# Object Embodiment

## in a Multimodal Simulation

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

## Overview

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 requirements on a visual simulation include the following components:

- A minimal embedding space (MES) for the simulation must be determined. This is the 3D region within which 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.

We believe that simulation can play a crucial role in communication. 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. 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

Following Generative Lexicon (GL) (Pustejovsky, 1995), lexical entries in the object language are given a feature structure consisting of a word's basic type, its parameter listing, its event typing, and its qualia structure. The semantics of an object will consist of the following:

- Atomic Structure (FORMAL ): objects expressed as basic nominal types
- Subatomic Structure (CONST ): mereotopological structure of objects
- Event Structure (TELIC and AGENTIVE ): origin and functions associated with an object
- Macro Object Structure : how objects fit together in space and through coordinated activities.

Objects can be partially contextualized through their qualia structure: a food item has a TELIC value of eat, an instrument for writing, a TELIC of write, a cup, a TELIC of hold, and so forth. For example, the lexical semantics for the noun *chair* carries a TELIC value of sit in:

$$\left[\begin{array}{l}\textbf{chair}\\ \text{AS} = \left[\text{ARG1} = x:e\right]\\ \text{QS} = \left[\begin{array}{l}\text{F} = phys(x)\\ \text{T} = \lambda z, e[sit\_in(e,z,x)]\end{array}\right]\end{array}\right]$$

The habitat for an object is built by first placing it within an *embedding space* and then contextualizing it. For example, in order to use a table, the top has to be oriented upward, the surface must be accessible, and so on. A chair must also be oriented up, the seat must be free and accessible, it must be able to support the user, etc.

$$\lambda x\left[\begin{array}{l}\textbf{chair}_{hab}\\ \text{F} = [phys(x), on(x,y_1), in(x,y_2), orient(x,up)]\\ \text{C} = [seat(x_1), back(x_2), legs(x_3), clear(x_1)]\\ \text{T} = \lambda z\lambda e[\mathcal{C} \rightarrow [sit(e,z,x)]\mathcal{R}_{sit}(x)]\\ \text{A} = [made(e',w,x)]\end{array}\right]$$

## VoxML: A Language for Concept Visualization

Experience and world knowledge can inform predicative force of a lexeme in language, but are usually left out of such encodings. We developed VoxML (Pustejovsky and Krishnaswamy 2016) to construct 3D visualizations of natural language expressions, focused on the visual instantiation a lexeme, a "voxeme."

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) program (Pustejovsky and Moszkowicz 2011a), or attribute properties.

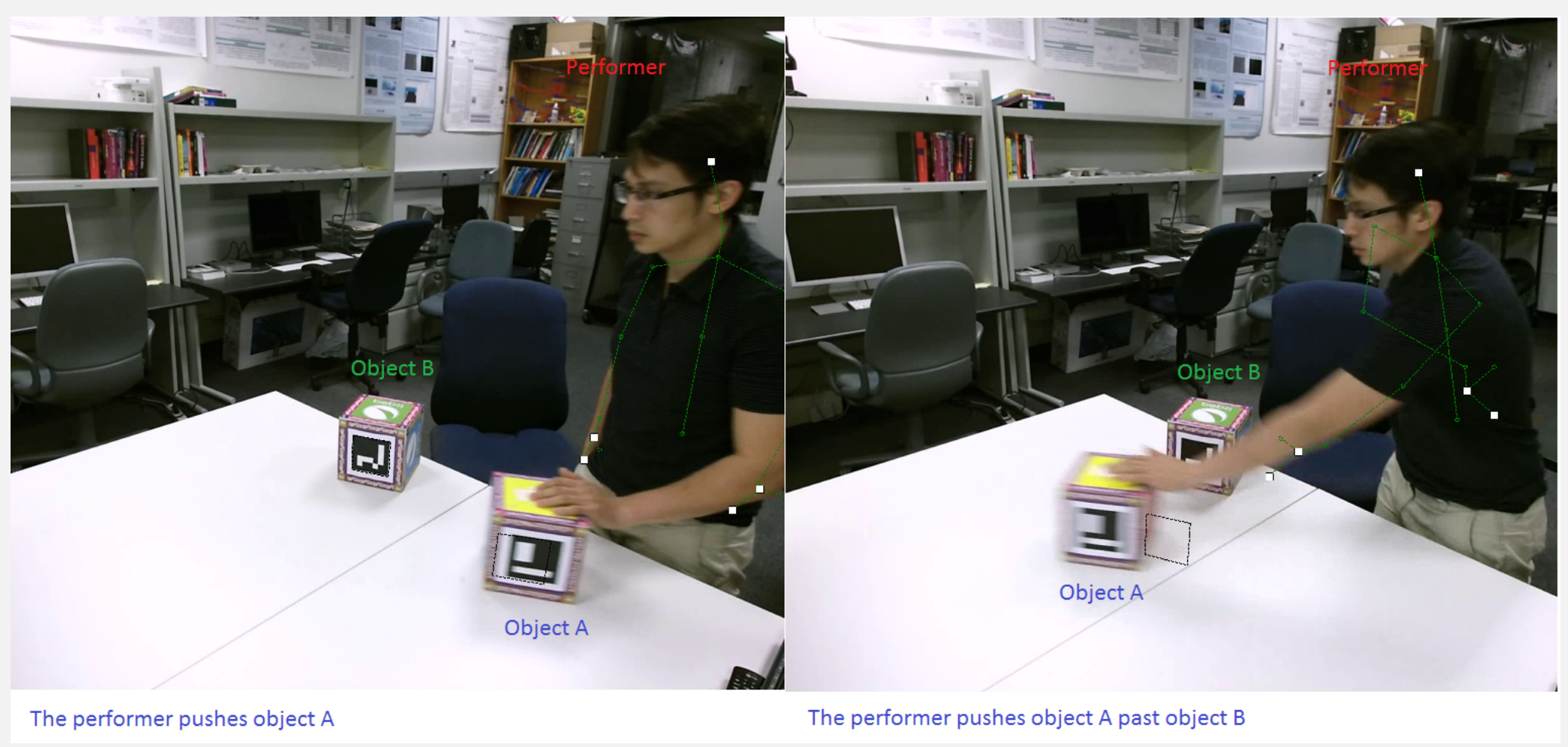
- Objects are logical constants;
- Programs are n-ary predicates that can take objects or other evaluated predicates as arguments;
- Logical types can be divided into attributes, relations, and functions, all predicates which take objects as arguments;
  - Attributes and relations evaluate to states, and
  - functions evaluate to geometric regions.

These entities can then compose into visualizations of natural language concepts and expressions. **VoxSim** is the implementation built on top of VoxML.

## Learning Events from Motion Data

VoxML can also be used to help detect and recognize events and actions in video. In order to do so, our lab is creating a dataset of videos annotated with event subevent relations using ECAT (Do, et al. 2016), an internally-developed video annotation tool. This 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.

In our dataset, 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.

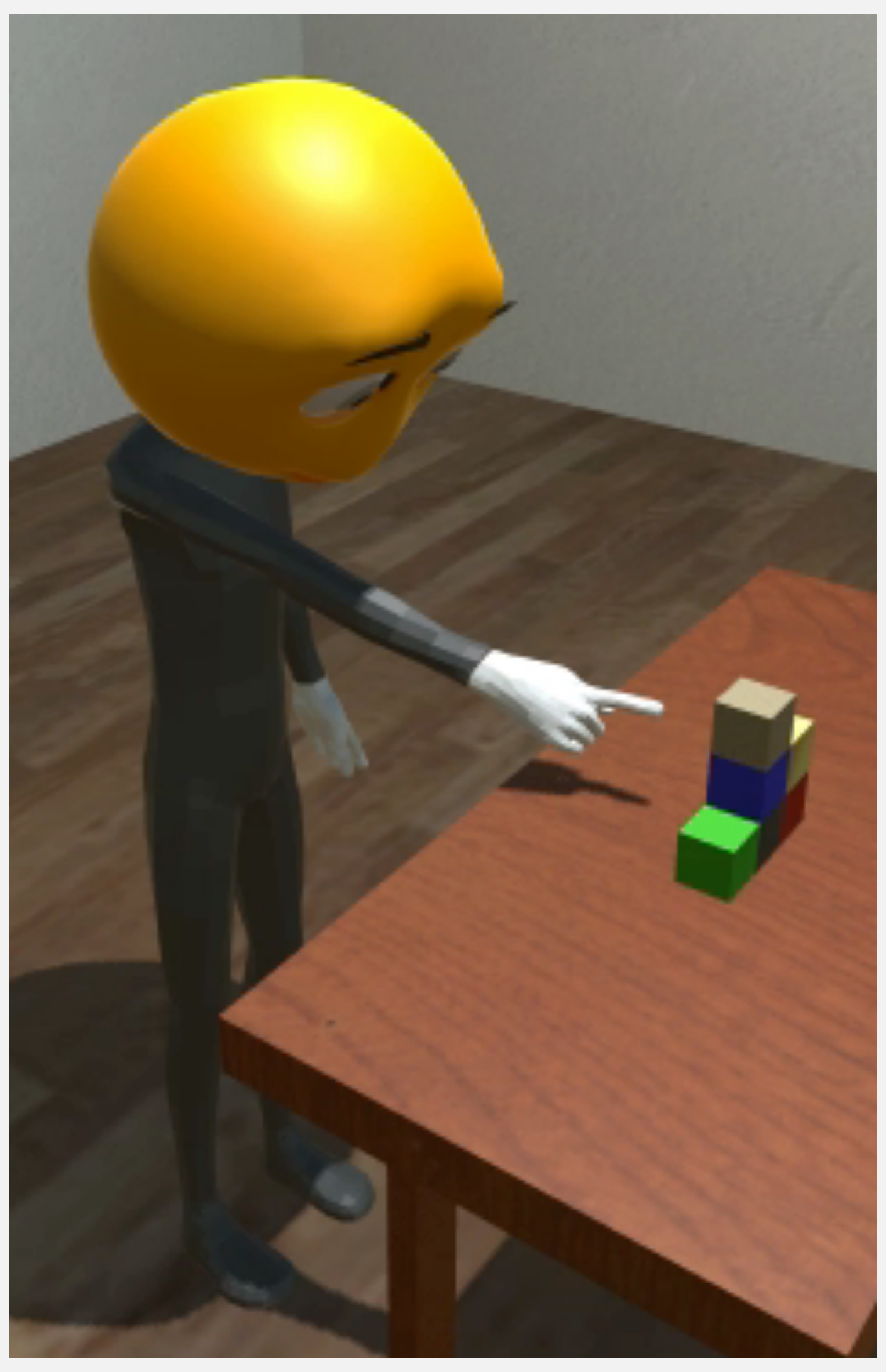


The performer pushes object A

The performer pushes object A past object B

## Gesture-Speech Act Mapping

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. 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.



## Examples of Voxemes

$$\left[\begin{array}{l}\textbf{wall}\\ \text{LEX} = \left[\begin{array}{l}\text{PRED} = \textbf{wall}\\ \text{TYPE} = \textbf{physobj}\end{array}\right]\\ \text{TYPE} = \left[\begin{array}{l}\text{HEAD} = \textbf{rectangular\_prism}\\ \text{COMPONENTS} = \textbf{nil}\\ \text{CONCAVITY} = \textbf{flat}\\ \text{ROTATSYM} = \{X,Y,Z\}\\ \text{REFLECTSYM} = \{XY,XZ,YZ\}\end{array}\right]\\ \text{HABITAT} = \left[\begin{array}{l}\text{INTR} = \left[\begin{array}{l}\text{UP} = align(Y,\mathcal{E}_Y)\\ \text{FRONT} = front(+Z)\\ \text{CONSTR} = Z \ll Y, Z \ll X\end{array}\right]\\ \text{EXTR} = ...\end{array}\right]\\ \text{AFFORD\_STR} = \left[\begin{array}{l}\text{A}_1 = H_1 \rightarrow [E_1]R\\ \text{A}_2 = \textbf{...}\\ \text{A}_3 = \textbf{...}\end{array}\right]\\ \text{EMBODIMENT} = \left[\begin{array}{l}\text{SCALE} = \textbf{>agent}\\ \text{MOVABLE} = \textbf{false}\end{array}\right]\end{array}\right]$$

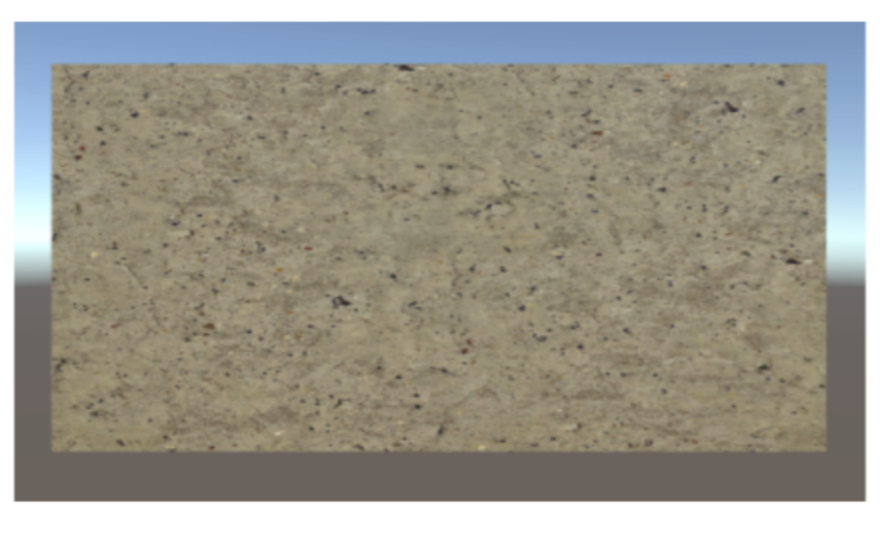


Figure 1: Wall voxeme instance

$$\left[\begin{array}{l}\textbf{table}\\ \text{LEX} = \left[\begin{array}{l}\text{PRED} = \textbf{table}\\ \text{TYPE} = \textbf{physobj}\end{array}\right]\\ \text{TYPE} = \left[\begin{array}{l}\text{HEAD} = \textbf{sheet [1]}\\ \text{COMPONENTS} = \textbf{surface [1], leg+}\\ \text{CONCAVITY} = \textbf{flat}\\ \text{ROTATSYM} = \{Y\}\\ \text{REFLECTSYM} = \{XY,YZ\}\end{array}\right]\\ \text{HABITAT} = \left[\begin{array}{l}\text{INTR} = \left[\begin{array}{l}\text{UP} = align(Y,\mathcal{E}_Y)\\ \text{TOP} = top(+Y)\end{array}\right]\\ \text{EXTR} = ...\end{array}\right]\\ \text{AFFORD\_STR} = \left[\begin{array}{l}\text{A}_1 = H_1 \rightarrow [E_1]R\\ \text{A}_2 = \textbf{...}\\ \text{A}_3 = \textbf{...}\end{array}\right]\\ \text{EMBODIMENT} = \left[\begin{array}{l}\text{SCALE} = \textbf{agent}\\ \text{MOVABLE} = \textbf{true}\end{array}\right]\end{array}\right]$$

Figure 2: Table voxeme instance

$$\left[\begin{array}{l}\textbf{slide}\\ \text{LEX} = \left[\begin{array}{l}\text{PRED} = \textbf{slide}\\ \text{TYPE} = \textbf{process}\end{array}\right]\\ \text{TYPE} = \left[\begin{array}{l}\text{HEAD} = \textbf{process}\\ \text{ARGS} = \left[\begin{array}{l}\text{A}_1 = \textbf{x:physobj}\\ \text{A}_2 = \textbf{y:physobj}\end{array}\right]\\ \text{BODY} = \left[\begin{array}{l}\text{E}_1 = while(EC(x,y),\\ \qquad move(x))\end{array}\right]\end{array}\right]\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{plate}\\ \text{LEX} = \left[\begin{array}{l}\text{PRED} = \textbf{plate}\\ \text{TYPE} = \textbf{physobj}\end{array}\right]\\ \text{TYPE} = \left[\begin{array}{l}\text{HEAD} = \textbf{sheet}\\ \text{COMPONENTS} = \textbf{surface, base}\\ \text{CONCAVITY} = \textbf{concave}\\ \text{ROTATSYM} = \{Y\}\\ \text{REFLECTSYM} = \{XY,YZ\}\end{array}\right]\\ \text{HABITAT} = \left[\begin{array}{l}\text{INTR} = {}_{[1]}\left[\begin{array}{l}\text{UP} = align(Y,\mathcal{E}_Y)\\ \text{TOP} = top(+Y)\end{array}\right]\\ \text{EXTR} = ...\end{array}\right]\\ \text{AFFORD\_STR} = \left[\begin{array}{l}\text{A}_1 = H[1] \rightarrow [put(x,y)]hold(y,x)\\ \text{A}_2 = \textbf{...}\\ \text{A}_3 = \textbf{...}\end{array}\right]\\ \text{EMBODIMENT} = \left[\begin{array}{l}\text{SCALE} = \textbf{< agent}\\ \text{MOVABLE} = \textbf{true}\end{array}\right]\end{array}\right]$$

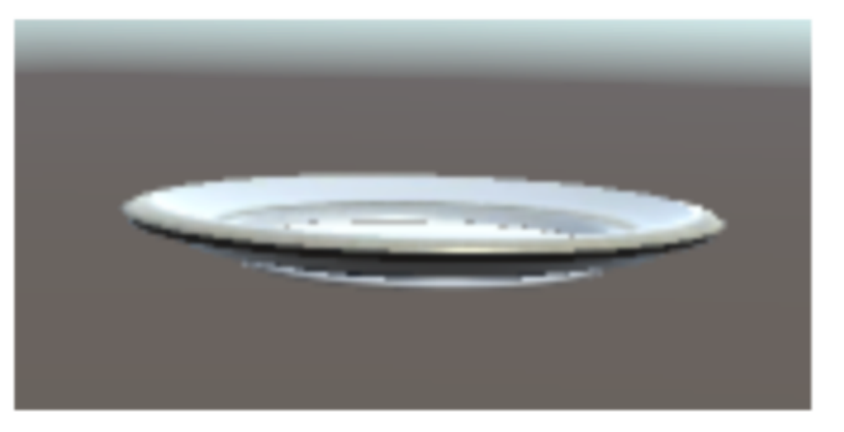


Figure 3: Plate voxeme instance

$$\left[\begin{array}{l}\textbf{top}\\ \text{LEX} = \left[\text{PRED} = \textbf{top}\right]\\ \text{TYPE} = \left[\begin{array}{l}\text{ARG} = \textbf{x:physobj}\\ \text{REFERENT} = \textbf{x}\rightarrow\text{HEAD}\\ \text{MAPPING} = \textbf{dimension(n):n-1}\\ \text{ORIENTATION} = \left[\begin{array}{l}\text{SPACE} = \textbf{world}\\ \text{AXIS} = \textbf{+Y}\\ \text{ARITY} = \textbf{x}\rightarrow\text{HABITAT}\rightarrow\\ \qquad \text{INTR}[top(axis)]:\\ \qquad intransitive\end{array}\right]\end{array}\right]\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{put}\\ \text{LEX} = \left[\begin{array}{l}\text{PRED} = \textbf{put}\\ \text{TYPE} = \textbf{transition\_event}\end{array}\right]\\ \text{TYPE} = \left[\begin{array}{l}\text{HEAD} = \textbf{transition}\\ \text{ARGS} = \left[\begin{array}{l}\text{A}_1 = \textbf{x:agent}\\ \text{A}_2 = \textbf{y:physobj}\\ \text{A}_3 = \textbf{z:location}\end{array}\right]\\ \text{BODY} = \left[\begin{array}{l}\text{E}_1 = grasp(x,y)\\ \text{E}_2 = [while(\\ \qquad hold(x,y),\\ \qquad move(x)]\\ \text{E}_3 = [at(y,z) \rightarrow\\ \qquad ungrasp(x,y)]\end{array}\right]\end{array}\right]\end{array}\right]$$

## VoxSim: A Visual Platform for Modeling Motion Language

VoxSim (Krishnaswamy and Pustejovsky 2016), is a semantically-informed visual event simulator built on top of the Unity game engine (Goldstone 2009). 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.

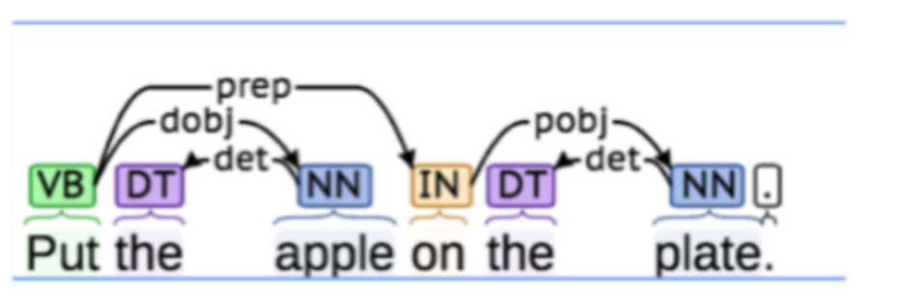


Figure 5: Dependency parse

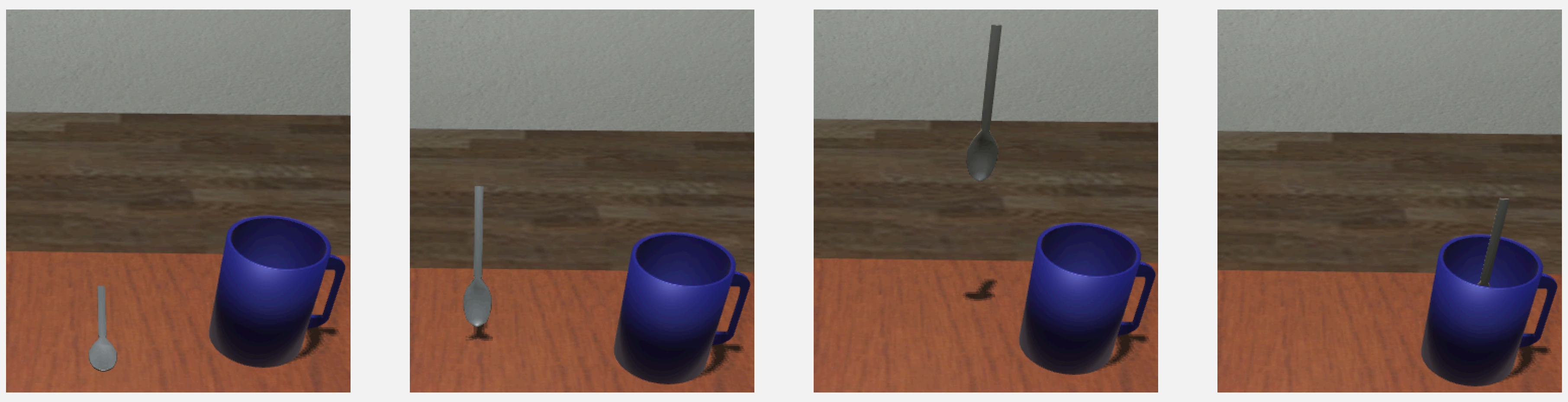
1. *pred := put(x,y)*
2. *x := apple*
3. *y := on(z)*
4. *z := plate*

*put(apple,on(plate))*

Figure 6: Transformation to logical form

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, meaningthat 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. The contained object must then be transformed to satisfy equivalent constraints, such as turning a spoon to sit in a cup (as shown below). If no such transformation is possible, VoxSim returns an error message.



## 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.