Mapping from Surface to Abstract Event Structures in Language

James Pustejovsky
Brandeis University

Beyond Time
University Colorado Boulder
April 7, 2017

Talk Outline

Beyond Time

Tense, Aspect, and Beyond

Beyond Time

Tense, Aspect, and Beyond

Beyond Time

Tense, Aspect, and Beyond

Beyond Time

"Oh, here are the events."

Extracting Events from Sparse Representations

- Increased popularity and availability of sparse forms
- Improved parsing results

Beyond Time
"Oh, here are the events."

EXTRACTING EVENTS FROM SPARSE REPRESENTATIONS

- Increased popularity and availability of sparse forms
- Improved parsing results
- however

Beyond Time

“Oh, here are the events.”

Extracting Events from Sparse Representations

- Increased popularity and availability of sparse forms
- Improved parsing results
- however
- Challenge to many richer semantic interpretations in language
- Including event structures

Surface Event Representations

- Surface event: Matrix predicate denotes an atomic event;
- Event is reified as first-order individual.

Mary ate the soup.

$\exists e[\mathrm{eat}(e, m, \mathrm{the\_soup})]$

Abstract Event Representations

- Abstract event: All predicates are event denoting, depending on the inference task

Mary cleaned the dirty table.

$$[mary(x), table(y)] : e$$

$$e \to \bar{e}_1 < e_2$$

$$\bar{e}_1 \to e_1 \circ e_3$$

$$e_1 : clean\_act(x, y) \qquad e_3 : \begin{bmatrix} \neg clean(y) \\ dirty(y) \end{bmatrix} \qquad e_2 : \begin{bmatrix} clean(y) \\ \neg dirty(y) \end{bmatrix}$$

Event Semantics

(1) a. Event Argument: Predicates in language have an *event variable* that can be treated as a first-order individual in the semantics, to enable logical inference (Davidson, 1967);
b. Aktionsarten: Predicates in language can be classified according to their event type or aspectual class, in order to specific capture grammatical and semantic behaviors (Vendler, 1967).

Davidson's Event Argument (EA) Theory

(2) a. $\lambda y \lambda x \lambda e[\text{eat}(e, x, y)]$
b. $\lambda z \lambda y \lambda x \lambda e[\text{give}(e, x, y, z)]$

Davidson is able to capture the appropriate entailments between propositions involving action and event expressions through the conventional mechanisms of logical entailment.

Davidson's Event Semantics

(3) a. Mary ate the soup.
b. Mary ate the soup with a spoon.
c. Mary ate the soup with a spoon in the kitchen.
d. Mary ate the soup with a spoon in the kitchen at 3:00pm.

(4) a. $\exists e[\text{eat}(e, m, \text{the\_soup})]$
b. $\exists e[\text{eat}(e, m, \text{the\_soup}) \wedge \text{with}(e, \text{a\_spoon})]$
c. $\exists e[\text{eat}(e, m, \text{the\_soup}) \wedge \text{with}(e, \text{a\_spoon})$
$\wedge \text{in}(e, \text{the\_kitchen})]$
d. $\exists e[\text{eat}(e, m, \text{the\_soup}) \wedge \text{with}(e, \text{a\_spoon}) \wedge$
$\text{in}(e, \text{the\_kitchen}) \wedge \text{at}(e, \text{3:00pm})]$

Aktionsarten – conceptual categories of event types

- Stative vs. Non-stative
- States -Conceived of as not changing over time, as well as extended in time and permanent.

(5) a. John is tall.
b. Mary knows the answer.
c. It is 8:00 p.m.
d. ! John is being tall.

Generally only compatible with simple present, but notice extended use of progressive and subtle meaning differences:

(6) . a. The statue stands in the square.
b. The statue is standing in the square.

Structural vs. Phenomenal distinction – Goldsmith and Woisetschlager (1979)

Temporary vs. permanent states

As seen with the English progressive marking before, states are not always permanent. Other languages also mark these differences (but not always for the same concepts).

- Spanish – *ser* vs. *estar*

  (7) a. Soy enfermo (I am a sickly person)
      b. Estoy enfermo (if I have a cold)

Processes

- Involve change and are extended in time. In present tense they need to be used in the progressive (unless habitual)

(8) . a. John ran a mile in under four minutes.
b. Sheila wrote three letters in an hour.
c. !John ran a mile for six minutes.
d. !Sheila ate an apple for ten minutes.

(9) a. John ran for twenty minutes.
b. Sheila ate apples for two days straight.
c. !John ran in twenty minutes.
d. !Sheila ate apples in two days.

Distinguishing Processes from Transitions

- Activities: Atelic i.e. have no natural endpoint or goal (e.g. *I'm running in the park*) Compatible with a durative adverbial (e.g. *for*) that profiles the amount of time the activity takes.
- Accomplishments: Telic i.e. have a natural endpoint of goal (e.g. *I'm running a mile*) Compatible with a container adverbial (e.g. *in*) that profiles the amount of time taken to reach the desired goal.

Typological Effects

Some languages are more systematic than English in distinguishing indicators of actual and potential terminal points. Thus Swedish use different prepositions:

(10) Jeg reser till Frankrike *på* två månader.
I('m) going to France for two months.

(11) Jeg reste i Frankrike *i* två månader.
I traveled in France for two months.

Achievements and points

Achievements: Events that are conceived of as instantaneous. Often, however, there is an underlying activity that causes a change of state. Their point-like nature tends to require them to be described in the past tense or narrative present.

(12) a. John shattered the window.
b. ! John shatters/is shattering the window.
c. The canals froze.
d. Mary found her keys.
e. *Mary is finding her keys.
f. John reached the top.

Achievements and points

Points: Similar to achievements in being conceived as instantaneous, but without the underlying run-up activity that characterizes gradual achievements

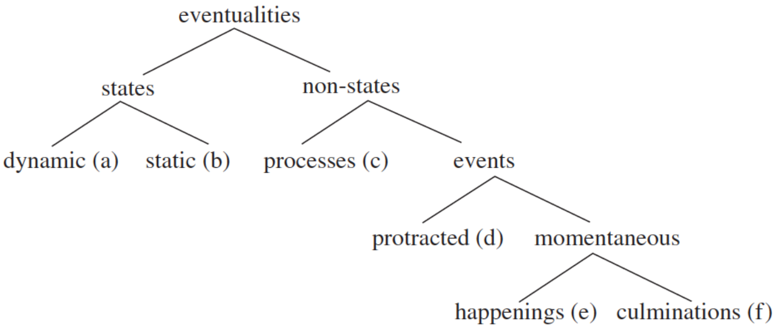
(13) a. Bill coughed.
b. The light flashed.
c. Bill is coughing.
d. The light is flashing.

(c) and (d) have an iterative interpretation. Compare with the gradual achievements *John is reaching the top* or *The canals are freezing*.

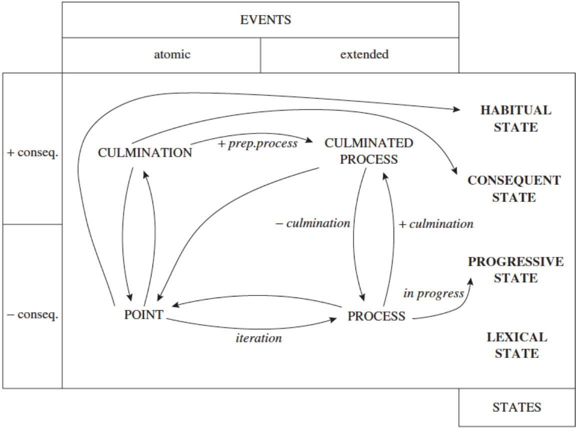
Vendler Event Classes + Semelfactive

- STATE: John loves his mother.
- ACTIVITY: Mary played in the park for an hour.
- ACCOMPLISHMENT: Mary wrote a novel.
- ACHIEVEMENT: John found a Euro on the floor.
- POINT: John knocked on the door (for 2 minutes).

Bach Eventuality Typology (Bach, 1986)



Event Transition Graph
Moens and Steedman (1988)



Subatomic Event Structure

Pustejovsky (1988, 1991)

(14) a. EVENT → STATE | PROCESS | TRANSITION
b. STATE: → e
c. PROCESS: → $e_1 \ldots e_n$
d. TRANSITION$_{ach}$: → STATE STATE
e. TRANSITION$_{acc}$: → PROCESS STATE

Incremental Theme Verbs

- "Certain NP's measure out the event. They are direct objects consumed or created in increments over time (cf. *eat an apple* vs. *push a chart*)" (Tenny 1994).
- In *Mary drank a glass of wine* "every part of the glass of wine being drunk corresponds to a part of the drinking event" (Krifka 1992)
- "Incremental themes are arguments that are completely processed only upon termination of the event, i.e., at its end point" (Dowty 1991).

Degree Achievements

- Verbs with variable aspectual behavior: they seems to be change of state verbs like other achievements , but allow durational adverbs (Dowty 1979, Hay, Kennedy and Levin 1999, Rappaport Hovav 2008).
- No implication that exactly the same change of state took place over and over again (no semelfactives).
- Scalar predicates: verbs which lexically specify a change along a scale inasmuch as they denote an ordered set of values for a property of an event argument (Hay, Kennedy and Levin 1999, Rappaport Hovav 2008).
- For example *cool*, *age*, *lenghten*, *shorten*; *descend*.
- *Let the soup cool for 10 minutes*.
- *I went on working until the soup cooled*.

Points

- Moens and Steedman 1988 analyze point expressions as those that are not normally associated to a consequent state (consequent state defined as no transition to a new state in the world – according to Moens and Steedman a point is an event whose consequences are not at issue in the discourse).
- Semelfactives (Smith 1990, Rothstein 2004).
- **arrived/landed for five minutes*, *knocked/tapped for five minutes*.
- Points admit iterative readings under coercive contexts (Moens and Steedman 1988).

Qualia Structure for Causative

Pustejovsky (1995)

$$
\left[\begin{array}{lll}
\textbf{kill} & & \\
\text{EVENTSTR} & = & \left[\begin{array}{lll} \text{E}_1 & = & \textbf{e}_1\textbf{:process} \\ \text{E}_2 & = & \textbf{e}_2\textbf{:state} \\ \text{RESTR} & = & <_\propto \\ \text{HEAD} & = & \textbf{e}_1 \end{array}\right] \\
\text{ARGSTR} & = & \left[\begin{array}{lll} \text{ARG1} & = & \boxed{\mathit{1}}\left[\begin{array}{lll} \textbf{ind} & & \\ \text{FORMAL} & = & \textbf{physobj} \end{array}\right] \\ \text{ARG2} & = & \boxed{\mathit{2}}\left[\begin{array}{lll} \textbf{animate\_ind} & & \\ \text{FORMAL} & = & \textbf{physobj} \end{array}\right] \end{array}\right] \\
\text{QUALIA} & = & \left[\begin{array}{lll} \textbf{cause-lcp} & & \\ \text{FORMAL} & = & \textbf{dead}(\textbf{e}_2,\ \boxed{2}\ ) \\ \text{AGENTIVE} & = & \textbf{kill\_act}(\textbf{e}_1,\ \boxed{1}\ ,\ \boxed{2}\ ) \end{array}\right]
\end{array}\right]
$$

Opposition Structure

Pustejovsky (2000)

(15) kill

e

$<$

e_1^* e_2

$kill\_act(x, y)$ $dead(y)$

$\neg dead(y)$

(16) break

e

$<$

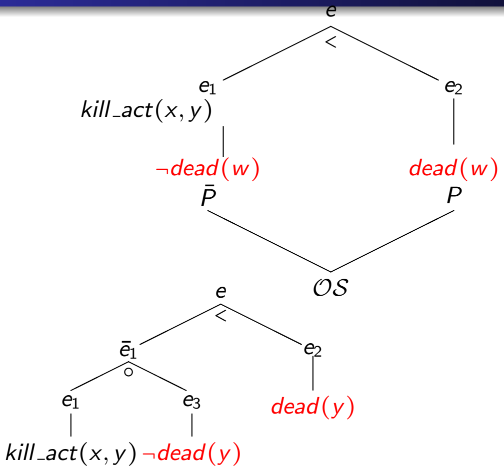
e_1 e_2

$break\_act(x, y)$ $broken(y)$

Qualia Structure with Opposition Structure

$$
\left[\begin{array}{l}
\textbf{kill} \\
\text{EVENTSTR} = \left[\begin{array}{l}
\mathrm{E}_0 = \textbf{e}_0\textbf{:state} \\
\mathrm{E}_1 = \textbf{e}_1\textbf{:process} \\
\mathrm{E}_2 = \textbf{e}_2\textbf{:state} \\
\text{RESTR} = <_{\propto} \\
\text{HEAD} = \textbf{e}_1
\end{array}\right] \\
\text{ARGSTR} = \left[\begin{array}{l}
\text{ARG1} = \boxed{1}\left[\begin{array}{l}\textbf{ind} \\ \text{FORMAL} = \textbf{physobj}\end{array}\right] \\
\text{ARG2} = \boxed{2}\left[\begin{array}{l}\textbf{animate\_ind} \\ \text{FORMAL} = \textbf{physobj}\end{array}\right]
\end{array}\right] \\
\text{QUALIA} = \left[\begin{array}{l}
\textbf{cause-lcp} \\
\text{FORMAL} = \textbf{dead}(\textbf{e}_2, \boxed{2}) \\
\text{AGENTIVE} = \textbf{kill\_act}(\textbf{e}_1, \boxed{1}, \boxed{2}) \\
\text{PRECOND} = \neg\textbf{dead}(\textbf{e}_0, \boxed{2})
\end{array}\right]
\end{array}\right]
$$

Opposition is Part of Event Structure



Constituent Parsing

In the beginning, was the word.

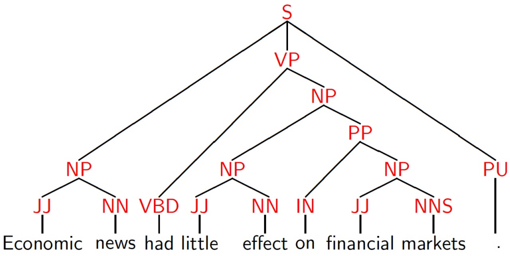
- Word forms are typed and syntagmatically structured;
- The encoding of hidden categories which express complex syntagmatic relations;
- Relations are labels between these categories with varying degrees of complexity (e.g., finite state, context-free, context-sensitive, r.e. sets).

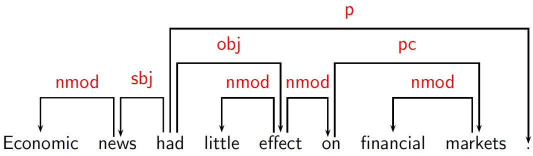
Dependency Parsing

In the end, was the word.

- Nominalist approach to representation of linguistic form;
- The encoding of function rather than syntagmatic relation;
- Relations are labels between words with no abstract categories proposed.

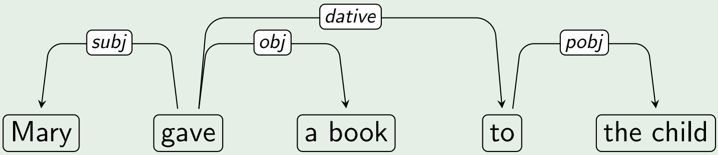
Dependency Parsing





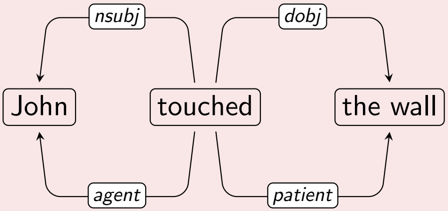
Dependency Parsing - Grammatical Relations

Mary gave a book to the child.



Dependency Parse - Enhanced with Semantic Roles

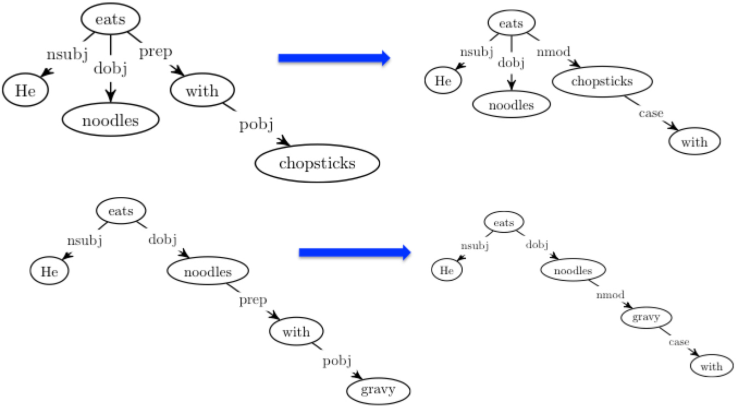
John touched the wall.



Universal Dependencies

- The primacy of content words: Dependency relations hold primarily between content words, rather than being indirect relations mediated by function words.
- Function words normally do not have dependents of their own. Multiple function words related to the same content words are typically siblings
- The dependency relations are described by a mixture of functional and structural notions: advmod vs nmod
- There is some machinery to account for word order variations: nsbj vs nsbjpass
- In coordination structures, the first conjunct is the head, and all other conjuncts depend on it. So are the coordinating conjunctions.

Primacy of Content Words



Making Event Structure Dynamic

- Treat events as programs
- Treat participants as conditions and constraints on the programs
- Event unfolds through time and space (path metaphor)

Qualia Structure for Verbs

- When applied to event-denoting expressions, QS receives the following different interpretation in the model.
- F characterizes predicates denoting stable and persistent verbal predicates, namely states such as *love* and *believe*.
- A refers to the manner in which something happens or changes, i.e. it introduces the causing act or the process of verbs such as *run* and *walk*.
- T introduces the purpose of actions with verbs denoting intentional acts, such as *build* and *clean*.

Formal for Verbs

F can be seen as characterizing predicates denoting stable and persistent verbal predicates, namely states such as *love* and *believe*.

STATE:

$$\left[\begin{array}{l} \textit{love} \\ \text{QUALIA} = \left[\text{F} = \textbf{love\_state} \right] \end{array}\right]$$

Agentive for Verbs

Intentional activities such as those denoted by the verbs *run* and *walk* can be characterized as Agentive Quale verbs.

ACTIVITY:

$$\begin{bmatrix} \textit{run} \\ \text{QUALIA} = \begin{bmatrix} \text{A} = \textbf{run\_act} \end{bmatrix} \end{bmatrix}$$

Qualia for Change-of-State verbs

Change-of-state verbs such as *break* and *open* can be modeled as denoting a static resulting state (Formal) brought about by an activity (Agentive).

CHANGE_STATE:

$$\begin{bmatrix} \textit{break} & \\ \text{QUALIA} = \begin{bmatrix} \text{F} = \textbf{broken} \\ \text{A} = \textbf{break\_act} \end{bmatrix} \end{bmatrix}$$

Telic for verbs

Intentional or directed events such as *build* and *clean* can be viewed as denoting a static intended goal state (Telic) brought about by an activity (Agentive).

ACCOMPLISHMENT:

$$\begin{bmatrix} \textit{build} \\ \text{QUALIA} = \begin{bmatrix} \text{T} = \textbf{build\_goal} \\ \text{A} = \textbf{build\_act} \end{bmatrix} \end{bmatrix}$$

Linguistic Approaches to Defining Paths

- Talmy (1985): Path as part of the Motion Event Frame
- Jackendoff (1983, 1990,1996): Minimal Path
- Langacker (1987): COS verbs as paths
- Goldberg (1995): way-construction introduces path
- Krifka (1998): Temporal Trace function
- Zwarts (2006): event shape: The trajectory associated with an event in space represented by a path.
- Pustejovsky and Moszkowicz (2011): Events as programs with objects leaving trails

GL Feature Structure

$$\begin{bmatrix} \alpha \\ \text{ARGSTR} = \begin{bmatrix} \text{ARG1} = x \\ \ldots \end{bmatrix} \\ \text{EVENTSTR} = \begin{bmatrix} \text{EVENT1} = e1 \\ \text{EVENT2} = e2 \end{bmatrix} \\ \text{QUALIA} = \begin{bmatrix} \text{CONST} = \textbf{what } x \textbf{ is made of} \\ \text{FORMAL} = \textbf{what } x \textbf{ is} \\ \text{TELIC} = e_2\textbf{: function of } x \\ \text{AGENTIVE} = e_1\textbf{: how } x \textbf{ came into being} \end{bmatrix} \end{bmatrix}$$

Inherent Dynamic Aspect of Qualia Structure

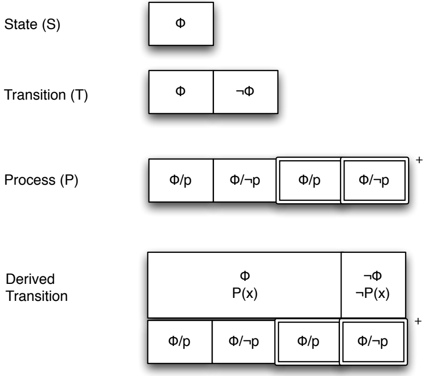
- Parameters of a verb, P, extend over sequential frames of interpretation (subevents).
- P is decomposed into different subpredicates within these events:

$$\text{Verb}(\text{Arg}_1\text{Arg}_2) \implies \lambda y \lambda x \boxed{P_1(x,y)}_A \boxed{P_2(y)}_F$$

Event Structure

- STATE: John loves his mother.
- ACCOMPLISHMENT: Mary wrote a novel.
- ACHIEVEMENT: John found a Euro on the floor.
- PROCESS: Mary played in the park for an hour.
- POINT: John knocked on the door (for 2 minutes).

Frame-based Event Structure



Frame-based Event Structure

Dynamic Interval Temporal Logic
(Pustejovsky and Moszkowicz, 2011)

- Formulas: ϕ propositions. Evaluated in a state, s.
- Programs: α, functions from states to states, $s \times s$. Evaluated over a pair of states, (s, s').
- Temporal Operators: $\bigcirc\phi$, $\Diamond\phi$, $\Box\phi$, $\phi\,\mathcal{U}\psi$.
- Program composition:
  1. They can be ordered, $\alpha;\beta$ (α is followed by β);
  2. They can be iterated, a^* (apply a zero or more times);
  3. They can be disjoined, $\alpha \cup \beta$ (apply either α or β);
  4. They can be turned into formulas
     $[\alpha]\phi$ (after every execution of α, ϕ is true);
     $\langle\alpha\rangle\phi$ (there is an execution of α, such that ϕ is true);
  5. Formulas can become programs, $\phi?$ (test to see if ϕ is true, and proceed if so).

Labeled Transition System (LTS)

The dynamics of actions can be modeled as a Labeled Transition Systems (LTS).

An LTS consists of a 3-tuple, $\langle S, Act, \rightarrow \rangle$, where

(17) a. S is the set of states;
b. Act is a set of actions;
c. $\rightarrow$ is a total transition relation: $\rightarrow \subseteq S \times Act \times S$.

An action, α provides the labeling on an arrow, making it explicit what brings about a state-to-state transition. As a shorthand for $(e_1, \alpha, e_2) \in \rightarrow$, we will also use:

(18) $e_1 \xrightarrow{\alpha} e_3$

If reference to the state content (rather than state name) is required for interpretation purposes, then as shorthand for: $(\{\phi\}_{e_1}, \alpha, \{\neg\phi\}_{e_2}) \in \rightarrow$, we use:

(19) $\boxed{\phi}_{e_1} \xrightarrow{\alpha} \boxed{\neg\phi}_{e_2}$

State Transition

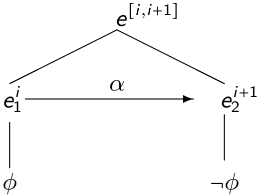
Frame-based representation:

(20) $\boxed{\phi}^{i}_{e_1} \boxed{\neg\phi}^{j}_{e_2}$

(21) $\boxed{\phi}^{i}_{e_1} \xrightarrow{\alpha} \boxed{\neg\phi}^{i+1}_{e_2}$

Dynamic Event Structure

(22)

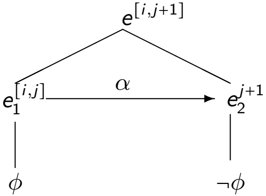


Dynamic Event Structure

(23) Mary awoke from a long sleep.

The state of being asleep has a duration, $[i, j]$, who's valuation is gated by the waking event at the "next state", $j + 1$.

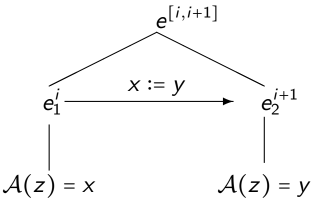
(24)



Dynamic Event Structure

(25) $x := y$ (ν-transition)
"x assumes the value given to y in the next state."
$\langle \mathcal{M}, (i, i+1), (u, u[x/u(y)]) \rangle \vDash x := y$
iff $\langle \mathcal{M}, i, u \rangle \vDash s_1 \wedge \langle \mathcal{M}, i+1, u[x/u(y)] \rangle \vDash x = y$

(26)



Processes

With a ν-transition defined, a *process* can be viewed as simply an iteration of basic variable assignments and re-assignments:

(27)

$$\begin{array}{c} e \\ \diagup \qquad \diagdown \\ e_1 \xrightarrow{\nu} e_2 \; \ldots \xrightarrow{\nu} e_n \end{array}$$

Capturing Motion as Change in Spatial Relations

Dynamic Interval Temporal Logic

- Path verbs designate a distinguished value in the change of location, from one state to another.
  The change in value is tested.
- Manner of motion verbs iterate a change in location from state to state.
  The value is assigned and reassigned.

Motion Leaving a Trail

(28) MOTION LEAVING A TRAIL:

a. Assign a value, y, to the location of the moving object, x.

$$loc(x) := y$$

b. Name this value b (this will be the beginning of the movement);

$$b := y$$

c. Initiate a path p that is a list, starting at b;

$$p := (b)$$

d. Then, reassign the value of y to z, where $y \neq z$

$$y := z, y \neq z$$

e. Add the reassigned value of y to path p;

$$p := (p, z)$$

e. Kleene iterate steps (d) and (e);

Leaving a Trail

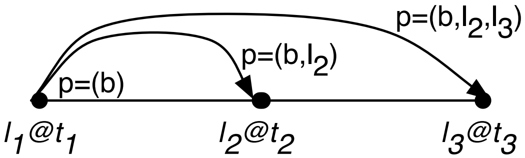


Figure: Directed Motion leaving a Trail

(29) a. The ball rolled 20 feet.
$\exists p \exists x[[roll(x,p) \wedge ball(x) \wedge length(p) = [20, foot]]$

b. John biked for 5 miles.
$\exists p[[bike(j,p) \wedge length(p) = [5, mile]]$

Directed Motion

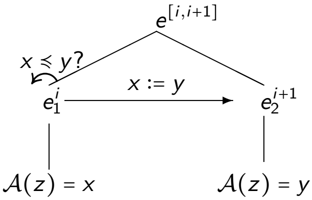
(30) $\overset{\overset{x \neq y?}{\curvearrowleft}}{\boxed{loc(z) = x}}_{e_1} \xrightarrow{\nu} \boxed{loc(z) = y}_{e_2}$

When this test references the ordinal values on a scale, $\mathcal{C}$, this becomes a *directed ν-transition* ($\vec{\nu}$), e.g., $x \preccurlyeq y$, $x \succcurlyeq y$.

(31) $\vec{\nu} =_{df} \overset{\overset{\mathcal{C}?}{\curvearrowleft}}{e_i} \xrightarrow{\nu} e_{i+1}$

Directed Motion

(32)



Change and Directed Motion

- Manner-of-motion verbs introduce an assignment of a location value:
  $loc(x) := y; y := z$
- Directed motion introduces a dimension that is measured against:
  $d(b, y) < d(b, z)$
- Path verbs introduce a pair of tests:
  $\neg\phi? \ldots \phi?$

Change and the Trail it Leaves

- The execution of a change in the value to an attribute $\mathcal{A}$ for an object x leaves a trail, τ.
- For motion, this trail is the created object of the path p which the mover travels on;
- For creation predicates, this trail is the created object brought about by order-preserving transformations as executed in the directed process above.

Dynamic Approach to Scalar Predicates

Klein (1980), Kennedy (1999, 2003)
Gradable adjectives are associated with measurement functions mapping individuals to degrees on a scale.

1. Incremental theme verbs:
   a. Sam ate ice cream. (atelic)
   b. Sam ate an ice cream cone. (telic)
2. Change of state verbs:
   a. The icicle lengthened (over the course of a week). (atelic)
   b. The icicle lengthened two inches. (telic)
3. Directed motion verbs:
   a. The plane ascended (for 20 minutes). (atelic)
   b. The plane ascended to cruising altitude. (telic)

Linguistic View on Scales

- PROPERTY SCALES: often found with change of state verbs
- PATH SCALES: most often found with directed motion verbs
- EXTENT SCALES: most often found with incremental theme verbs

Scale Theory: Stevens (1946), Krantz et al (1971)

- Nominal scales: composed of sets of categories in which objects are classified;
- Ordinal scales: indicate the order of the data according to some criterion (a partial ordering over a defined domain). They tell nothing about the distance between units of the scale.
- Interval scales: have equal distances between scale units and permit statements to be made about those units as compared to other units; there is no zero. Interval scales permit a statement of "more than" or "less than" but not of "how many times more."
- Ratio scales: have equal distances between scale units as well as a zero value. Most measures encountered in daily discourse are based on a ratio scale.

Measurement is a function of two Variables

- The Attribute Domain:
  - Extrinsic Relation: distance, orientation
  - Intrinsic Property: color, volume
- The Scale Theory that interprets it:
  - Nominal
  - Ordinal
  - Interval
  - Ratio

How Language Encodes Scalar Information

- Verbs reference a specific scale;
- We measure change according to this scale domain.
- Scales are introduced by predication (encoded in a verb);
- Scales can be introduced by composition (function application).
- Verbs may reference multiple scales.

Interpreting Events Dynamically

- Some verbs expressing change are associated with a scale while others are not (scalar vs. non-scalar change).

Interpreting Events Dynamically

- Some verbs expressing change are associated with a scale while others are not (scalar vs. non-scalar change).
- There is a single scale domain (ordinal scale), which varies with respect to mereological complexity (two-point vs. multi-point) and specificity of the end point (bounded vs. unbounded).

Interpreting Events Dynamically

- Some verbs expressing change are associated with a scale while others are not (scalar vs. non-scalar change).
- There is a single scale domain (ordinal scale), which varies with respect to mereological complexity (two-point vs. multi-point) and specificity of the end point (bounded vs. unbounded).
- Scales are classified on the basis of the attribute being measured: property (temperature, brightness, length, etc.), path, extent.

Interpreting Events Dynamically

- Some verbs expressing change are associated with a scale while others are not (scalar vs. non-scalar change).
- There is a single scale domain (ordinal scale), which varies with respect to mereological complexity (two-point vs. multi-point) and specificity of the end point (bounded vs. unbounded).
- Scales are classified on the basis of the attribute being measured: property (temperature, brightness, length, etc.), path, extent.
- Scales are associated with predicates (property scales with COS verbs, path scales with directed motion verbs, extent scales with ITVs).

Linguistic View on Scales

- Various scholars have observed that for certain scalar expressions the scale appears not to be supplied by the verb.

Linguistic View on Scales

- Various scholars have observed that for certain scalar expressions the scale appears not to be supplied by the verb.
- For example, Rappaport Hovav 2008, Kennedy 2009, L&RH 2010 claim that "the scale which occurs with ITVs (extent scale) is not directly encoded in the verb, but rather provided by the referent of the direct object".

Linguistic View on Scales

- Various scholars have observed that for certain scalar expressions the scale appears not to be supplied by the verb.
- For example, Rappaport Hovav 2008, Kennedy 2009, L&RH 2010 claim that "the scale which occurs with ITVs (extent scale) is not directly encoded in the verb, but rather provided by the referent of the direct object".
- This has lead them to the assumption that when nominal reference plays a role in measuring the change, V is not associated with a scale (denoting a non-scalar change).

Linguistic View on Scales

- Various scholars have observed that for certain scalar expressions the scale appears not to be supplied by the verb.
- For example, Rappaport Hovav 2008, Kennedy 2009, L&RH 2010 claim that "the scale which occurs with ITVs (extent scale) is not directly encoded in the verb, but rather provided by the referent of the direct object".
- This has lead them to the assumption that when nominal reference plays a role in measuring the change, V is not associated with a scale (denoting a non-scalar change).
- Locating the scale is only part of the problem. There may be multiple **sources of measurement** in a complex expression.

Challenge for Scalar Models

- Identify the source(s) of the measure of change.

Challenge for Scalar Models

- Identify the source(s) of the measure of change.
- What is the basic classification of the predicate with respect to its scalar structure?

Challenge for Scalar Models

- Identify the source(s) of the measure of change.
- What is the basic classification of the predicate with respect to its scalar structure?
- What is the exact contribution of each member of the linguistic expression to the measurement of the change?

Challenge for Scalar Models

- Identify the source(s) of the measure of change.
- What is the basic classification of the predicate with respect to its scalar structure?
- What is the exact contribution of each member of the linguistic expression to the measurement of the change?
- What is the role of nominal reference in aspectual composition?

Dynamic Event Structure 1/2

- Use multiple scalar domains and the "change as program" metaphor proposed in Dynamic Interval Temporal Logic (DITL, Pustejovsky 2011, Pustejovsky & Moszkowicz 2011).

Dynamic Event Structure 1/2

- Use multiple scalar domains and the "change as program" metaphor proposed in Dynamic Interval Temporal Logic (DITL, Pustejovsky 2011, Pustejovsky & Moszkowicz 2011).
- Define change as a transformation of state (cf. Galton, 2000, Naumann 2001) involving two possible kinds of result, depending on the change program which is executed:

Dynamic Event Structure 1/2

- Use multiple scalar domains and the “change as program” metaphor proposed in Dynamic Interval Temporal Logic (DITL, Pustejovsky 2011, Pustejovsky & Moszkowicz 2011).
- Define change as a transformation of state (cf. Galton, 2000, Naumann 2001) involving two possible kinds of result, depending on the change program which is executed:
- If the program is “change by testing”, Result refers to the current value of the attribute after an event (e.g., the house in build a house, the apple in eat an apple, etc.).

Dynamic Event Structure 1/2

- Use multiple scalar domains and the "change as program" metaphor proposed in Dynamic Interval Temporal Logic (DITL, Pustejovsky 2011, Pustejovsky & Moszkowicz 2011).
- Define change as a transformation of state (cf. Galton, 2000, Naumann 2001) involving two possible kinds of result, depending on the change program which is executed:
- If the program is "change by testing", Result refers to the current value of the attribute after an event (e.g., the house in build a house, the apple in eat an apple, etc.).
- If the program is "change by assignment", Result refers to the record or trail of the change (e.g., the path of a walking, the stuff written in writing, etc.).

Dynamic Event Structure 2/2

- Adopt Scale Shifting in the analysis.

Dynamic Event Structure 2/2

- Adopt Scale Shifting in the analysis.
- Scale Shifting is mapping from one scalar domain to another scalar domain.

  ordinal $\Rightarrow$ nominal
  nominal $\Rightarrow$ ordinal
  ordinal $\Rightarrow$ interval
  ...

Dynamic Event Structure 2/2

- Adopt Scale Shifting in the analysis.
- Scale Shifting is mapping from one scalar domain to another scalar domain.

  ordinal $\Rightarrow$ nominal

  nominal $\Rightarrow$ ordinal

  ordinal $\Rightarrow$ interval

  ...
- Scale Shifting may be triggered by:

Dynamic Event Structure 2/2

- Adopt Scale Shifting in the analysis.
- Scale Shifting is mapping from one scalar domain to another scalar domain.
  ordinal ⇒ nominal
  nominal ⇒ ordinal
  ordinal ⇒ interval
  ...
- Scale Shifting may be triggered by:
- Adjuncts: *for*/*in* adverbials, degree modifiers, resultative phrases, etc.

Dynamic Event Structure 2/2

- Adopt Scale Shifting in the analysis.
- Scale Shifting is mapping from one scalar domain to another scalar domain.
  ordinal $\Rightarrow$ nominal
  nominal $\Rightarrow$ ordinal
  ordinal $\Rightarrow$ interval
  ...
- Scale Shifting may be triggered by:
- Adjuncts: *for*/*in* adverbials, degree modifiers, resultative phrases, etc.
- Arguments (selected vs. non-selected, semantic typing, quantification).

Pustejovsky and Jezek 2012

- We analyze both *write*-verbs and *build*-verbs as *assignment predicates*.

Pustejovsky and Jezek 2012

- We analyze both *write*-verbs and *build*-verbs as *assignment predicates*.
- While both verb classes involve variable assignments over an *ordinal scale*, *build*-verbs also encode a predicate against which the ordinal scalar change is tested.

Pustejovsky and Jezek 2012

- We analyze both *write*-verbs and *build*-verbs as *assignment predicates*.
- While both verb classes involve variable assignments over an *ordinal scale*, *build*-verbs also encode a predicate against which the ordinal scalar change is tested.
- The test with *build*-verbs is provided by the (selected) direct object argument and makes reference to a *nominal scale*.

Pustejovsky and Jezek 2012

- We analyze both *write*-verbs and *build*-verbs as *assignment predicates*.
- While both verb classes involve variable assignments over an *ordinal scale*, *build*-verbs also encode a predicate against which the ordinal scalar change is tested.
- The test with *build*-verbs is provided by the (selected) direct object argument and makes reference to a *nominal scale*.
- *Write*-verbs may enter into test predications through **scale shifting** triggered by *argument introduction*.

Describing Motion Dynamically

Dynamic Interval Temporal Logic

- Path verbs designate a distinguished value in the change of location, from one state to another.
  The change in value is tested.
- Manner of motion verbs iterate a change in location from state to state.
  The value is assigned and reassigned.

Accomplishments Revisited

(33) a. Diana built a staircase.

b. Mary walked to the store.

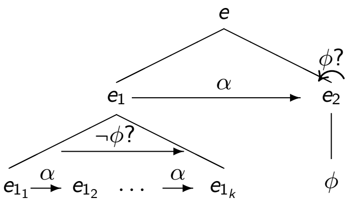
<table>
<tr><td>$build(x,z,y)$</td><td>$build(x,z,y)^{+}$</td><td>$build(x,z,y), y = v$</td></tr>
<tr><td colspan="2">$\neg staircase(v)$</td><td>$staircase(v)$</td></tr>
</table>

$_{\langle i,j \rangle}$

Table: Accomplishment: parallel tracks of changes

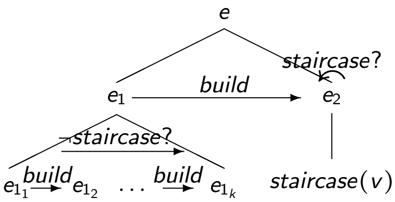
Dynamic Event Structure for Accomplishment

(34)



Parallel Scales define an Accomplishment

(35)



Revising VerbNet Frames

General assumptions

- Event E =[e1,e2,...,eN]
- Subevents
  - e2 - do/”guiding?” event
  - e3 - motion event
  - e1 - initial state
  - e4 - final state
- Temporally, e1<e2<e3<e4<….eN

Revising VerbNet Frames

Old VN	New VN

John pushed against the wall

Old VN	New VN
cause(Agent, E**)** **contact(**during(E), Agent, Theme**)** **exert_force(**during(E), Agent, Theme**)**	**do(**Agent, E**)** **contact(**E, Agent, Theme**)** **exert_force(**E, Agent, Theme**)**

John pushed the plate away

Old VN	New VN
cause(Agent, E**)** **motion(**during(E), Theme**)** **exert_force(**during(E), Agent, Theme**)** **not(Prep(**start(E), Theme, Destination**))** **Prep(**end(E), Theme, Destination**)**	**do(**Agent, E**)** **contact(**E, Agent, Theme**)** **exert_force(**E, Agent, Theme**)** **ch_event_str(**e1, Theme, Initial_Location) **ch_event_str(**e3, Theme, Trajectory) **ch_event_str(**e4, Theme, Destination) **do_motion(**Theme, e3) **eventType(**change_of_loc)* **meets(**e2, e3)

VerbNet *push-verbs*

Verb Specific features

Unified Verb Index: Search — 5/9/16, 3:15 AM

RETURN HOME | BACK | SEARCH   **Search**   VIEW OR MANAGE ALL COMMENTS | UNIVERSITY OF COLORADO

SEARCH REQUEST: **[PUSH]**

SEARCH: [ ] Go!

VERBNET MEMBERS

PUSH: CARRY-11.4-1-1
PUSH: FUNNEL-9.3-1-1
PUSH: PUSH-12-1-1
PUSH: SPLIT-23.2
PUSH: URGE-58.1.1-1

VERBNET CLASSES

PUSH-12: PUSH-12

ONTONOTES SENSE GROUPINGS

PUSH: PUSH.V
PUSH-VPC: PUSH-VPC.V
PUSH.N: PUSH.N.V

PROPBANK

PUSH: PUSH.V

FRAMENET

PUSH: MANIPULATION
PUSH: CAUSE_CHANGE_OF_POSITION_ON_A_SCALE
PUSH: CAUSE_MOTION

VerbNet *push-verbs*

Verb Specific features

VERBNET MEMBERS

PUSH: CARRY-11.4-1-1

PUSH: FUNNEL-9.3-1-1

PUSH: PUSH-12-1-1 DIR:away_from, MAN: touch

PUSH: SPLIT-23.2

PUSH: URGE-58.1.1-1

Revising VerbNet Event Structure

EXAMPLES "The lion tamer jumped the lion through the hoop."

SYNTAX **AGENT V THEME PP.TRAJECTORY**

SEMANTICS

DO(Agent, e2)
CAUSE(e2, e3)
GO(Theme, e3)
CH_EVENT_STR (e1, Theme, ?Initial_Location) - PRECONDITION
CH_EVENT_STR (e3, Theme, Trajectory)
CH_EVENT_STR (e4, Theme, ?Destination) - RESULT STATE
EVENT_TYPE(E,CHANGE_OF_LOC)

SPATIAL_INFORMATION(PrepPhrase)

Goals of VerbNet-GL Merging

- Enriching VerbNet's predicative representations.
  introduction of systematic predicative enrichment to the verb's predicate structure;
  explicit identification of the mode of opposition structure inherent in the predicate;
  GL-inspired semantic componential analysis over VerbNet classes.
- Modification of VerbNet's event representation, integrating aspects of the GL Dynamic event structure

VerbNet-GL Merging

- http://www.cs.brandeis.edu/~marc/cwc/verbnetgl/

Event Interpretations from Dependency Parses

Reddy et al (2016)

$$\begin{array}{lll} \text{acquired} & \Rightarrow & \lambda x.\,\text{acquired}(x_e) \\ \text{Disney} & \Rightarrow & \lambda y.\,\text{Disney}(y_a) \\ \text{Pixar} & \Rightarrow & \lambda z.\,\text{Pixar}(z_a) \end{array}$$

An example for a full sentence is as follows:

$$\begin{array}{l} \text{Disney acquired Pixar} \Rightarrow \\ \lambda x.\,\exists yz.\,\text{acquired}(x_e) \wedge \text{Disney}(y_a) \\ \qquad \wedge\, \text{Pixar}(z_a) \wedge \text{arg}_1(x_e, y_a) \wedge \text{arg}_2(x_e, z_a) \end{array}$$

Event Dependency Graphs

- Generate event representations directly from dependency structures;
- Assume dependencies are enhanced with semantic roles;
- Create subevent structures by reading dependency labels as partial functions over a macro-event;
- Enlist a lexicon of verbal event types, e.g., VerbNet-GL, Event Structure Frames (ESFs), Brandeis Semantic Ontology.
- Create a dependency graph that is directly interpreted as a dynamic event model.

Event Dependency Parsing

- Verbs are associated with a specific Aktionsarten classification (event structure frame);
- EDG-generation algorithm is a function from dependency relations and event frames to event dependency graphs: $DP(S) \times ESF \rightarrow EDG$

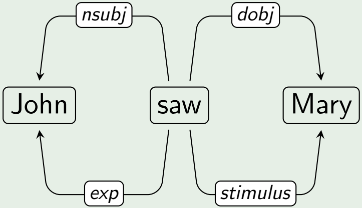
Event Dependency Parsing

Given a sentence S:

1. Perform dependency parse over S, $\mathcal{G}_D$.
2. Initialize a separate event graph, $\mathcal{G}_E$: Access a library of Event Structure Frames (ESFs) or an equivalent event structure representation from a lexical resource such as VerbNet.
3. Unify $\mathcal{G}_D$ and $\mathcal{G}_E$. Under constraints, perform role-specific transductions from semantic roles to partial event functions.

Example Derivation of Event Dependency

John saw Mary.



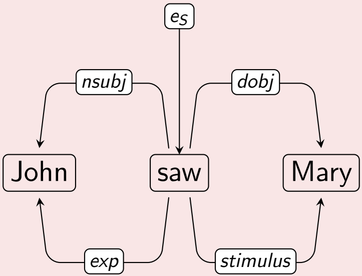
Example Derivation of Event Dependency

- Initialize the graph at the root, r, by inserting the subgraph for the ESF associated with the verbal root from the dependency parse, $DP_{root}(S)$. For example, for S = *John saw Mary*, we have $DP_{root}(S)$ = **saw**. Then, we assign the $ESF($**saw**$)$ as r, $r := e_S$, and attach it to the word form, **saw**:

see

Example Derivation of Event Dependency

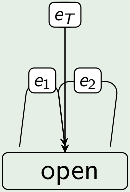
John saw Mary.



Example Derivation of Event Dependency

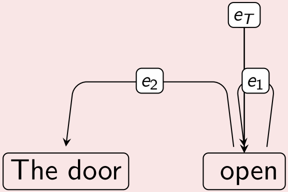
- If the ESF for the verbal root is a complex event (change-of-state, accomplishment, directed process), then initialize the graph with any subevents associated with the event type for the verb.

open



Example Derivation of Event Dependency

The door opened.

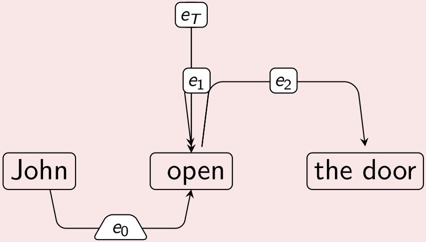


The subevent predication corresponds to the opposition structure of the verb: a verb typed as a transition, e_T, has an opposition associated with it, where e_1 denotes some $P(x)$, and e_2 denotes its opposition, $\neg P(x)$. Unaccusatives (*break* and *open*) alternate the **Patient** argument between subject and object position, and it is the **Patient** semantic role that is bound to e_2 in the EDG.

Example Derivation of Event Dependency

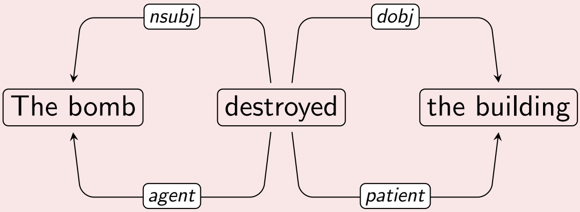
Given the presence of an **Agent** role from the dependency parse and the argument alternation of the **Patient** to direct object position, the e_2 subevent dependency remains attached, while a causing event, e_0 is introduced and attached to the subject **Agent** argument.

John opened the door.



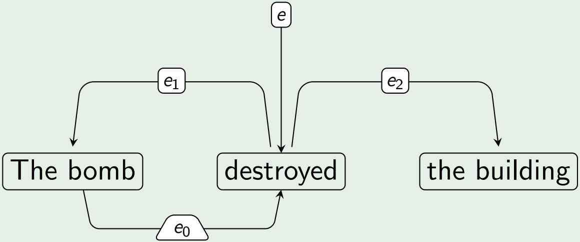
Example Derivation of Event Dependency

The bomb destroyed the building.



Example Derivation of Event Dependency

The bomb destroyed the building.

a. 

Outstanding Problems

- Type Coercion:
  John finished his coffee.
  Mary enjoyed her book.
- Light Verbs:
  Mary underwent an operation.
  John took a bath.
- Constructional Construal:
  Mary danced across the room.

Conclusion

- Computing causal graphs from linguistic expressions is both computationally expensive and typically ignores subevent properties associate with the causing and resulting events in the sentences.
- Encoding event structures directly as graphs (as with the Dynamic Event Models) provides a native data structure for algorithms that perform probabilistic causal reasoning over graphs.
- Providing a robust event graph generation algorithm from dependency parse structures facilitates the use of such graphs in reasoning.

References I

Antol, Stanislaw et al. (2015). "Vqa: Visual question answering". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425–2433.

Bunt, Harry, Robbert-Jan Beun, and Tijn Borghuis (1998). *Multimodal human-computer communication: systems, techniques, and experiments*. Vol. 1374. Springer Science & Business Media.

Chang, Angel et al. (2015). "Text to 3D Scene Generation with Rich Lexical Grounding". In: *arXiv preprint arXiv:1505.06289*.

Chao, Yu-Wei et al. (2015a). "HICO: A benchmark for recognizing human-object interactions in images". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1017–1025.

References II

Chao, Yu-Wei et al. (2015b). "Mining semantic affordances of visual object categories". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4259–4267.

Clay, Sharon Rose and Jane Wilhelms (1996). "Put: Language-based interactive manipulation of objects". In: *IEEE Computer Graphics and Applications* 16.2, pp. 31–39.

Coyne, Bob and Richard Sproat (2001). "WordsEye: an automatic text-to-scene conversion system". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, pp. 487–496.

Jacko, Julie A (2012). *Human computer interaction handbook: Fundamentals, evolving technologies, and emerging applications*. CRC press.

References III

Rautaray, Siddharth S and Anupam Agrawal (2015). "Vision based hand gesture recognition for human computer interaction: a survey". In: *Artificial Intelligence Review* 43.1, pp. 1–54.

Seversky, Lee M and Lijun Yin (2006). "Real-time automatic 3D scene generation from natural language voice and text descriptions". In: *Proceedings of the 14th ACM international conference on Multimedia*. ACM, pp. 61–64.

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

Turk, Matthew (2014). "Multimodal interaction: A review". In: *Pattern Recognition Letters* 36, pp. 189–195.