

Situational Grounding within Multimodal Simulations

James Pustejovsky and Nikhil Krishnaswamy

Department of Computer Science

Brandeis University

{jamesp, nkrishna}@brandeis.edu

Abstract

In this paper, we argue that simulation platforms enable a novel type of embodied spatial reasoning, one facilitated by a formal model of object and event semantics that renders the continuous quantitative search space of an open-world, real-time environment tractable. We provide examples for how a semantically-informed AI system can exploit the precise, numerical information provided by a game engine to perform qualitative reasoning about objects and events, facilitate learning novel concepts from data, and communicate with a human to improve its models and demonstrate its understanding. We argue that simulation environments, and game engines in particular, bring together many different notions of “simulation” and many different technologies to provide a highly-effective platform for developing both AI systems and tools to experiment in both machine and human intelligence.

Introduction

The concept of simulation has played an important role in both AI and cognitive science for over 40 years. There are two distinct uses for the term *simulation*, particularly as used in computer science and AI. First, simulation can be used as a description for *testing a computational model*. That is, variables in a model are set and the model is run, such that the consequences of all possible computable configurations become known. Examples of such simulations include models of climate change, the tensile strength of materials, models of biological pathways, and so on. We refer to this as *computational simulation modeling*, where the goal is to arrive at the best model by using simulation techniques.

Simulation can also refer to an environment which allows a user to interact with objects in a “virtual or simulated world”, where the agent is embodied as a dynamic point-of-view or avatar in a proxy situation. Such simulations are used for training humans in scripted scenarios, such as flight simulators, battle training, and of course, in video gaming: in these contexts, the software and gaming world assume an embodiment of the agent in the environment, either as a first-person restricted POV (such as a first-person shooter or RPG), or an omniscient movable embodied perspective (e.g., real-time or turn-based strategy). We refer to such ap-

proaches as *situated embodied simulations*. The goal is to simulate an agent within a situation.

Simulation has yet another meaning, however. Starting with Craik (1943), we encounter the notion that agents carry a mental model of external reality in their heads. Johnson-Laird (1987) develops his own theory of a mental model, which represents a situational possibility, capturing what is common to all the different ways in which the situation may occur (Johnson-Laird and Byrne 2002). This is used to drive inference and reasoning, both factual and counterfactual. Simulation Theory, as developed in philosophy of mind, has focused on the role that “mind reading” plays in modeling the mental representations of other agents and the content of their communicative acts (Gordon 1986; Goldman 1989; Heal 1996; Goldman 2006). Simulation semantics (as adopted within cognitive linguistics and practiced by Feldman (2010), Narayanan (2010), Bergen (2012), and Evans (2013)) argues that language comprehension is accomplished by means of such mind reading operations. Similarly, within psychology, there is an established body of work arguing for “mental simulations” of future or possible outcomes, as well as interpretations of perceptual input (Graesser, Singer, and Trabasso 1994; Barsalou 1999; Zwaan and Radvansky 1998; Zwaan and Pecher 2012). These simulation approaches can be referred to as *embodied theories of mind*. Their goal is to view the semantic interpretation of an expression by means of a simulation, which is either mental (a la Bergen and Evans) or interpreted graphs such as Petri Nets (a la Narayanan and Feldman).

In this position paper, we introduce a simulation framework, VoxWorld, that integrates the functionality and the goals of all three approaches above. Namely, we situate an embodied agent in a multimodal simulation, with *mind-reading* interpretive capabilities, facilitated through assignment and evaluation of object and context parameters within the environment being modeled.

For example, relations created by events persist after the completion of the event, and so in event simulation, they must also persist in order for the simulation to be considered accurate. Fig. 1 shows a number of objects at similar locations but in one case in orientations that, due to the effects of physics, would be considered “unstable” after the completion of a placement event. Object knowledge about things like *shape of cup*, *top of plate*, *default position of banana*

mean that human observers can judge the image on the left to be unsatisfactory results of placement events and that on the right to be more prototypical, due to the human ability to *simulate* what the result of an event in a given environment likely will be.

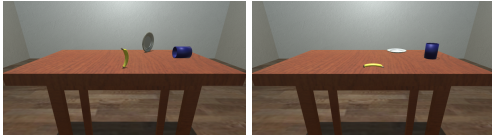


Figure 1: Objects in unnatural (L) and natural (R) positions

VoxWorld is based on the semantic scaffold provided by the VoxML modeling language (Pustejovsky and Krishnaswamy 2016), which provides a dynamic, interpretable model of objects, events, and their properties. This allows us to create visualized simulations of events and scenarios that are rendered analogues to the “mental simulations” discussed above. VoxSim (Krishnaswamy and Pustejovsky 2016a; 2016b) serves as the event simulator within which these simulations are created and rendered in real time, serving as the computer’s method of visually presenting its interpretation of a situation or event. Because modalities are modes of presentation, a multimodal simulation entails as many presentational modes as there are modalities being modeled. The visual modality of presentation (as in embodied gaming) necessitates “situatedness” of the agent, as do the other perceptual modalities. Therefore, when we speak of *multimodal simulations*, they are inherently situated. In a human-computer interaction using such a simulation, the simulation is a demonstration of the computational agent “mind-reading” capabilities (an *agent simulation*). If the two are the same (where the agent is a proxy for the player or user, then the “mind-reading” is just a demonstration of the scenario) If, on the other hand, the two are separate (agent is *not* proxy for the user), then the simulation/demonstration communicates the agent’s understanding of the user and the interaction. In this case, this demonstration entails the illustration of both epistemic and perceptual content of the agent.

We believe that simulation can play a crucial role in human-computer communication; it creates a shared epistemic model of the environment inhabited by a human and an artificial agent, and demonstrates the knowledge held by the agent publicly. Demonstrating knowledge is needed to ensure a shared understanding with its human interlocutor. If an agent is able to receive information from a human and interpret that relative to its current physical circumstances, it can create an epistemic representation of that same information. However, without a modality to express that representation independently, the human is unable to verify or query what the agent is perceiving or how that perception is being interpreted. In a simulation environment the human and computer share an epistemic space, and any modality of communication that can be expressed within that space (e.g., linguistic, visual, gestural) enriches the number of ways that a human and a computer can communicate within object and situation-based tasks, such as those investigated by Hsiao

et al. (2008), Dzifcak et al. (2009), and Cangelosi (2010), among others.

VoxWorld, and the accompanying simulation environment provided by VoxSim, includes the perceptual domain of objects, properties, and events. In addition, propositional content in the model is accessible to the simulation. Placing even a simple scenario, such as a blocks world setup, in a rendered 3D environment opens the search space to the all the variation allowed by an open world, as objects will almost never be perfectly aligned to each other or to a grid, with slight offsets in rotation caused by variations in interpolation, the frame rate, or effects of the platform’s physics. Nevertheless, when the rendering is presented to a user, the user can use their native visual faculty to quickly arrive at an interpretation of what is being depicted.

Situational embodiment takes place in real time, so in the case of a situation where there may be too many variables to predict the state of the world at time t from a set of initial conditions at time 0, situational embodiment within the simulation allows the reasoning agent to reason forward about a specific subset of consequences of actions that may be taken at time t , given the agent’s current conditions and surroundings. Situatedness and embodiment is required to arrive at a complete, tractable interpretation given any element of non-determinism. For example, an agent trying to navigate a maze from start to finish could easily do so with a map that provides them complete, or at least sufficient, information about the scenario. If, however, the scene includes a disruptor (e.g., the floor crumbles, or doors open and shut randomly), the agent would be unable to plot a course to the goal. It would have to start moving, assess the current circumstances at every timestep, and choose the next move or next set of n moves based on them. Situated embodiment allows the agent to assess next move based on the current set of relations between itself at the environment (e.g., ability to move forward but not leftward at the current state). This provides for reasoning that not only saves computational resources but performs more analogously to human reasoning than non-situated, non-embodied methods.

A Formal Interpretation of Simulations

Given the distinction above between interpretations for “simulation,” we have been developing an approach that integrates all three: a situated embodied environment built on a game engine platform. The computer, either as an embodied agent distinct from the viewer, or as the totality of the rendered environment itself, presents an interpretation (*mind-reading*) of its internal model, down to specific parameter values, which are often assigned for the purposes of testing that model.

We assume that a simulation is a contextualized 3D virtual realization of both the situational environment and the co-situated agents, as well as the most salient content denoted by communicative acts in discourse between them. VoxWorld and VoxML (Pustejovsky and Krishnaswamy 2016), provide the following characteristics: object encoding with rich semantic typing and action affordances; action encoding as multimodal programs; it reveals the elements of the com-

mon ground in interaction between parties, be they humans or artificially intelligent agents.

“Common ground” in a computational context relies on implementations of the following:

1. *Co-situatedness* and *co-perception* of the agents, such that they can interpret the same situation from their respective frames of reference. This might be a human and an avatar perceiving the same virtual scene from different perspectives (see Fig. 3), or a combined virtual-physical scene with the integration of computer vision technology; or a human sharing the perspective of a robot as it navigates through a disaster zone.
2. *Co-attention* of a shared situated reference, which allows more expressiveness in referring to the environment (i.e., through language, gesture, visual presentation, etc.). The human and avatar might be able to refer to objects on the table in multiple modalities with a common model of differences in perspective-relative references (e.g., “your left, my right”); or the human sharing the robot’s perspective might be able to direct its motion using reference in natural language (“go through the second door on the left”) or gesture (“go this way,” with pointing).
3. *Co-intent* of a common goal, such that adversarial relationships between agents reflect a breakdown in the common ground. Here, the human and avatar in interaction around a table might seek to collaborate to build a structural pattern known to one or both of them; or the human and robot sharing perspective both have a goal to free someone trapped behind a door in a fire. The robot informs the human about the situation and the human helps the robot problem-solve in real time until the goal is achieved.

The theory of common ground has a rich and diverse literature concerning what is shared or presupposed in human communication (Clark and Brennan 1991; Gilbert 1992; Stalnaker 2002; Asher 1998; Tomasello and Carpenter 2007; Pustejovsky 2018).

VoxML (Visual Object Concept Markup Language) forms the scaffold used to encode knowledge about objects, events, attributes, and functions by linking lexemes to their visual instantiations, termed the “visual object concept” or *voxeme*. In parallel to a lexicon, a collection of voxemes is termed a *voxicon*. There is no requirement on a voxicon to have a one-to-one correspondence between its voxemes and the lexemes in the associated lexicon, which often results in a many-to-many correspondence. That is, the lexeme *plate* may be visualized as a [[SQUARE PLATE]], a [[ROUND PLATE]], or other voxemes, and those voxemes in turn may be linked to other lexemes such as *dish* or *saucer*.

Each voxeme is linked to either an object geometry, a program in a dynamic semantics, an attribute set, or a transformation algorithm, which are all structures easily exploitable in a rendered simulation platform.

An OBJECT voxeme’s semantic structure provides *habitats*, which are situational contexts or environments conditioning the object’s *affordances*, which may be either “Gibsonian” affordances (Gibson, Reed, and Jones 1982)

or “Telic” affordances (Pustejovsky 1995; 2013). A habitat specifies how an object typically occupies a space. When we are challenged with computing the embedding space for an event, the individual habitats associated with each participant in the event will both define and delineate the space required for the event to transpire. Affordances are used as attached behaviors, which the object either facilitates by its geometry (Gibsonian) or purposes for which it is intended to be used (Telic). For example, a Gibsonian affordance for [[CUP]] is “grasp,” while a Telic affordance is “drink from.” This allows procedural reasoning to be associated with habitats and affordances, executed in real time in the simulation, inferring the complete set of spatial relations between objects at each frame and tracking changes in the shared context between human and computer.

It also allows the system to reason about objects and actions independently. When simulating the objects alone, the simulation presents how the objects change in the world. By removing the objects and presenting only the actions that the viewer would interpret as *causing* the intended object motion (i.e., a pantomime of an embodied agent moving an object without the object itself), the system can present a “decoupled” interpretation of the action, for example, as an animated gesture that traces the intended path of motion. By composing the two, it demonstrates that particular instantiation of the complete event. This allows an embodied situated simulation approach to easily compose objects with actions by directly interpreting at runtime how the two interact.

Reasoning within an Interpreted Simulation

VoxSim (Krishnaswamy and Pustejovsky 2016a; 2016b) implements the VoxML platform in the Unity game engine software by Unity Technologies¹. The current implementation of VoxSim provides scenes in a Blocks World domain, augmented with a set of more complicated or interesting everyday objects (e.g., cups, plates, books, etc.). There are scenes without an avatar where the user can direct the computer to manipulate objects in space (see Figs. 1 and 2) or with an avatar that can act upon objects and respond to the user’s input where it is ambiguous (see Fig. 3). *Vox-World* contains other software, models, and interfaces, e.g., to consume input from CNN-based gesture recognizers (Krishnaswamy et al. 2017), and to track and update the agent’s epistemic state or knowledge about what the human interlocutor knows.

It is a straightforward process to create new scenes with 3D geometries with packaged code that handles the creation and instantiation of voxemes, handles their interactions and performs basic spatial reasoning over them. We also provide a library of basic motion predicates and methods of composing them into more complex actions using VoxML.

Given the continuous tracking of object parameters such as position and orientation, facilitated by a game engine or simulation, and the knowledge of object, event, and functional semantics facilitated by a formal model, an entity’s interpretation at runtime can be computed in conjunction with the other entities it is currently interacting with and their

¹<https://unity3d.com/>

properties. One such canonical example would be placing an object [[SPOON]] in an [[IN]] relation with another object [[MUG]] (Fig. 2).

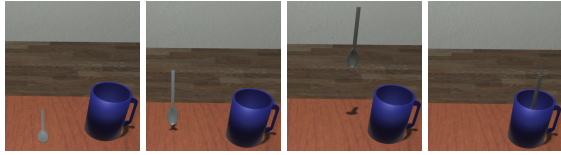


Figure 2: [[SPOON IN MUG]]

The mug has an intrinsic top, which is aligned with the upward Y-axis of the world or embedding space (denoted in VoxML as $\{align(Y, \mathcal{E}_Y), top(+Y)\}$). The mug is also a concave object, and the mug’s geometry (the [[CUP]], excluding the handle) has reflectional symmetry across its inherent (object-relative) XY- and YZ-planes, and rotational symmetry around its inherent Y-axis such that when the object is situated in its inherent *top* habitat, its Y-axis is parallel to the world’s. From this we can infer that the *opening* (e.g., access to the concavity) must be along the Y-axis. Encoding the object’s concavity also allows fast computation for physics and collisions using bounding boxes, while still facilitating reasoning over concave objects.

VoxSim performs reasoning over 3D variants of well-known spatial calculi (e.g., Albath et al. albath2010rcc) and interval/point calculi from libraries such as QSRLib (Gatsoulis et al. 2016), computing axial overlap with the Separating Hyperplane Theorem (Schneider 2014). In order to put object x in object y , while maintaining external contact with y ’s concave geometry, the placed object x must *fit inside* the concave object y . In the case of the mug, it can be reasoned as shown that its concavity opens along the Y-axis, so any computational reasoner must also determine that the object to be placed within it can fit in that same orientation. In the case of a spoon, normally lying flat on a surface, somewhere flush with the world’s XZ-plane, simply placing it at the point where it would touch the bottom of the inside of the mug would also cause it to interpenetrate the mug’s sides inappropriately, and so the spoon must first be turned (rotated) to align with the mug’s opening. The requirements on simulating *put the spoon in the mug* enforce the resulting state of this “turn” action as a precondition, which allows for intelligent decision making typically not learnable from a modality such as language or still images.

Deformation of the object is also possible, as long as it maintains the object’s topological isomorphism. Just as individual transformations over rigid object bounding boxes can be tested for relation or event satisfaction in a given context, transformations over individual vertices and edges can be performed to search for the set of deformations that satisfy a known constraint. For instance, the bounds of an object x and a containing space y can be extracted at each frame as deformations are performed over x such that a predicate like $contains(extents(x), extents(y))$ can be computed to test the satisfaction condition of $put(x, in(y))$ where x is deformable. This allows us, within VoxSim, to solve for deformations that describe events such as *crumple* or *fold*. Even

within the continuous open world of a game or simulation-based environment, searching for transformations over individuals within a finite set of vertices and edges that satisfy a predetermined condition keeps the search space tractable well within computational limits and facilitates the gathering of data for experiments that can teach an AI agent to solve decidable problems such as “how to fold a cloth” or “how to fit numerous items in a container.”

Search can be performed in an embodied, situated simulation environment through parameter setting of the type found in *computational simulation modeling*. In order to generate a visualization of an event, all variable parameters must have values assigned, otherwise the program will fail to run. This requirement on the game engine software also becomes a requirement on the creation of a fully-defined simulation model. The composition of objects and events provide much of the needed information but in cases where parameters still require values (e.g., the speed of a moving object described simply by the predicate “slide”), we can use Monte Carlo value assignment to set those values in the simulation environment. The rendered simulation including the stochastically-assigned value(s) is presented as the system’s interpretation of that situation being modeled formally.

We have performed a number of experiments in this area using the VoxSim software (Krishnaswamy 2017). For a set of motion predicates where various parameters are left underspecified in the linguistic description (e.g., “slide the block across the table” says nothing about speed or direction of motion; “put the block next to book” or “touch the block to the book” does not fully specify the relative placement of the two objects in 3D space), we generated multiple simulations of such events, captured them on video along with the specific parameter values used in each simulation, and had human evaluators choose the best simulation out of three for one description, and the best description out of three for one simulation. This allowed us generate a novel dataset of motion events and parameter values describing prototypical instances of them (according to evaluators), and such data can be used to train a model that captures contextual dependencies for better simulation generation and interpretation.

Learning by Communication

One of the things that an embodied simulation model for AI enables is *peer-to-peer communication*, specifically because of the requirement that the AI agent have some kind of situated embodiment in which it interprets its environment. This allows the creation of common ground between the human and the AI that allows them to communicate (Pustejovsky et al. 2017).

We use the example of learning to build a structure, namely a staircase. In a series of user studies, we had naive human subjects interact with an avatar (a screenshot of VoxSim is given in Fig. 3) to build a staircase out of six blocks. Due to subjects’ lack of skill using the system fluently, the generated structures all satisfied the user’s notion of a staircase, but across the 17 samples were often diverse and noisy (variants included spaces between blocks, blocks not properly aligned, or blocks rotated). Each of the 17 struc-



Figure 3: Screenshot of human-avatar interaction in VoxSim. The purple circle indicates the location interpreted as the target of the human subject’s pointing.

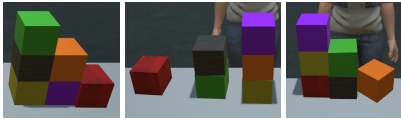


Figure 4: Example user-constructed staircases

tures is defined by ≈ 20 qualitative spatial relations, and each set is stored as an unordered list.

The qualitative relation sets that defined each structure, extracted directly from VoxSim via Unity, were then used to train a model for the avatar to use to build its own novel instance of a staircase structure. The avatar in the simulation places a block then, using a 4-layer convolutional neural network, chooses one of its known examples to begin approximating with the next step. Using a long short-term memory network of 3 layers with 32 nodes each, and trained up to 20 timesteps, the agent selects the most likely sequence of moves that would approximate the chosen example structures. These moves are then pruned using heuristic graph-matching and a move is chosen. Further details of the method and sample results are given in (Krishnaswamy, Friedman, and Pustejovsky 2019).

However, sometimes the agent generates a structure that does not comport with an observer’s understanding of that structure label. In previous experimentations using a similar framework for action learning (Do, Krishnaswamy, and Pustejovsky 2018) we encountered difficulty in using negative examples to steer the learning agent away from incorrect generations (cf. also (Dietterich 2000; Nguyen, Yosinski, and Clune 2015)). Here, the interaction within the simulation environment facilitates “learning by communication” that allows us to take a negative example and turn it into a corresponding positive example, and storing both increases the overall data size and gives a clear minimal pair between a good example and a bad one.

A correction interaction might proceed as follows:

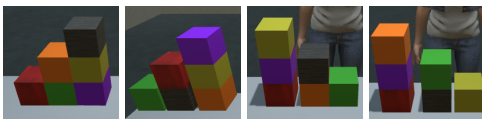


Figure 5: Example learned staircases

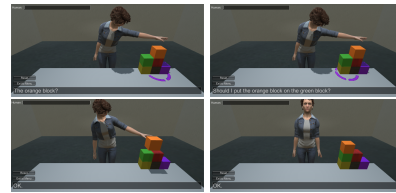


Figure 6: Correcting a generated structure with multimodal communication.

1. The system generates the “staircase” in the top left image shown in Fig. 6, which is one block off from a prototypical staircase. This gets marked as an incorrect case.
2. The user points to the orange block at the top of the structure. The agent clarifies this request.
3. The user indicates the green block and gestures for (or tells) the agent to move the orange block there.
4. The agent clarifies this request and makes the move. The result is marked as a correct case.

This is demonstrated in the simulation environment. Because of the simulation acting in the service of multimodal communication between a human and a computer, we can improve the model (understanding) of the computer’s concepts (here, a staircase) through learning by communication, and iterative demonstration of the agent’s model.

Conclusion

Across different fields and in the existing AI, cognition, and game development literature, there exist many different definitions of “simulation.” Nonetheless, we believe the common thread between them is that simulations as a framework facilitate both qualitative and quantitative reasoning by providing quantitative data (for example, exact coordinates or rotations) that can be easily converted into qualitative representations. This makes simulation an effective platform for both producing and learning from datasets.

When combined with formal encodings of object and event semantics, at a level higher than treating objects as collections of geometries, or events as sequences of motions or object relations, 3D environments provide a powerful platform for exploring “computational embodied cognition.” Recent developments in the AI field have shown that common-sense understanding in a general domain requires either orders of magnitude more training data than traditional deep learning models, or more easily decidable representations, involving context, differences in perspective, and grounded concepts, to name a few.

Technologies in use in the gaming industry are proving to be effective platforms on which to develop systems that afford gathering both traditional data for deep learning and representations of common sense, situated, or embodied understanding. In addition, game engines perform a lot of “heavy lifting,” providing APIs for UI and physics, among others, which allows researchers to focus on implementing truly novel functionality and develop tools for experimentation in simulation-based and qualitative understanding of

both human and machine cognition and intelligence.

References

- Asher, N. 1998. Common ground, corrections and coordination. *Journal of Semantics*.
- Barsalou, L. W. 1999. Perceptions of perceptual symbols. *Behavioral and brain sciences* 22(4):637–660.
- Bergen, B. K. 2012. *Louder than words: The new science of how the mind makes meaning*. Basic Books.
- Cangelosi, A. 2010. Grounding language in action and perception: from cognitive agents to humanoid robots. *Physics of life reviews* 7(2):139–151.
- Clark, H. H., and Brennan, S. E. 1991. Grounding in communication. *Perspectives on socially shared cognition* 13(1991):127–149.
- Craik, K. J. W. 1943. *The nature of explanation*. Cambridge University, Cambridge UK.
- Dietterich, T. G. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, 1–15. Springer.
- Do, T.; Krishnaswamy, N.; and Pustejovsky, J. 2018. Teaching virtual agents to perform complex spatial-temporal activities.
- Dzifcak, J.; Scheutz, M.; Baral, C.; and Schermerhorn, P. 2009. What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 4163–4168. IEEE.
- Evans, V. 2013. *Language and time: A cognitive linguistics approach*. Cambridge University Press.
- Feldman, J. 2010. Embodied language, best-fit analysis, and formal compositionality. *Physics of life reviews* 7(4):385–410.
- Gatsoulis, Y.; Alomari, M.; Burbridge, C.; Dondrup, C.; Duckworth, P.; Lightbody, P.; Hanheide, M.; Hawes, N.; Hogg, D.; Cohn, A.; et al. 2016. Qsrlib: a software library for online acquisition of qualitative spatial relations from video.
- Gibson, J. J.; Reed, E. S.; and Jones, R. 1982. *Reasons for realism: Selected essays of James J. Gibson*. Lawrence Erlbaum Associates.
- Gilbert, M. 1992. *On social facts*. Princeton University Press.
- Goldman, A. I. 1989. Interpretation psychologized*. *Mind & Language* 4(3):161–185.
- Goldman, A. I. 2006. *Simulating minds: The philosophy, psychology, and neuroscience of mindreading*. Oxford University Press.
- Gordon, R. M. 1986. Folk psychology as simulation. *Mind & Language* 1(2):158–171.
- Graesser, A. C.; Singer, M.; and Trabasso, T. 1994. Constructing inferences during narrative text comprehension. *Psychological review* 101(3):371.
- Heal, J. 1996. Simulation, theory, and content. *Theories of theories of mind* 75–89.
- Hsiao, K.-Y.; Tellex, S.; Vosoughi, S.; Kubat, R.; and Roy, D. 2008. Object schemas for grounding language in a responsive robot. *Connection Science* 20(4):253–276.
- Johnson-Laird, P. N., and Byrne, R. M. 2002. Conditionals: a theory of meaning, pragmatics, and inference. *Psychological review* 109(4):646.
- Johnson-Laird, P. 1987. How could consciousness arise from the computations of the brain. *Mindwaves. Oxford: Basil Blackwell* 247–257.
- Krishnaswamy, N., and Pustejovsky, J. 2016a. Multimodal semantic simulations of linguistically underspecified motion events. In *Spatial Cognition X: International Conference on Spatial Cognition*. Springer.
- Krishnaswamy, N., and Pustejovsky, J. 2016b. VoxSim: A visual platform for modeling motion language. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. ACL.
- Krishnaswamy, N.; Narayana, P.; Wang, I.; Rim, K.; Bangar, R.; Patil, D.; Mulay, G.; Ruiz, J.; Beveridge, R.; Draper, B.; and Pustejovsky, J. 2017. Communicating and acting: Understanding gesture in simulation semantics. In *12th International Workshop on Computational Semantics*.
- Krishnaswamy, N.; Friedman, S.; and Pustejovsky, J. 2019. Combining deep learning and qualitative spatial reasoning to learn complex structures from sparse examples with noise. In *33rd AAAI Conference on Artificial Intelligence*. AAAI.
- Krishnaswamy, N. 2017. *Monte-Carlo Simulation Generation Through Operationalization of Spatial Primitives*. Ph.D. Dissertation, Brandeis University.
- Narayanan, S. 2010. Mind changes: A simulation semantics account of counterfactuals. *Cognitive Science*.
- Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 427–436.
- Pustejovsky, J., and Krishnaswamy, N. 2016. VoxML: A visualization modeling language. In Chair), N. C. C.; Choukri, K.; Declerck, T.; Goggi, S.; Grobelnik, M.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA).
- Pustejovsky, J.; Krishnaswamy, N.; Draper, B.; Narayana, P.; and Bangar, R. 2017. Creating common ground through multimodal simulations. In *Proceedings of the IWCS workshop on Foundations of Situated and Multimodal Communication*.
- Pustejovsky, J. 1995. *The Generative Lexicon*. Cambridge, MA: MIT Press.
- Pustejovsky, J. 2013. Dynamic event structure and habitat theory. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)*, 1–10. ACL.

- Pustejovsky, J. 2018. From experiencing events in the action-perception cycle to representing events in language. *Interaction Studies* 19.
- Schneider, R. 2014. *Convex bodies: the Brunn–Minkowski theory*. Number 151. Cambridge university press.
- Stalnaker, R. 2002. Common ground. *Linguistics and philosophy* 25(5-6):701–721.
- Tomasello, M., and Carpenter, M. 2007. Shared intentionality. *Developmental science* 10(1):121–125.
- Zwaan, R. A., and Pecher, D. 2012. Revisiting mental simulation in language comprehension: Six replication attempts. *PloS one* 7(12):e51382.
- Zwaan, R. A., and Radvansky, G. A. 1998. Situation models in language comprehension and memory. *Psychological bulletin* 123(2):162.