Multimodal Semantic Grounding for Communicating with Computers

James Pustejovsky, Nikhil Krishnaswamy
Keigh Rim, Mark Hutchens, Ken Lai, Kelley Lynch
Brandeis University

CwC PI Meeting
Tufts@Boston, MA
October 28, 2019

Outline - Collaboration with CSU and UF

- Providing a multimodal semantics for communication
- Handling interruptions of agent actions
- Handling asynchronous input from multiple modalities
- System Refactoring and Redesign
  - VoxSim refactored
  - VoxWorld enabled as platform architecture
- Continuation Semantics for Multimodal Communication
  - Multimodal Simulation Grammar (MSG) - *Umami*
  - sequencing gestures and language to create meaningful compositions
  - linking gestures to speech
  - grounding contextualized actions to situated objects
- Future work
  - Adopting VoxWorld to Agent Bob
  - Fleshing out the MSG *Umami*

DARPA's Hallmarks of Communication

- Makes appropriate use of multiple modalities
  Machine vision, language, gesture
- Interaction has mechanisms to move the conversation forward
  Separate High-level and Task-related Dialogue Managers
- Each interlocutor can steer the course of the interaction
  Human specifies goals; avatar asks for clarification
- Both parties can clearly reference items in the interaction based on their frames of reference (FoR)
  Ensemble reference using deixis, language, and FoR
- Both parties demonstrate knowledge of the changing situation
  Visualizing the epistemic state (EpiSim) and showing affect
- Habitability: language generation and understanding is aligned
  Both generation and understanding are driven by same semantic representations
- Properties of Multimodal Communication:
  Asynchronous interactions and interruptions

Diana reacts to Agent's Affect

Diana reacts to Agent's Affect

Not as scary as

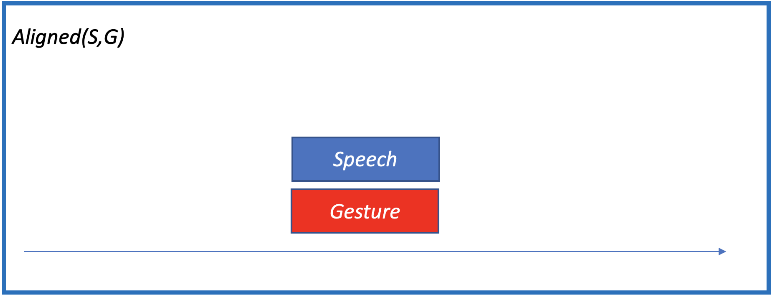
Our Approach - Situated Semantic Grounding

- When two or more people are engaged in dialogue during a shared experience, they share a common ground, which facilitates situated communication.
- Situated semantic grounding assumes shared perception of agents with co-attention over objects in a situated context, with co-intention towards a common goal.
- VoxWorld is a simulation framework for modeling multimodal situated interactions and communication between agents engaged in a shared goal or task (peer-to-peer communication).

Aligning Modalities in Communication

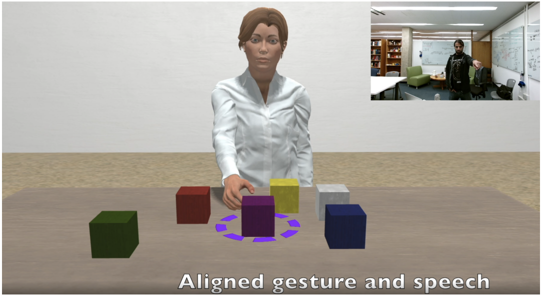
- When two or more people are engaged in dialogue during a shared experience, they share a common ground, which facilitates situated communication.
- Situated semantic grounding assumes shared perception of agents with co-attention over objects in a situated context, with co-intention towards a common goal.
- When a communicative act uses multiple modalities, such as language and gesture, these acts need to be aligned or coordinated, in oder to be interpretable.
    - An utterance, gesture, or action can be interrupted;
    - Channels can convey signals asynchronously

Aligning Modalities in Communication



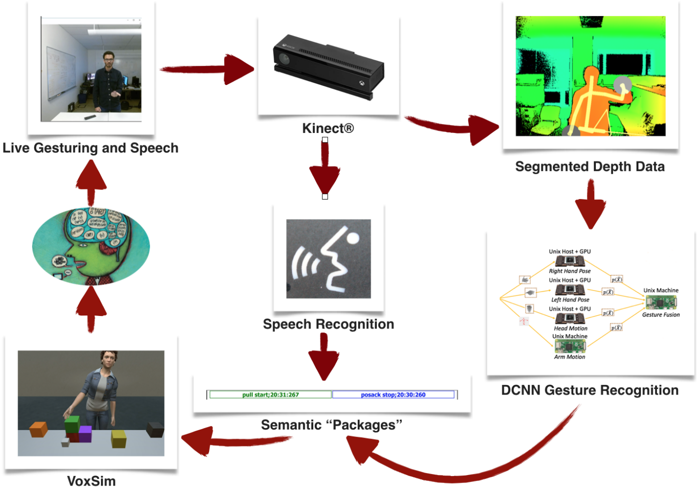
Current Synchronous Communication

Language and Gesture Align

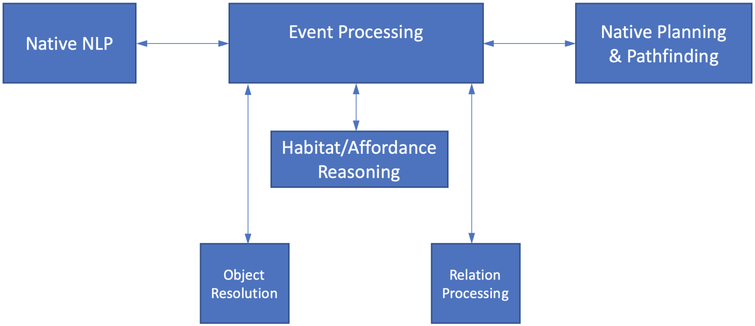


▸ Link

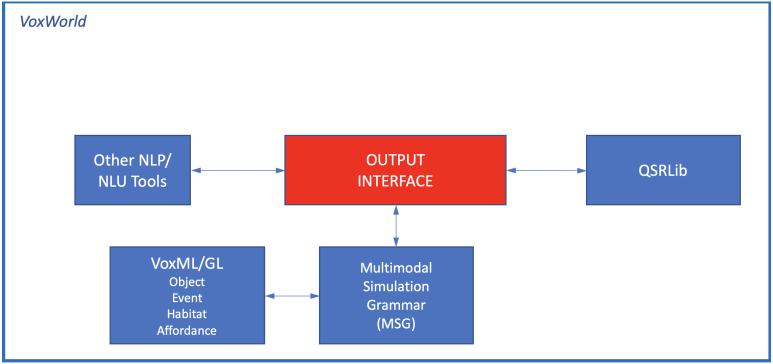
Previous Diana's World Architecture



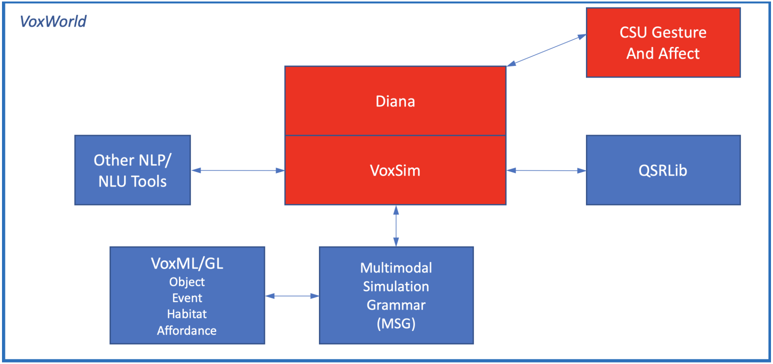
Refactoring VoxSim



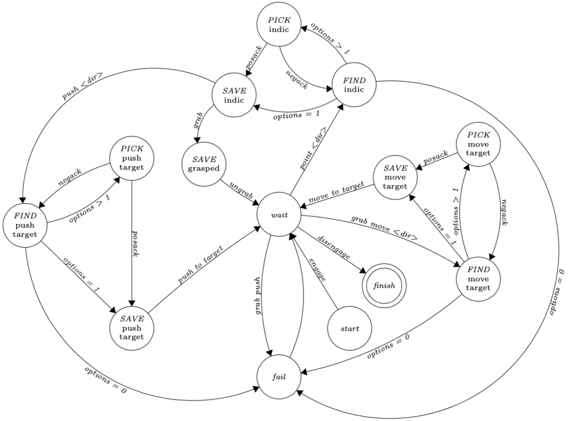
Refactoring VoxWorld as a Platform with Components



Refactoring VoxWorld as a Platform with Components

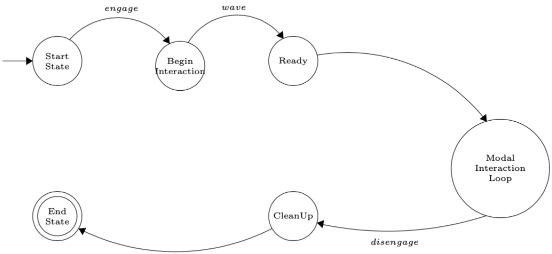


Previous Dialogue Manager



Refactored High-level Dialogue Manager

Separate from task-related interactions



Spatial Reasoning in VoxWorld

VoxML for Actions and Relations

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

$$
\begin{bmatrix}
\textbf{on} \\
\text{LEX} = \begin{bmatrix} \text{PRED} = \textbf{on} \end{bmatrix} \\
\text{TYPE} = \begin{bmatrix}
\text{CLASS} = \textbf{config} \\
\text{VALUE} = \textbf{EC} \\
\text{ARGS} = \begin{bmatrix} \text{A}_1 = \textbf{x:3D} \\ \text{A}_2 = \textbf{y:3D} \end{bmatrix} \\
\text{CONSTR} = \textbf{y}\rightarrow\text{HABITAT}\rightarrow\text{INTR}[align]
\end{bmatrix}
\end{bmatrix}
$$

Spatial Reasoning in C#

Qualitative Relations

```
// externally connected
public static bool EC(Bounds x, Bounds y) {
    bool ec = false;

    // if y and z dimensions overlap
    if (Mathf.Abs(x.center.y - y.center.y) * 2 < (x.size.y + y.size.y) &&
        (Mathf.Abs(x.center.z - y.center.z) * 2 < (x.size.z + y.size.z))) {
        if ((Mathf.Abs(x.min.x - y.max.x) < Constants.EPSILON * 2) || // if touching on x
            (Mathf.Abs(x.max.x - y.min.x) < Constants.EPSILON * 2)) {
            ec = true;
        }
        else {
            Debug.Log(Mathf.Abs(x.min.x - y.max.x));
            Debug.Log(Mathf.Abs(x.max.x - y.min.x));
        }
    }

    // if x and z dimensions overlap
    if (Mathf.Abs(x.center.x - y.center.x) * 2 < (x.size.x + y.size.x) &&
        (Mathf.Abs(x.center.z - y.center.z) * 2 < (x.size.z + y.size.z))) {
        Debug.Log(x.min.y);
        Debug.Log(y.max.y);
        if ((Mathf.Abs(x.min.y - y.max.y) < Constants.EPSILON * 2) || // if touching on y
            (Mathf.Abs(x.max.y - y.min.y) < Constants.EPSILON * 2)) {
            ec = true;
        }
    }

    // if x and y dimensions overlap
    if (Mathf.Abs(x.center.x - y.center.x) * 2 < (x.size.x + y.size.x) &&
        (Mathf.Abs(x.center.y - y.center.y) * 2 < (x.size.y + y.size.y))) {
```

Spatial Reasoning in C#

Motion Actions

```
GlobalHelper.PrintRDFTriples(rdfTriples);

if (prep == "_on") {
    // fix for multiple RDF triples
    if (args[0] is GameObject) {
        if (args[1] is Vector3) {
            GameObject theme = args[0] as GameObject; // get theme obj ("apple" in "put apple on plate")
            GameObject
                dest = GameObject.Find(rdfTriples[0]
                    .Item3); // get destination obj ("plate" in "put apple on plate")
            Voxeme voxComponent = theme.GetComponent<Voxeme>();

            //Renderer[] renderers = obj.GetComponentsInChildren<Renderer> ();
            /*Bounds bounds = new Bounds ();

            foreach (Renderer renderer in renderers) {
                if (renderer.bounds.min.y - renderer.bounds.center.y < bounds.min.y - bounds.center.y) {
                    bounds = renderer.bounds;
                }
            }*/

            List<GameObject> themeChildren = theme.GetComponentsInChildren<Renderer>().Where(
                    o => (GlobalHelper.GetMostImmediateParentVoxeme(o.gameObject) != theme)).Select(v => v.gameObject)
                .ToList();

            List<GameObject> destChildren = dest.GetComponentsInChildren<Renderer>().Where(
                    o => (GlobalHelper.GetMostImmediateParentVoxeme(o.gameObject) != dest)).Select(v => v.gameObject)
                .ToList();

            Debug.Log(GlobalHelper.VectorToParsable(GlobalHelper.GetObjectWorldSize(theme).size));
            Bounds themeBounds = GlobalHelper.GetObjectWorldSize(theme, themeChildren); // bounds of theme obj
            Bounds
                destBounds =
                    GlobalHelper.GetObjectWorldSize(
                        dest); // bounds of dest obj => alter to get interior enumerated by VoxML structure
            Debug.Log(GlobalHelper.VectorToParsable(themeBounds.size));

            //Debug.Log (Helper.VectorToParsable(bounds.center));
            //Debug.Log (Helper.VectorToParsable(bounds.min));

            float yAdjust = (theme.transform.position.y - themeBounds.center.y);
            Debug.Log("Y-size = " + (themeBounds.center.y - themeBounds.min.y));
            Debug.Log("put on: " + (theme.transform.position.y - themeBounds.min.y));
```

Spatial Reasoning in VoxWorld

VoxML for Actions and Relations

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

$$
\left[\begin{array}{l}
\textbf{on} \\
\text{LEX} = \left[\begin{array}{lll} \text{PRED} & = & \textbf{on} \end{array}\right] \\
\text{TYPE} = \left[\begin{array}{lll}
\text{CLASS} & = & \textbf{config} \\
\text{VALUE} & = & \textbf{EC} \\
\text{ARGS} & = & \left[\begin{array}{lll} \text{A}_1 & = & \textbf{x:3D} \\ \text{A}_2 & = & \textbf{y:3D} \end{array}\right] \\
\text{CONSTR} & = & \textbf{y}{\rightarrow}\text{HABITAT}{\rightarrow}\text{INTR}[align]
\end{array}\right]
\end{array}\right]
$$

Spatial Reasoning in VoxWorld

VoxML for Actions and Relations

```
<Type>
  <Head>transition</Head>
    <Args>
   <Arg Value="x:agent" />
   <Arg Value="y:physobj" />
   <Arg Value="z:location" />
  </Args>
  <Body>
   <Subevent Value="grasp(x,y)" />
   <Subevent Value="while(hold(x,y)^!
at(y,z)):move(x,y,z,'VoxSimPlatform.Pathfinding.AStarSearch.PlanPath',loc(y),z,y)" />
   <Subevent Value="if(at(y,z)):ungrasp(x,y)" />
  </Body>
 </Type>
```

```
<Type>
  <Class>config</Class>
  <Value>RCC8.EC</Value>
  <Args>
     <Arg Value="x:physobj" />
     <Arg Value="y:physobj" />
  </Args>
  <Constr>Y(x) &gt; Y(y)</Constr>
  <Corresps />
</Type>
```

QSRLib

Gatsoulis et al. (2016)

Table 1: Description of supported qualitative spatial relation families

qualitative spatial relation families	type	num of relations / variations	kind of entities
Qualitative Distance Calculus	distance	user specified	2D points
Probabilistic Qualitative Distance Calculus	distance	user specified	2D points
Cardinal Directions	direction	9	2D rectangles
Moving or Stationary	motion	2	2D points
Qualitative Trajectory Calculus	motion	B11: 9, C21: 81	2D points
Rectangle/Block Algebra	topology & direction	169/2197	2D/3D rectangles
Region Connection Calculus	topology	2, 4, 5, 8	2D rectangles
Ternary Point Configuration Calculus	direction	25	2D points

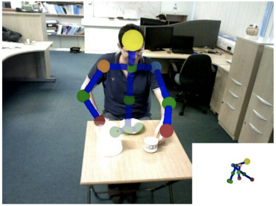


Figure 1: Activity recognition in a table top setting. Dyadic QSR relations between detected objects/skeleton points can be computed (bottom right inset).

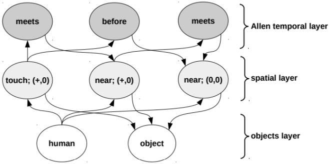
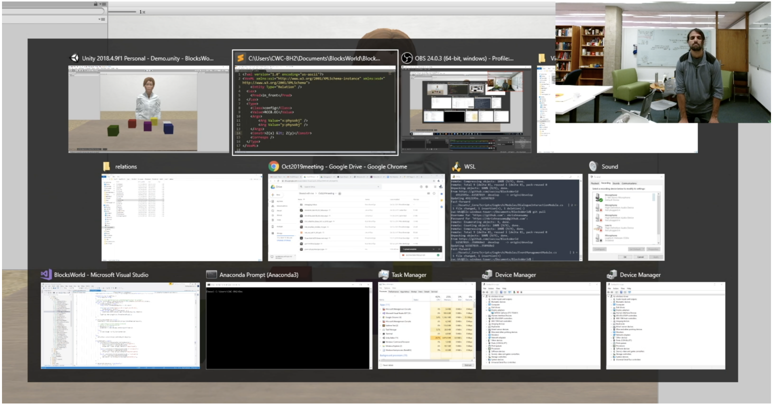


Figure 5: Example of a Qualitative Spatio-Temporal Activity Graph (QSTAG) between a human and an object; each spatial layer node encodes QSRs from two calculi: a QDC relation (touch/near) and a QTC_{B21} one ((+,0)/(0,0)).

VoxML Refactoring - Under the Hood



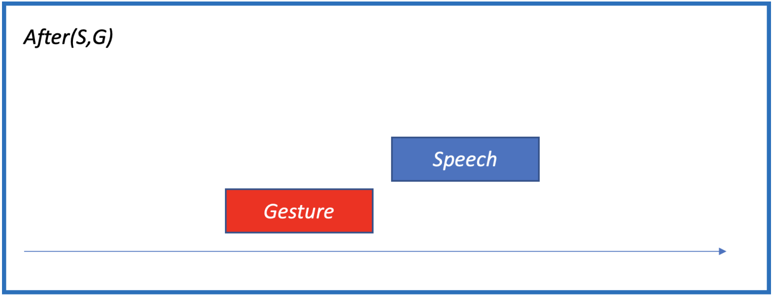
▸ Link

Interruption during Dialogue

Undoing Action on a Specific Block

Link

Non-aligned Modalities in Communication



Link

Multimodal Simulation Grammar (MSG)

Umami

- Grammar for multimodal communicative sequencing using:
  - language
  - gesture
  - action
- Vocabulary
  - VoxML objects (voxeme)
  - Common Ground Structures (cgs)
- Rules
  - Discourse Sequence Grammar
  - Gesture Grammar
  - Action Grammar

Multimodal Simulation Grammar (MSG)

Visual Object Concept Modeling Language (VoxML)

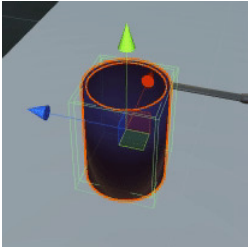
- Encodes afforded behaviors for each object
  - Gibsonian: afforded by object structure (Gibson,1977,1979)
    - grasp, move, lift, etc.
  - Telic: goal-directed, purpose-driven (Pustejovsky, 1995, 2013)
    - drink from, read, etc.
- Voxeme
  - Object Geometry: Formal object characteristics in R3 space
  - Habitat: Orientation, Situated context, Scaling
  - Affordance Structure:
    - What can one do to it
    - What can one do with it
    - What does it enable

Multimodal Simulation Grammar (MSG)

VoxML - cup

cup

- LEX = [PRED = **cup**, TYPE = **physobj, artifact**]
- TYPE = [HEAD = **cylindroid[1]**, COMPONENTS = **surface, interior**, CONCAVITY = **concave**, ROTATSYM = $\{Y\}$, REFLECTSYM = $\{XY, YZ\}$]
- HABITAT = [INTR = [2] [CONSTR = $\{Y > X, Y > Z\}$, UP = $align(Y, \mathcal{E}_Y)$, TOP = $top(+Y)$], EXTR = [3] [UP = $align(Y, \mathcal{E}_{\perp Y})$]]
- AFFORD_STR =
  - $A_1 = H_{[2]} \rightarrow [put(x, on([1]))]support([1], x)$
  - $A_2 = H_{[2]} \rightarrow [put(x, in([1]))]contain([1], x)$
  - $A_3 = H_{[2]} \rightarrow [grasp(x, [1])]$
  - $A_4 = H_{[3]} \rightarrow [roll(x, [1])]$
- EMBODIMENT = [SCALE = **<agent**, MOVABLE = **true**]



Multimodal Simulation Grammar (MSG)

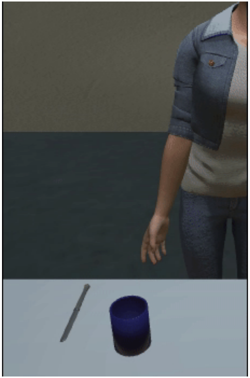
VoxML - grasp

$$
\begin{bmatrix}
\textbf{grasp} \\
\text{LEX} = \begin{bmatrix} \text{PRED} = \textbf{grasp} \\ \text{TYPE} = \textbf{transition\_event} \end{bmatrix} \\
\text{TYPE} = \begin{bmatrix} \text{HEAD} = \textbf{transition} \\ \text{ARGS} = \begin{bmatrix} \text{A}_1 = \textbf{x:agent} \\ \text{A}_2 = \textbf{y:physobj} \end{bmatrix} \\ \text{BODY} = \begin{bmatrix} \text{E}_1 = grasp(x, y) \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Multimodal Simulation Grammar (MSG)

VoxML - grasp cup

- Continuation-passing style semantics for composition
- Used within conventional sentence structures and between sentences in discourse in MSG



Multimodal Simulation Grammar (MSG)

Common Ground Structure (CGS)

(1) State Monad: $M\alpha = State \rightarrow (\alpha \times State)$

(2) a. **A**: The agents engaged in communication;
b. **B**: The shared belief space;
c. **P**: The objects and relations that are jointly perceived in the environment;
d. $\mathcal{E}$: The embedding space that both agents occupy in the communication.

(3) $$\left[\begin{array}{l} \textbf{A:}a_1, a_2 \quad \textbf{B:}\Delta \quad \textbf{P:}b \\ \hline \mathcal{S}_{a_1} = \text{“You}_{a_2}\text{ see it}_b\text{”} \end{array}\right]_{\mathcal{E}}$$

Multimodal Simulation Grammar (MSG)

Multimodal Configurations

- A communicative act, performed by an agent, a, is a tuple of expressions from the modalities available to a, involved in conveying information to another agent.
- We restrict this to the modalities of a linguistic utterance, S (either an intonational contour or speech), and a gesture, G. There are three possible configurations in performing C:
    1. $C_a = (G)$
    2. $C_a = (S)$
    3. $C_a = (S, G)$
- These modal channels can be aligned or unaligned in the input.

Multimodal Simulation Grammar (MSG)

Continuation-based Updating

- An agent's communicative act, C_i, (S_i, G_i)
- generated in a common ground context, cg_i;
- Updating context of the state monad, cg_i, through C_i is a Continuation-passing transformation.

- $D \rightarrow C_1\ C_2\ C_3$
- $[[\overline{(C_1 C_2)}]]^{M,cg} = \lambda k.\overline{C_1}(\lambda n.\overline{C_2}(\lambda m.k(m\ n)))$
- $[[\overline{(C_1 C_2 C_3)}]]^{M,cg} = \lambda k'.\overline{(C_1 C_2)}(\lambda r.\overline{C_3}(\lambda k.k'(k\ r)))$
- $[[\overline{(C_1 C_2)}]]^{M,cg} =$ $\lambda k_s \otimes k_g.\overline{C_1}(\lambda n_s \otimes n_g.\overline{C_2}(\lambda m_s \otimes m_g.k_s \otimes k_g(m_s \otimes m_g\ n_s \otimes n_g)))$
- $[[\overline{(C_1 C_2 C_3)}]]^{M,cg} =$ $\lambda k'_s \otimes k'_g.\overline{(C_1 C_2)}(\lambda r_s \otimes r_g.\overline{C_3}(\lambda k_s \otimes k_g.k'_s \otimes k'_g(k_s \otimes k_g\, r_s \otimes r_g)))$

Multimodal Simulation Grammar (MSG)

Object Affordances: Gibsonian and Telic

- Objects are antecedents to actions
  - block: Pick me up!, Move me!
  - cup: Pick me up!, Drink what's in me!
  - knife: Pick me up!, Cut that with me!
- Affordances are a subclass of continuations
  - $\lambda k_{Gib} \otimes k_{Telic}.k_{Gib} \otimes k_{Telic}(cup)$
    $grab \subseteq \textbf{sel}\ k_{Gib}$
    $drink \subseteq \textbf{sel}\ k_{Telic}$
  - $\lambda k_{Gib} \otimes k_{Telic}.k_{Gib} \otimes k_{Telic}(block)$
    $grab \subseteq \textbf{sel}\ k_{Gib}$
    $pick\_up \subseteq \textbf{sel}\ k_{Gib}$
    $move \subseteq \textbf{sel}\ k_{Gib}$

Actions as Described by Gesture

Kendon (2004), Lascarides and Stone (2009)

- $G = (prep); (prestrokehold); stroke; retract$

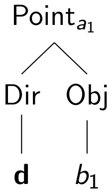
The stroke is the content-baring phase, **d**, and in a pointing gesture, will convey the deictic orientational information.

- $[[point]] = [[End(cone(\mathbf{d}))]]$
- Gestures can denote a range of primitive action types, including: grasp, hold, pick up, move, throw, pull, push, separate, and put together.

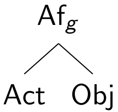
Gesture Grammar

Pustejovsky (2018)

(4) a. **Deixis**: $Point_g \rightarrow Dir \;\; Obj$

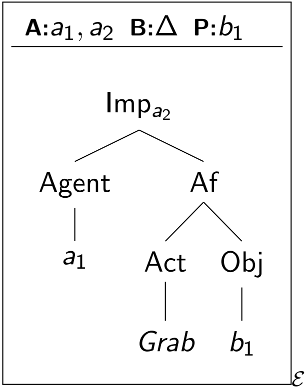


b. **Affordance**: $Af_g \rightarrow Act \;\; Obj$



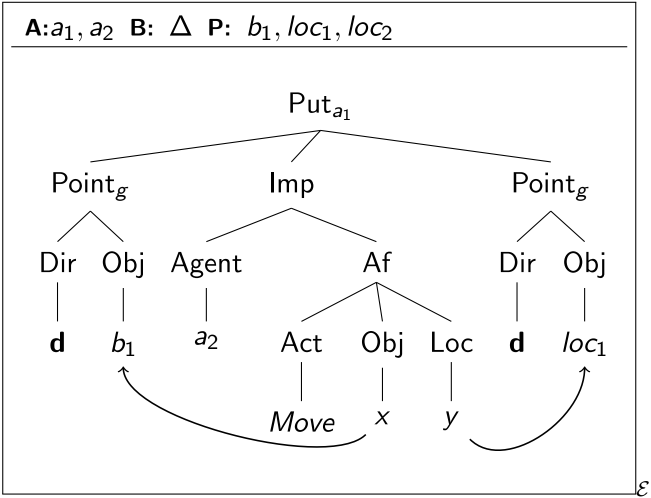
Gestures denoting Affordances

(5) a. $Grab_g \rightarrow Act \;\; Obj$
b. $Push_g \rightarrow Act \;\; Obj$
c. $Throw_g \rightarrow Act \;\; Obj$



(6) $\lambda k.k(grab)$

a_1: "That object b_1 move b_1 to there, the location loc_1."

(7) 

(8) $\lambda k'.\overline{(Point_1 Move)}(\lambda r.\overline{Point_2}(\lambda k.k'(k\ r)))$

Situated Communication

Multimodal Continuations

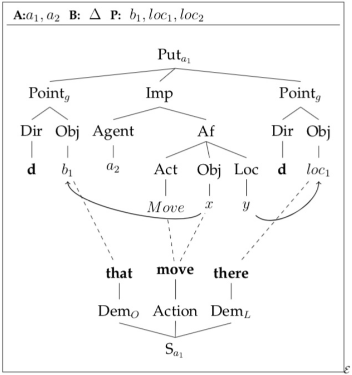
A multimodal communicative act, C, consists of a sequence of gesture-language ensembles, (g_i, s_i), where an ensemble is temporally aligned in the common ground:

(9) $C = (g_1, s_1); \ldots; (g_i, s_i); \ldots; (g_n, s_n)$.

(10) **Co-gestural Speech Ensemble**: multimodal communication with Gesture, $\mathcal{G}$, and Speech, $\mathcal{S}$:

$$\begin{bmatrix} \mathcal{G} & g_1 & g_i & g_n \\ \mathcal{S} & s_1 & s_i & s_n \end{bmatrix}$$

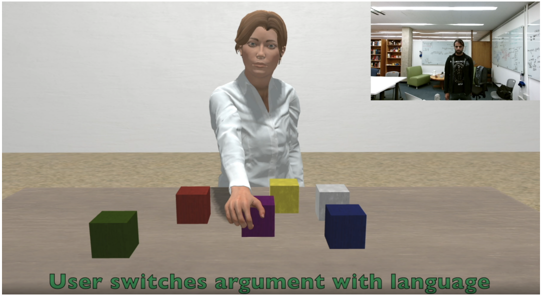
a_1: "**That** object b_1 **move** b_1 to **there**, location loc_1."



$$\lambda k'_s \otimes k'_g.\overline{(\langle \mathbf{that}, Point_1\rangle\langle \mathbf{move}, Move\rangle)}(\lambda r_s \otimes r_g.\overline{\langle \mathbf{that}, Point_2\rangle}$$
$$(\lambda k_s \otimes k_g.k'_s \otimes k'_g(k_s \otimes k_g r_s \otimes r_g)))$$

Interruption during Dialogue - Under the Hood

Correcting and Undoing Parameter Binding in Actions



▸ Link

Interruption during Dialogue - Under the Hood

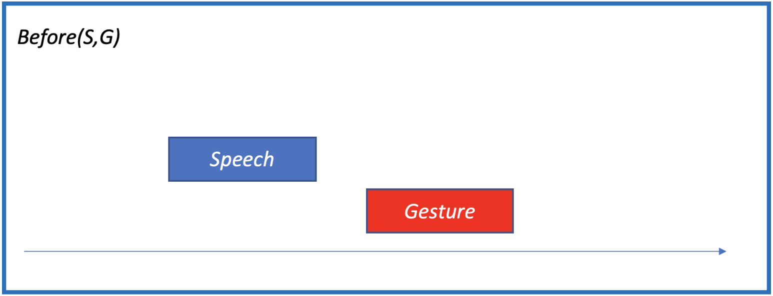
Correcting and Undoing Parameter Binding in Actions

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

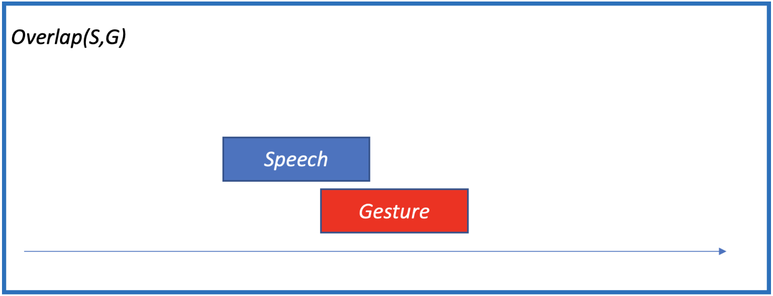
- $\lambda k_{Gib} \otimes k_{Telic}.k_{Gib} \otimes k_{Telic}(block)$
- $grab \subseteq \textbf{sel}\ k_{Gib}$
- $\lambda k.k(grab) \implies M, cg_1 \vDash grab(purple)$

- *"Wait, the yellow one."*
- **undo** $k = \lambda k.k(grab)$
- **Rewind** the state monad and **Reassign**:
- $\lambda k_{Gib} \otimes k_{Telic}.k_{Gib} \otimes k_{Telic}(block)$
- $grab \subseteq \textbf{sel}\ k_{Gib}$
- $\lambda k.k(grab) \implies$
- $M, cg_1 \vDash grab(yellow)$

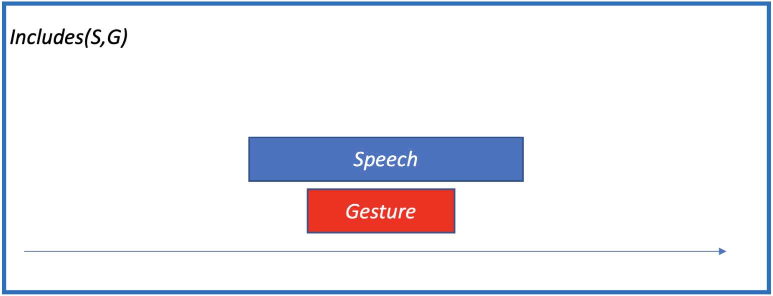
Aligning Modalities - Under the Hood



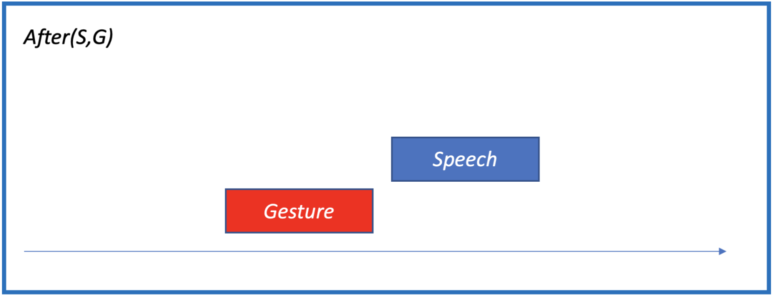
Aligning Modalities - Under the Hood



Aligning Modalities - Under the Hood

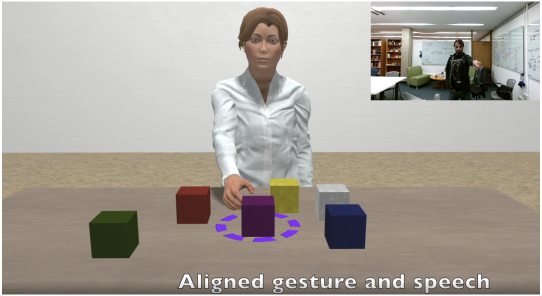


Aligning Modalities - Under the Hood



Aligning Modalities - Under the Hood

Language and Gesture are not Aligned



Link

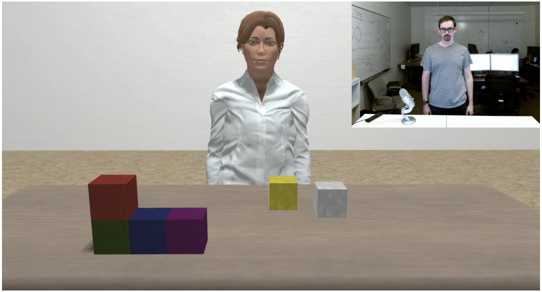
Communicating with Continuations 1/3

Object Gesture Denotation precedes Action Gesture Recognition

Link

Communicating with Continuations 2/3

Object Language Denotation precedes Location Gesture Denotation



Link

Communicating with Continuations 2/3

Principles [following de Groote(2006)]

$$
\begin{aligned}
[\![s]\!] &= \gamma \rightarrow (\gamma \rightarrow t) \rightarrow t \quad \triangleq \Omega \\
[\![np]\!] &= (e \rightarrow [\![s]\!]) \rightarrow [\![s]\!] \\
[\![n]\!] &= e \rightarrow [\![s]\!]
\end{aligned}
$$

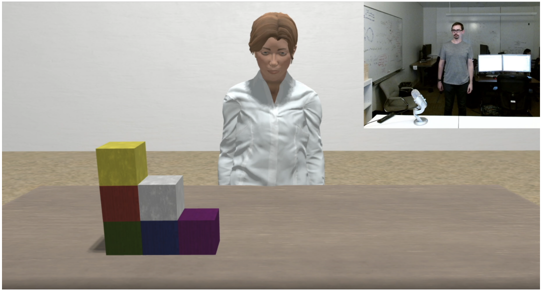
$$[\![T.S]\!] = \lambda i.\lambda k.[\![T]\!] \; i \; (\lambda i'.[\![S]\!] \; i' \; k)$$

Example

The blue block.	Move it.
$\lambda i.\lambda k.\text{the}(x).\,((\text{block}\ x) \wedge (\text{blue}\ x)) \wedge (k\,(x :: i))$	$\lambda i.\lambda k.(\text{move}\,(\texttt{sel}\ i)) \wedge (k\ i)$

Communicating with Continuations 3/3

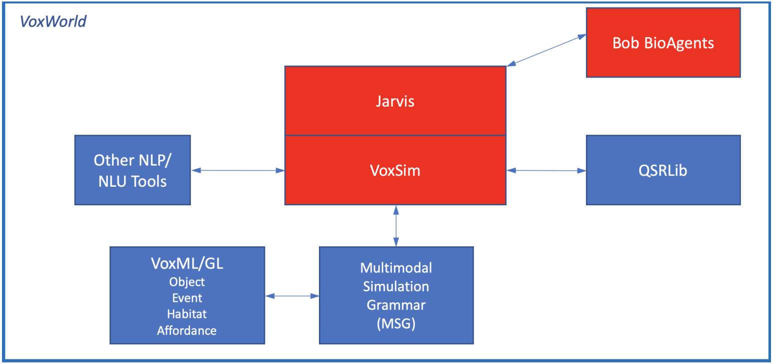
Action Recognition is inferred from Location Gesture Denotation



Link

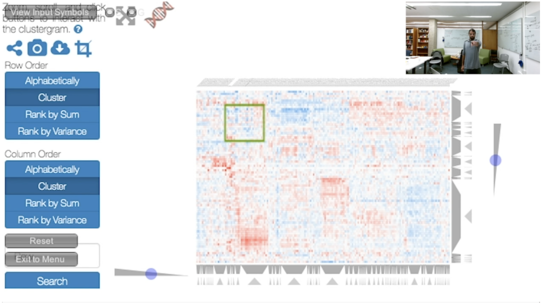
Adopting VoxWorld to another Agent

Interfacing VoxWorld to Agent Bob



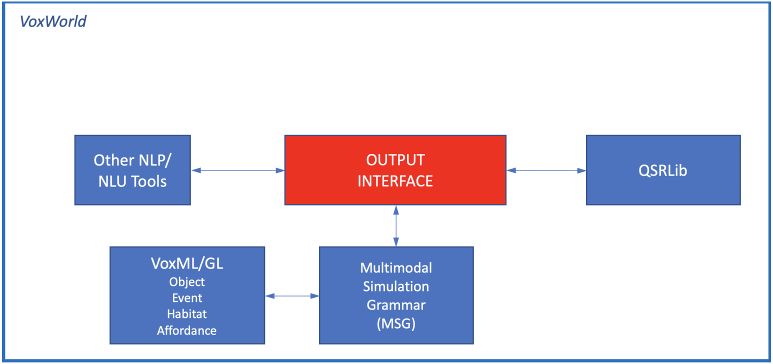
Interfacing VoxWorld to Agent Bob - *Hello Jarvis*

Multimodal Manipulation and Exploration of Data

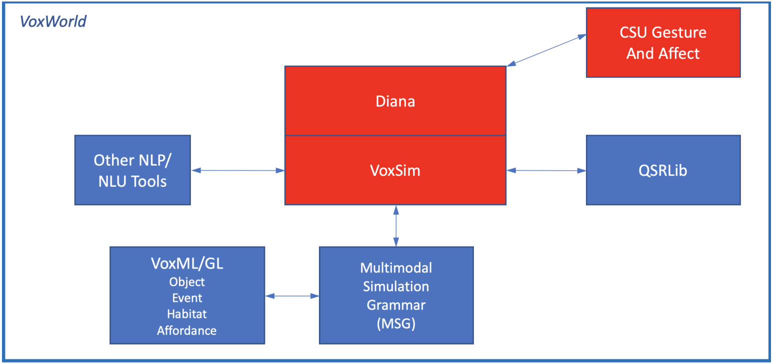


Link

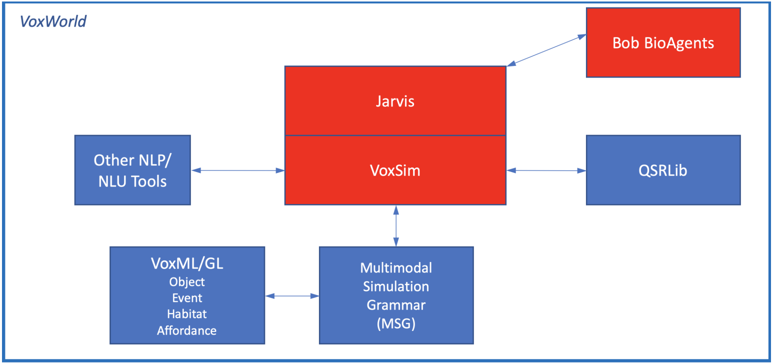
Refactoring VoxWorld as a Platform with Components



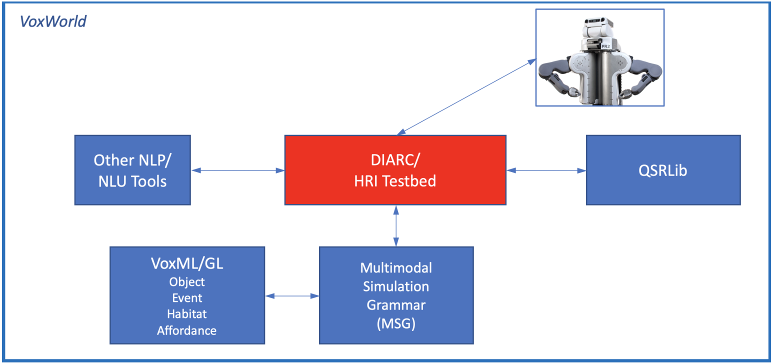
Refactoring VoxWorld as a Platform with Components



Refactoring VoxWorld as a Platform with Components



Refactoring VoxWorld as a Platform with Components



CwC Related Publications

- Krishnaswamy, N. and Pustejovsky, J. (2019). Multimodal Continuation-style Architectures for Human-Robot Interaction. In Workshop on Cognitive Vision: Integrated Vision and AI for Embodied Perception and Interaction. Cognitive Systems.
- Krishnaswamy, N. and Pustejovsky, J. (2019). Generating a Novel Dataset of Multimodal Referring Expressions. In International Workshop on Computational Semantics (IWCS). ACL.
- Lai, Kenneth, and James Pustejovsky. "A Dynamic Semantics for Causal Counterfactuals." In Proceedings of the 13th International Conference on Computational Semantics, pp. 1-8. 2019.
- Krishnaswamy, N., Friedman, S., and Pustejovsky, J. (2019). Combining Deep Learning and Qualitative Spatial Reasoning to Learn Complex Structures from Sparse Examples with Noise. In AAAI Conference on Artificial Intelligence, AAAI.
- Pustejovsky, J. and Krishnaswamy, N. (2019). Situational Grounding within Multimodal Simulations. In AAAI Workshop on Games and Simulations in AI (GameSim). AAAI.

CwC Related Publications

- Krishnaswamy, N. and Pustejovsky, J. (2018). Deictic Adaptation in a Virtual Environment. In Spatial Cognition XI: International Conference on Spatial Cognition. Springer.
- Pustejovsky, J. and Krishnaswamy, N. (2018). The Role of Event Simulation in Spatial Cognition. In Workshop on Models and Representations in Spatial Cognition (MRSC). Springer.
- Narayana, P., Krishnaswamy, N., Wang, I., Bangar, R., Patil, D., Mulay, G., Rim, K., Beveridge, R., Ruiz, J., Pustejovsky, J., and Draper, B. (2018). Cooperating with Avatars Through Gesture, Language and Action. In Intelligent Systems Conference, IEEE.
- Do, T., Krishnaswamy, N., Rim, K., and Pustejovsky, J. (2018). Multimodal Interactive Learning of Primitive Actions. In AAAI Fall Symposium: Artificial Intelligence for Human-Robot Interaction. AAAI.
- Pustejovsky, J. and Krishnaswamy, N. (2018). Every Object Tells a Story. In Workshop on Events and Stories in the News (EventStory). ACL.

Accepted for Presentation/Publication - 2019-2020

- Visually Grounded Interaction and Language Workshop (ViGIL)@ NeurIPS 2019
  Situated Grounding Facilitates Multi- modal Concept Learning for AI
  Nikhil Krishnaswamy and James Pustejovsky
- AAAI-20 Demonstrations Program:
  Diana's World: A Situated Multimodal Interactive Agent
  Nikhil Krishnaswamy, Pradyumna Narayana, Rahul Bangar, Kyeongmin Rim, Dhruva Patil, David White, Jaime Ruiz, Bruce Draper, Ross Beveridge and James Pustejovsky
- Avios Conversational Interaction Conference 2020
  Multimodal Communication with Computers and Robots
  James Pustejovsky and Nikhil Krishnaswamy

Future Work

- Enrich the Multimodal Simulation Grammar (Umami)
- Create a full prototype of Jarvis for Agent Bob
- Integrate structure and concept learning with ECIs for examinable models
- Integrate one-shot gesture learning into concept and action learning
- User modeling and learning from prior interactions

Thank You ☺

- Brandeis LLC lab members: Nikhil Krishnaswamy, Kyeongmin Rim, Mark Hutchens, Ken Lai, Kelley Lynch
- CSU Vision lab members: Ross Beveridge, Bruce Draper, Rahul Bangar, David White, Pradyumna Narayana, Dhruva Patil
- University of Florida lab members: Jaime Ruiz, Isaac Wang
- Funded by a grant from DARPA within the CwC Program

Thank You ☺

- Brandeis LLC lab members: Nikhil Krishnaswamy, Kyeongmin Rim, Mark Hutchens, Ken Lai, Kelley Lynch
- CSU Vision lab members: Ross Beveridge, Bruce Draper, Rahul Bangar, David White, Pradyumna Narayana, Dhruva Patil
- University of Florida lab members: Jaime Ruiz, Isaac Wang
- Funded by a grant from DARPA within the CwC Program

Thank You ☺

- Brandeis LLC lab members: Nikhil Krishnaswamy, Kyeongmin Rim, Mark Hutchens, Ken Lai, Kelley Lynch
- CSU Vision lab members: Ross Beveridge, Bruce Draper, Rahul Bangar, David White, Pradyumna Narayana, Dhruva Patil
- University of Florida lab members: Jaime Ruiz, Isaac Wang
- Funded by a grant from DARPA within the CwC Program