Group-Based Transactive Memory to implement Computationally-Plausible Social Cognition in Agents

Geoffrey P. Morgan
Kenneth Joseph
Kathleen M. Carley
Carnegie Mellon University
School of Computer Science
Institute for Software Research
5000 Forbes Ave
Pittsburgh, PA 15213
gmorgan@cs.cmu.edu, kjoseph@csc.mu.edu, kathleen.carley@cs.cmu.edu

Keywords:
Bounded Rationality, Transactive Memory, Social Agents, Network-Centric Simulation

ABSTRACT: There is a need for cognitively bounded implementations of transactive memory for agents. To do this, we use schema theory and tiered social cognition to implement Mead's Generalized Other (1925). We then compared our new implementation, Construct-ML, with a prior implementation of the same simulation, Construct-O. We were not able to replicate all of the patterns suggested by Construct-O's results. However, the pattern validity of Construct-ML improves as agents have more cognitive resources, which is suggestive and interesting.

1. Introduction

Social cognition is the ability to encode and retrieve information about other social entities. Humans, our social agents of interest, frequently find it useful to retain knowledge of other actors (social knowledge) as well as knowledge about the world (general knowledge).

Following in the traditions of the Carnegie School (Cyert & March, 1963; March & Simon, 1958; Simon, 1957), these social agents may not be able to access what they know all the time, they may not know what they know, and what they know may be wrong. Similarly, their understanding of other actors is error-prone and perception-based. Regardless, humans use what they think they know about other actors to inform their behavior. If for example, someone needs medical advice, do they consult a doctor or a carpenter? They consult the doctor, naturally, because they have made an inference that doctors tend to have, as a group, knowledge in the medical field. But the same actor would not always seek out the doctor, or if they did they would be ill-served as their house fell down around their ears for want of competent advice! We intend to take advantage of not only this important capability of humans, but also their inferential mechanisms, in this work.

Construct (Carley, 1990, 1991; Carley, Martin, & Hirshman, 2009) is a network-centric agent-based simulation of knowledge diffusion within groups. Agents communicate information to other agents. Agents may forget knowledge they possess. How information diffuses within a group depends on multiple factors: the preferences of individual agents, the initial knowledge of each agent, and the social ties between those agents.

Agents in Construct, like humans, both have knowledge about the world, represented as a knowledge bit array, and knowledge about what other agents know, called Transactive Memory (Wegner, 1995), represented as a per-ego matrix of alters by knowledge. An alter is a potential interaction partner of each individual. In a fully connected system of five actors, there are five (n) individuals and twenty-five ($n * n$) alters; Construct agents may interact with themselves. In earlier iterations of Construct, all agents had representations of all other possible communication partners.

Although in this work we use a very small population to allow for direct model comparison to prior work. Construct has successfully supported hundreds of agents existing in a simulation environment at a single time. However, even though computer hardware has gotten faster and ever more capable, the growth of computational expense limits earlier versions of Construct from being useful when considering simulations of large populations.

Construct's computational expense stems from two factors: the size of the knowledge array each agent may possess, and the size of each agent's transactive memory. Of these two, transactive memory is the dominant term for computational expense. Further, as the size of the agent population increases, the cognitive power ascribed to each agent becomes less plausible. Providing hard limits on the alter list of each agent is tenable, but tends to limit the applicability of the modeling technology to situations where spontaneous link formation is unlikely.

In this paper, we present a method for bounding the cost of transactive memory within Construct by implementing Mead's Generalized Other (1925). This change improves agent fidelity while capping the costs of transactive memory, allowing many more agents to exist simultaneously within a Construct environment. We suggest that other agent technologies which could take advantage of transactive memory may find our implementation useful and instructive to their own work. We also believe that this change may allow Construct to model many more phenomena than is currently feasible, but we reserve detailed discussion of those possibilities for other work (Joseph, Morgan, Martin, & Carley, 2014). For purposes of clarity to comparisons with older forms of Construct, we refer to the bounded transactive memory version of Construct as Construct-ML.

2. Prior Work

In this section, we describe the related work that has contributed to our approach towards conserving computational resources while also improving the model fidelity of these agents. We conclude with a summary of the extension's feature and their implications for our modeled individuals.

2.1 Transactive Memory

Individuals often find it valuable to retain an idea of the state of other individuals. We use Wegner's (1995) description of a networked file system as our narrative. In a networked file system, individual units are assumed to have finite storage capacity. Consequently, information is spread across many of these units, as capacity and demands allow. Units in such a system need some method of accessing information not stored locally in an efficient manner. One solution is that each unit may have information on what other units are likely to be able to access.

It is important, of course, that the depth of knowledge about other agents, which should improve the access of off-board storage, be balanced with the size constraints of that information. A unit must hold some non-trivial amount of local information as well as a representation of what other units may know.

We can think of Transactive Memory's representation as a three element tuple. Each agent *i* has for alter *j* some understanding of the amount of knowledge that alter *j* has about information set *s*. Previous work has varied representations of this *ijs* tuple. Palazzolo and his collaborators (2006), represented transactive memory as a single continuous value representing knowledge of set s (which they call topics) for each agent for each alter. In an alternative approach, Carley and Ren (2001) represented each sub-element of a topic *t* in each agent's representation of their alters. Thus, an agent's perception of an alter's mastery of a particular topic *t* is a proportion of the elements the agent believes the alter possesses which makes up *t*.

Each of these representations, considered naively, would prove onerous for a cognitively limited human agent. As the size of the population increases, the amount of information required for maintaining transactive memory rapidly dwarfs the amount of direct information present in each networked file system. The principal contribution of this work is a flexible mechanism that allows agents to make educated assessments of alters without needing to store representations of all alters. Our mechanism takes advantage of tiered social cognition, discussed in the next section.

2.2 Tiered Social Cognition

In our discussion of Transactive Memory, we identified that information processing units in an information-rich world need to either keep explicit state about the state of alters or must have the ability to generate useful predictions about the state of those alters. In this section, we discuss one way humans, our information processors of interest, generate useful predictions about the state of alters without needing to keep extensive state on those alters. They generate these predictions through group affiliation.

Our inspiration for this mechanism rests on work in ethics and social control by Mead (1925). Mead posited that people can make inferential statements on the nature of ethical behavior within their local context of the form, “people of type X tend to do thing Y”. A person can evaluate their own behavior by determining that they are a person of type X and therefore should consider doing Y. Mead called this aggregate of statements the ‘generalized other’, and allowed that many of these inferential statements could exist concurrently within a person’s mind.

These inferential statements may refer to concepts not only of what alters may be able to do, but also to what these alters may know or believe. As Mead (1925, pg. 275) states, “Social control depends, then, upon the degree to which the individuals in society are able to assume the attitudes of the others who are involved with them in common endeavor.” Just as some information processing units may differ on their chosen representations for alters, humans may have more or less nuanced constructs of other humans, and these constructs may include representations of actions, beliefs, and knowledge.

We focus on the last of these objects, and thus can narrow Mead’s statement, to say that “people of type X tend to have knowledge Y”. But how do we define types? Mead suggested that there exists both a large common group, called in his work “society”, but also, independently and concurrently, inferential statements of all groups of which the individual is aware. Thus, we can again transform the structure of the inferential statement to this form “People who are members of Group X tend to have knowledge Y”.

Mead’s argument for social perception is supported by concepts in schema theory (Rumelhart, 1978, 1980). A schema in schema theory is a data structure for representing the generic concepts stored in memory. In schema theory, each individual has a hierarchy of schema that may be applicable to any of various environmental conditions the individual encounters.

The concept can be clarified through examining a specific scenario: confronted with someone examining our neighbor’s wooden porch, we may ask ourselves, “Who is this person?” They may be a professional carpenter repairing the porch, a city inspector checking code, a burglar investigating a prospective target, or various other possibilities. We examine the person’s actions, their apparent attitude, their appearance, their clothes, and use that information, along with relevant historical knowledge, to make an educated guess to answer our own question. In the process of making that guess, we allow whatever knowledge we have that may be applicable to apply. On a social level, we may apply any of three levels of schema to help us answer the question.

- Personal: We know this specific person.
- Group: We don’t know this person, but we can infer that they are members of one or more relevant groups (to us).
- Global: We know this is a person.

We will use that answer to inform future action. We may confront the individual, we may mention it to our neighbor discreetly, or we may do nothing. Schema that help us understand who other people are and what they are likely to know are called “Social Schema” (Kuethe, 1962). These produced social schema are culturally dependent (Little, 1968), but we do not expect that the generative mechanism for these social schema to be culturally dependent.

In schema theory, the availability of schema is determined by environmental cues. Schemas are available if they are relevant. Irrelevant schema do not occupy the individual’s time. Schema-like representations in cognitive agent systems (Anderson, 1996) have found that it is possible for agents to have many schema (implemented as production rules) simultaneously and exhibit human-like cognition as they learn to perform tasks by activating the appropriate rule-sets for the task at hand. Work by Duong and Reilly (1995) used a hierarchy of neural-networks to implement schema theory and model Mead’s Symbolic Interactionism (Mead, 1922), producing a model of racial bias in hiring.

Anderson and his collaborators (2004) suggest, and give empirical evidence, that chunks of our memory are “activated” when they are used, and that this activation decays over time with non-use. We can thus associate schemas agents have with an “activation score”, allowing us to determine their likelihood of use by the agent.

These activation scores, according to Anderson, determine whether or not we are able to recall a chunk or not. If the agent cannot recall the chunk, they must do without it.

Our work takes advantage of the computational tractability suggested by Anderson's approach, but changes the granularity of the activated chunk. Rather than each chunk representing a single schema-object, of which there be many for a single alter, each chunk represents an alter or a membership group to which an alter can belong.

Thus, if we consider each interaction with an alter to be an "activation", alters which are frequently contacted will have high activation scores, as will groups to which those members belong or about which we receive information. Group schemas will tend to be sparser but more durable than individual schema.

3. Implementing Social Cognition via Bounded Transactive Memory

In the previous sections, we have discussed transactive memory, schema theory and memory activation and how these elements all play a part in an agent's social cognition. Through tiered social cognition, we can implement a computationally efficient and cognitively plausible form of transactive memory. Agents who have this capability must be able to be:

- Form expectations about groups, including the Generalized Other
- Revise these expectations
- Generalize about others based on group membership
- Keep track of specific alters of interest
- Revise their expectations of specific alters

In this section, we will discuss our implementation within Construct (Carley, 1991), a validated simulation of information diffusion. We modify the prior transactive memory implementation from Carley and Ren (2001) in three important ways:

- We add transactive memory elements for groups, of similar form to those for alters.
- Transactive memory elements have an activation score, which changes over time as agents interact
- Transactive memory elements may be lost through disuse
- We track the origination time of schemas, which allows for differential treatment of groups and individuals.

As in previous work (Carley & Ren, 2001), we represent a "schema" as a transactive memory vector – a series of *K* bits, where *K* represents the number of knowledge pieces, or "facts", in the system. Each bit represents the ego's perception of the knowledge of the associated alter, group or generalized other. In the previous work, there was one transactive memory vector for each alter an agent could potentially interact with. In this work, a transactive memory vector may exist for each alter, but also may exist for each group.

These schemas are, as mentioned, arranged hierarchically. An agent determines what an alter knows by starting at the lowest level of the schema hierarchy – the personal level. If that schema is activated above the threshold, then the agent uses this schema to understand the knowledge of the alter. If not, the agent will "construct" the knowledge of the alter based on the groups he is aware the alter is in. If the alter belongs to no groups, then the agent uses his knowledge of what he expects "everyone" to know, which we refer to as his transactive memory of the "generalized other".

As suggested, our new model still allows agents to determine a value their belief that each alter holds any knowledge set, and therefore trivially captures all five of the behaviors listed that transactive memory systems introduce into simulations. However, our model adds significant functionality in each of these categories, as described below:

- **Forgetting-** Activation equations provide a way for us to directly promote the concept of forgetting – at some point, we actually forget the knowledge of alters and must later reconstruct it
- **Means of Determining Whom to Interact with**-the new implementation now allows us to compute non-zero likelihoods of interaction for *all* agents, as opposed to only those we have specific perceptions of, by considering them as part of a group or the generalized other
- **Specialization –** Specialization can now extend to groups – a member of the "carpenter" group may have the specialized skills of a carpenter, and only be remembered as such
- **Hardening of Opinions**- In a naïve implementation of social cognition, the hardening of opinion was based solely on the fact that agents may not receive

a certain bit when interacting for a long time. In contrast, our implementation allows for a much more robust and cognitively plausible notion of the hardening of opinions – if I label you as a member of group A, it will be difficult for my mind to change that you do not hold all of the attributes of group A – in this instantiation of the model, until you become one of my strong ties.
- **Bounded rationality**- Agents now have an even more limited perception of the knowledge of others, and hence bounded rationality is only increased.

As mentioned, we implement our new model in place of a previous naïve implementation of agent social cognition in Construct, an empirically validated social simulation tool. This allows us to focus solely on our implementation, and avoids a lengthy discussion on the full details of the tool or the model used. For full details on the tool itself, we refer the reader to a useful technical report (Lanham, Joseph, Morgan, & Carley, 2014).

4. Docking with Construct

In this work, we focus on the question of whether Construct-ML is able to replicate results of prior versions of Construct. To do this, we are using a docking analysis (Axtell, Axelrod, Epstein, & Cohen, 1996) to compare Construct-ML with prior versions of Construct. We are choosing to replicate experiments of a previous work that focused on group behavior, documented in Carley and Hill (2001), and referred to then as Construct-O (O is for organizations). The capabilities Construct-O added were later folded back into Construct. When making comparisons, we will refer to the Carley and Hill (2001) iteration as Construct-O, and continue to refer to our extension as Construct-ML.

4.1 Virtual Experiment Methodology

Carley and Hill (2001) introduced to Construct the idea of a second driver of human interaction, the desire for expertise. Earlier iterations of Construct focused on the homophily preference – where agents preferred to interact with people like themselves. Although the homophily drive is more powerful in many social situations, the addition of an expertise preference, a desire to interact with agents with rare knowledge, broadens the applicability of the simulation to include ones where work requires new knowledge to be gained from interaction.

They also wanted to examine the effect of group size on group performance, so their simulations always had two groups. The two groups may be equal or asymmetric sizes.

In short, the Virtual Experiment could be summarized as so:

Table 1. The Virtual Experiment Design

Parameter	**Values**	**# of Values**
Population Size (PopSize)	10, 20, 30	3
Expertise Drive Weight*	0.0, 0.25, 0.5, 0.75, 1.0	5
Group Size	Undifferentiated, Differentiated	2
Individual Memory Threshold**	0, -1, -2	3
Constants		
Knowledge Size	2 x PopSize	1
Knowledge Assignment	Random	1
Groups	2	1
Group Membership	One group	1
	Total Combinations Original Experiment	**30**
	Total Combinations This Experiment	**90**

*In this experiment, Expertise and Homophily preferences sum to 1.
** This parameter was added to allow variation in cognitive resources available to agents.

Population size (PopSize) is the number of actors in the simulation. Expertise Drive Weight indicates the relative weighting of homphily and the expertise drive in agents. Group Size is whether the groups are of equal or unequal sizes. In the Carley and Hill (2001) experiments, the amount of knowledge was always scaled to the population size; there were always two groups; each agent was a member of only one group; and knowledge was randomly assigned.

We have taken this basic experimental structure and also manipulated one parameter related to our new transactive memory implementation, individual memory threshold, how long individuals and groups remain in memory. The range of values provided (0, -1, and -2) suggest a wide range of cognitive resources available to agents.

In this experiment, and sympathetic to what was originally done in Carley and Hill (2001), we paid attention to the following outcome variables over the course of the simulation:

- **Knowledge Diffusion** - Of all available information, how much has been distributed to all agents. Mathematically, the sum of all binarized ties in the Agent x Knowledge matrix divided by the total cells in this matrix.
- **Task Performance** - Each group performs a binary classification task 50 times each turn based on a random sampling of 50 knowledge bits (note that bits may, and often will be, represented multiple times in the classification task). Each member votes and majority rules. Group accuracy is reported and averaged for overall performance.
- **Task Consensus** - Based on the same binary classification task, the number of members that agree with the group's decision for each task is recorded and averaged.
- **Triad Count** - Given node A,B, and C, a triad exists if the probability of interaction between A & B, B & C, and A & C, *in either direction*, is above the average probability of interaction calculated across all actors.

These values are calculated at each time-point. Carley and Hill (2001) focused on the length of time required for each outcome metric to reach 90% of the achieved maximum, as these values are much less sensitive to random noise than the time to reach Maximum.

Carley and Hill (2001)'s results suggested that in Construct-O:

- Small groups reach benchmarks faster
- That time to reach 90% of maximum tends to follow this pattern for outcomes, 1. Diffusion, 2. Performance, 3. Consensus, and 4. Triad development
- That increasing the weight of expertise in the agent's drives tends to reduce these times, except when expertise weighting is 100% (which vastly increases these times)

We will use these findings to inform a pattern-level analysis of Construct-ML, described in the next section.

4.2 Docking Results

In these results, we are investigating the relational, or pattern, validity of Construct-ML to Construct-O. Thus, although we report the actual metrics, the material at interest is the pattern match for each outcome based on Construct-O's findings.

We have three findings from Construct-O we wanted to investigate. We first explore the issue of group size, with our time to reach 90% for each of our four outcome variables. Table 2 includes pattern match values on group size and outcome patterns.

Table 2. Population and Outcome Metrics

		Threshold = 0		
Pop. Size	**10**	**20**	**30**	**Group Size Pattern Match**
Diffusion	94	153	210	3/3
Performance	40	31	27	0/3
Consensus	20	72	101	3/3
Triad	122	113	162	2/3
Outcome Pattern Match	3/6	3/6	3/6	**17/30**
		Threshold = -1		
Pop. Size	**10**	**20**	**30**	**Group Size Pattern Match**
Diffusion	80	160	236	3/3
Performance	34	32	28	0/3
Consensus	19	80	107	3/3
Triad	269	137	162	1/3
Outcome Pattern Match	3/6	3/6	3/6	**16/30**
		Threshold = -2		
Pop. Size	**10**	**20**	**30**	**Group Size Pattern Match**
Diffusion	76	170	256	3/3
Performance	33	39	32	**1**/3
Consensus	19	84	126	3/3
Triad	273	286	326	3/3
Outcome Pattern Match	3/6	**4**/6	**4**/6	**21/30**

We represent this table graphically in Figure 1, next page.

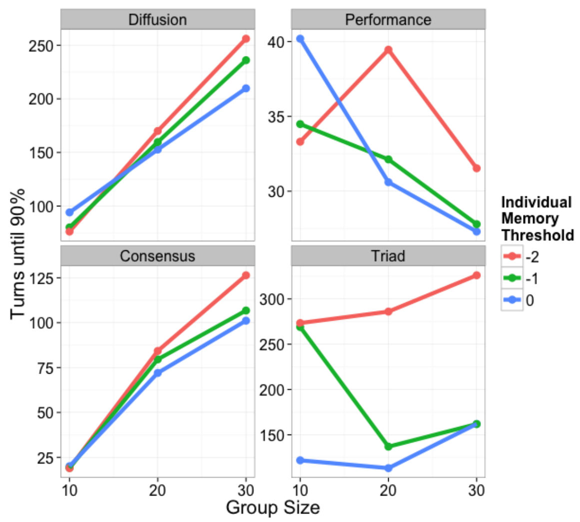

Figure 1. Average Turns to reach 90% of Maximum for each Outcome Metric. Lines colored by Threshold, Group Size along the X-Axis.

Note that Construct-ML agents have more cognitive bounds than Construct-O agents. Pattern validity to Construct-O, in general, improves as the agents' social space and cognitive resources increases.

Our simulation with the settings as given here matches Construct-O's outcome pattern in relationship to size for the outcomes of Diffusion and Consensus Formation, but not for Performance and Triadic closure.

For performance, this may be because of the implicit parameters built into the task performance evaluation (50 tasks, with a task of size 50) do not well match the original settings, which are not given in the original work. We will investigate the impact of task size on the performance outcome in future work.

The triad outcome is more interesting in that it does not appear to be obviously arbitrary in relation to settings given. We are still investigating the implications behind this triad stability pattern.

Because knowledge informs the stereotypes developed for each group, it's possible the random nature of knowledge assignment affects these simulations in ways that the prior implementation did not. If pattern validity improves with group-based knowledge assignment, then that will suggest that the model has, in one sense, improved, as random knowledge assignment is not very realistic!

The final outcome we wanted to compare against Construct-O was the impact of interaction drive.

Table 3. Expertise and Outcome Variables, averaged across threshold settings

Expertise	**0**	**0.25**	**0.5**	**0.75**	**1**
Diffusion	170	165	160	164	123
Performance	37	36	34	37	18
Consensus	77	75	75	79	30
Triad	231	224	199	210	153

Again, we do not see full pattern validity to Construct-O. Although we find in Construct-ML, as we did in Construct-O, that more expertise drive weight tends to decrease times to reach stability, we do not find, as Construct-O did, that agents who are only concerned with expertise perform less effectively, instead there is often a large drop in the amount of time required to get to the 90% benchmark. This may be due to an interaction of drive-weight and the transactive memory of groups that was not previously modeled.

Although these results suggest that Construct-ML will not predict similar group outcomes as Construct-O – it does not say that these new outcomes may not be, in practice, more realistic. Comparison to human small-group data is clearly indicated.

5. Conclusions and Future Work

In this paper we have discussed the need for a cognitively bounded implementation of transactive memory for agents. We have described the theory behind our approach, and have discussed the requirements of such a system. We then compared the findings of our model with an earlier iteration of Construct-O, and we were not able to replicate all of the patterns suggested by Construct-O's results. However, the pattern validity of Construct-ML improves as agents have more cognitive resources, which is suggestive and interesting.

Further work will involve systematic exploration of the robustness of these findings using other, more realistic, knowledge assignment procedures, and also changing the size of the binary classification task vector.

We are happy to discuss implementation details of the transactive memory system described in this work, but have refrained for reasons of space.

References

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. American Psychologist, 51(4), 355.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. Psychological review, 111(4), 1036.

Axtell, R., Axelrod, R., Epstein, J. M., & Cohen, M. D. (1996). Aligning simulation models: A case study and results. Computational & Mathematical Organization Theory, 1(2), 123-141.

Carley, K. M. (1990). Group Stability: A Socio-Cognitive Approach. In E. Lawler, B. Markovsky, C. Ridgeway, and H. Walker (Ed.), Advances in Group Processes: Theory and Research (Vol. Vol III, pp. 1-44). Greenwhich, CT: JAI Press.

Carley, K. M. (1991). A Theory of Group Stability. American Sociological Review, 56(3), 331-354.

Carley, K. M., Martin, M. K., & Hirshman, B. R. (2009). The etiology of social change. Topics in Cognitive Science, 1(4), 621-650.

Carley, K. M., & Ren, Y. (2001). Tradeoffs between performance and adaptability for C3I architectures. Paper presented at the Proceedings of the 6th International Command and Control Research and Technology Symposium.

Cyert, R. M., & March, J. G. (1963). A behavioral theory of the firm (2nd ed. Vol. 2). Cambridge, MA: Blackwell.

Duong, D. V., & Reilly, K. D. (1995). A system of IAC neural networks as the basis for self-organization in a sociological dynamical system simulation. Behavioral science, 40(4), 275-303.

Joseph, K., Morgan, G. P., Martin, M. K., & Carley, K. M. (2014). On the Coevolution of Stereotype, Culture, and Social Relationships: An Agent-Based Model. Social Science Computer Review, 32(3), 295-311. doi: 10.1177/0894439313511388

Kuethe, J. L. (1962). Social schemas. The Journal of Abnormal and Social Psychology, 64(1), 31.

Lanham, M., Joseph, K., Morgan, G. P., & Carley, K. M. (2014). Developing and building CASOS' Construct Simulation Development Environments. Pittsburgh, PA: Institute for Software Research.

Little, K. B. (1968). Cultural variations in social schemata. Journal of personality and social psychology, 10(1), 1.

March, J. G., & Simon, H. A. (1958). Organizations. New York, NY: Wiley.

Mead, G. H. (1922). A behavioristic account of the significant symbol. The Journal of Philosophy, 157-163.

Mead, G. H. (1925). The genesis of the self and social control. International journal of Ethics, 251-277.

Palazzolo, E. T., Serb, D. A., She, Y., Su, C., & Contractor, N. S. (2006). Coevolution of communication and knowledge networks in transactive memory systems: Using computational models for theoretical development. Communication Theory, 16(2), 223-250.

Rumelhart, D. E. (1978). Schemata: The building blocks of cognition: Center for Human Information Processing, University of California, San Diego.

Rumelhart, D. E. (1980). Schemata: The building blocks of cognition. In R. Spiro, B. Bruce, & W. Brewer (Eds.), Theoretical Issues in Reading Comprehension: Hillsdale, NJ: Erlbaum.

Simon, H. A. (1957). Models of man: social and rational; mathematical essays on rational human behavior in society setting. New York, NY: Wiley.

Wegner, D. M. (1995). A computer network model of human transactive memory. Social cognition, 13(3), 319-339.

Author Biographies

GEOFFREY P MORGAN is a PhD student in Carnegie Mellon's Computation, Organization, and Society program. Morgan has experience in developing autonomous robotic systems, self-learning systems, cognitive models, and user-assistive systems. He is interested in organizational impacts on individual decision-making and on optimizing organizational performance.

KENNETH JOSEPH is a 5th year Ph. D. student in the Computation, Organization and Society program in the School of Computer Science at Carnegie Mellon University. He is interested in a better understanding of the dynamics and cognitive representations of prejudice and its interrelationships with sociocultural structure. In his work, he utilizes both machine learning and agent-based modeling approaches.

KATHLEEN M CARLEY is a Professor and the director of the Center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University. Her research combines cognitive science, social networks and computer science to address complex social and organizational problems.