Object Embodiment in a Multimodal Simulation

James Pustejovsky, Nikhil Krishnaswamy, Tuan Do
Department of Computer Science
Brandeis University
Waltham, MA 02453 USA
{jamesp,nkrishna,tuandn}@brandeis.edu

Abstract

In this paper, we introduce a multimodal environment and semantics for facilitating communication and interaction with a computational agent, as proxy to a robot. To this end, we have created an embodied 3D simulation enabling both the generation and interpretation of multiple modalities, including: language, gesture, and the visualization of objects moving and agents acting in their environment. Objects are encoded with rich semantic typing and action affordances, while actions themselves are encoded as multimodal expressions (programs), allowing for contextually salient inferences and decisions in the environment.

Motivation

In order to facilitate communication with a computational agent, we have been pursuing a new approach to modeling the semantics of natural language: *Multimodal Semantic Simulations (MSS)*. This framework assumes both a richer formal model of events and their participants, as well as a modeling language for constructing 3D visualizations of objects and events denoted by linguistic expressions. The Dynamic Event Model (DEM) encodes events as programs in a dynamic logic with an operational semantics, while the language VoxML, Visual Object Concept Modeling Language, is being used as the platform for multimodal semantic simulations in the context of human-computer communication, as well as for image- and video-related content-based grounding and querying.

The simulation environment we describe is presently configured for joint activity and communication between a human and a computational agent. But because of the nature and design of the VoxML model, we believe it can be used as the conceptual platform for robotic representation, reasoning, and concept learning.

While the workshop is mainly aimed at issues of knowledge acquisition over large datasets with multiple sensory modalities, the present paper focuses on the representation of multiple modalities that interface to language input, for the purpose of facilitating communication and activity with a computational agent. The datasets are currently small, since we are creating the specification and semantics of the modeling language that we believe is necessary to allow such an interaction in the first place.

We believe that simulation can play a crucial role in communication: namely, it creates a shared epistemic model of the environment inhabited by a human and an artificial agent or robot. Further, the simulation demonstrates the knowledge held by the agent or robot publicly. Demonstrating knowledge is needed to ensure a shared understanding with the humans involved in the activity. But why create a simulation model, if the goal is to interact with a robot? If a robotic agent is able to receive information from a human commander or collaborator in a linguistic modality and interpret that relative to its current physical circumstances, it is able to create an epistemic representation of the same information provided by the human. However, in the absence of any modality of expressing that representation independently, the human is unable to verify or query what the robot agent is actually perceiving or how that perception is being interpreted. A simulation environment provides an avenue for the human and robot to 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 robotic agent can communicate on object and situation-based tasks such as those investigated by (Hsiao et al. 2008), (Dzifcak et al. 2009), (Cangelosi 2010).

Embodiment and Affordances

Central to understanding the integration of multiple sensory modalities by an agent is the notion of embodiment. Prior work in visualization from natural language has largely focused on object placement and orientation in static scenes (Coyne and Sproat 2001; Siskind 2001; Chang et al. 2015). In previous work (Pustejovsky and Krishnaswamy 2014; Pustejovsky 2013a), we introduced a method for modeling natural language expressions within a 3D simulation environment, Unity. The goal of that work was to evaluate, through explicit visualizations of linguistic input, the semantic presuppositions inherent in the different lexical choices of an utterance. This work led to two additional lines of research: an explicit encoding for how an object is itself situated relative to its environment; and an operational characterization of how an object changes its location or how an agent acts on an object over time. The former

has developed into a semantic notion of situational context, called a *habitat* (Pustejovsky 2013a; McDonald and Pustejovsky 2014), while the latter is addressed by dynamic interpretations of event structure (Pustejovsky and Moszkowicz 2011b; Pustejovsky 2013b; Mani and Pustejovsky 2012; Pustejovsky 2013a). The requirements on a "multimodal simulation semantics" include, but are not limited to, the following components:

- A minimal embedding space (MES) for the simulation must be determined. This is the 3D region within which the state is configured or the event unfolds;
- Object-based attributes for participants in a situation or event need to be specified; e.g., orientation, relative size, default position or pose, etc.;
- An epistemic condition on the object and event rendering, imposing an implicit point of view (POV);
- Agent-dependent embodiment; this determines the relative scaling of an agent and its event participants and their surroundings, as it engages in the environment.

In the discussion that follow, we outline briefly the components of a multimodal simulation environment to address the needs stated above to provide multi-sensory representations for robots.

A humanoid skeleton in a 3D environment is a directed rooted graph with nodes laid out in the rough configuration of a human, representing the positions of the major joins. Even though a robotic agent may not be laid out in a humanoid shape (and nearly also simpler robots are not), the same graph structure can be used to represent the locations of major pivot points on the robot's external structure, such as those of graspers or robotic limbs.

We can thus create a 3D representation of a robotic agent that operates in the real world and give it a skeleton structure that reflects the actual robot's join configuration. Assuming the physical robot and its 3D representation are isomorphic, this then allows us to simulate events in the 3D world that represent real-world events (such as moving the simulated robot around a simulated table that has simulated blocks on it in a configuration that is generated from the positioning of real blocks on a real table). The event simulation then generates a set of position and orientation information for each object in the scene at each time step t, which isomorphic to the real-world configuration in the same way that the robot's virtual skeleton is isomorphic to its actual join structure. This allows the real robot, acting as an agent, to be fed a set of translation and rotation "moves" by its virtual embodiment that is a nearly exact representation of the steps it would need to take to satisfy a real world goal, such as navigating to a target or grasping an object.

By default, the camera in a simulated world is independent of any agents, allowing the human user to freely roam and interpret the virtual space, but it is trivial to add a switch that would allow the user to move the camera to the location on the virtual agent that corresponds to the location of the sensors on the physical robotic agent. A human watching the simulation can view a representation of what the robotic agent perceives, and then has a way of looking inside the agent's "brain."

VoxML: a Language for Concept Visualization

While both experience and world knowledge about objects and events can influence our behavior as well as our interpretation of said events, such factors are seldom involved in representing the predicative force of a particular lexeme in a language. Some representations, such as Qualia Structure (Pustejovsky 1995) do provide additional information that can be used to map a linguistic expression to a minimal model of the event, and then from there to a visual output modality such as one that may be produced by a computer system, and so requires a computational framework to model it. Still, such languages are not in themselves rich enough to create useful minimal models.

To remedy this deficit, we have developed a modeling language, VoxML (Visual Object Concept Markup Language), for constructing 3D visualizations of natural language expressions (Pustejovsky and Krishnaswamy 2016). VoxML forms the scaffold used to link lexemes to their visual instantiations, termed the "visual object concept" or *voxeme*. In parallel to a lexicon, a collection of voxemes is 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]][1], or other voxemes, and those voxemes in turn may be linked to other lexemes such as *dish* or *saucer*.

Each voxeme is linked to an object geometry (if a noun—OBJECT in VoxML), a dynamic logic program (if a verb or VoxML PROGRAM), an attribute set (VoxML ATTRIBUTEs), or a transformation algorithm (VoxML RELATIONs or FUNCTIONs). VoxML is used to specify the "episemantic" information beyond that which can be directly inferred from the linked geometry, Dynamic Interval Temporal Logic (DITL) progrsm (Pustejovsky and Moszkowicz 2011a), or attribute properties.

In order to demonstrate the composition of the linguistic expression plus the VoxML encoded information into a fully-realized visual output, we have developed, **VoxSim** (Krishnaswamy and Pustejovsky 2016), a semantically-informed visual event simulator built on top of the Unity game engine (Goldstone 2009).[2]

VoxSim procedurally composes the properties of voxemes in parallel with the lexemes to which they are linked. Input is a simple natural language sentence, which is part-of-speech tagged, dependency-parsed, and transformed into a simple predicate-logic format.

From tagged and parsed input text, all noun phrases are indexed to objects in the scene, so a reference to *a/the block* causes the simulator to attempt to locate a voxeme instance

[1]Note on notation: discussion of voxemes in prose will be denoted in the style [[VOXEME]] and should be taken to refer to a visualization of the bracketed concept.

[2]The VoxSim Unity project and source may be found at `https://github.com/nkrishnaswamy/voxicon`.

in the scene whose lexical predicate is "block." Attributive adjectives impose a sortal scale on their heads, so *small block* and *big block* single out two separate blocks if they exist in the scene, and the VoxML-encoded semantics of "small" and "big" discriminates the blocks based on their relative size. *red block* vs. *green block* results in a distinction based on color, a nominal attribute, while *big red block* vs. *small red block* can be used to disambiguate two distinct red blocks by iteratively evaluating each interior term of a formula such as *big(red(block))* until the reference can be resolved into a single object instance that has all the signaled attributes[3]. The system may ask for clarification (e.g., *"Which block?"*) if the object reference is still ambiguous.

An OBJECT voxeme's semantic structure provides *habitats*, situational contexts or environments which condition the object's *affordances*, which may be either "Gibsonian" or "telic" in nature (Gibson, Reed, and Jones 1982; Pustejovsky 1995; 2013a). Affordances are used as attached behaviors, which the object either facilitates by its geometry (Gibsonian) (Gibson, Reed, and Jones 1982), or for which it is intended to be used (telic) (Pustejovsky 1995). For example, a Gibsonian affordance for [[CUP]] is "grasp," while a telic affordance is "drink from." Conventionally, agents of a VoxML PROGRAM must be explicitly singled out in the associated implementation by belonging to certain entity classes (e.g., humans). Thus affordances describe what *can be done to* the object, and not what actions it *itself* can perform. Therefore, an affordance is notated as HABITAT $\rightarrow$ [EVENT]RESULT. $H_{[2]} \rightarrow [put(x, on([1]))]support([1], x)$ can be paraphrased as "In habitat-2, an object x can be put on component-1, which results in component-1 supporting x." This procedural reasoning from habitats and affordances, executed in real time, allows VoxSim to infer the complete set of spatial relations between objects at each state and track changes in the shared context between human and computer. Thus, simulation becomes a way of tracing the consequences of linguistic spatial cues through a narrative.

A VoxML entity's interpretation at runtime depends on the other entities it is composed with. In order to contain another object, a cup must be currently situated in a *habitat* which allows objects to be placed partially or completely inside it (represented by partial overlap and tangential or non-tangential proper part relations—PO, TPP, or NTPP according to the Region Connection Calculus (Randell, Cui, and Cohn 1992)). In VoxML, [[CUP]] is encoded as a concave object with rotational symmetry around the Y-axis and reflectional symmetry across the XY and YZ planes, meaning that it opens along the Y-axis. Its HABITAT further situates the opening along its positive Y-axis, meaning that if the cup's opening along its +Y is currently unobstructed, it affords containment. Previously established habitats, i.e., "The cup is flipped over," may activate or deactivate these and other affordances.

The spatial relations operating within the context of a verbal program, such as "put the spoon in the cup," enforce constraints that require a test against the current situational context before a value assignment can be made. Given *put*, if the "placed object" is of a size that is too large to fit inside the mentioned object, VoxSim conducts a series of calculations to see if the object, when reoriented along any of its three orthogonal axes, will be situated in a configuration that allows it to fit inside the region bounded by the ground object's containing area. The containing area is situated relative to one of the ground object's orthogonal axes. For example, the symmetrical and concave properties of [[CUP]] compose to situate the cup's opening along its *positive* Y-axis. So, to place a [[SPOON]] in a [[CUP]], assuming objects of typical size, [[SPOON]] must be reoriented so that its world-space bounding box aligning with the [[CUP]]'s Y-axis is smaller than the bounds of the [[CUP]]'s opening in that same configuration.

Figure 1: Spoon on table vs. spoon in cup

Learning Events from Motion Data

Now let us turn the language-to-visualization strategy on its head. VoxML can also be used to help detect and recognize events and actions in video. This task has received increasing attention in the scientific community, due to its relevance to a wide variety of applications (Ballan et al. 2011) and there have been calls for annotation infrastructure that includes video (Ide 2013).

Our lab has begun bootstrapping a dataset of videos annotated with event-subevent relations using ECAT, an internally-developed video annotation tool (Do, Krishnaswamy, and Pustejovsky 2016). This annotation tool allows us to annotate videos of labeled events with object participants and subevents, and to induce what the common subevent structures are for the labeled superevent. Using the Microsoft Kinect®, we are currently recording videos of a test set of human actions interacting with simple objects, such as blocks, cylinders, and balls. Both human bodies (rigs) and these objects can be tracked and annotated as participants in a recorded motion event; this labeled data can then be used to build a corpus of *multimodal semantic simulations* of these events that can model object-object, object-agent, and agent-agent interactions through the event duration (Figure 2). This library of simulated motion events can serve as a novel resource directly linking natural language to event visualization, indexed through the multimodal lexical representation for the event, its voxeme.

In turn, this facilitates a mechanism to fill in the miss-

[3]See (Pustejovsky and Krishnaswamy forthcoming) for details on discriminating and referencing objects through sortal and scalar descriptions.

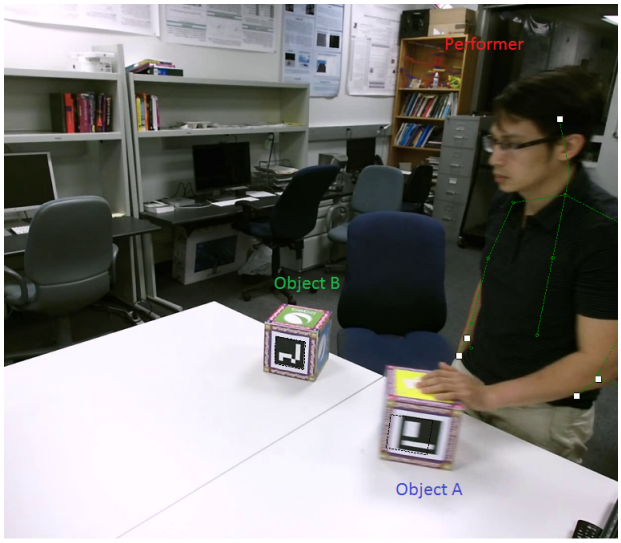

Figure 2: Event capture with fine-grained annotation

ing pieces in a simulation of *underspecified motion events*. Specifically, we want to use our captured and annotated data to learn a *generative* model that combines programmatic representation of events in VoxML and sequential learning methods. It would be used to distinguish *process* events, such as "I slide the cube" from its *completive* form, such as "I slide the cube to the cylinder". In a more ambitious scenario, we plan to put the human agent in a *pedagogic role*, allowing our robotic agent to learn to perform gradually more complex events, such as when there are dynamic *spatio-temporal* interactions between objects, including affordance information (Bohg and Kragic 2009). The generativity of a learning model would allow the robot to search for both an economical path to the goal and the required movements and interactions with the objects (given an appropriate transformation between human and robotic kinematics) (Billard et al. 2008).

We are also interested in learning the mapping between communicative gestures and their *speech acts*. In a situation when a human agent has to give directions to a robot in order to achieve a specific task (such as building a structure), using gestures in a multimodal (coverbal) manner can be more economical than language alone. However, it turns out that a single human gesture can be interpreted as several different speech acts, depending on local context. For example, a gesture "hands joined then move apart horizontally" could mean "build a row of blocks", "space out the blocks you are constructing", or "the structure to be built has a flat base". Using the same annotation methodology, we can map between *(current_configuration, verbal command, coverbal gesture)* and a speech act of type *(target_configuration)*. Given the current configuration in its simulated environment, by receiving a linguistic expression, a coverbal gesture, or the simultaneous articulation of both, the robot can learn to generate the appropriate target configuration.

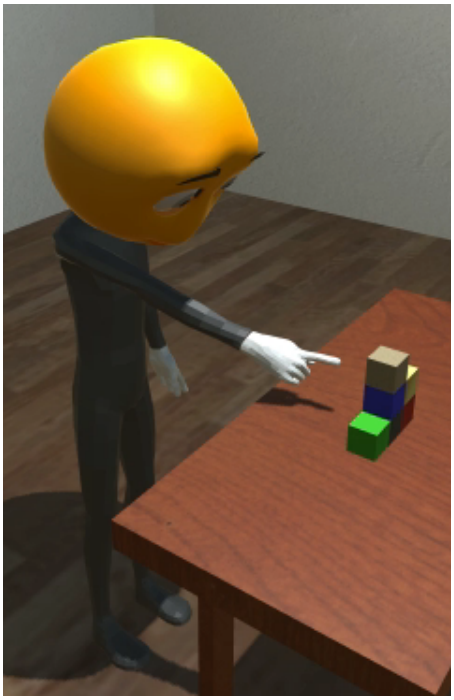

Figure 3: Sample gesture-driven interaction

Conclusion and Future Directions

We have described initial steps towards the design and development of a multimodal simulation environment, based on a modeling language that admits of multiple representations from different modalities. These are not just linked data streams from diverse modalities, but are semantically integrated and interpreted representations from one modality to another. The language VoxML and the resource Voxicon are presently being used to drive simulations using multiple modalities within the DARPA Communicating with Computers program. Our proposal here has been to propose this environment as a platform for representing the environment of an embodied robotic agent. The appropriateness and adequacy of such a model for actual robotic interactions has not yet been explored, but we welcome the opportunity to present the model for consideration to this audience.

Acknowledgements

This work is supported by a contract with the US Defense Advanced Research Projects Agency (DARPA), Contract W911NF-15-C-0238. Approved for Public Release, Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. We would like to thank Scott Friedman, David McDonald, Marc Verhagen, and Mark Burstein for their discussion and input on this topic. All errors and mistakes are, of course, the responsibilities of the authors.

References

Ballan, L.; Bertini, M.; Del Bimbo, A.; Seidenari, L.; and Serra, G. 2011. Event detection and recognition for semantic annotation of video. *Multimedia Tools and Applications* 51(1):279–302.

Billard, A.; Calinon, S.; Dillmann, R.; and Schaal, S. 2008. Robot programming by demonstration. In *Springer handbook of robotics*. Springer. 1371–1394.

Bohg, J., and Kragic, D. 2009. Grasping familiar objects using shape context. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, 1–6. IEEE.

Cangelosi, A. 2010. Grounding language in action and perception: from cognitive agents to humanoid robots. *Physics of life reviews* 7(2):139–151.

Chang, A.; Monroe, W.; Savva, M.; Potts, C.; and Manning, C. D. 2015. Text to 3d scene generation with rich lexical grounding. *arXiv preprint arXiv:1505.06289*.

Coyne, B., and Sproat, R. 2001. Wordseye: an automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 487–496. ACM.

Do, T.; Krishnaswamy, N.; and Pustejovsky, J. 2016. Ecat: Event capture annotation tool. *Proceedings of ISA-12: International Workshop on Semantic Annotation*.

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.

Gibson, J. J.; Reed, E. S.; and Jones, R. 1982. *Reasons for realism: Selected essays of James J. Gibson*. Lawrence Erlbaum Associates.

Goldstone, W. 2009. *Unity Game Development Essentials*. Packt Publishing Ltd.

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.

Ide, N. 2013. An open linguistic infrastructure for annotated corpora. In *The People's Web Meets NLP*. Springer. 265–285.

Krishnaswamy, N., and Pustejovsky, J. 2016. Multimodal semantic simulations of linguistically underspecified motion events. *Proceedings of Spatial Cognition*.

Mani, I., and Pustejovsky, J. 2012. *Interpreting Motion: Grounded Representations for Spatial Language*. Oxford University Press.

McDonald, D., and Pustejovsky, J. 2014. On the representation of inferences and their lexicalization. In *Advances in Cognitive Systems*, volume 3.

Pustejovsky, J., and Krishnaswamy, N. 2014. Generating simulations of motion events from verbal descriptions. *Lexical and Computational Semantics (* SEM 2014)* 99.

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., and Krishnaswamy, N. forthcoming. Envisioning language: The semantics of multimodal simulations.

Pustejovsky, J., and Moszkowicz, J. 2011a. The qualitative spatial dynamics of motion. *The Journal of Spatial Cognition and Computation*.

Pustejovsky, J., and Moszkowicz, J. L. 2011b. The qualitative spatial dynamics of motion in language. *Spatial Cognition & Computation* 11(1):15–44.

Pustejovsky, J. 1995. *The Generative Lexicon*. Cambridge, MA: MIT Press.

Pustejovsky, J. 2013a. 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. 2013b. Where things happen: On the semantics of event localization. In *Proceedings of ISA-9: International Workshop on Semantic Annotation*.

Randell, D.; Cui, Z.; and Cohn, A. 1992. A spatial logic based on regions and connections. In Kaufmann, M., ed., *Proceedings of the 3rd Internation Conference on Knowledge Representation and Reasoning*, 165–176.

Siskind, J. M. 2001. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *J. Artif. Intell. Res.(JAIR)* 15:31–90.