An Abstract Specification of VoxML as an Annotation Language

Kiyong Lee
Dept. of Linguistics
Korea University, Seoul
ikiyong@gmail.com

Nikhil Krishnaswamy
Dept. of Computer Science
Colorado State University
nkrishna@colostate.edu

James Pustejovsky
Dept. of Computer Science
Brandeis University
jamesp@brandeis.edu

Abstract

VoxML is a modeling language used to map natural language expressions into real-time visualizations using commonsense semantic knowledge of objects and events. Its utility has been demonstrated in embodied simulation environments and in agent-object interactions in situated multimodal human-agent collaboration and communication. It introduces the notion of object affordance (both Gibsonian and Telic) from HRI and robotics, as well as the concept of habitat (an object's context of use) for interactions between a rational agent and an object. This paper aims to specify VoxML as an annotation language in general abstract terms. It then shows how it works on annotating linguistic data that express visually perceptible human-object interactions. The annotation structures thus generated will be interpreted against the enriched minimal model created by VoxML as a modeling language while supporting the modeling purposes of VoxML linguistically.

1 Introduction

As introduced by Pustejovsky and Krishnaswamy (2016), VoxML is a modeling language encoding the spatial and visual components of an object's conceptual structure.[1] It allows for 3D visual interpretations and simulations of objects, motions, and actions as minimal models from verbal descriptions. The data structure associated with this is called a *voxeme*, and the library of voxemes is referred to as a *voxicon*.

VoxML elements are conceptually grounded by a conventional inventory of semantic types (Pustejovsky, 1995; Pustejovsky and Batiukova, 2019). They are also enriched with a representation of how and when an object affords interaction with another object or an agent. This is a natural extension of Gibson's notion of object affordance (Gibson, 1977) to functional and goal-directed aspects of Generative Lexicon's Qualia Structure (Pustejovsky, 2013; Pustejovsky and Krishnaswamy, 2021), and is situationally grounded within a semantically interpreted 4D simulation environment (temporally interpreted 3D space), called VoxWorld (McNeely-White et al., 2019; Krishnaswamy et al., 2022).

VoxML has also been proposed for annotating visual information as part of the ISO 24617 series of international standards on semantic annotation schemes, such as ISO-TimeML (ISO, 2012) and ISO-Space (ISO, 2020). VoxML, as an annotation language, should be specified in abstract terms, general enough to be interoperable with other annotating languages, especially as part of such ISO standards, while licensing various implementations in concrete terms. In order to address these requirements, this paper aims to formulate an abstract syntax of VoxML based on a metamodel. It develops as follows: Section 2, Motivating VoxML as an Annotation Language, Section 3, Specification of an Annotation Scheme, based on VoxML, Section 4, Interpretation of Annotation-based Logical Forms with respect to the VoxML Minimal Model, and Section 5, Concluding Remarks.

2 Motivating VoxML as an Annotation Language

Interpreting actions and motions requires situated background information about their agents or related objects, occurrence conditions, and enriched lexical information. The interpretation of base annotation structures, anchored to lexical markables for annotating visual perceptions, depends on various sorts of parametric information besides their associated dictionary definitions.

A significant part of any model for situated com-

[1] VoxML represents a *visual object concept structure (vocs)* modeling language.

Proceedings of the 19th Joint ACL - ISO Workshop on Interoperable Semantic Annotation, pages 66–74
June 20, 2023. ©2023 Association for Computational Linguistics

munication 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 Generative Lexicon (GL) with the Qualia Structure, for example, (Pustejovsky, 1995), as well as *object affordances* (Gibson, 1977) is useful to help ground expression meaning to speaker intent. As an illustration, consider first how such information is encoded and then exploited in reasoning. Knowledge of objects can be partially contextualized through their *qualia structure* (Pustejovsky and Boguraev, 1993), where each Qualia role can be seen as answering a specific question about the object it is bound to: *Formal*, the IS-A relation; *Constitutive*, an object PART-OF or MADE-OF relation; *Agentive*, the object's CREATED-BY relation; and *Telic*: encoding information on purpose and function (the `used-for` or FUNCTIONS-AS relation).

While such information is needed for compositional semantic operations and inferences in conventional models, it falls short of providing a representation for the *situated grounding* of events and their participants or of any expressions between individuals involved in a communicative exchange. VoxML provides just such a representation. It further encodes objects with rich semantic typing and action affordances and actions themselves as multimodal programs, enabling contextually salient inferences and decisions in the environment. To illustrate this, consider the short narrative in (1) below.

(1) Mary picked up the glass from the table and put it in the dishwasher to wash and dry it.

VoxML provides the means to better interpret these events as situationally grounded in interactions between an agent and objects in the world.

In order to create situated interpretations for each of these events, there must be some semantic encoding associated with how the objects relate to each other physically and how they are configured to each other spatially. For example, if we associate the semantic type of "container" with glass, it is situationally important to know how and when the container capability is activated: i.e., the orientation information is critical for enabling the use or function of the glass *qua* container. VoxML encodes these notions that are critical for Human-Object Interaction as: *what* the function associated with an object is (its affordance), and just as critically, *when* the affordance is active (its habitat). It also explicitly encodes the dynamics of the events bringing about any object state changes in the environment, e.g., change in location, time, and attribute.

3 Specification of the Annotation Scheme

3.1 Overview

VoxML is primarily a modeling language for simulating actions in the visual world. Still, it can also be used as a markup language for (i) annotating linguistic expressions involving human-object interactions, (ii) translating annotation structures in shallow semantic forms in typed first-order logic, and then (iii) interpreting with the minimal model simulated by VoxML by referring to the voxicon, or set of voxemes, as shown in Figure 1.

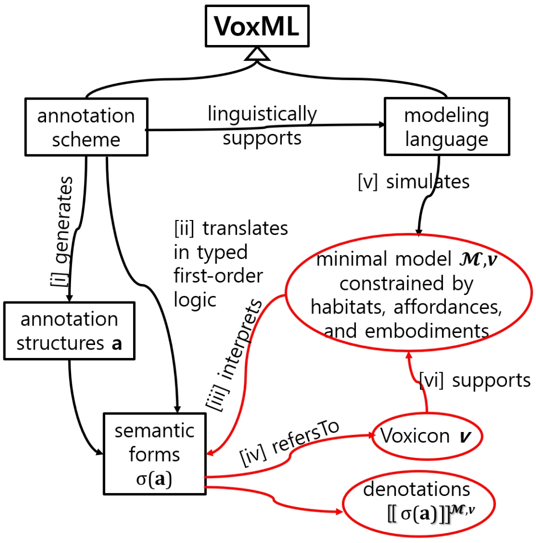
VoxML
annotation scheme
linguistically supports
modeling language
[i] generates
[ii] translates in typed first-order logic
[v] simulates
minimal model $\mathcal{M},v$ constrained by habitats, affordances, and embodiments
annotation structures **a**
[iii] interprets
[vi] supports
semantic forms σ(**a**)
[iv] refersTo
Voxicon v
denotations $[\![\sigma(\mathbf{a})]\!]^{\mathcal{M},v}$

Figure 1: How VoxML operates

This section formally specifies the VoxML-based annotation scheme, with a metamodel (3.2), an abstract syntax (3.3), a concrete representation of annotation structures (3.4), and their translation to semantic forms in typed first-order logic (3.5).

3.2 Metamodel of the VoxML-based Annotation Scheme

A metamodel graphically depicts the general structure of a markup language. As pointed out by Bunt (2022), a metamodel makes the specification of annotation schemes intuitively more transparent, thus becoming a *de facto* requirement for constructing semantic annotation schemes. The metamodel, represented by Figure 2, focuses on interactions between entities (objects) and humans, while the *dynamic paths*, triggered by their actions, trace the visually perceptible courses of those actions. The VoxML-based annotation scheme, thus represented, is construed to annotate linguistic expressions for human-object interactions (cf. Henlein et al. (2023)).

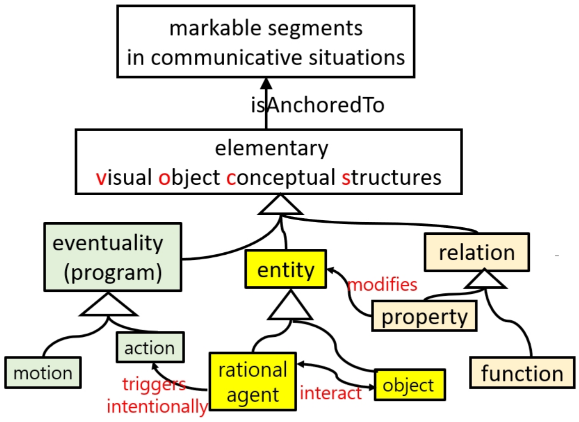

Figure 2: Metamodel of VoxML

We view the VoxML model or world as inhabited by only three categories of entities: *event (program):action*, *object*, and *relation*. Each of them has subcategories, as represented by the hollow triangles in Figure 2.[2] Because of its key role in VoxML, category *action* is introduced as a subcategory of category *event*. This model represents a *small minimal* world, focused on actions, (physical) objects, and their interrelations, which together constitute the larger ontology such as SUMO (Niles and Pease, 2001). Unlike other types of eventuality, agents intentionally trigger actions, and these agents can be humans or other rational agents. These agents also interact with objects as participants in actions.

Category *relation* has two subcategories, *property* and *function*. As unary relations, properties modify entities (objects), as in *big table*. Functions are particular relations mapping one object to another. The function loc for *localization*, for instance, maps physical objects (e.g., *table*) to spatial locations where some other objects like apples can be placed. As introduced by Katz (2007), the runtime function τ maps eventualities to times such that $\tau(e)$ refers to the occurrence time of the event e. We may also introduce a function *seq* that forms paths by ordering pairs $t@l$ of a time t and a location l. The VoxML annotation language has no category such as location, time, or path, but can introduce time points to discuss, for instance, their temporal ordering: e.g., $\tau(e_1) \prec \tau(e2)$. Binary or any other n-ary relations, such as *in* or *between*, are of category *relation* and are also introduced into VoxML.

VoxML, as a modeling language, views physical objects and actions as forming visually perceptible conceptual structures called *voxemes*. Applied to language and its constituent expressions, the VoxML-based annotation scheme takes them as *markables*, anchored to a word, an image, a gesture, or anything from communicative actions that consist of verbal descriptions, gestures, and surrounding backgrounds.

3.3 Abstract Syntax

An abstract syntax defines a specification language and rigorously formulates its structures. In constructing natural language grammars (Lee, 2016, 2023), the abstract syntax of a semantic annotation scheme is defined as a tuple in set-theoretic terms. The abstract syntax $\mathcal{AS}yn_{voxml}$ of the VoxML-based annotation scheme is also defined as a set-theoretic tuple, as in Definition 2:

(2) Definition of $\mathcal{AS}yn_{voxml}$:
Given a finite set D, or data, of communicative segments in natural language, the abstract syntax $\mathcal{AS}yn_{voxml}$ of VoxML is defined to be a triplet $<M, C, @>$, where:

- M is a nonnull subset of D that contains (possibly null or non-contiguous) strings of communicative segments, called *markables*, each delimited by the set B of base categories.
- C consists of base categories B and relational categories R:
  - Base categories B and their subcategories, as depicted in Figure 2: [i] *event*:*action*, [ii] *entity (object)* and [iii] *relation*:{*property*, *function*}.
  - Relational categories R: unspecified for $\mathcal{AS}yn_{voxml}$.
- $@_{cat}$ is a set of assignments from attributes to values specified for each category *cat* in C.

For every base category *cat* in B, the assignment $@_{cat}$ has the following list of attributes as required to be assigned a value:

(3) **Assignment** $@_{cat}$ in Extended BNF:
```
attributes =
identifier, target, type, pred;
identifier = categorized prefix
  + a natural number;
target = markable;
type = CDATA;
pred = CDATA|null;
  (* predicative content *)
```

[2] In UML, a hollow triangle represents a subcategorization relation.

Each category may have additional required or optional attributes to be assigned a value. For instance, the assignment $@_{action}$ is either a *process* or *transition* type. Category *action* has the attribute `@agent`, which triggers it.

3.4 Representing Annotation Structures

The annotation scheme, such as $\mathcal{AS}_{voxml}$, generates annotation structures based on its abstract syntax. These annotation structures have two substructures: *anchoring* and *content* structures. In pFormat[3], these two structures are represented differently by representing anchoring structures by their values only, but content structures as attribute-value pairs.

The first part of Example (1) is annotated as follows:

(4) a. Base-segmented Data:

Mary$_{x1,w1}$ picked up$_{e1,w2-3}$ the glass$_{x2,w5}$ from$_{r1,w6}$ the table$_{x3,w8}$.

b. Annotation Structures:

object `(x1, w1,`
`type="human", pred="mary")`
action `(e1, w2-3,`
`type="transition", pred="pickUp",`
`agent="#x1", physObj="#x2")`
object `(x2, w5,`
`type="physobj", pred="glass")`
relation `(r1, w6,`
`type="spatial", source="#x3")`
object `(x3, w8,`
`type="physobj", pred="table")`

In base-segmented data, each markable is identified by its anchoring structure $<cat_i, w_j>$ (e.g., `x1, w1`), where cat_i is a categorized identifier and w_j is a word identifier. The agent which triggered the action of picking up the glass is marked as Mary$_{x1}$, and the object glass$_{x2}$ is related to it.

Interoperability is one of the adequacy requirements for an annotation scheme. Here, we show how the VoxML-based annotation scheme is interoperable with other annotation schemes, such as ISO-TimeML (ISO, 2012) and the annotation scheme on anaphoric relations (see Lee (2017) and ISO (2019)). The rest of Example (1) can also be annotated with these annotation schemes. It is first word-segmented, while each markable is tagged with a categorized identifier and a word identifier as in (5):

(5) a. Primary Data:

Mary picked up the glass from the table and put it in the dishwasher to wash and dry it.

b. Base-segmented Data:

Mary$_{x1,w1}$ [picked up]$_{e1,w2-3}$ the glass$_{x2,w5}$ from$_{r1,w6}$ the table$_{x3,w8}$ and put$_{e2,w10}$ it$_{x4,w11}$ in$_{r2,w12}$ the dishwasher$_{x5,w14}$ to wash$_{e3,w16}$ and dry$_{e4,w18}$ it$_{x6,w19}$.

Second, each markable is annotated as in (6):

(6) Elementary Annotation Structures:

action `(e2, w10`
`type="transition", pred="put"`
`agent="#x1", relatedTo="#x4")`
object `(x4, w11,`
`type="unknown", pred="pro")`
relation `(r2, w12`
`type="spatial", pred="in")`
object `(x5, w14,`
`type="physobj, artifact",`
`pred="dishwasher")`
action `(e3, w16,`
`type="process", pred="wash",`
`agent="#5, theme="#x6")`
action `(e4, w18,`
`type="process", pred="dry",`
`agent="#x5", theme="#x6")`
object `(x6, w19,`
`type="unknown", pred="pro")`

The first two actions *pick up* and *put* are triggered by the human agent *Mary*, whereas the actions of *wash* and *dry* are triggered by the dishwasher, which is not human.

The annotation scheme $\mathcal{AS}_{voxML}$ for actions annotates the temporal ordering of these four actions by referring to ISO-TimeML, as in (7):

(7) a. Temporal Links (`tLink`):

tLink `(tL1, eventID="#e2",`
`relatedToEventID="#e1",`
`relType="after")`
tLink `(tL2, eventID="#e3",`
`relatedToEventID="#e2",`
`relType="after")`
tLink `(tL3, eventID="#e4",`
`relatedToEventID="#e3",`
`relType="after")`

[3] pFormat is a predicate-logic-like annotation format for replacing XML, thus being constrained to introduce embedded structures into annotations.

b. Semantic Representation:
$[\mathit{pickUp}(e_1), \mathit{put}(e_2), \mathit{wash}(e_3), \mathit{dry}(e_4),$
$\tau(e_1) \prec \tau(e_2) \prec \tau(e_3) \prec \tau(e_4)]$[4]

The annotation scheme $\mathcal{AS}_{voxML}$ can also refer to the subordination link (`sLink`) in ISO-TimeML (ISO, 2012) to annotate subordinate clauses such as *to wash and dry it* in Example (1).

(8) a. Subordination Link (`sLink`):

```
sLink(sL1, eventID="#e2",
relatedTo="{#e3,#e4}",
relType="purpose")
```

b. Semantic Representation:
$[\mathit{put}(e_2), \mathit{wash}(e_3), \mathit{dry}(e_4),$
$\mathit{purpose}(e_2, \{e_3, e_4\})]$

The subordination link (8) relates the actions of *wash* and *dry* to the action of *put* by annotating that those actions were the *purpose* of *putting* the glass in the dishwasher.

By referring to the annotation schemes proposed by Lee (2017) or ISO (2019), the VoxML-based annotation scheme can annotate the anaphoric or referential relations involving pronouns. The two occurrences of the pronoun *it* refer to the noun *the glass* are annotated as in (9):

(9) a. Annotation of Coreferential Relations:

```
object(x2, w5,
type="physobj, artifact",
pred="glass")
anaLink(aL1, x4, x2, identity)
anaLink(aL2, x6, x2, identity)
```

b. Semantic Representation:
(i) $\sigma(x2) := [\mathit{glass}(x_2)]$,
$\sigma(aL1) := [x_4{=}x_2]$,
$\sigma(aL2) := [x_6{=}x_1]$
(ii) $[\mathit{glass}(x_2), x_4{=}x_2, x_6{=}x_2]$

Semantic Representation (ii) is obtained by unifying all the semantic forms in (i). It says that the two occurrences of the pronoun *it* both refer to the glass.

3.5 Annotation-based Semantic Forms

The annotation scheme translates each annotation structure $\mathbf{a}_4$ into a semantic form $\sigma(\mathbf{a}_4)$, as in (10).

(10) a. Base Semantic Forms σ:[5]
$\sigma(x1) := \{x_1\}[\mathit{human}(x_1), \mathit{mary}(x_1)]$
$\sigma(x2) := \{x_2\}[\mathit{physObj}(x_2),$
$\mathit{glass}(x_2)]$
$\sigma(x3) := \{x_3\}[\mathit{physObj}(x_3),$
$\mathit{table}(x_3)]$
$\sigma(e1) := \{e_1\}[\mathit{action}(e_1),$
$\mathit{transition}(e_1),$
$\mathit{pickUp}(e_1),$
$\mathit{agent}(e_1, x_1),$
$\mathit{theme}(e_1, x_2)]$
$\sigma(r1) := \{r_1\}[\mathit{relation}(r_1),$
$\mathit{source}(r_1, x_3)]$

b. Composition of the Semantic Forms:
$\sigma(\mathbf{a}_4) := \oplus\{\sigma(x1), \sigma(x2), \sigma(x3), \sigma(e1), \sigma(r1)\}$

By unifying all of the semantic forms in (10a), we obtain the semantic form $\sigma(\mathbf{a_1})$ of the whole annotation structure $\mathbf{a}_1$. This semantic form roughly states that Mary picked up a glass (see $\sigma(e1)$), which moved away from the table. This interpretation is too shallow to view how Mary's picking up the glass from the table happened. It was on the table, but now it is no longer there. It is in the hand of Mary, who grabbed it. It didn't move by itself, but its location followed the path of the motion how Mary's hand moved.

3.6 Interpreting Annotation-based Semantic Forms

To see the details of the whole motion, as described by Example (1a), we must know the exact sense of the verb *pick up*. WordNet Search - 3.1 lists 16 senses, most rendered when the verb is used with an Object as a transitive verb. Picking up a physical object like a glass or a book means taking it up by hand, whereas picking up a child from kindergarten or a hitchhiker on the highway means taking the child home or giving the hitchhiker a ride. Such differences in meaning arise from different agent-object interactions. The VoxML-based annotation scheme refers to Voxicon that consists of voxemes and interprets the annotation-based semantic forms, such as (10), with respect to a VoxML model.

4 Interpretation with respect to the VoxML Minimal Model

Voxemes in VoxML create a minimal model. Each of the annotation-based semantic forms, as in (10),

[4] These semantic forms can be represented in DRS validly. See Lee (2023).

[5] As noted earlier, DRS (Kamp and Reyle, 1993) represents these semantic forms in an equivalent way.

is interpreted with respect to this minimal model by referring to its respective voxemes.

4.1 Interpreting Objects

There are four objects mentioned in Example (1): $mary(x_1)$, $glass(x_2)$, $table(x_3)$, and $dishwasher(x_5)$.[6] The semantics forms in (10) say very little. For instance, the semantic form $\sigma(x2)$ of the markable *glass* in (10) says it is a *physical object* but nothing else.

In addition to the lexical information, as given by its annotation structure and corresponding semantic form, each entity of category *object* in VoxML is enriched with information with the elaboration of [i] its *geometrical type*, [ii] the *habitat* for actions, [iii] the *affordance structures*, both Gibsonian and telic, and [iv] the agent-relative *embodiment*.

In a voxicon, such information is represented in a typed feature structure. An example is given in Figure 3 for the object *glass*.[7]

$$\begin{bmatrix}
\textbf{glass} & \\
\text{LEX} = & \begin{bmatrix} \text{PRED} = \textbf{glass} \\ \text{TYPE} = \textbf{physobj, artifact} \end{bmatrix} \\
\text{TYPE} = & \begin{bmatrix} \text{HEAD} = \textbf{cylindroid[1]} \\ \text{COMPONENTS} = \textbf{surface, interior} \\ \text{CONCAVITY} = \textbf{concave} \\ \text{ROTATSYM} = \{Y\} \\ \text{REFLECTSYM} = \{XY, YZ\} \end{bmatrix} \\
\text{HABITAT} = & \begin{bmatrix} \text{INTR} = {\scriptstyle [2]} \begin{bmatrix} \text{CONSTR} = \{Y > X, Y > Z\} \\ \text{UP} = align(Y, \mathcal{E}_Y) \\ \text{TOP} = top(+Y) \end{bmatrix} \\ \text{EXTR} = {\scriptstyle [3]} \begin{bmatrix} \text{UP} = align(Y, \mathcal{E}_{\perp Y}) \end{bmatrix} \end{bmatrix} \\
\text{AFFORD\_STR} = & \begin{bmatrix} \text{A}_1 = H_{[2]} \to [put(x, on([1]))]support([1], x) \\ \text{A}_2 = H_{[2]} \to [put(x, in([1]))]contain([1], x) \\ \text{A}_3 = H_{[2]} \to [grasp(x, [1])] \\ \text{A}_4 = H_{[3]} \to [roll(x, [1])] \end{bmatrix} \\
\text{EMBODIMENT} = & \begin{bmatrix} \text{SCALE} = \textbf{<agent} \\ \text{MOVABLE} = \textbf{true} \end{bmatrix}
\end{bmatrix}$$

Figure 3: VoxML representation for object *glass*

The TYPE structure in Figure 3 contains definitions of rotational symmetry ROTATSYM and reflectional symmetry REFLSYM. The rotational symmetry ROTATSYM of a shape gives the major axis of an object such that when the object is rotated around that axis for some interval of less than or equal to 180 °, the shape of the object looks the same. Examples of shapes with rotational symmetry are *circle*, *triangle*, etc. The reflectional symmetry REFLSYM is a type of symmetry which is with respect to reflections across the plane defined by the axes listed, e.g., a *butterfly* assuming vertical orientation would have reflectional symmetry across the YZ-plane.

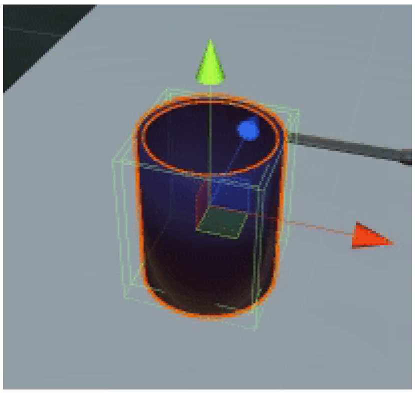

Figure 4: Rendering of object *glass* (cf. Figure 3) showing orthogonal axes.

Figure 4 shows a 3D rendering of a glass object as defined by the structure Figure 3, taken from the VoxWorld platform (Pustejovsky et al., 2017; Krishnaswamy et al., 2022). The object is shown with the 3 major orthogonal axes of the 3D world The green axis is the Y-axis, which is the axis of rotational symmetry. The glass is also symmetric across the XY-plane (defined by red and green axes) and the YZ-plane (defined by the green and blue axes).

Under the HABITAT structure in Figure 3, the variables X, Y, and Z correspond to extents in standard Cartesian coordinates, representing the dimensions, such as areas, required to represent 3D objects in space. From these areas, the radii or circumferences of the bottom and the top areas and the height of the glass are obtainable. Note that the top of a glass has its top area open as a container. Unlike the solid cylindroid, the glass consists of two sheets for the closed bottom and the side such that the circumference of the top area only stands for the width of the side sheet. Note also that the size of the circumference of the top Y, which is the brim of a glass, may equal or be larger than that of the bottom X.

The *habitat* describes environmental and configurational constraints that are either inherent to the object ("intrinsic" habitats, such as a glass having an inherent top, regardless of its placement in the environment), or required to execute certain activ-

[6] The variables x_4 and x_6 are assigned to the two occurrences of the pronoun *it*.

[7] Taken from the Voxicon in Krishnaswamy and Pustejovsky (2020).

ities with it ("extrinsic" habitats, such as a glass needing to be placed on its side to be rollable).

This representation provides the necessary information for its full interpretation. It says the object is glass, a physical artifact having the shape of a concave cylindroid and other geometrical features. It should be standing concave upward to hold liquid. Thus, it can be placed on the table, contain water or wine, and be grasped by a hand. It may roll if it falls sideways, but it does only if it does not have something like a handle or is not designed like a wine glass. The embodiment says it is smaller than the one holding it and can move.

4.2 Interpreting Agents

A voxeme for an agent may refer to an actual human agent or an AI agent of any form (humanoid, robotic, or without distinct form). Other entities, or rational agents, may function as agents as long as they are capable of executing actions in the world (Krishnaswamy, 2017; Pustejovsky et al., 2017) Examples developed using the VoxWorld platform include collaborative humanoid agents that interact with humans and objects, including interpreting VoxML semantics in real time to exploit and learn about object affordances (Krishnaswamy et al., 2017, 2020; Krishnaswamy and Pustejovsky, 2022), navigating through environments to achieve directed goals (Krajovic et al., 2020), and also self-guided exploration where the VoxML semantics "lurk in the background" for the agent to discover through exploratory "play" (Ghaffari and Krishnaswamy, 2022, 2023). The physical definition of agents conditions their actions (Pustejovsky and Krishnaswamy, 2021). For instance, a humanoid agent with defined *hand* $\sqsubseteq_c$ *arm* $\sqsubseteq_c$ *torso* is enabled to execute the act of grasping, while a robotic agent defined with *wheels* $\sqsubseteq_c$ *chassis* $\sqsubseteq_c$ *self* is enabled for the act of locomotion. This has implications for the semantics of how the agent is interacted with: the humanoid can *pick up* objects while the robot can *go to* them.

4.3 Interpreting Actions as Programs

Actions are viewed as *programs* that can formally implement them as processes, (dynamic) sequences of sub-events or states, recursions, algorithms, and execution (see Mani and Pustejovsky (2012) and de Berg et al. (2010)).

The voxemes for actions are much simpler than those for objects. They consist of three attributes: [i] Lex for lexical information, [ii] Type for argument structure, and [iii] Body for subevent structure. The information conveyed by [i] and [ii] is provided by the annotation structures for predicates with their attributes @type, @pred, @agent, and @physObj.

(11) Annotation Structure:
action(a1, w2-3,
type="transition", pred="pickUP",
agent="#x1", physObj="#x2")

As being of type *transition*, the action of picking up involves two stages of a motion, [i] the initial stage of *grasping* the glass and [ii] the ensuing process of *moving* to some direction while *holding* it. This involvement is stated by part of the voxeme for the predicate *pick up*, as in (12):[8]

(12) Embodiment for *pick up*:
a. $E_1 = grasp(x, y)$
b. $E_2 = [while(hold(x, y),$
$move(x, y, vec(\mathcal{E}_Y)))]$

The embodiment E_2 states that the agent x moves the glass y, as her hand and arm move together, along the path or vector $\mathcal{E}_Y$ while holding it (see Harel et al. (2000) for *while* programs or *tail recursion*).

4.4 Interpreting the Role of Relations

The preposition *from* functions as a spatial relation between the object *glass* and the table on which it was located and supported. Then, as the hand of the agent *Mary* holding the glass moves, the glass is no longer on the table but moves away along the path that the hand moves. Hence, the relation *from* marks the initial point of that path or vector.

5 Concluding Remarks

The paper specified the VoxML-based annotation scheme in formal terms. The example of the action of Mary picking up a glass from the table showed how that particular example was annotated and how its logical forms were interpreted with a VoxML model while referring to the voxicon. Each voxeme in the Lexicon, especially that of objects, contains information enriched with the notions of habitat, affordance, and embodiment. As the voxicon develops into a full scale, the task of interpreting

[8] This information is derived from the voxeme for *lift* in Krishnaswamy and Pustejovsky (2020) and applied to the predicate *pick up*.

annotated language data involving complex interactions between humans and objects can easily be managed.

For purposes of exposition, the discussion here focused on the annotation of one short narrative in English involving one verb, *pick up*, and one object, *glass*. The proposed VoxML-based annotation scheme needs to be applied to large data with a great variety to test the effectiveness of interpreting its annotation structures and corresponding semantic forms against the VoxML model. At the same time, such an application calls for the need to enlarge the size and variety of the voxicon for modeling purposes as well. The evaluation of the VoxML-based annotation scheme and the extension of the voxicon remain as future tasks.

Acknowledgments

This paper has been revised with the constructive comments from four reviewers. We thank them all for their suggestions.

References

Harry Bunt. 2022. Intuitive and formal transparency in semantic annotation schemes. In *Proceedings of the 18th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-18)*, pages 102–109, Workshop at LREC2022, Marseilles, France.

Mark de Berg, Otfried Choeng, Marc van Kreveld, and Mark Overmars. 2010. *Computational Geometry: Algorithms and Applications*. Springer, Berlin. Third Edition.

Sadaf Ghaffari and Nikhil Krishnaswamy. 2022. Detecting and accommodating novel types and concepts in an embodied simulation environment. In *Proceedings of the Tenth Annual Conference on Advances in Cognitive Systems*.

Sadaf Ghaffari and Nikhil Krishnaswamy. 2023. Grounding and distinguishing conceptual vocabulary through similarity learning in embodied simulations. In *Proceedings of the 15th International Conference on Computational Semantics*. ACL.

James Jerome Gibson. 1977. The theory of affordances. *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, pages 67–82. Reprinted as chapter 8 of Gibson (1979).

David Harel, Dexter Kozen, and Jerzy Tiuryn. 2000. *Dynamic Logic*. The MIT Press, Cambridge, MA.

Alexander Henlein, Anju Gopinath, Nikhil Krishnaswamy, Alexander Mehler, and James Pustejovksy. 2023. Grounding human-object interaction to affordance behavior in multimodal datasets. *Frontiers in Artificial Intelligence*, 6(1084740):01–12. Doi:10.3389/frai.2023.1084740.

ISO. 2012. *ISO 24617-1 Language resource management – Semantic annotation framework – Part 1: Time and events*. International Organization for Standardization, Geneva.

ISO. 2019. *ISO 24617-9 Language resource management – Semantic annotation framework – Part 9: Reference annotation framework (RAF)*. International Organization for Standardization, Geneva.

ISO. 2020. *ISO 24617-7 Language resource management – Semantic annotation framework – Part 7: Spatial information*. International Organization for Standardization, Geneva. 2nd edition.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logical and Discourse Representation Theory (Studies in Linguistics and Philosophy)*. Kluwer, Dordrecht.

Graham Katz. 2007. Towards a denotational semantics for TimeML. In Frank Schilder, Graham Katz, and James Pustejovsky, editors, *Annotation, Extracting, and Reasoning about Time and Events*, pages 88–106. Springer, Berlin.

Katherine Krajovic, Nikhil Krishnaswamy, Nathaniel J Dimick, R Pito Salas, and James Pustejovsky. 2020. Situated multimodal control of a mobile robot: Navigation through a virtual environment. *RoboDial*.

Nikhil Krishnaswamy. 2017. *Monte Carlo Simulation Generation Through Operationalization of Spatial Primitives*. Ph.D. thesis, Brandeis University.

Nikhil Krishnaswamy, Pradyumna Narayana, Rahul Bangar, Kyeongmin Rim, Dhruva Patil, David McNeely-White, Jaime Ruiz, Bruce Draper, Ross Beveridge, and James Pustejovsky. 2020. Diana's world: A situated multimodal interactive agent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 34.9, pages 13618–13619.

Nikhil Krishnaswamy, Pradyumna Narayana, Isaac Wang, Kyeongmin Rim, Rahul Bangar, Dhruva Patil, Gururaj Mulay, Ross Beveridge, Jaime Ruiz, Bruce Draper, et al. 2017. Communicating and acting: Understanding gesture in simulation semantics. In *IWCS 2017–12th International Conference on Computational Semantics–Short papers*.

Nikhil Krishnaswamy, William Pickard, Brittany Cates, Nathaniel Blanchard, and James Pustejovsky. 2022. The VoxWorld platform for multimodal embodied agents. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1529–1541.

Nikhil Krishnaswamy and James Pustejovsky. 2020. VoxML specification 1.0. *Unpublished*.

Nikhil Krishnaswamy and James Pustejovsky. 2022. Affordance embeddings for situated language understanding. *Frontiers in Artificial Intelligence*, 5.

Kiyong Lee. 2016. An abstract syntax for ISO-Space with its <moveLink> reformulated. In *Proceedings of the 12th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-13)*, pages 107–118, Workshop at the 12th International Conference on Computational Semantics (IWCS 2017), Montpellier, France.

Kiyong Lee. 2017. Semantic annotation of anaphoric links in language. *Linguistics and Literature Studies*, 5.4:248–270. DOI: 10.131189/lls.2017.050403.

Kiyong Lee. 2023. *Annotation-Based Semantics for Space and Time in Language*. Cambridge University Press, Cambridge, UK.

Inderjeet Mani and James Pustejovsky. 2012. *Intepreting Motion: Grounded Representation for Spatial Language*. Oxford University Press, Oxford.

David G McNeely-White, Francisco R Ortega, J Ross Beveridge, Bruce A Draper, Rahul Bangar, Dhruva Patil, James Pustejovsky, Nikhil Krishnaswamy, Kyeongmin Rim, Jaime Ruiz, et al. 2019. User-aware shared perception for embodied agents. In *2019 IEEE International Conference on Humanized Computing and Communication (HCC)*, pages 46–51. IEEE.

Ian Niles and Adam Pease. 2001. Toward a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*. Ogunquit, Maine.

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

James Pustejovsky. 2013. Dynamic event structure and habitat theory. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)*, pages 1–10. Association for Computational Linguistics, Pisa, Italy.

James Pustejovsky and Olga Batiukova. 2019. *The lexicon*. Cambridge University Press.

James Pustejovsky and Bran Boguraev. 1993. Lexical knowledge representation and natural language processing. *Artificial Intelligence*, 63:193–223.

James Pustejovsky and Nikhil Krishnaswamy. 2016. VoxML: A visualization modeling language. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4606–4613, Portorož, Slovenia. ELRA. ACL anthology L16-1730.

James Pustejovsky and Nikhil Krishnaswamy. 2021. Embodied human-computer interaction. *KI-Künstliche Intelligenz*, 35(3-4):307–327.

James Pustejovsky, Nikhil Krishnaswamy, and Tuan Do. 2017. Object embodiment in a multimodal simulation. In *AAAI Spring Symposium: Interactive Multisensory Object Perception for Embodied Agents*.