



Embodied Human Computer Interaction

James Pustejovsky¹ · Nikhil Krishnaswamy²

Received: 7 December 2020 / Accepted: 11 May 2021

© Gesellschaft für Informatik e.V. and Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

In this paper, we argue that embodiment can play an important role in the design and modeling of systems developed for Human Computer Interaction. To this end, we describe a simulation platform for building Embodied Human Computer Interactions (EHCI). This system, VoxWorld, enables multimodal dialogue systems that communicate through language, gesture, action, facial expressions, and gaze tracking, in the context of task-oriented interactions. A multimodal simulation is an embodied 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 a discourse. It is built on the modeling language VoxML (Pustejovsky and Krishnaswamy in VoxML: a visualization modeling language, proceedings of LREC, 2016), which encodes objects with rich semantic typing and action affordances, and actions themselves as multimodal programs, enabling contextually salient inferences and decisions in the environment. VoxWorld enables an embodied HCI by situating both human and artificial agents within the same virtual simulation environment, where they share perceptual and epistemic common ground. We discuss the formal and computational underpinnings of embodiment and common ground, how they interact and specify parameters of the interaction between humans and artificial agents, and demonstrate behaviors and types of interactions on different classes of artificial agents.

Keywords Multimodal embodiment · Simulation · Artificial agent · Situated grounding

1 Introduction

One of the most persistent and challenging problems facing both the areas of Human–Computer Interaction (HCI) and Human-Robot Interaction (HRI) involves communicating intentions, goals, and attitudes through multiple modalities beyond language, including gesture, gaze, facial expressions, and situational awareness [14, 32, 51, 64, 84, 98]. For example, within the HRI community, there has been a growing interest in how to contextually resolve ambiguities that may arise from communication in situated dialogues, ranging

from discussions on how HRI dialogues should be designed [31, 57, 85], how perception and grounding can be integrated into language understanding [58, 67], to more recent work on task-oriented dialogues [90]. This is the problem of identifying and modifying the *common ground* between speakers [2, 17, 88, 91]. It has long been recognized that a linguistic utterance’s meaning is subject to contextualized interpretation; but this is also the case with gestures in task-oriented dialogues. Depending on the situation, for example, an oriented hand gesture could refer either to an action request (“move it”) or a dismissive response (“forget it”) [101]. A pointing gesture might designate a specific object, a location, or a direction, as illustrated in Fig. 1. Even a request for action can be underspecified, denoting either a continuous movement or a movement to a specific location.

Similarly, depending on the situation, the definite description in the command “Open the box.” may uniquely refer or not, depending on how many boxes are in the context. These and similar miscommunications or the need for clarification in dialogue have been called *situated grounding problems* [63], and can be viewed as problematic in a model that appeals to and encodes both a visual modality

This work was supported by the US Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO) under contract #W911NF-15-C-0238 at Brandeis University. It was first presented in [79], on which this discussion is based.

✉ James Pustejovsky
jamesp@brandeis.edu
Nikhil Krishnaswamy
nkrishna@colostate.edu

¹ Brandeis University, Waltham, MA, USA

² Colorado State University, Fort Collins, CO, USA

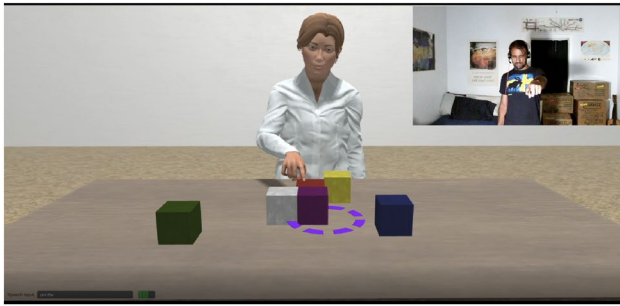


Fig. 1 Diana, an Embodied Artificial Agent, engaging in an embodied HCI with a human user. The purple circle around the red block shows where the user (top right panel) is pointing



Fig. 2 Mother and child baking

and situational information into the dialogue state. What the occurrence of these issues makes apparent is the complexity underlying the interpretation of referential expressions in actual situated dialogues. The richness provided by situationally grounding computer or robot behaviors brings to the surface interpretive questions similar to those of a human in the same scenario.

We will argue that natural human–computer interactions involving intelligent virtual agents (IVAs) require not only that the artificial agent itself be embodied, but that the entire *interaction* between the human and the IVA must be embodied, in order to fully establish the common ground that both agents share to communicate fluently. We will refer to this as an *embodied Human–Computer Interaction*, and develop this idea below.

In typical task-oriented interactions between humans, (as shown in Fig. 2), actions, gesture, and language are situated within a common ground. For such situations, the common ground includes the following characteristics:

- *Co-situatedness* and *co-perception* of the agents, such that they can interpret the same situation from their respective frames of reference.
- *Co-attention* of shared situated references, allowing richer expressiveness in referring to the environment (i.e., using 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.
- *Co-intent* or agreement of the common goals in a dialogue. It is important to recognize the intent of other agents, to facilitate the interpretation of their expressions.

In order to achieve these goals, human–computer/robot interactions require robust recognition and generation of expressions through multiple modalities (language, gesture, vision, action), and the encoding of **situated meaning**: this entails three aspects of common ground interpretation: (a) the situated *grounding* of expressions in context; (b) an interpretation of the expression contextualized to the *dynamics* of the discourse; and (c) an appreciation of the *actions and consequences* associated with objects in the environment.

With this in mind, many HCI researchers have adopted the notion of “embodiment” in order to better understand user expectations when interacting with artificial agents. Embodied agents or avatars add new dimensions to human–agent interactions compared to voice- or text-only conversational artificial agents. Embodied agents can express emotions and perform gestures, two crucial non-verbal modes of human communication. Potentially, this enables such artificial agents to have more human-like, peer-to-peer interactions with users. Unfortunately, embodiment alone does not avoid some of the key limitations of conversational artificial agents. Even embedded in an avatar, most artificial agents won’t know what you are pointing at. As with verbal conversations, visual communication mechanisms like gestures, expressions, and body language need to be two-way.

In this paper, we present a model of embodiment that is implemented in VoxWorld, a simulation platform for modeling and building embodied Human–Computer Interactions. We describe the formal and computational properties of VoxWorld, including how it encodes and deploys embodiment centrally in interpreting dialogue, intentions, goals, and attitudes, and in executing actions in a situated context. We also provide examples of how differently-embodied artificial agents can use the same underlying formal and computational model to reason about and execute different behaviors based on the particulars of their respective embodiments. Throughout the paper, we discuss how the features of our model of embodiment compare to related approaches modeling situated grounding and perception, such as that taken

in [26], with the application of Type Theory with Records (TTR) [19, 25].

2 The Meaning of Embodiment

There are many dimensions to the meaning of “embodiment” in cognitive psychology, AI, and communication [1, 15, 48, 64, 87]. In the context of human–computer and human–robot interaction, there are at least three aspects of embodiment for an artificial agent, beyond the conventional agent architecture entailing belief and intent, that we believe are crucial for effective communication. These include representations for the following factors:¹

- *Self-embodiment* of the artificial agent: it is registered with a “spatial presence” within the domain or space of the interaction (the embedding space); it has a skeletal form, explicit effectors for action, and explicit sensors for audio and visual input². Constraints on its behavior are imposed by the physical extents and limitations of the embodiment (e.g., how far it can reach, degrees of freedom on the joints, etc.).
- The *human’s embodiment*; recognition of the partner’s linguistic and gestural expressions, facial expressions, and actions. Just as the human perceives the artificial agent’s embodiment and real and virtual space and interprets this cognitively, the artificial agent continuously receives inputs through which it constructs and maintains a representation of its human partner’s embodiment.
- *Situated meaning* for the objects and actions in the environment; an elementary understanding of how objects behave relative to each other and as a consequence of the agent’s actions (affordances, action dynamics, etc).

- a. **Atomic Structure** (*formal*) : objects expressed as basic nominal types
- b. **Subatomic Structure** (*const*) : mereotopological structure of objects
- c. **Event Structure** (*telic and agentive*) : origin and functions associated with an object
- d. **Macro Object Structure** : how objects fit together in space and through coordinated activities.

This may also extend to notions like frame of reference and an agent’s knowledge (or lack thereof) that its interpretation of certain propositions or predicates (e.g., “ x is left of y ”) may differ from that of other agents, based on factors like frame of reference. For instance, if a human says “left” and points to their own right, that should signal to the artificial agent that the human has adopted the artificial agent’s egocentric frame of reference.

¹ This recalls the question of how to best model *situated action* [16, 97].

² See Sect. 5 for details on integrating various sensor types and their relationships with the particulars of the artificial agent’s embodiment.

When these conditions are present in the artificial agent, along with the properties of the common ground discussed above, we will say that an interaction between an artificial agent and a human is “embodied.”

Not part of these conditions but related is a notion of *situated pseudo-embodiment* where the artificial agent has no physical embodiment but is represented by a dynamic point of view or camera in virtual space. This may be a floating first-person camera that can move and turn and is hence limited in its ability to interact with the environment but can present to the human interlocutor what it sees and can conduct certain reasoning about the environment. For obvious reasons, this is only possible with virtual artificial agents although can be approximated by physical agents in certain situations (see Sect. 11).

3 Semantic Typing and Qualia Structure

A significant part of any model for situated communication is an encoding of the semantic type, functions, purposes, and uses introduced by the objects under discussion. For example, a semantic model of perceived *object teleology*, as introduced by Qualia Structure, for example [73], as well as *object affordances* [34] is needed to help ground expression meaning to speaker intent.

Let us assume, following Generative Lexicon (GL) [73] and other type-driven semantic approaches [3, 18, 49], that 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:

Objects can be partially contextualized through their *qualia structure* [77]. Each Qualia role can be seen as answering a specific question about the object it is associated with:

- *Formal*: encoding taxonomic information about the lexical item (the *is-a* relation);
- *Constitutive*: encoding information on the parts and constitution of an object (*part-of* or *made-of* relation);
- *Telic*: encoding information on purpose and function (the *used-for* or *functions-as* relation);
- *Agentive*: encoding information about the origin of the object (the *created-by* relation).

Taken together, the answers to these questions can help elucidate the meanings of words in the language. We can view the qualia structure of a lexical item, α , as the four features below in (2), where F=FORMAL, C=CONST, T=TELIC and A=AGENTIVE.

$$\left[\begin{array}{l} \alpha \\ \text{QUALIA} = \left[\begin{array}{l} \text{F} = \text{what } \alpha \text{ is} \\ \text{C} = \text{what } \alpha \text{ is made of} \\ \text{T} = \text{function of } \alpha \\ \text{A} = \text{origin of } \alpha \end{array} \right] \end{array} \right] \quad (2)$$

Objects under discussion in discourse (cf. [36]) can be partially contextualized through their semantic type and 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 GL lexical semantics for the noun *chair* in (3) carries a TELIC value of *sit_in*, while the concept of *letter* in (4) carries a TELIC value of *read* and an AGENTIVE value of *write*.³

$$\lambda x \left[\begin{array}{l} \text{chair} \\ \text{AS} = \left[\text{ARG1} = x : e \right] \\ \text{QS} = \left[\begin{array}{l} \text{F} = \text{phys}(x) \\ \text{T} = \lambda z, e [\text{sit_in}(e, z, x)] \end{array} \right] \end{array} \right] \quad (3)$$

$$\lambda x \left[\begin{array}{l} \text{letter} \\ \text{AS} = \left[\text{ARG1} = x : e \right] \\ \text{QS} = \left[\begin{array}{l} \text{F} = \text{phys}(x) \\ \text{T} = \lambda z, e_2 [\text{read}(e_2, z, x)] \\ \text{A} = \lambda w, e_1 [\text{write}(e_1, w, x)] \end{array} \right] \end{array} \right] \quad (4)$$

However, while an artifact may be designed for a specific purpose, this can only be achieved under specific circumstances. To account for this context-dependence, [74] enriches the lexical semantics of words denoting artifacts (the TELIC role specifically) by introducing the notion of an object's *habitat*, which encodes these circumstances. For example, an object, x , within the appropriate habitat (or context) \mathcal{C} , performing the action π will result in the intended or desired resulting state, \mathcal{R} , i.e., $\mathcal{C} \rightarrow [\pi]\mathcal{R}$. That is, if the habitat \mathcal{C} (a set of contextual factors) is satisfied, then every time the activity of π is performed, the resulting state \mathcal{R} will occur. It is necessary to specify the precondition context \mathcal{C} , since this enables the local modality to be satisfied. An illustration of what the resulting knowledge structure for the habitat of a chair is shown in the QS entry below.

$$\lambda C \lambda x \left[\begin{array}{l} \text{chair} \\ \text{F} = [\text{phys}(x), \text{on}(x, y_1), \text{in}(x, y_2), \text{orient}(x, up)] \\ \text{C} = [\text{seat}(x_1), \text{back}(x_2), \text{legs}(x_3), \text{clear}(x_1)] \\ \text{T} = \lambda z \lambda e [\mathcal{C} \rightarrow [\text{sit}(e, z, x)] \mathcal{R}_{\text{sit}}(x)] \\ \text{A} = [\text{made}(e', w, x)] \end{array} \right] \quad (5)$$

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.

The embedding space for an activity is meant to delineate the dynamic spatial region referenced when an agent is engaged in any self or joint interaction in its environment. Since it is functionally as well as spatially defined [35, 78], it shares aspects with Coventry's work on functional relations in space [21]. While embedding space is unique to our approach, it is possible to relate it to aspects of spatial description within the TTR model, as developed in [25]. Assume we identify the region defined by the set of constraints on the "spatial templates" associated with each agent in an interaction. This would correspond to the dependent type *RegionObject*, which includes both the interlocutors and the salient 3D convex hull including the objects under discussion.

4 VoxML and Situated Meaning

4.1 Objects and Affordances

The notion of habitat described above and the attached behaviors that are associated with an object are further developed in [78], where an explicit connection to Gibson's ecological psychology is made [35], along with a direct encoding of the *affordance structure* for the object [34]. The affordance structure available to an agent, when presented with an object, is the set of actions that can be performed with it. We refer to these as GIBSONIAN affordances, and they include "grasp", "move", "hold", "turn", etc. This is to distinguish them from more goal-directed, intentionally situated activities, what we call TELIC affordances.

Extending this notion, we define a habitat as a representation of an object situated within a simulation, a partial minimal model [12, 50, 53]; in this sense, it is a directed enhancement of the qualia structure. Multi-dimensional affordances determine how habitats are deployed and how they modify or augment the context, and compositional operations include procedural (simulation) and operational (selection, specification, refinement) knowledge.

The language used to construct this simulation is called VoxML (Visual Object Concept Modeling Language) [78]. VoxML is a modeling language for constructing 3D visualizations of concepts denoted by natural language expressions, and is being used as the platform for creating multimodal semantic simulations in the context of human-computer and human-robot communication [54]. It adopts the basic semantic typing for objects and properties from Generative Lexicon and the dynamic interpretation of event structure

³ AS = argument structure; QS = qualia structure.

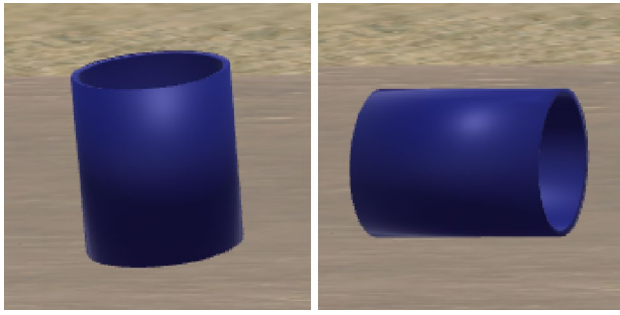


Fig. 3 Cup in different habitats allowing sliding and holding (left) and rolling (right)

developed in [80], along with a continuation-based dynamic interpretation for both sentence and discourse composition [5, 9, 23].

VoxML forms the scaffolding we use to encode knowledge about objects, events, attributes, and functions by linking lexemes to their visual instantiations, termed the “visual object concept” or *voxeme*.

Entities modeled in VoxML can be OBJECTS, programs, or logical types. 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. For example, the attributes associated with objects such as *cup*, *chair*, and *block*, include the following:

LEX	OBJECT’s lexical information
TYPE	OBJECT’s geometrical typing
HABITAT	OBJECT’s habitat for actions
AFFORD_STR	OBJECT’s affordance structure
EMBODIMENT	OBJECT’s agent-relative embodiment

The LEX attribute contains the subcomponents PRED, the predicate lexeme denoting the object, and TYPE, the object’s type according to Generative Lexicon.

Voxemes representing humans or IVAs are lexically typed as *agents*, but artificial agents, due to their embodiments, ultimately inherit from physical objects and so fall under objects in the taxonomy. 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]]⁴,

⁴ Beginning in [52], voxemes have been denoted [[VOXEME]].

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’s voxeme structure provides *habitats*, which are situational contexts or environments conditioning the object’s *affordances*, which may be either “Gibsonian” affordances [34] or “Telic” affordances [73, 74]. 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.⁵

For example, the object geometry for the concept [[CUP]], along with the constraints on symmetry, is illustrated below.

$$\left[\begin{array}{l} \text{cup} \\ \text{TYPE} = \left[\begin{array}{l} \text{HEAD} = \text{cylindroid}[1] \\ \text{COMPONENTS} = \text{surface, interior} \\ \text{CONCAVITY} = \text{concave} \\ \text{ROTATIONAL_SYMMETRY} = \{Y\} \\ \text{REFLECTION_SYMMETRY} = \{XY, YZ\} \end{array} \right] \end{array} \right] \quad (6)$$

Consider now the various habitats identified with [[CUP]].

$$\left[\begin{array}{l} \text{cup} \\ \text{HABITAT} = \left[\begin{array}{l} \text{INTRINSIC} = [2] \left[\begin{array}{l} \text{CONSTR} = \{Y > X, Y > Z\} \\ \text{UP} = \text{align}(Y, \mathcal{E}_Y) \\ \text{TOP} = \text{top}(+Y) \end{array} \right] \\ \text{EXTRINSIC} = [3] \left[\text{UP} = \text{align}(Y, \mathcal{E}_{\perp Y}) \right] \end{array} \right] \end{array} \right] \quad (7)$$

Finally, given these habitats, we can identify the associated behaviors that are enabled (afforded) in such situations:

$$\left[\begin{array}{l} \text{cup} \\ \text{AFF_STR} = \left[\begin{array}{l} A_1 = H[2] \rightarrow [\text{put}(x, \text{on}([1]))] \text{support}([1], x) \\ A_2 = H[2] \rightarrow [\text{put}(x, \text{in}([1]))] \text{contain}([1], x) \\ A_3 = H[2] \rightarrow [\text{grasp}(x, [1])] \text{hold}(x, [1]) \\ A_4 = H[3] \rightarrow [\text{roll}(x, [1])] \mathcal{R} \end{array} \right] \end{array} \right] \quad (8)$$

Indeed, object properties and the events they facilitate are a primary component of situational context. In Fig. 3, we understand that the cup in the orientation shown can be *rolled* by a human. Were it not in this orientation, it

⁵ It should be noted that Gibsonian affordances might be construed as the goal of an activity in some contexts.

might be able to be only *slid* across its supporting surface (cf. (9)).

This voxeme for `[[CUP]]` gives the object appropriate lexical predicate and typing (a *cup* is a `PHYSICAL OBJECT` and an `ARTIFACT`). It denotes that the cup is roughly cylindrical and concave, has a surface and an interior, is symmetrical around the Y-axis and across associated planes (VoxML adopts 3D graphics conventions, where the Y-axis is vertical), and is smaller than and movable by the artificial agent. The remainder of VoxML typing structure is devoted to habitat and affordance structures, which we discuss below.

$$\begin{aligned} \text{cup} \\ \text{LEXICAL} = & \left[\begin{array}{l} \text{PREDICATE} = \text{cup} \\ \text{TYPE} = \text{physobj} \bullet \text{artifact} \end{array} \right] \\ \text{TYPE} = & \left[\begin{array}{l} \text{HEAD} = \text{cylindroid}[1] \\ \text{COMPONENTS} = \text{surface, interior} \\ \text{CONCAVITY} = \text{concave} \\ \text{ROTATIONAL_SYMMETRY} = \{Y\} \\ \text{REFLECTION_SYMMETRY} = \{XY, YZ\} \end{array} \right] \\ \text{HABITAT} = & \left[\begin{array}{l} \text{INTRINSIC} = [2] \left[\begin{array}{l} \text{CONSTR} = \{Y > X, Y > Z\} \\ \text{UP} = \text{align}(Y, \mathcal{E}_Y) \\ \text{TOP} = \text{top}(+Y) \end{array} \right] \\ \text{EXTRINSIC} = [3] \left[\text{UP} = \text{align}(Y, \mathcal{E}_{\perp Y}) \right] \end{array} \right] \\ \text{AFF_STR} = & \left[\begin{array}{l} A_1 = H_{[2]} \rightarrow [\text{put}(x, \text{on}([1]))] \text{support}([1], x) \\ A_2 = H_{[2]} \rightarrow [\text{put}(x, \text{in}([1]))] \text{contain}([1], x) \\ A_3 = H_{[2]} \rightarrow [\text{grasp}(x, [1])] \text{hold}(x, [1]) \\ A_4 = H_{[3]} \rightarrow [\text{roll}(x, [1])] \mathcal{R} \end{array} \right] \\ \text{EMBOD} = & \left[\begin{array}{l} \text{SCALE} = \text{agent} \\ \text{MOVABLE} = \text{true} \end{array} \right] \end{aligned} \quad (9)$$

(6–8) respectively show the typing, habitat, and affordance structure of `[[CUP]]`, which are brought together in the complete VoxML encoding in (9). Bracketed numbers, (e.g., [1]) are reentrancy indices; terms annotated with the same number refer to the same entity. For instance, in habitat 2 ($H_{[2]}$), the intrinsic habitat where the cup has an upward orientation, if an agent puts some x inside the cup's cylindroid geometry ([1]), the cup contains x .

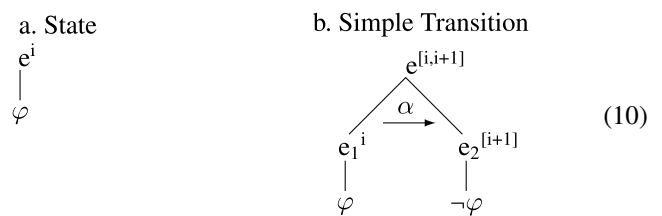
4.2 Actions and Their Consequences

VoxML treats actions and events within a dynamic event semantics as programs [62, 80]. Event structure is enriched to not only *encode* but dynamically *track* those object attributes modified in the course of the event (the location of the moving entity, the extent of a created or destroyed entity, etc.). The resulting event structure representation is called a *Dynamic Event Structure* [74]. Starting with the view that subevents of a complex event can be modeled as a sequence of states (containing formulae), a dynamic event structure explicitly labels the transitions that move an event from state to state, i.e., programs [10, 30, 70].

A dynamic approach to modeling updates makes a distinction between formulae, φ , and programs, π . A formula is interpreted as a classical propositional expression, with assignment of a truth value in a specific state in the model

[44]. For our purposes, a state is a set of propositions with assignments to individual variables at a specific frame. We can think of atomic programs as input/output relations, i.e., relations from states to states, and hence interpreted over an input/output state-state pairing. The model encodes three kinds of representations: (i) predicative **content** of a frame; (ii) **programs** that move from frame to frame; and **tests** that must be satisfied for a program to apply. These include: pre-tests, while-tests, and result-tests.

In this model, there are only two primitive event types: *states*, which are simply propositions describing a snapshot in time; and *transitions*, which are pairs of states connected by a function that moves from the first state to the second state (in some ways similar to the situation calculus representation). These two event types are illustrated in (10).



The structure in (10a) represents a **state** as a snapshot of the world in time, e^i , with the propositional content, φ . The event structure in (10b) illustrates how the program α takes the world from the state in e^i with content φ , to the adjacent state, $e_2^{[i+1]}$, where the propositional content has been negated, $\neg\varphi$. This structure corresponds directly to **achievements**. The other two Vendlerian classes can be generated from these two types:

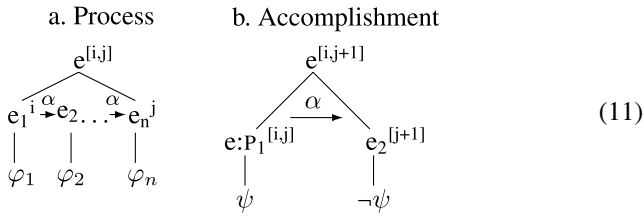
Processes can be modeled as an iteration of simple transitions, where two conditions hold: the transition is a change in the value of an identifiable attribute of the object; every iterated transition shares the same attribute being changed. This is illustrated in (11a).

Finally, **accomplishments** are built up by taking an underlying process event, $e:p$, denoting some change in an object's attribute, and synchronizing it with an achievement (simple transition): that is, $e:p$ is unfolding while ψ is true, until one last step of the program α makes it such that $\neg\psi$ is now true. This can be seen in the event structure in (11b).

The advantage of adopting a dynamic interpretation of events is that one can map linguistic expressions directly into simulations through an operational semantics [66].

Models of processes using updating typically make reference to the notion of a state transition [43]. Each event, such as *put* in (13), can be seen as a traced structure over a Labeled Transition System. The approach is similar in many respects to that developed in [30], and is integrated

into the framework of Type Theory with Records (TTR) [19] for modeling robot action control and communication [26].⁶



This approach allows the system to reason about objects and actions independently. In the case of unspecified objects, VoxSim’s parameter assignment requirement can be satisfied by a “transform object” that has no geometry but can be tracked by position and rotation, resulting in a pantomime, (or “simulation”) of the action within the simulation. 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, the system presents 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 a 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.

For the simulation to run, all parameters (e.g., object location, agent motion, etc.) must have values assigned. The simulation environment itself facilitates the calculation of these values, including a common path that the object and agent’s manipulator must follow while completing an action; adhering to these common paths and positional values keeps the two synchronized.

$$\left[\begin{array}{l} \text{grasp} \\ \text{LEX} = \left[\begin{array}{l} \text{PRED} = \text{grasp} \\ \text{TYPE} = \text{transition_event} \end{array} \right] \\ \text{TYPE} = \left[\begin{array}{l} \text{HEAD} = \text{transition} \\ \text{ARGS} = \left[\begin{array}{l} A_1 = \text{x:agent} \\ A_2 = \text{y:physobj} \end{array} \right] \\ \text{BODY} = \left[E_1 = \text{grasp}(x, y) \right] \end{array} \right] \end{array} \right] \quad (12)$$

⁶ TTR encodes actions (such as *put* and *grasp* above) as finite-state sequences of subevents (cf. [72]), but the computational effect of applying the updating functions over the current *RobotState*, given an action, are similar to our interpretation of events as state-transformers; e.g., mapping from *RobotState* to *RobotState*.

Events as Programs:

$$\left[\begin{array}{l} \text{put} \\ \text{LEX} = \left[\begin{array}{l} \text{PRED} = \text{put} \\ \text{TYPE} = \text{transition_event} \end{array} \right] \\ \text{TYPE} = \left[\begin{array}{l} \text{HEAD} = \text{transition} \\ \text{ARGS} = \left[\begin{array}{l} A_1 = \text{x:agent} \\ A_2 = \text{y:physobj} \\ A_3 = \text{z:location} \end{array} \right] \\ \text{BODY} = \left[\begin{array}{l} E_1 = \text{grasp}(x, y) \\ E_2 = [\text{while}(\text{hold}(x, y), \text{move}(x, y))] \\ E_3 = [\text{at}(y, z) \rightarrow \text{ungrasp}(x, y)] \end{array} \right] \end{array} \right] \end{array} \right] \quad (13)$$

The logic of event structure encodes minimal temporal constraints on how the subevents interact or play out. The rendering engine itself maintains a floating frame rate and regulates the time needed to conduct movements, obviating the need to regularly model this temporal aspect in operationally defined events in VoxML, although scalar attributives like *faster* or *slower* can provide temporal modifiers.

VoxML aims to be as generic as possible, while also relying on various mechanisms for their respective strengths, like the simulated environment or spatial calculi, to specify what may be underspecified, as is discussed subsequently.

5 VoxWorld: An Embodied Interaction Platform

Our platform built on the expressive capabilities of VoxML is *VoxWorld*. VoxWorld supports embodied HCI wherein artificial agents consume different sensor inputs for awareness of not only their own virtual space but also the surrounding physical space. It brings together three definitions of simulation from computer science and cognitive science:

1. In *computational simulation modeling*, variables are set in a model and it is run to discover the consequences of possible computable configurations. Examples include models of climate change, structural engineering, biological pathways, etc.
2. *Situated embodied simulations* provide embodiment via a dynamic point-of-view. These are used in training simulators (e.g., combat or flight simulation), and video games.
3. Craik [22] and Johnson-Laird [45] develop *embodied theory of mind*, wherein agents carry a mental model of external reality in their heads. Simulation Theory in philosophy of mind focuses on the role of “mind reading” in modeling the representations and communications of other agents (e.g., [38, 39]). The goal here is semantic interpretation of an expression by means of a simulation, which is either mental [11, 28] or interpreted graphs such as Petri Nets [29, 69].

VoxWorld brings together the model testing of (1), the situated embodiment of (2), and the modeling machinery of (3). There may be a large space of specific operationalizations that satisfy a given action label. The specifics may depend on the objects involved, and may contain many underspecified variable values (e.g., speed of motion, exact path—depending on the verb, etc.). The embodied demonstration of action thereby demonstrates the computational interpretation of it.

At the center of VoxWorld is VoxSim [54], a visual three-dimensional event simulator built on the Unity game engine. VoxSim contains native natural language processing capabilities, VoxML encodings and GL knowledge as interpreted through the multimodal semantics, and some built-in 3rd-party libraries, e.g., QSRLib [33]. VoxML representations are qualitative in nature, and are grounded to “primitives” in 3D space, such as calculated locations, or sequences of *move_to* and *turn_to* actions. These interpretations are made at runtime using a C# VoxML interpreter, and the resultant grounded actions and locations are specific enough to be passed to the animation engine or robotic actuators.⁷

VoxWorld extends VoxSim with connections to arbitrary 3rd-party endpoints; these are usually external sensors or services like speech recognition or vision clients, but can be other platforms like the open-source Robot Operating System, ROS. Developers can therefore use VoxWorld to design and build artificial agent behaviors using these inputs.

The interactive avatar Diana is an output interface that can also include 3rd-party endpoints; in the case of Diana, this is custom gesture and affect recognition [68].

Diana can speak, gesture, track, move, and emote [56, 68]. She is connected to depth sensors running custom gesture recognition and commercial affect recognition software [65], as well as speech recognition via Google Cloud ASR. These inputs let her sense the physical world around her, including the user and how they move. Diana knows when the user is attending to her, as opposed to doing something else, she can observe the user’s emotions, and most importantly she can understand both the user’s speech and gestures. As a result, visual communication joins verbal communication as a two-way process, wherein the artificial agent and the human sense and can communicate with each other similarly.

The architecture of Diana as implemented in VoxWorld system is shown in Fig. 4.

The user, of course, can see Diana act within the VoxSim virtual environment. Such shared perception is a critical component of human communication. When people work together on a physical task, they can each perceive what the others are doing and do not have to describe all their actions. The technology exists to create visual simulations while it is not yet at the requisite level to create simulations for all other senses yet.⁸ Thus, when Diana moves a (virtual) block, she does not have to

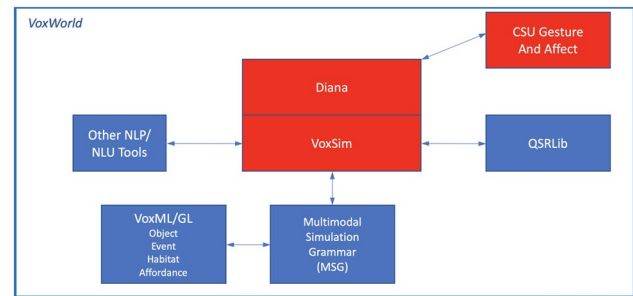


Fig. 4 VoxWorld Architecture schematic for Diana

tell the user she’s doing it; the user can see it happening. This simplifies communication. It also enables visually-grounded reasoning, where the feasibility of actions is determined by the visualization/simulation of the action in the 3D environment perceived by both human and artificial agent. The human can see what actions Diana may take in the current situation and direct their instructions that way. If the human tells Diana to do something impossible, say, interact with a nonexistent object or put an object at an impossible location, Diana can survey the environment, determine the infeasible command and its nature, and communicate this back to the user, e.g., “there is no pink block here,” or “I can’t do that, the purple block is in the way.” Some of Diana’s utterances, such as “OK” when she successfully interprets an input, simply make communication more natural. These orthogonal elements such as dialogue policies help smooth the interaction. Diana must respond in some way to make her behavior seem natural, as has been uncovered through user studies on the Diana system.

Diana is therefore more than an embodied conversational agent. She combines self-embodiment with perception of human embodiment to create a two-way conversational and visual agent. By being situated in a visualized world, she and the user also share perception. The combination results in an interface that feels qualitatively new. Even though the user knows that Diana is an artificial agent and her avatar need not be particularly life-like, she has enough capabilities to establish a conceit of peer-to-peer interaction.

As Diana appears on a screen, her embodiment and the human’s are partially grounded in different worlds. Thus humans adopt slightly different conventions with her than with each other in a shared environment. For example, the elicitation study from which Diana’s gesture recognizers were developed found that humans interacting with each other would point to objects and surfaces in their own environment [99], while we later found that users interacting with Diana pointed at the screen she was displayed on [55].

⁷ VoxSim source can be found [here](#).

⁸ Shared aural perception is possible, while haptic technology is rapidly advancing. We expect that much of the semantics presented here would be suitable for modeling extra-visual shared perception. This is the topic of ongoing research, beginning with haptics in VR.

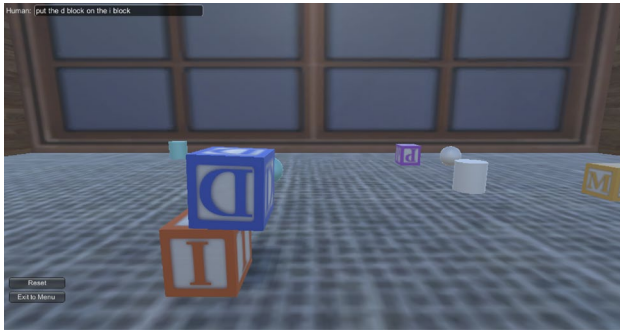


Fig. 5 Situated pseudo-embodied HCI

The current implementation of Diana provides scenes in a Blocks World domain, as well as scenes augmented with a set of more complex everyday objects (e.g., cups, plates, books, etc.). We will revisit embodied human computer interaction with Diana later in this paper.

VoxWorld itself can be used to create other types of artificial agent behaviors, including interactions without an avatar where the user can direct the computer to manipulate objects in space (Fig. 5), interaction with visualized data, or real robots (see Sect. 11).

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, after the simulation has been running for an arbitrary number of timesteps, situational embodiment allows the artificial agent to reason forward about the consequences of specific 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 artificial agent trying to navigate a maze could easily do so with a map that provides 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 artificial 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 artificial agent to assess the next move based on the current set of relations between itself and 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.

6 Embodiment within the Common Ground

The theory of common ground has a rich and diverse literature concerning what is shared or presupposed in human communication [2, 17, 37, 88, 91]. With the presence of a common ground during shared experiences, embodied communication assumes agents can understand one another in a shared context, through the use of co-situational and co-perceptual anchors, and a means for identifying such anchors, such as gesture, gaze, intonation, and language. In this section, we develop a computational model of common ground for multimodal communication.

We assume generally a model of discourse semantics as proposed in [4], as it facilitates the adoption of a continuation-based semantics for our phrase-level compositional semantics [9], as well for discourse, as outlined in [5] and [23]. For the present discussion, however, we will not refer to Segmented Discourse Representation Theory (SDRT) representations, but focus instead on the semantics of integrated multimodal expressions in the context of task oriented dialogue, as presented first in [75] and extended here.

Here, we introduce the *common ground structure* (CGS), the information associated with a state in a dialogue or discourse. We model this as a state monad [93], as in (14):

$$\text{State Monad} : \mathbf{M}\alpha = \text{State} \rightarrow (\alpha \times \text{State}) \quad (14)$$

This monad corresponds to computations that read and modify a state in the discourse. $\mathbf{M}\alpha$ specifies the type of those programs which return α -typed values. These values correspond to the following elements in the dialogue state:

- The communicative act, C_a , performed by an agent, a :
a tuple of expressions from the diverse modalities involved. Broadly, this includes the modalities of a linguistic utterance, $S(\text{speech})$, gesture, G , facial expression, F , gaze, Z , and an explicit action, A .
 $C_a = \langle S, G, F, Z, A \rangle$. For our present discussion, we restrict this to a linguistic utterance, $S(\text{speech})$ and a gesture, G . There are hence three possible configurations in performing a
 $C : C_a = \{(G), (S), (S, G)\}$
- \mathbf{A} : The agents engaged in communication;
- \mathbf{B} : The salient shared belief space;
- \mathbf{P} : The objects and relations that are jointly perceived in the environment;
- \mathcal{E} : The embedding space that both agents occupy in the communication.

Given these attributes, we initialize the common ground for the dialogue based on shared beliefs and dynamic perceptual content for each of the agents. To appreciate the role that the common ground structure plays in creating an embodied interpretation of a communicative expression, let us consider how linguistic expressions are typically interpreted relative to a discourse. The conventional semantic interpretation of a linguistic expression, S , is computed relative to a model, \mathcal{M} , and the relevant assignment functions, e.g., g : that is, $\llbracket S \rrbracket^{\mathcal{M},g}$. For example, “A blue block is on the table.” will have an interpretation, $\llbracket . \rrbracket$, relative to our model specified, which will have the relevant discourse elements, e.g., blocks, a table, the agents present, etc. This will typically be accompanied by dynamic updating operations to enable discourse level anaphoric binding of linguistic expressions introduced over multiple sentences in the dialogue [5, 23, 24, 42, 46]. For example, in the dialogue below, there is a natural coreference between the linguistic antecedent *the blue block* and the pronoun in the subsequent sentence, *it*.

DIALOGUE 1: CO-REFERENCE ACROSS MULTIPLE SENTENCES

HUMAN₁: $S =$ Pick up *a blue block*₁.
HUMAN₂: $S =$ Move *it*₁ there.

Following [5, 23] and further developments in [96], we represent a context as a stack of items and the type of left contexts to be lists of entities, $[e]$. Right contexts will be interpreted as continuations: a discourse that requires a left context to yield a truth value. The type of a right context is therefore $[e] \rightarrow t$. Hence, context transitions get the type $[e] \rightarrow [e] \rightarrow t$; they are characteristic functions of binary relations on contexts; the type for an utterance would be, $[e] \rightarrow ([e] \rightarrow t) \rightarrow t$.

The information state is updated in the dialogue through continuation-passing, as in [5]. We apply a continuation-passing style (CPS) transformation to arrive at the continued type for each expression, notated as an overlined expression [96]. Given the current discourse, D , and the new utterance, S , S integrates into D as follows:

$$\llbracket (\overline{D.S}) \rrbracket^{\mathcal{M},cg} = \lambda i \lambda k. \llbracket \overline{D} \rrbracket i (\lambda i'. \llbracket \overline{S} \rrbracket i' k) \quad (16)$$

This states that the current discourse has two arguments, its left context i (where we are), and what is expected later in the discourse, k . The anaphoric pronoun (*it*) in Dialogue 1 above in the second sentence is interpreted relative to the introduction of the linguistic expression (*a blue block*) in the previous sentence, and as a result, it has a logical antecedent that it can refer to. The first sentence is the context within which the second is interpreted, resulting in the pronoun *it* taking *a blue block* as its antecedent.

While this is a dynamic interpretation of the linguistic expression, it does not capture the interpretation of other modalities in the communication that convey denotative information (such as gesture), and it fails to provide a situated grounding for the expressions within the dialogue state of the current context. This dialogue state is the CGS mentioned above. By treating the common ground as a state monad, we can continue the composition above the level of the sentence as well. We will extend the analysis of continuation passing for linguistic expressions to multimodal processing, as it allows for an *informational distribution* among the expressions from different modalities being used in composition to form larger meanings.

The CGS can be represented as in (17), where an agent, a_i , makes a communicative act through either gesture, as in (17a), or linguistically, as in (17b).⁹

$$\begin{array}{l} \text{a.} \quad \begin{array}{|l} \hline \mathbf{A}:a_1, a_2 \quad \mathbf{B}:\Delta \quad \mathbf{P}:b \quad \mathcal{E}:E \\ \hline \mathcal{G}_{a_1} = \text{“grab } it_b \text{”} \\ \hline \end{array} \\ \text{b.} \quad \begin{array}{|l} \hline \mathbf{A}:a_1, a_2 \quad \mathbf{B}:\Delta \quad \mathbf{P}:b \quad \mathcal{E}:E \\ \hline \mathcal{S}_{a_1} = \text{“You}_{a_2} \text{ see } it_b \text{”} \\ \hline \end{array} \end{array} \quad (17)$$

(17a) specifies that two agents, a_1 and a_2 , co-inhabiting an embedding space, E , within which the experience is embodied, share a set of beliefs, Δ , where they can both see the object, b . Given this representation, the gesture is now situated to refer to objects and knowledge within the CG structure. In (17b), the linguistic expression, \mathcal{S}_{a_1} , is grounded relative to the parameters of common ground, where the indexical *you* will denote the agent, a_2 , and the pronoun *it* will denote the object, b .

To account for an interpretation that enables situated meaning in context, we introduce a *simulation* within which communicative expressions are interpreted. We define this simulation, \mathcal{S} , as a triple, $\langle \mathcal{M}, \mathcal{E}, \mathcal{CG} \rangle$, consisting of a conventional model, \mathcal{M} , an embedding space, \mathcal{E} , together with a common ground structure, \mathcal{CG} . This definition brings together the three types of simulation discussed in Sect. 5.

Hence, we will refer to an interpretation of an expression, α , within a simulation, as $\llbracket \alpha \rrbracket^{\mathcal{S}}$. For example, the CGS in (17b) provides a situated grounding for the linguistic expression, “You see it”:¹⁰

$$\llbracket \text{see}(you, it) \rrbracket^{\mathcal{S},g} = \text{see}(a_2, b) \quad (18)$$

⁹ This is similar in many respects to the representations introduced in [20, 27] and [37] for modeling action and control with robots.

¹⁰ The theory of semiotic schemas introduced in [83] attempts to encode the perceptual context of a linguistic utterance as well, to resolve reference.

However, in order to appreciate the interpretation of gesture (as in (17a)), multimodal expressions, and action within a simulation model, it is necessary to understand: (a) how different actions are embodied; and (b) how gestures denote in the common ground. We turn to these questions in the next two sections.

7 Agent Capabilities Determined by Embodiment

Agent embodiment is defined relative to the actions that can be performed by the agent. What this means is that, while an agent can have a disembodied semantics for many possible actions, it can have an embodied interpretation for only those actions it can execute, in principle. We make a basic distinction between actions that are executable/not executable by the agent, and those actions that are afforded by the environment and the situation.

Consider the two artificial agents discussed below in Sects. 10 and 11, Diana, the interactive avatar, and Kirby the robot. Diana has a humanoid upper body frame and rigging, including legs but legs that do not move. She has two arms and hands and is able to grasp, reach, flex, push, slide, roll, flip, and stack. She has a humanoid head, with a synthetic vision sensor located above her nose (reading from the Kinect camera). She can turn her head but not her torso. Hence, we define these as the *modally executable actions* available to Diana. Kirby, a mobile robot with wheels, a camera, and a LIDAR, but no graspers, has a different set of modally executable actions, e.g., go-to, find, move, turn.

Given the executable actions available to an agent, this determines the subsequent range of other computations, including affordances. But affordable actions are only active when the appropriate object is situated in the environment. For example, a block or a cup can be grasped, picked up, and moved. Hence, these modally executable actions are only possible when the object is present.

Now consider how these actions are actually executed by the artificial agent. All of these are actions that are embodied primarily in the hand (grasper), and by extension (transitivity), in the attached local rig (the arm), and the attached torso. Hence, there is a complete *action path* for the embodied semantics associated with these actions. This has the consequence of forcing a modeling of forward kinematics (FK) and inverse kinematics (IK) for the artificial agent¹¹,

when interpreting expressions entailing such actions. For example, “grab the blue block” presupposes the appropriate proximity to reaching the block, and this may entail moving towards the block. Hence, the *embodied semantics* for *grab* must include the IK required to perform the basic action.

We adopt the notion of *COMPONENT*, as used in type-based ontologies and semantic resources, where the *COMPONENT OBJECT* relation exhibits transitivity: i.e., if $A \sqsubseteq_c B$ and $B \sqsubseteq_c C$, then $A \sqsubseteq_c C$ [76, 102]. For Diana, we define $hand \sqsubseteq_c arm$, and $arm \sqsubseteq_c torso$, hence $hand \sqsubseteq_c torso$.

Given these observations, we see that the semantics encoded in the VoxML representations for objects and actions enables a more embodied interpretation for the entity in question. That is, *hand* is typed as a *grasper* instrument, encoded as a Gibsonian affordance action (cf. Sect. 4.1).

$$\begin{aligned} \llbracket \text{grab it} \rrbracket^{S_g} &= \text{grab}(a_2, b) \\ &= \exists x, y, z [\text{grab}(x, b) \wedge \text{hand}(x) \wedge \text{arm}(y) \\ &= \wedge \sqsubseteq_c (x, y) \wedge \text{torso}(z) \wedge \sqsubseteq_c (y, z)] \end{aligned} \quad (19)$$

Similar remarks on embodied semantics hold for Kirby, but relative to a different set of modally executable actions. For instance Kirby’s wheels allow him to move, such that when a destination is supplied, the wheels, as a component of the chassis and the entire robot as a whole ($wheels \sqsubseteq_c chassis \sqsubseteq_c self \rightarrow wheels \sqsubseteq_c self$) execute the act of locomotion until the robot (self) evaluates its location to be the target.

$$\begin{aligned} \llbracket \text{go there} \rrbracket^{S_g} &= \text{go_to}(a_2, b) \\ &= \exists x, y, z [\text{go_to}(x, b) \wedge \text{wheels}(x) \wedge \text{chassis}(y) \\ &= \wedge \sqsubseteq_c (x, y) \wedge \text{self}(z) \wedge \sqsubseteq_c (y, z)] \end{aligned} \quad (20)$$

8 Modeling the Semantics of Gesture

As mentioned in Sect. 6, a conversation between two agents assumes a common ground, within which we create situated groundings for the communicative expressions used and the actions being performed. Here we introduce a dynamic interpretation for gesture that explicitly references the common ground structure in discourse. We extend the approach taken in [47] and [60], where gestures are simple schemas consisting of distinct sub-gestural phases, where **Stroke** is the content-bearing phase of the gesture.

$$G \rightarrow (\text{Prep}) (\text{Pre_stroke Hold}) \text{Stroke Retract} \quad (21)$$

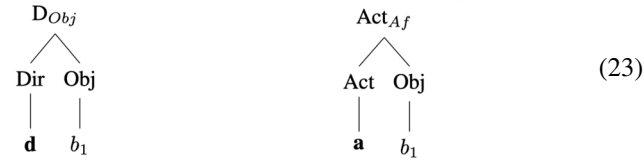
In the context of multimodal dialogues and interactions with artificial agents, a gesture’s **Stroke** will denote a range of primitive action types, *ACT*, e.g., *grasp*, *pick up*, *move*,

¹¹ Forward kinematics computes the position of the end-effector from the joint parameters. Inverse kinematics computes the joint parameters from the position of the effector.

throw, pull, push, separate, and put together. There are many ways to convey intent to carry out these actions, but they all involve two characteristics: (a) the action's object is an embodied reference in the common ground; and (b) the gesture sequence must be interpreted dynamically, to correctly compute the end state of the event. To this end, we model two kinds of gestures in our dialogues: (a) establishing a reference; and (b) depicting an action-object pair.

- a. **Deixis** : $D_{obj} \rightarrow Dir \text{ Obj}$
 b. **Action** : $G_{Af} \rightarrow Act \text{ Obj}$ (22)

We introduce the notion of an interpreted gesture tree in (23a), which indicates that the gesture D_{obj} functionally consists of a deictic orientation, Dir , with the demonstratum, \mathbf{d} , and the referenced or denoting entity, Obj , denoting b_1 .

- a. **Deixis**: $D_{obj} \rightarrow Dir \text{ Obj}$ b. **Action**: $G_{Af} \rightarrow Act \text{ Obj}$
 (23)

As gesture is intended for visual interpretation, it is directly interpretable by the artificial agents in the context if and only if the value is clearly evident in the common ground, most likely through visual inspection. Directional or orientational information conveyed in a gesture identifies a distinct object or area of the embedding space, E , by directing attention to the *End* of the designated pointing ray (or cone) trace [60, 61, 75].

$$\llbracket \mathbf{D}_{obj} \rrbracket = \llbracket End(ray(\mathbf{d})) \rrbracket \quad (24)$$

We model the interpretation function, $\llbracket \cdot \rrbracket$, as fully determining the value of the deixis in the context, supplied by the common ground, which we discuss below. In (23b), the action gesture type, G_{Af} , consists of an action-object pairing, where the action, \mathbf{a} , is applied to the object, b_1 , in some prototypical manner. The available gesture sequence strategies (or “gesture phrases,” GPs) are outlined in (25–27).

- a. *action – object* : e.g., *grab* [**Object**]
 b. $GP_1 \rightarrow G_{Af} D_{obj}$ (Action Focus)
 $\rightarrow D_{obj} G_{Af}$ (Object Focus) (25)

- a. *action – result* : e.g., *put* [**Object**]at [**Location**]
 b. $GP_2 \rightarrow G_{Af} D_{obj} D_{loc}$ (Action Focus)
 $\rightarrow D_{obj} G_{Af} D_{loc}$ (Object Focus)
 $\rightarrow D_{obj} D_{loc} G_{Af}$ (Transition Focus) (26)

- a. *action – result* : e.g., *move* [**Object**] [**Direction**]
 b. $GP_3 \rightarrow G_{Af} D_{obj} D_{dir}$ (27)

As mentioned above, the deictic gesture in (22a) and (23a) actually serves to indicate both a location and objects within that location, suggesting that deixis denotes a *dot object*, viz., $PHYSOBJ \bullet LOCATION$ [6, 73]. Either of these type components may be exploited by the deictic reference, which is then interpreted in context, either as a selection (exploiting the $PHYSOBJ$) or as a destination (exploiting either). For example, should an object b_1 already be selected through a deixis \mathbf{d}_a , as in (23a), a subsequent deixis \mathbf{d}_b may be interpreted as selecting a destination location in isolation (in which case the interpretation exploits the $LOCATION$ of \mathbf{d}_b), or as selecting a location relative to another object (exploiting the $PHYSOBJ$ type of \mathbf{d}_b). We discuss this further below.

When content-bearing gestures are generated as a sequence, as discussed above, they can assume a continuative semantics similar to that used *within* a linguistic expression. The continuative semantics for gesture phrases is in (28).^{12,13,14,15}

- a. $S_G \rightarrow (NP) GP$
 b. $GP_1 \rightarrow G_{af} \mathbf{D}_{obj}$
 c. $GP_2 \rightarrow G_{af} \mathbf{D}_{obj} \mathbf{D}_{Loc}$
 d. $GP_3 \rightarrow G_{af} \mathbf{D}_{obj} \mathbf{D}_{Dir}$ (28)

As before, since the common ground is modeled as a state monad, we can continuize the sequence of gestures in the dialogue accordingly. We apply a CPS transformation to arrive at the continuized type for each expression, notated as an overlined expression [96]. Given the current gesture discourse, D , and a new gesture, C , we take the integration of C into D as follows:

$$\llbracket (\overline{D.C}) \rrbracket^{M, cg} = \lambda k. \llbracket \overline{D} \rrbracket (\lambda n. \llbracket \overline{C} \rrbracket (\lambda m. k(m \ n))) \quad (29)$$

DIALOGUE 2: SINGLE MODALITY (GESTURE) IMPERATIVE

HUMAN₁: $\mathcal{G} = [\text{points to the purple block}]_{t1}$
 HUMAN₂: $\mathcal{G} = [\text{makes move gesture}]_{t2}$
 HUMAN₃: $\mathcal{G} = [\text{points to the red block}]_{t3}$

Given a description of the gesture grammar as used in our multimodal dialogues, let us explore a communicative act that exploits a combination of both speech and gesture, (S, G). We identify three configurations for how a language-gesture *ensemble* can be interpreted, depending on which modality carries the majority of semantic content: (a)

¹² $\llbracket S \rrbracket = (\llbracket NP \rrbracket \llbracket GP \rrbracket)$.

¹³ $\llbracket GP_1 \rrbracket = \lambda j. (\llbracket \mathbf{D}_{obj} \rrbracket; \lambda j'. ((\llbracket G_{af} \rrbracket') j))$.

¹⁴ $\llbracket GP_2 \rrbracket = \lambda k. (\llbracket \mathbf{D}_{Loc} \rrbracket; \lambda j. (\llbracket \mathbf{D}_{obj} \rrbracket; \lambda j'. ((\llbracket G_{af} \rrbracket') j) k))$.

¹⁵ $\llbracket GP_3 \rrbracket = \lambda k. (\llbracket \mathbf{D}_{Dir} \rrbracket; \lambda j. (\llbracket \mathbf{D}_{obj} \rrbracket; \lambda j'. ((\llbracket G_{af} \rrbracket') j) k))$.

language with *co-speech gesture*, where language conveys the bulk of the propositional content and gesture adds situated grounding, affect, effect, and presuppositional force [13, 59, 86]; (b) *co-gestural speech*, where gesture plays this role [75]; and (c) a truly mixed-modality expression, where both language and gesture contribute equally to the meaning. In practice, while many of the interactions in our dialogues have this property, the discourse narrative is broadly guided by gesture. For this reason, we model the multimodal interactions as content-bearing gesture with *co-gestural speech*.

Language and gesture are considered separate modal channels, but they operate interdependently. Deictic gesture acts like a demonstrative in a referring expression, and embodied gesture, when enacted, becomes part of the embedding space. The embodied artificial agent can interpret and generate expressions like “this/that block,” accompanied by deixis, and can do the same when referring to the embodiment of its interlocutor (e.g., “my/your arm”). While agents in the interaction are considered separately from objects in the model, the GL and VoxML typing of embodied agents (e.g., HUMAN • PHYSOBJ) show they have properties of physical objects (e.g., a convex hull, an interaction with the physics of the world, etc.), and so can be discussed in similar terms.

In our theory, a multimodal communicative act, C , consists of a sequence of gesture-language ensembles, (g_i, s_i) , where an ensemble is temporally aligned in the common ground. Let us assume that a linguistic subexpression, s , is either a word or full phrase in the utterance, while a gesture, g , comports with the gesture grammar described above.

Co – gestural Speech Ensemble :

$$\begin{bmatrix} \mathcal{G} & g_1 & \dots & g_i & \dots & g_n \\ \mathcal{S} & s_1 & \dots & s_i & \dots & s_n \end{bmatrix} \quad (30)$$

We assume an aligned language-gesture syntactic structure, for which we provide a continuized semantic interpretation. Both of these are contained in the common ground state monad introduced above in (15). For each temporally indexed and aligned gesture-speech pair, (g, s) , we have a continuized interpretation, as shown below. Each modal expression carries a continuation, k_g or k_s , and we denote the alignment of these two continuations as $k_s \otimes k_g$, seen in (31).

$$\begin{aligned} & \lambda k_s.k_s(\llbracket s \rrbracket) \\ & \lambda k_g.k_g(\llbracket g \rrbracket) \\ & \lambda k_s \otimes k_g.k_s \otimes k_g(\llbracket (s, g) \rrbracket) \end{aligned} \quad (31)$$

Each of these modalities will contribute information if it is present. We bind co-gestural speech to specific gestures in the communicative act, within a common ground, CGS. A dashed line in an ensemble expression indicates that a

co-gestural speech element, \mathcal{S} , is aligned with a particular gesture, \mathcal{G} . For example, the CG structure for this expression,

$$\begin{bmatrix} \mathcal{G} & D_{Obj} & Grab_g \\ \mathcal{S} & that & _ \end{bmatrix}, \text{ is shown in (32).} \quad (32)$$

$$\begin{aligned} & \llbracket \langle that, D_{Obj} \rangle . \langle _, Grab \rangle \rrbracket \\ & = \lambda k_s \otimes k_g.(\llbracket D_{Obj} \rrbracket; \lambda j_g.((\llbracket Grab \rrbracket j_g)k_s \otimes k_g)) \end{aligned}$$

Given the theory of two-level affordances proposed here (Gibsonian and Telic), we can naturally think of objects as *antecedents to the actions performable on them*. For example, for each object in (33), we can identify the attached behaviors.

- a. **block** : Pick me up!, Move me!
- b. **cup** : Pick me up!, Drink what’s in me! (33)
- c. **knife** : Pick me up!, Cut that with me!

This naturally suggests that affordances are a subclass of continuations. For example, both $\llbracket CUP \rrbracket$ and $\llbracket BLOCK \rrbracket$ have similar Gibsonian affordance values, but quite distinct Telic affordance values. This can be distinguished by the nature of their respective Telic continuation sets as follows, where **sel** is a function that selects a suitable discourse antecedent inside the continuation set [5]:

- a. $\lambda k_{Gib} \otimes k_{Telic}.k_{Gib} \otimes k_{Telic}(cup) :$
 $grab \subseteq \mathbf{sel} k_{Gib},$
 $drink \subseteq \mathbf{sel} k_{Telic},$
- b. $\lambda k_{Gib} \otimes k_{Telic}.k_{Gib} \otimes k_{Telic}(block) :$ (34)
 $grab \subseteq \mathbf{sel} k_{Gib},$
 $pick_up \subseteq \mathbf{sel} k_{Gib},$
 $move \subseteq \mathbf{sel} k_{Gib}.$

This is the subject of ongoing research within our studies.

9 Tracking Beliefs and Perception in Context

Common ground updates will also include executing modal operations over the belief space \mathbf{B} , where each new element from the discourse is introduced via a *public announcement logic* (PAL) formula, and each new perceived object or relation is introduced into \mathbf{P} via an analogous *public perception logic* (PPL) formula [71, 94, 95]. We will use $[\alpha]\varphi$ to denote that an agent “ α knows φ ”. Public announcements are implemented as: $[\alpha]\varphi_1\varphi_2$. Any proposition, φ , in the common knowledge held by two agents, α and β , is computed as: $[(\alpha \cup \beta)^*]\varphi$. Agent knowledge is encoded as sets of accessibility relations between situations.

This model allows us to distinguish information in the common ground that is shared by the agents from new assertions accompanying a request or command in a dialogue.

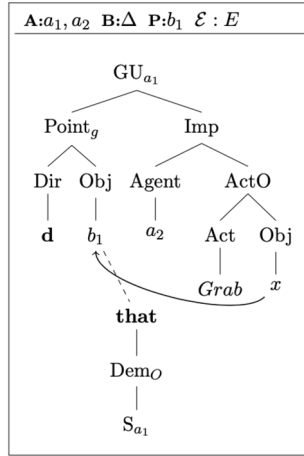


Fig. 6 Common-ground structure for “that” (ensemble) + “grab” (speech) (equivalent to (32))

This is called “assertion in the common ground”, and can be stated as follows: $[(\alpha \cup \beta)^*] \varphi_p \wedge \psi$. This says that agents α and β share the information φ_p , and the common ground is updated with the new information ψ .

In a similar fashion, an agent’s perception is encoded as sets of accessibility relations, α , between situations. What is seen in a situation is encoded as either a proposition, φ , or existential statement of an object x , \hat{x} . $[\alpha]_\sigma \varphi$ denotes that agent “ α perceives that φ ”. $[\alpha]_\sigma \hat{x}$ denotes that agent “ α perceives that there is an x .” Some expressions of co-perception in the common ground include:^{16,17,18,19}

- In order to co-attend, two agents direct gaze towards an object, x_i , or event e_i ;
- Each agent sees the other attend;
- Each agent sees that the other sees her attend;
- The co-perception for α and β includes φ (“*Everyone can see that φ .*”).

Objects and events are realized in terms of their localized embedding space \mathcal{E} , so gazing at “the same” entity is considered to be gazing toward points within the same \mathcal{E} .

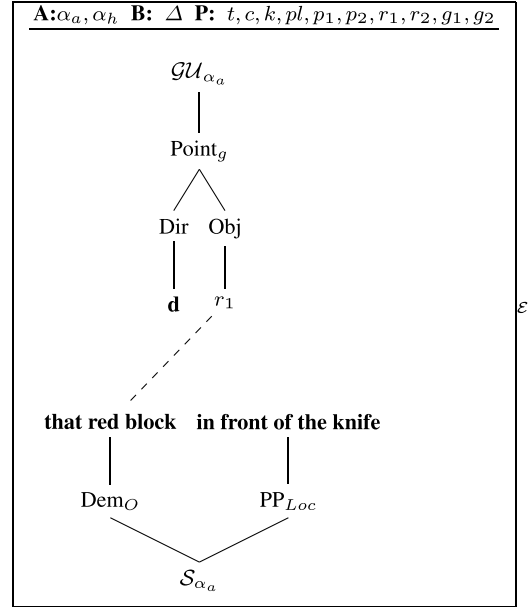
Currently, the update functions in the model described here, for both perception (from action or directly perceived acts) and for beliefs (from public announcements and the result of perception), are largely non-probabilistic in nature, unlike that outlined in [26]. This is due in part to our focus on creating a general architecture for multimodal embodiment. It is clear, however, that both perceptually-derived categorization of the environment and the general (un)certainly of an agent’s belief will be probabilistically determined, as

¹⁶ $[\alpha]_\sigma(x_i \vee e_i), [\beta]_\sigma(x_i \vee e_i)$.

¹⁷ $[\alpha]_\sigma([\beta]_\sigma(x_i \vee e_i)), [\beta]_\sigma([\alpha]_\sigma(x_i \vee e_i))$.

¹⁸ $[\beta]_\sigma([\alpha]_\sigma([\beta]_\sigma(x_i \vee e_i))), [\alpha]_\sigma([\beta]_\sigma([\alpha]_\sigma(x_i \vee e_i)))$.

¹⁹ $[(\alpha \cup \beta)^*]_\sigma \varphi$.



$\lambda k_s \otimes k_g (\text{that}(x)[\text{block}(x) \wedge \text{red}(x) \wedge \text{in_front}(x, k, v)] \wedge k_s \otimes k_g(x))$, where $v = \alpha_a$

Fig. 7 Common-ground structure for “that red block in front of the knife” (cf. Fig. 8). The semantics of the RE includes a continuation for each modality, k_s and k_g , which applies over the object in subsequent moves in the dialogue

well as dynamically updated, as also argued in [7, 41, 81, 92]. This would involve constructing user models for distinct

interlocutors, which we anticipate building in VoxWorld (Figs. 6, 7, 8).

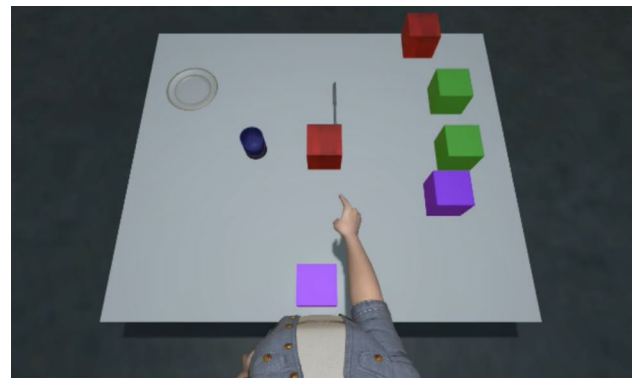
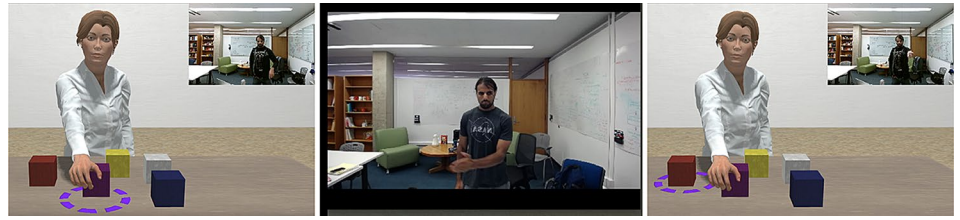


Fig. 8 Agent Points to Block in Common Ground

Fig. 9 Embodied interaction with language and gesture



10 Demonstrations of Embodied Common Ground

10.1 Undoing Actions

Semantically, undoing an action is fundamentally unbinding and correcting parameters of the action. Assuming a generic semantics as in (36), the human may initially indicate (37) and instruct Diana to place the purple block on the yellow block. However, should they decide that this is not actually correct, they can use the correction as in (38), which treats “on the white one” as replacement content, while keeping the remainder of the semantic content the same. This undoes the action in progress, rewinds the state monad and performs a reassignment with the replacement content. The result is that Diana puts the purple block on the white block instead.

$$\lambda k. \overline{C_1}(\lambda n. \overline{C_2}(\lambda m. k(m\ n))) \quad (36)$$

$$\begin{aligned} & \text{a. } \lambda k_{Gib} \otimes k_{Telic}. k_{Gib} \otimes k_{Telic}(block) \\ & \text{b. } grab \subseteq \text{sel } k_{Gib} \\ & \text{c. } put \subseteq \text{sel } k_{Gib} \\ & \text{d. } \lambda k. k(put) \Rightarrow M, cg_1 \models on(yellow, purple) \end{aligned} \quad (37)$$

$$\begin{aligned} & \text{a. "Wait, on the white one."} \\ & \text{b. } \text{undo } k = \lambda k. k(put) \\ & \text{c. } \textbf{Rewind} \text{ the state monad and } \textbf{Reassign} : \\ & \text{d. } \lambda k_{Gib} \otimes k_{Telic}. k_{Gib} \otimes k_{Telic}(block) \\ & \text{e. } put \subseteq \text{sel } k_{Gib} \\ & \text{f. } \lambda k. k(put) \Rightarrow \\ & \text{g. } M, cg_1 \models on(yellow, white) \end{aligned} \quad (38)$$

Depending on the exact nature of the replacement content, the resulting action may differ. In (38) the replacement content is a prepositional phrase, and so it is the destination that is reassigned. If the human had said “wait, the white one,” instead then the subject of the action would have been reassigned after rewinding the state monad. The result here would have been Diana putting down the purple block and putting the white one on the yellow block.

Where corrected references are available for anaphora (e.g., shifting antecedents of third-person pronouns, cf. [40]), *it* (looking for a *PHYSOBJ*) would remain resolved to the object, while *she* (looking for an *AGENT*) would resolve



Fig. 10 Diana interacts with an unknown object through recognizing its affordances. The human points to the cup and says “What is that?” to which Diana replies, “That’s a cup.” When the human indicates the bottle, Diana says, “I don’t know what it’s called, but I can grasp it like a cup”

to Diana. However, if the language was one like Bengali or Turkish, without gendered 3rd-person singular pronouns, additional language processing techniques would be required.

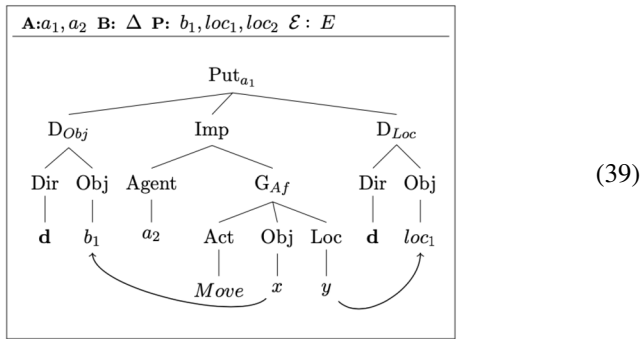
10.2 Aligning Gesture and Language

Figure 9 illustrates an embodied HCI, where deixis (pointing) and action-affordance gestures from the human are situated in an embodied space shared by both the IVA and the human. These are accompanied by aligned co-gestural language expressions, such as “that one”, “the purple one”, etc.

To show how continuations help interpretation of gesture sequences, consider a single modality gesture imperative.

Through its own continuation, the referent identified in the first deixis, \mathbf{D}_{Obj} , is passed to the action ($\lambda k. k(\llbracket \text{Move} \rrbracket)$), while the continuized interpretation of the action delays the computation of its argument until the appropriate binding has been identified. Finally, the goal location for the movement selected for by the *move* gesture is identified through the action of the continuized location deixis, \mathbf{D}_{Loc} . This is illustrated in (40), along with the common ground structure that is computed, shown in (39).²⁰

²⁰ A video demo can be viewed here <http://www.voxicon.net/wp-content/uploads/2020/07/DARPA-CwC-Brandeis-CSU-July-2020.mp4>.



$$\begin{aligned}
 & \llbracket D_{Obj} \cdot \text{Move} \cdot D_{Loc} \rrbracket \\
 &= \lambda k. (\llbracket D_{Loc} \rrbracket; \lambda j. (\llbracket D_{Obj} \rrbracket; \lambda j'. ((\llbracket \text{Move} \rrbracket j') k)))
 \end{aligned}
 \tag{40}$$

10.3 Transferring Object Properties using Affordances

Because Diana is embodied and situated within an embodied HCI environment, this facilitates transfer learning of object affordances between objects, as illustrated in Fig. 10. For this configuration, we assume that Diana has no semantics for the object we recognize as a bottle. In embodied interaction with the human, Diana is able to observe certain similarities in the shape and habitats of $[[CUP]]$ and $[[BOTTLE]]$ (e.g., current upright orientation, similar symmetry and size constraints), and infer that they might share some behaviors, which leads her to infer that a way to grasp the bottle would be like she grasps the cup. The close association between habitats and affordances and the structured encoding provided by VoxML allows us to perform this kind of transfer learning by inferring a likely missing behavior given a novel object and an encoding of its current configuration.

Given that similar habitats serve as necessary (but not, in isolation, sufficient) preconditions to certain behaviors (e.g., in order to be rolled, an apple, a cup, and a bottle must all be turned on their sides), the ability to assess an unknown object relative to known ones allows an artificial agent to computationally transfer properties of known objects to unknown ones, in a way that gives it a handle on interacting with and discussing a novel object.

Our method involves training 200-dimensional embeddings over 22 VoxML habitat and affordance encodings using a Skip-Gram method. Objects are represented as averaged habitat or affordance vectors, which are fed into two models, a 7-layer MLP and a 4-layer (1D) CNN. Given an object representation and a desired behavior, the models pick the known object most similar to an unlabeled embedding vector with respect to the desired behavior. For example, an affordance vector representing a *plate* was predicted to be similar to a *cup* or *bottle* due to its containment affordance.

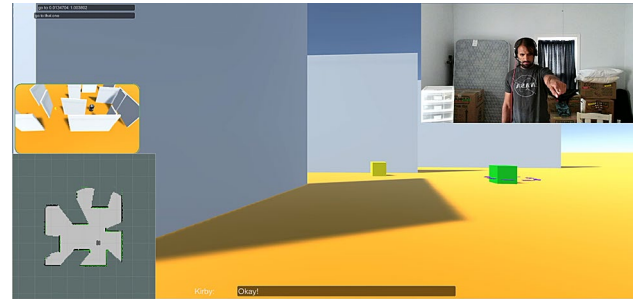


Fig. 11 Communicating with a mobile robot. In this figure, the robot itself is a simulated robot running in the ROS Gazebo simulator environment

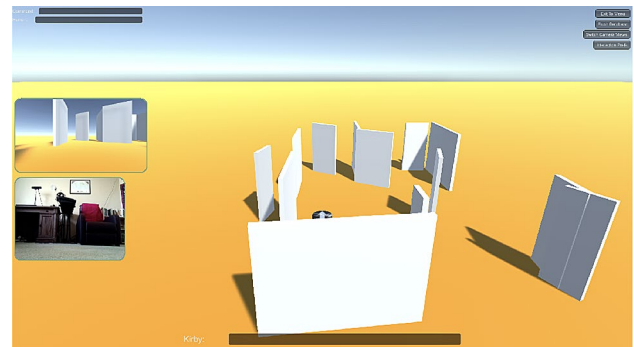


Fig. 12 Communicating with a mobile robot. This figure features a real robot navigating through a real environment

Diana observes similarities in the habitats of the cup and the bottle (e.g., upright orientation, similar symmetry and size constraints), and infers they may share behaviors, which leads her to grasp the bottle like the cup. Links between habitats and affordances in VoxML allows us to infer similar objects and behaviors in the current situation.

Putting a formal representation in distributional space is a hard problem, central to neurosymbolic AI, for which VoxWorld is well-suited. We rely on the representation of affordances as a habitat H in which an event $[E]$ leads to a result R (e.g., see (8)). This vocabulary of affordances is vectorized like words: by turning cooccurrence of habitats and affordances into a distributional representation.

To learn about novel *events*, the same principles could be applied to different (e.g., sequential) models, such as LSTMs instead of CNNs. Transitions may likely be handled similarly; this is in need of empirical exploration.

11 Embodied HCI and Robot Control

We are exploring an additional application of embodied HCI in the context of communication and control of a mobile robot. Specifically, we have used VoxWorld for navigation in



Fig. 13 Custom GoPiGo3

novel environments using coordinated gesture and language. A human partner can deliver instructions to the robot using spoken English and gestures relative to the simulated environment, to guide the robot through navigation and exploration tasks.

In our system, a human user and a robot (called “Kirby”) exist in a co-situated space that is mediated by a virtual environment displayed on a screen, such that the human can see a virtual rendition of the environment the robot has explored, and of the robot’s current perspective view. The human can then gesture to objects and locations on the screen, either in a perspective or omniscient view, and speak about them in English, e.g., “go there,” “go to that wastebasket and turn around,” or “find the blue block.” Deictic gestures are grounded to coordinates on the screen which are transformed to equivalent coordinates in the robot’s ROS environment, allowing the robot to execute native navigation commands, e.g., *go_to(x, y)*. The robot can likewise communicate status updates back to the human which are then spoken out through text-to-speech.

Figure 11 shows the 3D rendering of the robot’s environment from its perspective (main panel), an omniscient view (center left), and the visualized LIDAR data (bottom left).

The robot might hear the instruction “go here/to that one,” be able to see which object the user is indicating, and go to it. In another scenario, imagine the user is viewing the omniscient perspective and pointing to a different object outside the robot’s field of view, and gives the same instruction. There, the denotation of “here” or “that one” is not available to the robot in the common ground, because the demonstrative has not been grounded to a location. The robot will have to ask for clarification (“I can’t see where you’re pointing”)

or turn around to scan until it sees the location of deixis in order to interpret the instruction.

Figure 12 shows the VoxWorld rendering of an environment being explored by a real robot (see Fig. 13) using the same omniscient and perspective views, plus an additional view that streams the onboard camera from the robot into the VoxWorld view (left side, below the perspective view). Note how the obstacles detected by the LIDAR and rendered into VoxWorld as walls match real objects visible in the live camera stream. Because of the planar nature of the LIDAR, only obstacles at LIDAR-level are detected, like the cabinet, chair, bookshelf, and piano stand. Obstacles above LIDAR-level, such as the desk surface, are invisible to it. In omniscient view (here the main panel), we can see that the robot has detected other obstacles outside of the camera frame, such as other walls of the room. Its situationally grounded knowledge of these obstacles informs how it navigates. For instance, to execute a *go forward* command the robot may have to navigate around detected obstacles.

Figure 13 shows the real navigating robot used in this example. This is a GoPiGo3 robot by Dexter Industries that has been customized by placing a LIDAR on top, attaching a high-capacity power pack, and installing a Raspberry Pi Camera Module v2 (8 megapixels). Object detection is currently accomplished using fiducial detection with markers placed on objects or instantiated in the ROS Gazebo simulator. Integrating real object detection using the onboard camera stream is the topic of ongoing research.

Here VoxWorld connects to one external sensor—the onboard camera—directly to stream the feed. It also connects to a cross-platform robotic services bridge. This consumes output either directly from the robot such as its position and orientation, or from routines running over its sensor data, such as LIDAR feedback converted into line segments [8, 89].

12 Embodiment, Common Ground, and Agent Capabilities

The capabilities of a given artificial agent are dependent on the particulars of its embodiment in that having or lacking certain effectors allow it to or disallow it from participating in certain behaviors. Diana has two hands and with them can manipulate objects. Upon entering into an interaction with her, the user, perceiving that she has hands, and given the situation in which she is placed before blocks on the table, typically assumes that Diana is capable of manipulating the blocks and will give her instructions pertaining to that task. The mere fact of perceiving the nature of her effectors introduces an assumption into the common ground that Diana can manipulate the blocks somehow. Perception of the artificial agent’s embodiment populates the common ground with

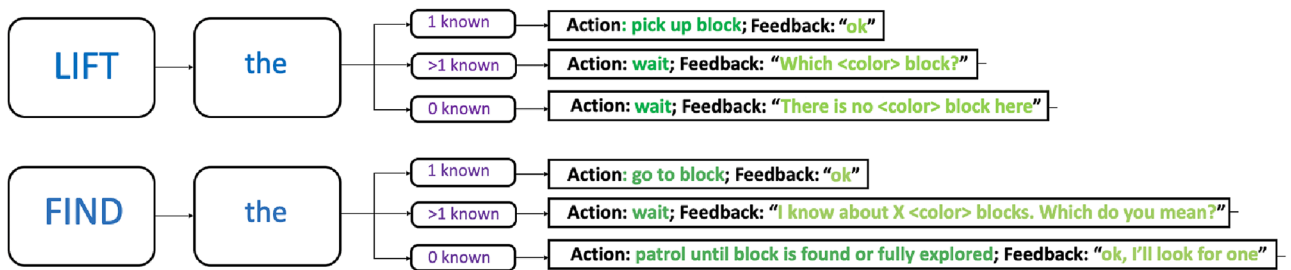


Fig. 14 Equivalent sections of Diana's (top) and Kirby's (bottom) respective dialogues, showing effects of embodiment and information state on their respective actions and utterances. On the left is the action, in the center are conditions on the number of known objects

that match the characteristics given in the instruction, and on the right are the actions taken and feedback given in each case. Other quantifiers trigger different action and dialogue sequences based on the situated context

some propositional content that can then be confirmed or refuted as the interaction proceeds.

In the robot control examples from Sect. 11, the live camera feed serves as a kind of *situated pseudo-embodiment* discussed in Sect. 2. This moving isolated camera is not physically possible without the embodiment of the robot and the structures that support the camera, i.e., the chassis and the wheels that move it. In these examples we demonstrate two distinct kinds of embodiment—the pseudo-embodiment of the camera and the *situated self-embodiment* of the robot's presence in the simulated world that is directly derived from real-world sensor feedback. This computational embodiment facilitates reasoning and the camera feed and rendering of the robot's current interpretation presents that reasoning process for the human to interact with.

Here and in Sect. 7, we have demonstrated two distinct kinds of interactions with two distinct types of artificial agents. Diana the interactive avatar exists solely in the virtual world, cannot locomote, and has hands. Kirby the robot maintains an embodied presence in the VoxWorld virtual environment that is derived from sensor data coming from a real robot navigating through the real world. Diana can manipulate objects but cannot walk or move to other locations. Kirby can go to locations and find objects but cannot manipulate them. This in turn has a profound effect on the types of instructions these different artificial agents can take and the types of interactions they can participate in.

Figure 14 shows snippets of Diana's and Kirby's dialogue. These flowcharts show two distinctions:

1. Due to the different self-embodiments of the artificial agents, they participate in different activities entirely. Diana can *lift* blocks while Kirby can *find* them.
2. Because of Kirby's ability to move and explore the scene, which Diana lacks, Kirby begins with an assumption that he does not have perfect information about the scene and that there may be items he hasn't discovered yet.

As shown in the action/feedback pairs on the right, Kirby can explore the scene if he does not know of any objects matching what he's looking for. Diana, assuming more complete information about the scene, will state that "there is no <COLOR> block here" if she has not found and does not currently see one. Both sequences use the same underlying semantics based on content introduced into the common ground. Differences in ability and articulation (i.e., the knowledge the artificial agents introduce into the common ground) depend on their respective embodiments.

These differences allow for testing embodiment's effects with respect to language and reasoning. Diana and Kirby may have to solve to same problem: for example, catching water from a leak. Both can identify the need for a container y (cf. affordance embeddings in Sect. 10.3). Then, they need to move it to the location of the leak. They have two actions in their common vocabular. Having graspers, Diana can satisfy "grasp," a precondition to the "move" subevent of both actions, so she can perform both as encoded. Kirby, with no graspers, cannot satisfy "grasp," but he can reason about certain other subevents of $[[SLIDE]]$, such as moving y while keeping it in contact with its supporting surface sfc ²¹. Where Diana could grasp and move or slide the container to its destination, Kirby can solve the problem through second-order reasoning over the VoxML predicate $[[SLIDE]]$ by determining subevents he is capable of satisfying through other means, such as simply pushing the object.

13 Conclusion

In this paper, we have discussed the role that embodiment of artificial agents and common ground between artificial agents and humans plays in creating rich, interactive,

²¹ VoxML encodes relations using a number of common spatial reasoning calculi, including the Region Connection Calculus [82], where this would be encoded $EC(y, sfc)$.

intelligent behaviors. VoxWorld facilitates experiments with IVAs in embodied HCI contexts, using multiple modalities in diverse settings. An embodied HCI, as is enabled by VoxWorld, allows the human and artificial agent to share an epistemic space, and any communicative modality that can be expressed within that space (e.g., linguistic, visual, gestural) enriches the ways a human and a computer or robot can communicate about objects, actions, and situated tasks.

These different task-appropriate behaviors serve to examine the capabilities of artificial agents and behavior of humans in different situations, and those interactions can be evaluated with regard to the same underlying semantic structures, including the parameters of an artificial agent's embodiment and the common ground between interlocutors.

A Diana-like artificial agent, capable of sight, speech, and situational understanding (including understanding of the user's situatedness) represents a potential step forward in interacting with smart devices; the ability to see and understand the environment, including the user's gesture, and interpret that in conjunction with language, gives rise to the potential of truly situated smart artificial agents.

Embodied HCI with physically embodied robotic agents opens up the possibility of assistive robotics in situations dangerous or inaccessible to humans. The robotic agent can navigate the space and potentially interact with it, given appropriate effectors, while communicating back to the human partner to receive instructions, direction, and relying on the human's experience and situational knowledge.

These different interactions can use the same type of semantic processing vis-à-vis the common ground. The particulars of the artificial agent's embodiment then condition what it can do and discuss, and how it in turn introduces knowledge into the common ground.

This combination of embodiment and common ground in intelligent behaviors lays the groundwork for novel kinds of ubiquitous computing à la Weiser ([100]) wherein the artificial agent makes the environment an inextricable part of its reasoning and communicates that reasoning back to its human partner(s) in terms of that same environment.

Acknowledgements We would like to thank Ross Beveridge, Bruce Draper, Francisco R. Ortega, and their team at Colorado State University, and Jaime Ruiz and his team at the University of Florida, without whose contribution the Diana System would not be a reality. We would also like to thank Katherine Krajovic, R. Pito Salas, and Nathaniel J. Dimick for their work on the Kirby implementation. Particular thanks to Ms. Krajovic for assembling the dialogue flowcharts in Fig. 14. We would also like to thank Ken Lai for his discussion regarding common ground structure. This work was supported by the US Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO) under contract #W911NF-15-C-0238 at Brandeis University. This work was also supported in part by a grant to James Pustejovsky from the IIS Division of National Science Foundation (1763926) entitled "Building a Uniform Meaning Representation for Natural Language Processing". The points of view expressed herein

are solely those of the authors and do not represent the views of the Department of Defense or the United States Government. Any errors or omissions are, of course, the responsibility of the authors.

References

1. Anderson ML (2003) Embodied cognition: a field guide. *Artif Intell* 149(1):91–130
2. Asher N (1998) Common ground, corrections and coordination. *J Semant*
3. Asher N (2008) A type driven theory of predication with complex types. *Fund Inf* 84(2):151–183
4. Asher N, Lascarides A (2003) *Logics of conversation*. Cambridge University Press, Cambridge
5. Asher N, Pogodalla S (2010) Sdrt and continuation semantics. In: *JSAI international symposium on artificial intelligence*, Springer, New York, pp 3–15
6. Asher N, Pustejovsky J (2006) A type composition logic for generative lexicon. *J Cognit Sci* 6:1–38
7. Baker CL, Jara-Ettinger J, Saxe R, Tenenbaum JB (2017) Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nat Hum Behav* 1(4):1–10
8. Ballard DH (1981) Generalizing the hough transform to detect arbitrary shapes. *Pattern Recogn* 13(2):111–122
9. Barker C, Shan CC (2014) *Continuations and natural language*, vol 53. Oxford Studies in Theoretical Linguistics
10. van Benthem JFAK (1991) Logic and the flow of information
11. Bergen BK (2012) *Louder than words: the new science of how the mind makes meaning*. Basic Books
12. Blackburn P, Bos J (2003) Computational semantics. *Theor Int J Theory Hist Found Sci* pp 27–45
13. Cassell J, Stone M, Yan H (2000a) Coordination and context-dependence in the generation of embodied conversation. In: *Proceedings of the first international conference on Natural language generation-Volume 14*, ACL, pp 171–178
14. Cassell J, Sullivan J, Churchill E, Prevost S (2000b) *Embodied conversational agents*. MIT Press, New York
15. Chrisley R (2003) Embodied artificial intelligence. *Artif Intell* 149(1):131–150
16. Clancey WJ (1993) Situated action: A neuropsychological interpretation response to vera and simon. *Cogn Sci* 17(1):87–116
17. Clark HH, Brennan SE (1991) Grounding in communication. *Perspect Soc Share Cognit* 13(1991):127–149
18. Cooper R (2005) Records and record types in semantic theory. *J Logic Comput* 15(2):99–112
19. Cooper R (2017) Adapting type theory with records for natural language semantics. In: *Modern perspectives in type-theoretical semantics*, Springer, New York, pp 71–94
20. Cooper R, Ginzburg J (2015) Type theory with records for natural language semantics. *The handbook of contemporary semantic theory* p 375
21. Coventry K, Garrod SC (2005) Spatial prepositions and the functional geometric framework. *Towards a classification of extra-geometric influences*
22. Craik KJW (1943) *The nature of explanation*. Cambridge University, Cambridge
23. De Groote P (2001) Type raising, continuations, and classical logic. In: *Proceedings of the thirteenth Amsterdam Colloquium*, pp 97–101
24. Dekker PJ (2012) Predicate logic with anaphora. In: *Dynamic Semantics*, Springer, New York, pp 7–47

25. Dobnik S, Cooper R (2017) Interfacing language, spatial perception and cognition in type theory with records. *J Lang Modell* 5(2):273–301
26. Dobnik S, Cooper R, Larsson S (2012) Modelling language, action, and perception in type theory with records. In: *International workshop on constraint solving and language processing*, Springer, New York, pp 70–91
27. Dobnik S, Cooper R, Larsson S (2013) Modelling language, action, and perception in type theory with records. In: *Constraint solving and language processing*, Springer, New York, pp 70–91
28. Evans V (2013) *Language and time: a cognitive linguistics approach*. Cambridge University Press, Cambridge
29. Feldman J (2010) Embodied language, best-fit analysis, and formal compositionality. *Phys Life Rev* 7(4):385–410
30. Fernando T (2009) Situations in ltl as strings. *Inf Comput* 207(10):980–999
31. Fischer K (2011) How people talk with robots: designing dialog to reduce user uncertainty. *AI Magn* 32(4):31–38
32. Foster ME (2007) Enhancing human–computer interaction with embodied conversational agents. In: *International conference on universal access in human–computer interaction*, Springer, New York, pp 828–837
33. 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
34. Gibson JJ (1977) The theory of affordances. *Perceiving, acting, and knowing: toward an ecological psychology*, pp 67–82
35. Gibson JJ (1979) *The ecological approach to visual perception*. Psychology Press
36. Ginzburg J (1996) Interrogatives: questions, facts and dialogue. *The handbook of contemporary semantic theory*. Blackwell, Oxford pp 359–423
37. Ginzburg J, Fernández R (2010) Computational models of dialogue. *The handbook of computational linguistics and natural language processing* 57:1
38. Goldman AI (1989) Interpretation psychologized*. *Mind Lang* 4(3):161–185
39. Gordon RM (1986) Folk psychology as simulation. *Mind Lang* 1(2):158–171
40. Gregoromichelaki E, Kempson R, Howes C (2020) Actionism in syntax and semantics. *Dial Percept* pp 12–27
41. Griffiths TL, Chater N, Kemp C, Perfors A, Tenenbaum JB (2010) Probabilistic models of cognition: exploring representations and inductive biases. *Trends Cogn Sci* 14(8):357–364
42. Groenendijk J, Stokhof M (1991) Dynamic predicate logic. *Linguist Philos* pp 39–100
43. Harel D (1984) Dynamic logic. In: Gabbay M, Gunthner F (eds) *Handbook of philosophical logic, volume II: extensions of classical logic*, Reidel, p 497–604
44. Harel D, Kozen D, Tiunyn J (2000) *Dynamic logic*, 1st edn. The MIT Press, New York
45. Johnson M (1987) *The body in the mind: the bodily basis of meaning, imagination, and reason*. University of Chicago Press, Chicago
46. Kamp H, Van Genabith J, Reyle U (2011) Discourse representation theory. In: *Handbook of philosophical logic*, Springer, New York, pp 125–394
47. Kendon A (2004) *Gesture: visible action as utterance*. Cambridge University Press, Cambridge
48. Kiela D, Bulat L, Vero AL, Clark S (2016) Virtual embodiment: A scalable long-term strategy for artificial intelligence research. *arXiv preprint [arXiv:161007432](https://arxiv.org/abs/161007432)*
49. Klein E, Sag IA (1985) Type-driven translation. *Linguist Philos* 8(2):163–201
50. Konrad K (2004) 4 minimal model generation. In: *Model generation for natural language interpretation and analysis*, Springer, New York, pp 55–56
51. Kopp S, Wachsmuth I (2010) *Gesture in embodied communication and human–computer interaction*, vol 5934. Springer, New York
52. Krishnaswamy N (2017) Monte-carlo simulation generation through operationalization of spatial primitives. PhD thesis, Brandeis University
53. Krishnaswamy N, Pustejovsky J (2016a) Multimodal semantic simulations of linguistically underspecified motion events. In: *Spatial Cognition X*, Springer, New York, pp 177–197
54. Krishnaswamy N, Pustejovsky J (2016b) VoxSim: a visual platform for modeling motion language. In: *Proceedings of COLING 2016, the 26th international conference on computational linguistics*, ACL
55. Krishnaswamy N, Pustejovsky J (2018) Deictic adaptation in a virtual environment. In: *Spatial cognition XI*, Springer, New York, pp 180–196
56. Krishnaswamy N, Narayana P, Wang I, Rim K, Bangar R, Patil D, Mulay G, Ruiz J, Beveridge R, Draper B, Pustejovsky J (2017) Communicating and acting: Understanding gesture in simulation semantics. In: *12th International workshop on computational semantics*
57. Kruijff GJM, Lison P, Benjamin T, Jacobsson H, Zender H, Kruijff-Korbayová I, Hawes N (2010) Situated dialogue processing for human–robot interaction. In: *Cognitive systems*, Springer, pp 311–364
58. Landragin F (2006) Visual perception, language and gesture: a model for their understanding in multimodal dialogue systems. *Signal Process* 86(12):3578–3595
59. Lascarides A, Stone M (2006) Formal semantics for iconic gesture. In: *Proceedings of the 10th workshop on the semantics and pragmatics of dialogue (BRANDIAL)*, pp 64–71
60. Lascarides A, Stone M (2009) A formal semantic analysis of gesture. *J Semant* p ffp004
61. Lücking A, Pfeiffer T, Rieser H (2015) Pointing and reference reconsidered. *J Pragmat* 77:56–79
62. Mani I, Pustejovsky J (2012) *Interpreting motion: grounded representations for spatial language*. Oxford University Press, Oxford
63. Marge M, Rudnicki AI (2013) Towards evaluating recovery strategies for situated grounding problems in human–robot dialogue. In: *2013 IEEE RO-MAN, IEEE*, pp 340–341
64. Marshall P, Hornecker E (2013) Theories of embodiment in hci. *SAGE Handb Digit Technol Res* 1:144–158
65. McNeely-White DG, Ortega FR, Beveridge JR, Draper BA, Bangar R, Patil D, Pustejovsky J, Krishnaswamy N, Rim K, Ruiz J, Wang I (2019) User-aware shared perception for embodied agents. In: *2019 IEEE international conference on humanized computing and communication (HCC)*, IEEE, pp 46–51
66. Miller GA, Johnson-Laird PN (1976) *Language and perception*. Belknap Press, Cambridge
67. Muller P, Prévot L (2009) Grounding information in route explanation dialogues
68. Narayana P, Krishnaswamy N, Wang I, Bangar R, Patil D, Mulay G, Rim K, Beveridge R, Ruiz J, Pustejovsky J, Draper B (2018) Cooperating with avatars through gesture, language and action. In: *Intelligent systems conference (IntelliSys)*
69. Narayanan S (2010) Mind changes: a simulation semantics account of counterfactuals. *Cognit Sci*
70. Naumann R (2001) Aspects of changes: a dynamic event semantics. *J Semant* 18:27–81
71. Plaza J (2007) *Logics of public communications*. Synthese 158(2):165–179

72. Pustejovsky J (1991) The syntax of event structure. *Cognition* 41(1–3):47–81
73. Pustejovsky J (1995) *The generative Lexicon*. MIT Press, New York
74. Pustejovsky J (2013) Dynamic event structure and habitat theory. In: *Proceedings of the 6th international conference on generative approaches to the Lexicon (GL2013)*, ACL, pp 1–10
75. Pustejovsky J (2018) From actions to events: communicating through language and gesture. *Interact Stud* 19(1–2):289–317
76. Pustejovsky J, Batiukova O (2019) *The lexicon*. Cambridge University Press, Cambridge
77. Pustejovsky J, Boguraev B (1993) Lexical knowledge representation and natural language processing. *Artif Intell* 63(1–2):193–223
78. Pustejovsky J, Krishnaswamy N (2016) Voxml: a visualization modeling language. *Proceedings of LREC*
79. Pustejovsky J, Krishnaswamy N (2020) Embodied human-computer interactions through situated grounding. In: *IVA '20: proceedings of the 20th international conference on intelligent virtual agents*, ACM
80. Pustejovsky J, Moszkowicz JL (2011) The qualitative spatial dynamics of motion in language. *Spatial Cognit Comput* 11(1):15–44
81. Qing C, Goodman ND, Lassiter D (2016) A rational speech-act model of projective content. In: *Proceedings of cognitive science*, pp 1110–1115
82. Randell D, Cui Z, Cohn A, Nebel B, Rich C, Swartout W (1992) A spatial logic based on regions and connection. In: *KR'92. Principles of knowledge representation and reasoning: proceedings of the 3rd international conference*, Morgan Kaufmann, San Mateo, pp 165–176
83. Roy D (2005) Semiotic schemas: a framework for grounding language in action and perception. *Artif Intell* 167(1–2):170–205
84. Schaffer S, Reithinger N (2019) Conversation is multimodal: thus conversational user interfaces should be as well. In: *Proceedings of the 1st international conference on conversational user interfaces*, pp 1–3
85. Scheutz M, Cantrell R, Schermerhorn P (2011) Toward human-like task-based dialogue processing for human robot interaction. *AI Magn* 32(4):77–84
86. Schlenker P (2020) Gestural grammar. *Nat Lang Linguist Theory* pp 1–50
87. Shapiro L (2014) *The Routledge handbook of embodied cognition*. Routledge, England
88. Stalnaker R (2002) Common ground. *Linguist Philos* 25(5–6):701–721
89. Tavares JMRS, Padilha AJMN (1995) A new approach for merging edge line segments. In: *Proceedings RecPad'95*, Aveiro
90. Tellex S, Gopalan N, Kress-Gazit H, Matuszek C (2020) Robots that use language. *Annu Rev Control Robot Auton Syst* 3:25–55
91. Tomasello M, Carpenter M (2007) Shared intentionality. *Dev Sci* 10(1):121–125
92. Ullman TD, Goodman ND, Tenenbaum JB (2012) Theory learning as stochastic search in the language of thought. *Cogn Dev* 27(4):455–480
93. Unger C (2011) Dynamic semantics as monadic computation. In: *JSAI international symposium on artificial intelligence*, Springer, New York, pp 68–81
94. Van Benthem J (2011) *Logical dynamics of information and interaction*. Cambridge University Press, Cambridge
95. Van Ditmarsch H, van Der Hoek W, Kooi B (2007) *Dynamic epistemic logic*, vol 337. Springer, New York
96. Van Eijck J, Unger C (2010) *Computational semantics with functional programming*. Cambridge University Press, Cambridge
97. Vera AH, Simon HA (1993) Situated action: a symbolic interpretation. *Cognit Sci* 17(1):7–48. [https://doi.org/10.1016/S0364-0213\(05\)80008-4](https://doi.org/10.1016/S0364-0213(05)80008-4)
98. Wahlster W (2006) Dialogue systems go multimodal: The smartkom experience. In: *SmartKom: foundations of multimodal dialogue systems*, Springer, New York, pp 3–27
99. Wang I, Narayana P, Patil D, Mulay G, Bangar R, Draper B, Beveridge R, Ruiz J (2017) EGGNOG: A continuous, multi-modal data set of naturally occurring gestures with ground truth labels. In: *To appear in the Proceedings of the 12th IEEE international conference on automatic face & gesture recognition*
100. Weiser M (1999) The computer for the 21st century. *ACM SIGMOBILE Mob Comput Commun Rev* 3(3):3–11
101. Williams T, Bussing M, Cabrol S, Boyle E, Tran N (2019) Mixed reality deictic gesture for multi-modal robot communication. In: *2019 14th ACM/IEEE international conference on human-robot interaction (HRI)*, IEEE, pp 191–201
102. Winston ME, Chaffin R, Herrmann D (1987) A taxonomy of part-whole relations. *Cognit Sci* 11(4):417–444