM) for the process of using such features to detect collusion or other forms of miscreant behavior.

One such feature has to do with agent actions. Each action can be characterized as being beneficial, neutral, or harmful to each other agent in the system, as well as to the system as a whole. If an agent’s actions give a disproportionate benefit to another agent, then collusion between the two should be suspected.

Disproportionate benefit analysis can be used to detect possible collusion between agents. Actions are labeled as beneficial, neutral, or harmful by summing their effects over time on each other agent and comparing the effects to the average level of benefit of all actions in the society. A side-effect of this analysis will be to also identify an agent’s aggressiveness even if collusion is not present. For example, we may find that some agent is always more harmful or more helpful than is the norm to all other agents, or some particular class of agents. This can be used to help detect non-collusive miscreants at the society level, and at the individual agent level, it can be used to adjust an agent’s responses to the aggressive or altruistic agent.

Another feature that can indicate collusion is uncharacteristic behavior by one or more agents, particularly when paired with other changes in the situation. For example, if a normally neutral agent suddenly becomes more aggressive or docile when a new agent enters the system, then a reasonable hypothesis is that the two occurrences are linked, possibly due to collusion between the agents.

Other features that could be indicative of collusion have to do with detecting paired features of agents. For example, implicit collusion could be found by measuring the correlation between actions and comparing that correlation with the societal mean, say by “ganging up on” an opponent, making similar bids, or paired postures (e.g., aggressive/submissive). Although this can happen in explicit collusion as well, here there is a richer set of behavior available due to the ability of the colluding agents to agree on their roles and the ability to arrange to share risks and rewards. This might allow agents to agree to take complementary actions, for example. Of course, sophisticated agents might attempt to obfuscate collusion, and consequently any pairing of actions or postures would have to be detected statistically, over time.

Other attributes of the MAS agents can also serve as features predictive of possible collusion. For example, agents owned by the same organization or even arriving together into society might be worth watching for signs of collusion.

3 Using Contextual Knowledge

Miscreant behavior, including collusion, is more likely in some situations than others, and it presents differently in different situations as well. Consequently, the approach we take to miscreant detection is based on context-mediated behavior (CoMB), which uses a priori contextual knowledge to determine appropriate behavior in a particular situation. Here, that behavior is the process of detecting miscreants.

For now, we assume that there are one or more trusted agents in the multi-agent system that are tasked with (or at least capable of) detecting miscreant behavior. Since these agents primarily use a society-level analysis of motive, we call them SLAM agents. When a SLAM agent detects such behavior, its responsibility is to alert the rest of the MAS or in some other way take action. What action to take, while itself context-dependent, is MAS-dependent. Consequently, it is not the focus of this paper.

In our approach, a SLAM agent makes use of knowledge about the context to detect miscreant behavior. We make a distinction between a *situation* and a *context*. For a MAS, its situation is the sum total of all features, observable or otherwise, of itself, others, and its environment. Situations can be grouped, with the members of each group all having the same or nearly the same implications for the MAS in terms of predictions about outcomes of actions or events, likelihood of miscreant behavior and the type of that behavior, its agents' appropriate behavior, and other inferences that can be made about it, its agents, or its environment.

These groups of situations are what we mean by *context*: A context is a class of situation with important implications for a MAS or its agents. In our example contexts might be: detecting an aggressive agent, recognizing when two known compatriots are present, interacting when there is a reputation system in place, and so forth. Note that a given situation may be an instance of more than one context; for example, the MAS might be in the context in which two known compatriots are present and there is also a reputation system in place.

Our approach explicitly represents known contexts as knowledge structures called *contextual schemas* (c-schemas). A c-schema describes a class of situation as well as provides predictions about possibly unseen features of the situation which can help the MAS or its agents appropriately disambiguate new information. The features we discussed above define the space of contexts that are important, and they form the basis for the representation of the c-schemas.

Context assessment is the process of determining of which contexts a given situation is an instance. It is a diagnostic process, with features of the situation playing the role of signs and symptoms and contexts playing the role of "diseases". The descriptive knowledge contained in c-schemas is used to diagnose the situation as being an instance of one or more contexts the c-schemas represent. In past work, we have used an abductive diagnostic process based on the work of Miller, Pople, and Myers [4].

The c-schemas comprising the context assessment help the agent or MAS decide how to behave. In general, we are interested in using context to help agents in the MAS decide how to participate in the society as well as how to detect miscreant behavior. Our prior work in CoMB addresses the former; here, we focus on the latter.

A particular context assessment can directly serve as a hypothesis that miscreant behavior is occurring. That is, the context might be, essentially, "a miscreant agent is present" or "collusion is occurring". Depending on the degree of belief associated with the hypothesis, which will depend both on how strongly the agent believes its assessment and how strongly the c-schema in question predicts the offending behavior, it may or may not need to gather additional evidence to confirm the hypothesis.

Context can also focus attention on salient features of the situation, for example, unusual communication or the possibility of a covert channel. Although these might be noticed otherwise with additional reasoning, using c-schemas allows the features to be noticed automatically as a side-effect of context assessment.

C-schemas can also tailor an agent’s reasoning techniques to fit the situation. For example, in our current work on SLAM, we intend to build a dynamic decision network (DDN) for each other agent in the system, adjusting its belief about their attitudes each time an action is observed. Instead of building these structures from scratch when a new agent enters the system, we can store parameters of the networks in c-schemas representing contexts involving the agents. Then, when the agent is seen again, the c-schemas will be retrieved and will allow the networks to be immediately re-instantiated.

It is important for an agent to always maintain an accurate view of the context, even as the situation changes. This can be done in several ways. First, the agent can re-assess its context on a regular basis, and when the assessment produces a new set of c-schemas, the new merged c-schemas become the new context representation and behavior changes automatically. Second, we can identify a set of context change-inducing events (CCIE) that agents should look for. When these are detected, then context assessment would be done. Examples of CCIEs we have so far identified include: detection of collusion; change in composition of the MAS due to an agent entering or exiting; detection of unusual message traffic or covert channels; and, for some kinds of domains, temporal characteristics, such as immediately following startup or at a critical stage of a game or problem-solving session. Finally, c-schemas can themselves suggest when the context should change by predicting features or events that indicate the c-schema is no longer a good fit for the situation or that suggest other c-schemas that are better.

While we have not yet addressed the question of the origin of c-schemas, CoMB itself has the position that c-schemas are actually generalized cases of problem solving and that they can be learned from experience via similarity-based or other kinds of learning. Learning is not the focus of this work, however, and we anticipate, initially at least, hand-crafting c-schemas for our agents to use.

4 Conclusion and Future Work

In this paper, we have described a context-based approach to the problem of miscreant behavior detection that is being developed, in which known contexts are explicitly represented and reasoned about, and in which the context representations (c-schemas) contain context-appropriate knowledge the agents can use to guide their decision making. We believe that this approach will allow agents to more easily detect such behavior by using knowledge that is appropriate to the current context.

Although work on context-mediated behavior, on which this work is based, is somewhat mature, the work reported in this paper is still in the early stages and is the subject of a PhD dissertation that is in progress. The short term will see a more complete design and implementation of the context-based SLAM approach described.

In the longer term, we anticipate this approach being the basis for more general context-aware multiagent systems. In such systems, context would itself be a first-class object for discussion and reasoning about by the agents, and context assessment would be a shared activity among at least a subset of the agents. Such systems would be able to adjust their behavior and possibly their form to fit their evolving context.

References

1. Durfee, E.H.: Distributed problem solving and planning. In: Weiss, G. (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA (1999)
2. Falcone, R., Castelfranchi, C.: Social trust: A cognitive approach. In: Castelfranchi, C., Tan, Y. (eds.) *Trust and Deception in Virtual Societies*, pp. 55–90. Kluwer Academic Publishers (2001)
3. Izquierdo, L., Izquierdo, S.: Dynamics of the Bush–Mosteller learning algorithm in 2x2 games. In: Weber, C., Elshaw, M., Mayer, N. (eds.) *Reinforcement Learning: Theory and Applications*, p. 424. I-Tech Education and Publishing, Vienna (2008)
4. Miller, R.A., Pople, H.E., Myers, J.D.: INTERNIST–1, an experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine* 307, 468–476 (1982)
5. Moskowitz, I., Kang, M.: Covert channels—here to stay? In: *Computer Assurance, 1994. COMPASS '94 Safety, Reliability, Fault Tolerance, Concurrency and Real Time, Security*. Proceedings of the Ninth Annual Conference on. pp. 235–243 (jun-1 jul 1994)
6. Mui, L., Mohtashemi, M., Halberstadt, A.: Notions of reputation in multi-agents systems: a review. In: *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. pp. 280–287. AAMAS '02, ACM, New York, NY, USA (2002), <http://doi.acm.org/10.1145/544741.544807>
7. Sandholm, T.: Distributed rational decision making. In: Weiss, G. (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA (1999)
8. Turner, R.M.: *Adaptive Reasoning for Real-World Problems: A Schema-Based Approach*. Lawrence Erlbaum Associates, Hillsdale, NJ (1994)
9. Turner, R.M.: Context-mediated behavior for intelligent agents. *International Journal of Human–Computer Studies* 48(3), 307–330 (March 1998)
10. Whitsel, L.T.: *A Simulator for Rule-based Agent Trust Decisions*. Master's thesis, University of Maine, Department of Computer Science, Orono, Maine (2010)