Computational Organizational Theory

Geoffrey P Morgan and Kathleen M Carley, Carnegie Mellon University, Pittsburgh, PA, USA

Published by Elsevier Ltd.

Abstract

Research in computational organizational theory explores the complex interactions between organizations and their members. Organizations are legally autonomous entities that structure the efforts of individuals to achieve large goals. Interesting behavior often emerges from the interactions and demands of modeled primitives. This introduction describes the common ground and recent advances in multimodeling, multilevel modeling, and rapid model development. We conclude this summary with discussion of issues of model fidelity and model validation.

Introduction

Computational organization theory (COT) is a research discipline, which develops formal computational models to understand, extend, and examine organizational theory. Specific questions of interest to researchers in this field include organizational decision making, adaptation, and organizational learning. Often, these researchers are interested in the complex interdependencies between individuals and the organizations of which they are members. Organizations need individuals to make decisions, yet individuals benefit from membership in an organization: the organization is not merely the sum of its members.

Models developed by COT researchers tend to embody theory, represented as mathematical and computational structures and processes, of how an organization may behave. These models are intended to reproduce behavior analogous to the behavior of real-world organizations, but are implemented as computer programs. These models can be created and tested through low-cost simulation methodologies – although validation usually requires some form of comparison with either empirical data or with another model, which has been compared to empirical data (e.g., see Schreiber and Carley, 2013).

Organizational models have been implemented in many diverse ways – most often in the way that best suited the theoretical premises of the modeler and with which the modeler was fluent. These methods vary from formal mathematical models to boundedly rational agent models. Organizational models tend to focus on either processes (where the people are implicit) or people (where the processes are often implicit). This article focuses on organizational models that place the individual as an organizational decision maker, a people-oriented perspective (although subgroups may also be represented in such models). Thus, these models are multi-agent simulations.

There are other methods for modeling organizations. For a thorough introduction to the art of modeling organizations as collections of intertwined processes, see Carlsen (1997) or Jørgensen (2004). In contrast, some researchers are primarily interested in organizational ecologies (Hannan and Freeman, 1977) – where the primary unit of analysis are organizations, rather than people within those organizations. Recent work in multilevel modeling (discussed below) is attempting to unite these various perspectives.

This article will frequently draw examples and grounding by presenting ideas from Construct (Carley, 1991a; Carley et al., 2009). Construct is a turn-based network-centric simulation of information and belief diffusion, based on the theory of Constructuralism (Carley, 1991b), and inheriting the information processing perspective of the Carnegie School (Simon, 1957; Cyert and March 1963).

Computational models, instantiated as simulations, are valuable to researchers in multiple ways. Computational methods allow theoreticians to explore the sources of complex and interesting organizational behavior, often finding that relatively simple rules when localized and in aggregate can produce these phenomena. This idea, that the interactions of simple processes produce complex and interesting behavior, is often called *emergence*. These methods also allow theoreticians to explore how large classes of organizations may behave in a particular environment of interest compared to other organizations from the same classification scheme. In general, the translation of theory into computational model tends to require that theory to be expressed with more clarity and precision than that required by textual description, which improves the resulting descriptions of the theory.

For applied research, computational models allow researchers to explore counterfactual scenarios for specific organizations of interest. These counterfactual scenarios allow researchers to answer 'what-if' questions, which it would be impossible to explore practically or ethically in the real world. Model results, of course, cannot be assumed to provide a specific and valid prediction of outcomes, but can often (if the theoretical model has been validated) be useful to examine from a trend analysis perspective.

COT makes these important assumptions about organizations and phenomena:

- Phenomena are modelable. If we cannot model it, we cannot examine it.
- Uncertainty is important. Individuals make mistakes. Environmental conditions can rapidly shift.
- Humans are boundedly rational. Because of cognitive and physical limitations, humans do not know everything about the world they live in, what they know may be wrong, and they may not be able to access what they know at all times.
- Organizations are organized to do work. Most metrics focus, explicitly or implicitly, on task performance.

- Outcomes are path dependent. Choices made in the past affect the present and future of an organization.
- Organizations may be designed and manipulated. Humans can create an initial plan for an organization, and these plans can be adjusted during an organization's life.

We will begin by defining the concept of an organization. From there, we will discuss the common elements of multi-agent simulation, and the representational choices that inform these models. After we describe important elements of models, we will discuss methods used to both inform and test models. Finally, we will explore common issues that face all modelers and model consumers.

What is an Organization?

The definition of an organization, as opposed to other collections of people, can be difficult to clearly enumerate. We find it useful to characterize organizations as entities that have:

- Multiactor membership: actors may or may not be human.
- Task-orientation: organizations are structured to perform work.
- Goal-driven behavior: tasks support goals. Goals may change over time.
- Agency: organizations can alter themselves and their environment, and be affected by external changes.
- Independent knowledge: organizational knowledge is not directly tied to the knowledge of individuals.
- Legal autonomy: individuals are not directly responsible for the actions of the organization.

As such, organizations are multilevel entities such that the structure of the organization can be represented as a series of interconnected networks (Carley, 2005). One of these networks, and perhaps the most studied network, is the human-to-human network, which typically operates at a formal or authority level and at an informal or friendship, advice and collaboration level. Another network that is particularly important from a cyber perspective is the information technology (IT)-to-IT network, which is comprised of all such technology, associated databases, and the electronic modes of communication among these. Connecting the humans and the IT is another network often realized over multiple types of media. Then there are the knowledge network that links the agents to the data available and the activity network which links the agents to their tasks. And there is the traditional task network, familiar to operations research, showing what needs to be done before or in conjunction with what. A coherent collection of networks that describe a single organization is referred to as a meta-network (Carley, 2002). The analysis of these meta-networks for assessing organizations is a growing field, and there are analytical tools, such as ORA (Carley, 2014), optimized for this type of analysis. Using such tools organizational units can be designed and their performance assessed.

Organizations exist, from a functional perspective, to organize individuals and resources to achieve goals not within the capabilities of any single individual. As such, collections of individuals must be brought together. Individuals cooperate and perform tasks in service of the organization's goals. The goals of an organization are distinct from the goals of any of its members, although they may be well aligned. Goals are set by organizational processes, usually involving the input of multiple decision makers. In the pursuit of their goals, organizations change themselves and the environment around them. These changes may or may not be in their favor. Over time, as the organization achieves goals, unique organizational knowledge is formed. This organizational knowledge is not a description of the current knowledge of its individuals, but is instead distilled from individual contributions over time and is often stored in an organization's standard operating procedures. In order to organize individuals to do work toward large goals, particularly those that may have consequences for others, members of organizations must feel protected from the legal consequences of the organization's actions – the organization must have legal autonomy.

As stated earlier, organizations exist to organize the efforts of individuals toward large goals. How do organizations attempt this? They do this by structuring the responsibilities, interactions, and incentives of individuals to maximize the alignment of an individual's capabilities and goals with the organization's needs. This is sometimes referred to as the 'design' of an organization. Because there are complex interactions between the environment in which an organization operates, the capabilities of its members, and the goals of an organization, there is no one optimal design. One goal of COT's research is in understanding the interactions between all of these factors and an organization's design.

Now that we share a common understanding of an organization, we now discuss multiagent simulations and the representational choices required of developers of these simulations.

Defining a Multiagent Simulation

To define a working organizational computational model, the modeler/theoretician must establish representations of the environment.

The environment can be roughly characterized based on four questions.

- Is time continuous or discrete?
- Is space physical or virtual?
- What is in the environment?
- How could the environment change over time?

The following subsections will attempt to describe common modeling choices to answer each of these questions.

Representations of Time

Time, in simulation, can be either continuous (like the real world) or discrete, where it can be separated into equal-sized pieces. We call these pieces 'turns.' In discrete-time (or turn-based) simulation, everything that can act may have an opportunity to do so once per turn. In continuous-time simulations, agents may take new actions as computing resources allow. Some effort is usually made (in competitive environments in particular, see below) to ensure that agents have roughly equivalent computational resources, but not necessarily. Games like risk, chess, and monopoly can all be viewed as turn-based simulation.

Turn-based simulation may seem like a gross simplification of the real world, rendering such simulations suspect, but in practice it is a relatively harmless simplification that makes problems of agent concurrency and computational resource access trivial. Turn-based simulations are appropriate when (1) actors should be expected to have similar opportunities to participate, and (2) when the phenomenon of interest occurs over long periods of time: hours, days, or months, as opposed to seconds or minutes.

Representations of Space

Space in simulations may be physical or virtual. If space is physical, then agents are placed at points on a map. Physical space may, like time, be continuous or discrete. Discrete space involves dividing up the area of the map into equal-sized chunks, often called tiles or blocks. Cellular automata exist in such tile-worlds, and Schelling's Segregation Model (Schelling, 1971) also used such a world, see **Figure 1**. Tile-worlds may or may not be connected at their opposite edges – if both edges are connected, a tile-world forms a ring torus, which can remove problematic edge effects.

Physical space in simulations is useful if the theory requires it. A simulation examining, for example, how people interact as they move about an office building will require some idea of physical space and of entities moving through it and interacting with other entities. Typically, simulations with physical space use it to constrain agent perception and interaction.

Often in organizational models, space is virtual – entities are not placed in physical space. Instead, connections are drawn between entities that indicate who may affect and/or interact with who. These connections can be represented as networks, matrices of who is connected to whom, where '0' indicates no connection and '1' indicates a connection, as shown in **Figure 2**. Modern simulations often use multiple network semantics to enrich the virtual environment.

Virtual environments are useful when physical adjacency is not a primary driver of interaction in the simulation. For example, interactions between people doing knowledge work may be more driven by work responsibilities rather than physical adjacency. Virtual environments, like physical ones, are used to constrain agent perception and interaction possibilities.

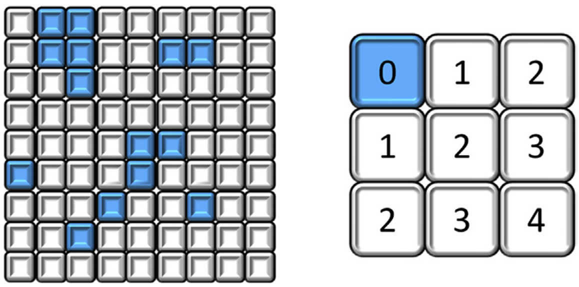

Figure 1 A tile-world, with agents in blue. On the right, Manhattan distance is shown relative to an agent on the top-left.

	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	0	0
3	0	1	0	1	1
4	0	0	1	0	1
5	0	0	0	1	0

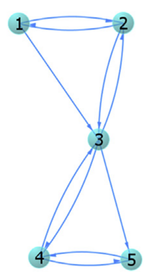

Figure 2 Networks are a convenient and useful representation of virtual space environments.

Elements in the Environment

What exists within the simulation environment? This is a fundamental choice that should be driven by theory. Because organizations are goal-driven entities, organizational learning models usually include some representation of agents, knowledge, and tasks. Although an organization, as a whole, may be said to have some particular piece of information, that information may not be where it can be usefully applied (as defined by tasks). Thus, one common method of evaluating simulated organizational performance is through examining the agreement between three networks (although they may not be represented, explicitly, as networks by the modelers). These three networks are the knowledge network (who knows what), the assignment network (who should do what), and the requirements network (what knowledge is needed for what), as shown in **Figure 3**.

Not all organizational models include explicit representations of these three objects, but their presence is often implied in the theoretical descriptions of the model.

Agents

Usually, agents are representations of humans, but some simulations allow the set of entities to include nonhuman objects, such as IT systems, and may also include entities at different levels of granularity, an agent representing a single individual may interact with an agent representing a team of individuals.

Entities that can affect their own state or the state of others have agency. We refer to these entities as agents. Entities without agency will usually be referred to as objects. Agents may act on objects, but objects do not act directly on

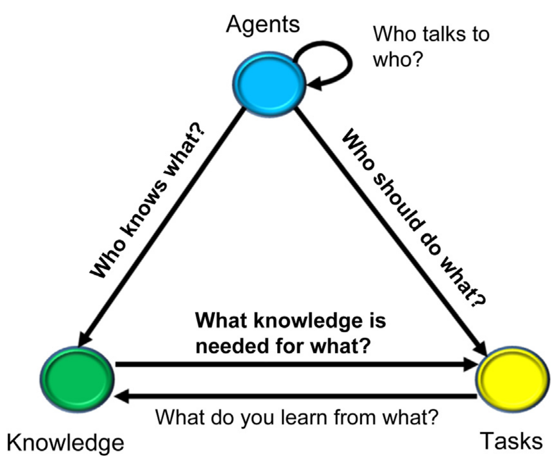

Figure 3 Three common objects within an organizational model are agents, knowledge, and tasks.

agents, although their presence may alter the actions chosen by an agent.

Agents have bounded rationality, limited access to the state of the larger world in which they inhabit. There are various limitations that can be prescribed to these agents, usually based on the larger theory of interest and drawn from human limitations. Agents may be only able to access state for things they can directly observe. They may forget knowledge they acquired earlier. They may have a limited number of refinement cycles before a decision must be made. Their perception of state may be error prone. Agents should have one or more of these limitations to be useful in a multiagent simulation.

Following directly from bounded rationality, agents also have limits to the possibilities available to them in their social interactions (Simon, 1991). Formal and informal mechanisms frequently prevent individuals from interacting with each other. Although it may be theoretically possible for the mailroom assistant to interact with the CEO of a Fortune 500 company, such interactions are unlikely to be routine and often may be safely discounted. Organizations often seek to impose structure to effectively distribute coercive, reward, and legitimate power (French and Raven, 1959). Individuals, in order to protect their own social capital, often seek to prune social ties for their own benefit (Burt, 1992, 1997). Thus, useful models of organizations will frequently limit interaction possibilities for their agents.

Agents must have actions available to them to be classified as agents. Again proceeding from theory, a modeler must define the actions available to agents and the decision heuristics used to select actions. Actions may be destructive (attacking another agent) or nondestructive (sharing knowledge with another agent), but all actions must have consequences that may alter future behavior of this or other agents.

Some models have entities that represent individuals but do not allow these representations to take independent action. Instead, a single model controller, which represents the organization, acts upon these agents. March's Mutual Learning Model (1991) is an influential example of such a model.

In Construct, agents may choose an interaction partner from all-available alters. They share knowledge with these partners, which may alter the partner's perception of the agent. The choice of who to interact with is weighted by two dynamic drives: (1) the desire to interact with alters similar to the agent, and (2) the desire to interact with alters who appear to have rare knowledge. Because of these two drives, the act of sharing knowledge will change the probabilities for both agent and alter of who they will interact with in the future.

Knowledge

Knowledge is information useful to the agents, either informing their choices of actions or helping them to complete tasks. Knowledge may be represented in a variety of ways; some methods include access to resources, rules that constrain an agent's actions, elements of the state of the environment that the agent can perceive, and as bits in a bit string. Organizational knowledge and individual knowledge may have different representations in the same simulation. In simulations where knowledge is represented as an explicit object, it is usually assumed to have some evidentiary basis – it is not merely an opinion. The transfer of knowledge, unlike other material resources, does not deplete the donor's store. Agents may not know, however, what they know.

In Construct, agents have two separate knowledge representations. The first is the model's ground state, what agents knows what, represented as multimode Agent x Knowledge matrix. Construct agents also have a perceptual matrix of what they think other people know, each agent thus has their own Agent x Knowledge matrix.

Tasks

Tasks are activities that are expected to have some useful larger purpose for the organization. They are not identical to goals, but may be instrumental in completing goals. Alternatively, they may be present in simulations where goals are never modeled or discussed.

Task performance may be dictated by agent actions, or may be inferred from agent characteristics. Task success is usually not guaranteed: it may be informed by agent knowledge, by exogenous probabilistic interference, or by third-party constraints.

In Construct, task performance is determined by access to knowledge relevant to the task. Tasks are assigned to agents at simulation run-time, and their performance is assessed each turn. Because tasks in Construct are assumed to be knowledge intensive, they are modeled as binary information tasks, if the agent has a majority of the required information for the task, the agent succeeds. If not, they fail at the task.

Current Research

In this section, we introduce some of the new trends in COT research. We discuss three trends: multimodeling, multilevel modeling, and rapid model development.

Multimodeling and Model Interaction

Multimodeling is an effort to leverage previous modeling work by incorporating one or more models into a single unified process. There are four ways models can be linked together on the basis of how they are connected by inputs and outputs, these are (1) integration, (2) docking, (3) interoperability, and (4) collaboration (Carley et al., 2012) as shown in **Figure 4**.

Integration, which involves refactoring models so that they work within a single common system envelope, is the most obvious but also most expensive method of integrating models. Representations of the phenomena must be aligned for all models, and this process often involves significant investment in new development work.

Docking (Axtell et al., 1996) is an approach to model validation where two models of the same phenomenon are directly compared on their features and their results. Docking two models require those models to have a common input and a common output. Model results are compared on three increasing levels of equivalence: relational, statistical, and numerical. Beginning with numerical equivalence, this is where one model produces the exact same numbers as the comparison partner, the two models can be, essentially, swapped interchangeably across tested values. Statistical equivalence is a characteristic of two models that they produce

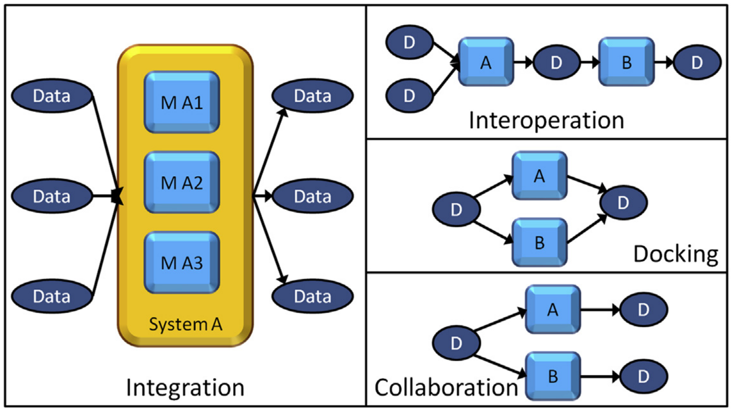

Figure 4 Multimodeling methods 'D' stands for data. A line is drawn from data to a model if it is input to that model, and a line is drawn from model to data if the data is an output of that model.

functionally identical results across tested values. Relational equivalence is where two models do not produce identical values, but instead produce similar patterns of output value across tested input values. Relational equivalence is usually considered sufficient to say that two models are docked to each other, because model inputs, although notionally similar, are rarely perfectly matched in the granularity and responsiveness of the input data.

Collaboration, like docking, involves two or more models that take in a common input, but do not produce similar outputs. Instead, both models take the input data, along with other data, and use that to predict different new data. Models can be easily made collaborative by matching the representations of their input data.

Interoperability is the idea that the outputs of one model can serve as inputs to another model. This is a useful and often inexpensive technique, particularly when building one model to extend and leverage the power of an older, more established model, because the data representations both required by and outputted by the older model are well understood. It is one approach that can lead toward Integration (where the models are in the same system envelope), but can be useful as part of a looser confederation of systems.

Multilevel Modeling

Multilevel modeling is an effort to account for both group and individual level effects concurrently within a model. Issues in multilevel representation have been central to organization science research (Rousseau, 1985). Statistical methods have found some success in grouping observations according to larger real-world structures (Snijders and Bosker, 2012). These methods help account for observations that, due to the inherent grouping of subjects within the frame of the research question, were not completely independent, such as students within a common classroom.

Within the modeling conducted in COT, multilevel modeling is used to indicate that there are multiple levels of dynamic description within a single model. For example, the model may have 42 individuals, 5 teams, and 2 multiteam organizations, but the number or characteristics of higher-order descriptions (the teams and organizations) should change over the length of the simulation. These descriptive characteristics are important to the model's mechanisms.

There may be multiple levels of agency within a single model. The individual may take actions within their purview, and the groups, teams, and companies of which they are members may also take actions separately as autonomous entities. The actions of individuals may influence the actions of the team, but they are not the same action. The actions allowed to each level of actor are usually distinct, although there may be parallels.

Adversarial models, where individuals are members of teams hostile to each other, could be described as a multilevel model, but only if the condition of the larger group unit has an effect on the actions of the individual, such as in Morgan et al., 2010. Adversarial models, where the state of the group has no direct impact on the individual's state should not be characterized as multilevel models.

In Construct, individuals use their knowledge of a not well-known actor's group affiliations to interpret what that actor is likely to know. As they interact with these actors, their perception of these groups will change, which will affect their likelihood of interacting with other not well-known alters. In this way, Construct is a multilevel model.

Rapid Model Development

Models take time to build. As models move from highly theoretical and intellective exercises to an interest in modeling specific scenarios of interest, the timeline for development shrinks even as the modeling requirements increase. Models can be made completely irrelevant by events if their results are predicated on assumptions no longer valid. For example, the organization has restructured, or a key leader has left the company. Thus, this research work is driven heavily by the needs of applied model users.

Rapid model development takes lesson from object-oriented and data-driven development paradigms. Rather than explicitly enumerating all state objects within a given system, the model software defines interface standards and file formats that can be used to provide this state. This decouples the model, which specifies how agents will interact with their environment, from the model's initial state. Modelers must still define behavior for both agents and the environment.

This decoupling provides practical advantages when attempting to apply a theoretical model to a given applied context. Data used to inform the model can either be generated from theoretically useful test conditions, or created by translation of real-world data into a format digestible by the model. This translation process, although nontrivial, provides an opportunity for channeling the enormous data created by social media into models.

On a theoretical basis, it also provides an important distinction between the model as an environment and the initial state of the model. The validation of a model without a data-driven element requires evaluation not only of the model's behavior but also of the model's initial state. Separating these elements allows for concerns about the model to be more easily addressed.

For example, Construct models have been informed by the analysis of news articles (Pfeffer and Carley, 2012), by field studies of teams (Schreiber and Carley, 2007), by doctrinal texts on organizations (Lanham et al., 2011a), from combinations of doctrinal and text-analysis work (Lanham et al., 2014), and from subject matter expert estimates (Lanham et al., 2011b).

Model Fidelity and Model Validation: Common Concerns

Any discussion of COT is incomplete without discussing the highly related issues of model fidelity and model validation. The decisions modelers take in choosing to represent a phenomena are not guided, intrinsically, by deductive processes. Rather, the process is a creative one. Modelers may make poor decisions. Understanding how to assess models is important for consuming model results appropriately.

Model Fidelity

Model fidelity focuses on the issues of actor and state representation. Models exist on a fidelity axis, ranging from highly theoretical intellective models, which simplify and abstract the environment of phenomena to their essentials, to highly emulative models, which attempt to reproduce as realistic an environment as possible, as shown in **Figure 5**. All models, of course, must abstract at some point. A perfect reproduction of the modeled object is not a model, it is the thing itself.

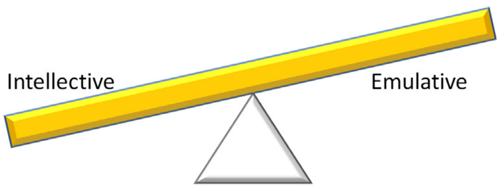

Figure 5 Balancing the fidelity needs of the simulation is an important modeling effort. Most computational organizational theory models are intellective.

High model fidelity (emulative simulation) is particularly important to simulations for use in training (Hays and Singer, 1989), although even intellective simulations can be useful for training in the business context (Van Ackere et al., 1993). The vast majority of COT models are intellective in nature.

Data-oriented design (as discussed in the rapid model development section above) blurs the line between emulative and intellective models. Models can be built with relatively simple behaviors available to both agents and the environment, but be instantiated with very complex and detailed initial states.

Questions of model fidelity that concern both a model developer and a model consumer are:

- Does this model include the factors I think are important to exploring this phenomena?
- Does this model include factors that I do not think are important to exploring this phenomena? Is it unnecessarily complex?
- Does this model ignore important interaction possibilities between agents and the environment?

Model Validation

Where model fidelity focuses on the questions of representation, model validation asks the question 'is this a good model?' By this, we mean 'does the model, given an initial state, produce reasonable behavior?' Of course, the answer is almost always 'it depends, how will the model's results be used?' Validation is thus tied, even more intrinsically than model fidelity, to the intended purposes of the model. Models expected to guide decision makers should be subjected to much more rigorous validation that models intended merely for discussion.

In the organizational theory domain, standard experimental assumptions do not hold. When making interventions on a real-world organization, it is rare for the experiment to be easily repeatable or for the results to be easily replicated. Experimental controls can be difficult to clarify, and gathering all necessary data is often infeasible or unethical. The difficulty of traditional experimentation makes a strong case for COT, but we should approach testing the resulting computational models with a similar level of rigor, evaluating them with 'virtual experiments.'

Virtual experiment methodology also makes some assumptions. It assumes the model's code is accurate, that the model has sufficient fidelity to the phenomena, and that standard statistical tests can be used to compare outcomes. Thus, questions of a model's fidelity should be addressed before a model's validation process even begins. Hopefully the modeler has considered the model's fidelity multiple times over the model's development period.

Validation, it should be noted, is hard. Model validation can require as much or more effort than the model's development. Human evaluators usually have some bias toward their products, and neutralizing this bias is difficult. Validation is knowledge intensive, requiring a deep understanding of the empirical phenomena being modeled (Louie and Carley, 2008). Validating agent-based COT simulations is also difficult because agent behavior is not predetermined and path dependent – many simulation runs may be necessary to understand the full scope of possible system behaviors. Finally, the empirical data for validation may simply not be available at an appropriate granularity.

Validation can occur at multiple levels (Maxwell and Carley, 2009). The type of validity necessary for a model depends on its purpose (Burton and Obel, 1995). These levels are:

- **Internal validity**: the model's code is error free.
- **Parameter validity**: the model's parameters allow expression of all critical variables of the phenomenon.
- **Process validity**: the behavior of agents and the environment fit with current understanding of the phenomenon.
- **Face validity**: the results of the simulation, given the initial state, match the opinions of experts.
- **Pattern validity**: the outputs of the models show similar valence and magnitude of change across tested parameters to empirical data. It matches the pattern of the available empirical data. If abstracted, this is sometimes known as 'stylized fact validation.'
- **Value validity:** the outputs of the models show very similar values to those found in empirical data.
- **Theoretical validity:** the model correctly expresses a theory that explains the phenomena of interest.

Docking (Axtell et al., 1996; Burton and Obel, 1995) is also a model validation technique. A model that is docked to another model and has established relational equivalence (as discussed above), can make some claim to the pattern validity of the original model. Value equivalence, if achieved, would allow the docked model to inherit the value validity of the original model.

In multimodeling environments, validation of all the models at once can be difficult or impossible. Instead, inputs and outputs of these models are validated against real-world data individually, a process called *validation-in-parts*. This idea inherits from modern manufacturing process validation, which focuses on the outputs of the manufacturing effort, not on the mechanics of that effort.

Summary

In this article, we have introduced the area of COT. Research in COT explores the complex interactions between organizations and their members. Interesting behavior often emerges from the interactions and demands of modeled primitives, whether people or processes, although this introduction has focused on people-oriented simulation.

We have defined organizations as legally autonomous entities that attempt to achieve goals not within the capacity of any individual member. An organization's pursuit of its goals will change both the organization and the environment in which it operates, affecting other organizations (and themselves) either beneficially or adversely. Members of an organization do tasks, and both require and generate knowledge in the performance of these tasks.

To provide a useful introduction to COT research, we explored the requirements of defining and instantiating a multiagent simulation, using Construct (Carley, 1991a) as an illustrative example throughout. We discuss time, space, and elements within an environment, focusing on the importance of identifying agents and what actions those agents can take.

After identifying important elements to any single model, we discussed some important current trends in COT research. Multimodeling is an effort to leverage multiple models to provide a more complete picture of complex phenomena. Multilevel modeling allows actors at multiple levels of granularity to have important effects on actors at other levels. Rapid model development leverages data-driven approaches to software development: divorcing model behavior from a model's initial state.

Finally, we discuss important points every COT model consumer and developer should be aware of, issues of model fidelity and model validation. Model fidelity explores the issue of whether a model's representation of phenomena is appropriate, too simple, or unnecessarily complex for the phenomena of interest. Model validation asks whether the model is sufficient to the purpose for which it will be applied.

COT is a valuable and interesting area of research with important theoretical and applied areas. We hope this introduction has been useful.

See also: Behavioral Theories of Organization; Computational Approaches to Model Evaluation; Decision and Choice: Bounded Rationality; Dynamic Decision Making; Ecology: Organizations; Emergent Properties; Group Processes in Organizations; Interorganizational Relationships and Networks; Knowledge Representation; Network Analysis, History of; Organizational Control; Social Simulation: Computational Models; Stochastic Models.

Bibliography

Axtell, R., Axelrod, R., Epstein, J.M., Cohen, M.D., 1996. Aligning simulation models: a case study and results. Computational and Mathematical Organizational Theory 1, 123–141.

Burt, R.S., 1992. Structural Holes. Harvard University Press, Boston, MA.

Burt, R.S., 1997. The contingent value of social capital. Administrative Science Quarterly 42, 339–365.

Burton, R.M., Obel, B., 1995. The validity of computational models in organization science: from model realism to purpose of the model. Computational and Mathematical Organization Theory 1 (1), 57–71.

Carley, K.M., 1991a. Designing organizational structures to cope with communication breakdowns: a simulation model. Industrial Crisis Quarterly 5, 19–57.

Carley, K.M., 1991b. A theory of group stability. American Sociological Review 56 (3), 331–354.

Carley, K.M., 2002. Smart agents and organizations of the future. In: Lievrouw, L., Livingstone, S. (Eds.), The Handbook of New Media. Sage, Thousand Oaks, CA, pp. 206–220 (Chapter 12).

Carley, K.M., 2005. Organizational design and assessment in cyberspace. In: Rouse, W.B., Boff, K.R. (Eds.), Organizational Simulation. John Wiley Sons Inc., Hoboken, NJ.

Carley, K.M., 2014. ORA: a toolkit for dynamic network analysis and visualization. In: Alhajj, R., Rokne, J. (Eds.), Encyclopedia of Social Network Analysis and Mining, Springer.

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

Carley, K.M., Morgan, G.P., Lanham, M.J., Pfeffer, J., 2012. Multi-modeling and socio-cultural complexity: Reuse and validation. In: Salvendy, G., Karwowski, W. (Eds.), Advances in Design for Cross-cultural Activities, vol. 2. CRC Press, Boca Raton, FL, pp. 128–137.

Carlsen, S., 1997. Conceptual modeling and composition of flexible workflow models (Doctoral Dissertation). www.idi.ntnu.no/grupper/su/publ/phd/carlsen-thesis.pdf.
Cyert, R., March, J.G., 1963. A Behavioral Theory of the Firm, second ed. Blackwell, Cambridge, MA.
French, J.R.P., Raven, B.H., 1959. The bases of social power. In: Cartwright, D. (Ed.), Studies in Social Power. Institute for Social Research, Ann Arbor, MI, pp. 150–167.
Hannan, M.T., Freeman, J., 1977. The population ecology of organizations. American Journal of Sociology 82 (5), 929–964.
Hays, R.T., Singer, M.J., 1989. Simulation Fidelity in Training System Design: Bridging the Gap between Reality and Training. Springer Verlag, New York.
Jørgensen, H.D., 2004. Interactive process models (Doctoral Dissertation). http://ntnu.diva-portal.org/smash/get/diva2:125137/FULLTEXT01.
Lanham, M.J., Morgan, G.P., Carley, K.M., 2011a. Data-Driven Diffusion Modeling to Examine Deterrence. IEEE Network Science Workshop, West Point, NY.
Lanham, M.J., Morgan, G.P., Carley, K.M., 2011b. Social network modeling and simulation of integrated resilient command and control in contested cyber environments. In: Levis, A., Carley, K.M., Karsai, G. (Eds.), Resilient Architectures for Integrated Command and Control in a Contested Cyber Environment. The Volgenau School of Engineering Department of Electrical and Computer Engineering System Architectures Laboratory, Technical Report SAL/FR-11-02, George Mason University, Fairfax, VA, pp. 63–94.
Lanham, M.J., Morgan, G.P., Carley, K.M., 2014. Social network modeling and agent-based simulation in support of Crisis De-escalation. IEEE Transactions on Systems, Man, and Cybernetics: Systems 44 (1), 103–110.
Louie, M.A., Carley, K.M., 2008. Balancing the criticisms: validating multi-agent models of social systems. In a Special Issue on "Simulating Organizational Processes" in Simulation Modeling Practice and Theory 16 (2), 242–256.
March, J.G., 1991. Exploration and exploitation in organizational learning. Organizational Science 2 (1), 71–87.
Maxwell, D., Carley, K.M., 2009. Principles for effectively representing heterogeneous populations in multi-agent simulations. In: Tolk, A. (Ed.), Complex Systems in Knowledge Based Environments. Springer-Verlag, pp. 199–228 (Chapter 8).
Morgan, J.H., Morgan, G.P., Ritter, F.E., 2010. A preliminary model of participation for small groups. Computational and Mathematical Organization Science 16, 246–270.
Pfeffer, J., Carley, K.M., 2012. Rapid modeling and analyzing networks extracted from pre-structured news articles. Computational and Mathematical Organizational Theory 18 (3), 280–299.
Rousseau, D.M., 1985. Issues of level in organizational research: multi-level and cross-level perspectives. Research in Organizational Behavior 7 (1), 1–37.
Schelling, T.C., 1971. Dynamic models of segregation. Journal of Mathematical Sociology 1 (2), 143–186.
Schreiber, C., Carley, K.M., 2007. Agent interactions in construct: an empirical validation using calibrated grounding. In: Proceedings of 2007 BRIMS, Norfolk, VA, 26–29 March, pp. 1–6.
Schreiber, C., Carley, K.M., 2013. Validating agent interactions in construct against empirical communication networks using the calibrated grounding technique. IEEE Transactions on Systems, Man and Cybernetics – Part A 43 (1), 208–214.
Simon, H.A., 1957. Models of Man: Social and Rational – Mathematical Essays on Rational Human Behavior in a Social Setting. Wiley, New York.
Simon, H.A., 1991. Bounded rationality and organizational learning. Organizational Science 2 (1), 125–134.
Snijders, T.A.B., Bosker, R., 2012. Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling, second ed. SAGE, Thousand Oaks, CA.
Van Ackere, A., Larsen, E.R., Morecroft, J.D.W., 1993. Systems thinking and business process redesign: an application to the beer game. European Management Journal 11 (4), 412–423.