Editorial contact

lel-redazione@ilc.cnr.it

Lingue e Linguaggio is indexed in Web of Science (ESCI), Scopus Bibliographic Database, MLA Directory of Periodicals, Linguistics and Language Behavior Abstracts, ERIH Plus, Articoli italiani di periodici accademici (AIDA), JournalSeek, Catalogo italiano dei periodici (ACNP), Google Scholar, Primo Central (Ex Libris), EDS (EBSCO), and is ranked as "INT2" by ERIH Revised Lists 2011, and "A" by Anvur (Agenzia nazionale di valutazione della ricerca universitaria).

LINGUE E LINGUAGGIO

ANNO XVIII, N. 2, DICEMBRE 2019

TABLE OF CONTENTS

DYNAMIC INTERPRETATION OF PREDICATE-ARGUMENT STRUCTURE

ELISABETTA JEZEK JAMES PUSTEJOVSKY

ABSTRACT: We propose a new representation of predicate-argument structure, which encodes how the participants to the event expressed by the verb change as the event unfolds. This involves a dynamic interpretation of the predicate and their arguments, and the development of a new representation designed to encode such information, which we call *dynamic predicate-argument structure*.

As a case study, we focus on verbs of creation and destruction, such as *build* and *destroy*. We refine and extend the proposal of dynamic typing presented in Jezek & Pustejovsky 2017, and provide an integrated account of the predicate-argument structures associated with these verbs and their classes. This account includes the specification of several pieces of information: the role and type of each argument; the predicates inherent in the verb semantics (called *modes*), which are responsible for the changes in the participants; the temporal/aspectual information that the verb encodes, expressed in terms of *assignment* and *test* of values of attributes of the participants during the event. With this information, we are able to represent how, with creation and destruction predicates, the exploitation of the resource argument results in the initiation or termination of a new object, i.e. the result. Moreover, we address and solve one of the long standing problems of scalar approaches to verb classification, namely that of defining the exact contribution of each member of a linguistic expression to the measurement of the change. We do so by allowing compositional shifts in the scale of interpretation, a phenomenon we refer to as *scale shifting*.

KEYWORDS: argument structure, dynamic semantics, semantic roles, creation and destruction predicates, compositionality.

1. INTRODUCTION[1]

The representation of predicate-argument structures is a key concern of linguistic studies since the very early days of modern linguistics. In this paper we propose a representation that is new in many ways with respect to existing

[1] We would like to thank two reviewers for their useful comments. All remaining errors and infelicities are our own.

ones, and overcomes many of the difficulties observed in previous studies, in particular: accounting for the complex interplay between semantics and syntax in the realization of arguments of verbs of change, and capturing at the same time the temporal and aspectual development of the event. Within this framework, we introduce a new representation for verbs of change, which aims at encoding how the participants change as the event unfolds. We call this representation *dynamic predicate-argument structure*. The kinds of change we focus on encompass those expressed by creation and destruction predicates, i.e., predicates that denote the "coming into being" (e.g. *create*, *build*) and the "going out of being" of an entity (e.g. *disappear*, *destroy*). The first systematic classification of change, what Aristotle calls "generation and corruption" in *de Generatione* (McKeon1968), can be summarized as follows:

(1) Coming into being:
   a. The creation of an object through exploitation of resource:
      - natural growth;
      - aggregation;
      - manipulation of resource.
   b. The creation of something through representation of resource.

(2) Going out of being:
   a. The destruction of an object into existing material:
      - disaggregation;
      - termination.

Both creation and destruction verbs have long posed a problem for linguistic classification (see among others Dowty 1991; Tenny 1994) and the interplay between their event structure, and the syntactic expressibility of their arguments is still not well understood (Bisetto & Melloni 2007; Jezek 2014). Creation verbs, for example, exhibit considerable variation in the syntactic realization of the created entity. This is illustrated in the examples below, where we see that, depending on the predicate, the argument realizing the created object may be: optional, as in (3), mandatory, as in (4); subcategorized as object only, as in (5); as PP only, as in (6); as object or PP, as in (7); or remain hidden as an entailed inference of the sentence (i.e., the resulting photograph in (8)) (so called *representation source theme* verbs, cf. Dowty 1979).

(3) a. *John wrote a new book.*
   b. *Sophie has been writing for hours.*

(4) a. *John built a wooden bookcase.*
   b. **John has been building for weeks.*[2]

[2] Note that it is not impossible to make up a context in which *build* is constructed without direct object as in (4b), but differently from *write* in (3b), the referent of the argument must in this

(5) a. *John built a wall (out of bricks).*
b. **John built the bricks into a wall.*

(6) a. *Mary stacked the blocks into a tower.*
b. **Mary stacked the tower.*

(7) a. *John assembled the sofa.*
b. *John assembled the individual parts into a complete sofa.*

(8) *They have already photographed the scene*.

Works on scalar change (Beavers 2008; Levin & Rappaport 2010, among others) and dynamic event semantics (Naumann 2001) suggest a new understanding of the interplay between verb meaning, event semantics, and argument structure with these predicates, by focusing on the measurement of the change in value of the properties of the participants in each intermediate state during the event. However, they make conflicting statements regarding the attribute available to construct the gradience scale for creation verbs, and for incremental theme verbs more generally. The attribute usually employed as a measure of change for creation verbs is *extent*, but it is recognized that the change in volume of the created entity (as captured by the *extent* scale) is not sufficient to fully characterize an act of creation. Moreover, no general consensus has been reached regarding the role played by arguments in the aspectual profile of the predicate, i.e. whether the scale is lexicalized in the predicate (Hay *et al.* 1999) or introduced in composition by the object (Rappaport Hovav 2008; Kennedy 2009; Levin & Rappaport 2010). Finally, their accounts do not provide an explanation of the contrastive data on argument variation introduced in (3) to (8) above.

In the following discussion, we propose a new analysis of these predicates, based on a dynamic interpretation of their event and of the changes the participants undergo while the event unfolds over time, as well as their syntactic realization. We identify the parameters of behavior dictating the expression of change for creation and destruction predicates in language, which are adequate for this purpose. We claim that these parameters are the semantic and syntactic linguistic dimensions as defined by the answers to the following questions:

(9) Is the created/destroyed object realized in the syntax?
What is the semantic type of the resulting object in a creation?
What counts as a result of a destruction act?

---

case be retrieved from the immediate situational or discourse context (cf. *John has been building for weeks, but that shed still isn't finished*); it is in other words a pragmatic deletion; moreover, such contexts would be exceptions and not the norm, as disclosed by the corpus analysis we performed on a sample 250 concordances for the verb (details on the corpus analysis can be found in section 4.1.3).

Does the creation act make use of existing resources, such as material or substance? How are these exploited in the event?
Are the resources that are used in the event realized in the syntax?
What is the temporal unfolding of the event and the mode of change inherent in the predicate?

The answers to these questions will allow us to create a coherent classification of creation and destruction predicates in language. The major contributions of this study is to characterize each class in terms of its *dynamic predicate-argument structure*, a representation, the features of which correspond to how the questions in (9) above are answered.

The structure of the paper is as follows. The theoretical background is introduced in section 2, while the analysis of members for each of the proposed classes is provided in section 3 for creation classes and 4 for destruction ones. Section 5 reports our concluding observations and briefly outlines how our proposed analysis can be successfully extended to other verb classes.

2. THEORETICAL BACKGROUND

In our study we rely on Generative Lexicon as our theoretical background (Pustejovsky 1995), and propose an extension thereof. Generative Lexicon theorizes that words in our mental lexicon contain four intertwined types of information, represented as *structures*. *Qualia Structure* encodes the most idiosyncratic aspects of the meaning a word; these aspects are conceived as *relations* that the word holds with the concepts expressed by other words, for example the relation between *bread* and *bake*. Specifically, the *Formal quale* encodes the relation between the object denoted by the word and the larger set of objects is it a type of (*bread* is a type of *food*), the *Constitutive quale* encodes what the entity is made of (*flour*, etc.), the *Telic quale* encodes the purpose of the entity (*be eaten*), and the *Agentive quale* encodes what causes its existence (*be baked*). While all words in the lexicon encode values for the *Qualia relations,* such as *food*, *flour*, *be eaten* and *be baked* for *bread*, words denoting events encode additional information types. Specifically, verbs and event-denoting nouns such as *party* or *meeting* encode information about the internal temporal structure of the event they denote (*event structure*), and information about the participants of such event (*argument structure*). The event types proposed in Generative Lexicon, summarized in (10), capture the conventional Vendlerian *Aktionsart*-based typology for events: the primitive events are *States*, *Processes* (Vendlerian Activities) and *Transitions* (subsuming both punctual (Achievements) and durative (Accomplishments) telic changes).

(10) a. EVENT → STATE | PROCESS | TRANSITION

Event structure may include distinguishable subevents, i.e. temporal parts representing "portions" of the main event; for example, the event structure of a simple Transition (Achievement) includes an initial State and an end State (11c), whereas the event structure of a durative Transition (Accomplishment) includes an initial Process followed by an end State corresponding to the state of the object at the end of the event (11d).

(11) a. STATE: → e
b. PROCESS: → $e_1 \ldots e_n$
c. TRANSITION_{ach}: → STATE STATE
d. TRANSITION_{acc}: → PROCESS STATE

As regards *Argument Structure*, the standard inventory of argument types in GL includes three argument types: *true* arguments, *default* arguments, and *shadow* arguments. *True* arguments are obligatorily expressed in the syntax and cannot be omitted (as with the object of *lock*), *default* arguments may remain unexpressed under certain conditions (as with the object of *eat* and *drink*), and *shadow* arguments are incorporated in the predicate and cannot be expressed (as with *phone* in **she called with the phone*) - unless they are further specified. In the following, we will consider an additional argument type, *hidden* arguments (cf. Jezek & Melloni 2011, *inter alia*), which is very relevant for the verb classes we are investigating and for the type of representation we propose.

Turning now to semantic composition, Generative lexicon assumes that the information encoded in the information structures outlined above interact in composition at a sublexical level. According to this view, the meaning of complex linguistic expressions is not strictly compositional in the traditional fregean sense but it is built compositionally through processes of mutual adjustments between the meanings of the co-occurring words. In this paper, we will touch at two of these processes: *coercion* and *co-composition*. Informally, *coercion* occurs for example when a verb composing with an argument induces a change in the semantics of the argument to adjust it to its selectional requirements, as with the expression *finish the wine*, which - under a default interpretation - means *finish the drinking of the wine*, due to the process of coercion initiated by the verb, in which the value of the *Telic quale* of *wine* ('drink') is exploited to satisfy the requirement of *finish* to have an *Activity* as Object.[3] C*o-composition*, on the other hand, is the process through which all

[3] Linguistic and situational contexts obviously play a crucial role in the interpretation of coercions. For example, in the expression *the winemakers added tannin by finishing the wine in oak*, the context triggers a different interpretation for *wine* (preparing, making) compared to the example above.

elements – not only verbs – are active and behave as functors in semantic composition. For example, in the composition between the noun *meeting* and the verb *open*, the argument acts functionally and co-determines the meanings of the verb, which in the context of *meeting* acquires the metaphorical reading of 'start'.

In the standard theory, the levels of information presented above (*Qualia*, *Event*, and *Argument Structure*) are usually represented as components of a typed-feature structure; for the purposes of our study we will represent them using a formalism based on trees, as will be shown in the next section.

3. DYNAMIC PREDICATE-ARGUMENT STRUCTURE

After introducing the theoretical framework in which our analysis of creation and destruction predicates is couched, let us now focus on examining how the change taking place during the event expressed by those verbs can be encoded in a dynamic way. This entails reinterpreting the traditional event types presented in (10) above in terms of dynamic event structures. We will do so by resorting to dynamic temporal logic and by interpreting change as the *assignment* of distinct values to relevant attributes of the participants[4] in the event over time.[5]

In a dynamic approach to modeling changes (cf. Harel *et al.* 2000), a distinction is made between formulas, ϕ, and programs, π. A formula is interpreted as a proposition, evaluated relative to a specific state in the model. Programs are either atomic or complex, where atomic programs are input-output relations interpreted over state-to-state pairings.

Within this framework, a *state* is defined as either a single frame (event) containing a proposition, or an extended sequence of temporally ordered frames containing the same propositional content; this can be evaluated as holding true over multiple adjacent states, which can be handled with an operation of concatenation, +.

The simplest change of state, ϕ to $\neg\phi$ is called a *simple transition* and involves an atomic program, α, that changes the content in the first state to its negation in the next state. This corresponds to the basic event structure for *Achievements*, as in Figure 1, where ϕ stands for the proposition, and ϕ? and

[4] The relevant attributes and their values vary depending on the nature of the event; for example, for directed motion verbs the relevant attribute is "path" and the values are different positions along this path.

[5] For the present discussion we rely on the proposal for a dynamically interpreted event structure presented in Pustejovsky & Moszkowicz 2011, which captures the scalar change associated with motion events. This proposal has been extended to the class of incremental change predicates in Pustejovsky & Jezek 2011, and located within a global model of dynamic event structure representation in Pustejovsky 2013.

the backward arrow indicate the testing program being performed on the proposition, as described below. Note that in all of the event trees below, temporal sequencing of subevents is encoded directly in the tree configuration.

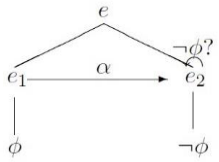

FIGURE 1: SIMPLE TRANSITION.

Following Pustejovsky 2013, we call verbs that encode this event type *test predicates*, because the program they encode is a *test* that checks whether the change occurring during the execution of the event ϕ? is identical to the distinguished endpoint encoded in the event semantics ϕ, and stops when this point is reached. This checking operation is signalled by the backward arrow in Figure 1. In our model, a test program references a *nominal scale* (cf. Krantz *et al.* 1971) that is, a scale that introduces a dichotomy between what did not exist before the event and what exists at the end of it: nominal scales indicate transformations by any one-to-one function. In this view, a test is a program that constantly checks whether the changes against the goal defined by the nominal scale encoded in the verb is attained.[6]

According to Pustejovsky (2013) we define a *Process*, in dynamic terms, as a sequence of transitions, where at each transition, the value of an attribute α associated with an argument is assigned and reassigned (¬ϕ?), as there is no terminal test in the proposition ϕ expressed by the verb (cf. Figure 2). The incremental change in the attribute associated with the argument is encoded as an *assignment* program that makes reference to an *ordinal* scale, that is, a scale that does not introduce a dichotomy as the nominal scale but rather indicates increasing ordered changes over a defined domain (Kranz *et al.* 1971). Verbs encoding processes are *assignment* predicates.

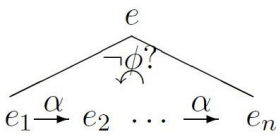

FIGURE 2: PROCESS.

[6] In our use of the term *scale*, we depart from the conventional linguistic terminology, according to which 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) (cf. Rappaport & Hovav 2008), and adopt the distinction between *nominal* and *ordinal* scales as discussed in Krantz *et al.* (1971).

Finally, an *Accomplishment* is an event that makes reference to a preparatory phase consisting of an iteration of changes (ordinal scalar predication), followed by a nominal scalar predication in a resulting state. The iterated assignment stops when the predefined goal or intended state is achieved. This is achieved by testing (ϕ? in Figure 3) whether a proposition ϕ is true: the process continues while the test is not satisfied – $\neg\phi$? – until it finally is. This is represented in Figure 3. Accomplishments verbs encode both an *assignment* (e_1) and a *test* (e_2) component.

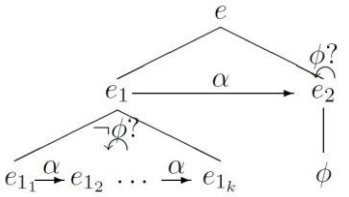

FIGURE 3: ACCOMPLISHMENT.

After introducing the dynamic event theory we rely on, we focus now on the participants of the event. In Jezek & Pustejovsky (2017) we argue that the semantics of change, as expressed in verbal predicates of creation and destruction, must include specification of how the event participants behave throughout the event: that is, a participant can stay the same, be modified, come into existence, or be terminated. Moreover, participants play roles, and such roles should be specified as well.

We distinguish four primary *modes of change* that a participant may undergo in creation and destruction events.[7] They are:

(12) *Modes of change*
  a. Modification (mod)
  b. Initiation (init)
  c. Termination (term)
  d. Transfer (trans)

An argument undergoes *modification* (mod) if there is an identifiable attribute whose value is changed during the event. An argument undergoes *initiation* (init) if it is brought into existence as predicated by the verb, and undergoes *termination* (term) if the converse is true. An argument undergoes *transfer* (trans) when it represents the transfer of information from a *source* through a *medium* (both these semantic roles, *source* and *medium*, are defined below).

[7] An additional mode, namely *Persistence*, identifies the property of an argument that is unaffected by the change predicated by the verb throughout the event, such as the Agent in change of state predications. For the current purposes of capturing change in creation and destruction predicates, we will leave this mode aside in the present discussion.

In the representations that follow, we also consider the *existence* mode. This is not a mode of change, therefore we do not include it in (12): it is a mode that qualifies the resource that exists prior to and independently of the event, corresponding to the available material out of which an object is created (with creation predicates) or to the object which is terminated (with destruction predicates).[8]

Concerning the notion of *semantic role*, as in most theories where semantic roles are adopted, role assignments are usually unique for each argument to the verb. But as we demonstrate in the discussion below, when events are interpreted dynamically, labels to each argument will reflect the role it plays in that phase of the event, which may change as it unfolds, for example an argument that plays the role of *resource* at the beginning of the event may play the role of *result* as a consequences of the changes occurring during the event. A participant in an event of creation or destruction can be characterized as fulfilling one of the following roles:[9]

(13) *Semantic Roles of Arguments*

a. *Resource*: the material or object used to bring about the change of state or result.
b. *Result*: the outcome of the change of state brought about by the event.
c. *Source*: Dowty's source of representation.
d. *Medium*: the physical used by an animate object acting intentionally in an event that brings about a new object through the representation of an existing one, for example characters for writing.[10]

There is a clear relation between the roles, particularly between the *resource* and the *result*; in section 4 we will illustrate how this is captured formally.

In order to keep track of the change occurring in the participants throughout the process, we adopt and extend the notion of *dynamic program variable*, used in Pustejovsky & Moszkowicz 2011 for motion predicates and in

[8] Notice that there exists an implication between the modes in (12), such that *termination* presupposes *initiation*, which presupposes *modification*, which presupposes *esistence*. As we will see in section 4, this implication is encoded in the dynamic representation in terms of temporal ordering of the subevents; the subevent associated with the *existence* mode happens before the subevent associated with the *modification* mode, and so forth.

[9] The roles we resort to in our analysis are widely used in resources such as *FrameNet* and *VerbNet*. In the following we provide a definition for each role tailored to address the specific properties of arguments of creation and destruction predicates.

[10] According to this definition, a *medium* could be seen as a subtype or *resource*. As we will show in section 4.2, and particularly in the discussion of the examples in (28) and the representations in Figure 12 and 13, it is more appropriate to see it as a component of the *resource*, that together with the *source*, forms the *resource* of verbs denoting the creation of a representation, such as *write* or *photograph*.

Pustejovsky & Jezek 2011 for general scalar predicates. A dynamic program variable is a variable that keeps track of the current state of what has been "acted on". It may designate either the values of the attribute of the resource being modified – with change of state verbs –, or the intermediate stages of the "result-to-be" – with creation or destruction predicates. It is annotated with a vector, $\vec{x}$. For example, in the unfolding process of building a house, the dynamic program variable representing the intermediate stages of the "coming-into-existence house" is encoded as $res\vec{u}lt$.

As an additional feature in the representation, we introduce the *argument type*, which encodes the conditions under which an argument is expressed in the syntax. As anticipated in section 2, we will consider four possible values for this feature.

(14) *Argument Types*
   a. *True Argument* (T-ARG)
   b. *Defaulted Argument* (D-ARG)
   c. *Shadow Argument* (S-ARG)
   d. *Hidden Argument* (H-ARG)

Recall from section 2, that True arguments (T-ARG) are complements of the verb that in the absence of specific pragmatic constructions are mandatory from a syntactic point of view, such as the direct object of *lock*. Defaulted arguments (D-ARG) are complements that are licensed by the predicate but are omitted in the syntax, while remaining present at the level of the semantic representation: an example is the direct object of *eat*.[11] Shadow arguments (S-ARG) are complements that are incorporated in the predicate, such as the expressions for sensory organs (eyes, ears) with perception verbs *see* and *hear*). They cannot be expressed (see the oddness of "I saw it with the eyes") unless the expressions they are part of carries additional information about the incorporated type ("I saw it with my own eyes"). Hidden arguments (H-ARG) encode participants that can never be expressed. An example of hidden argument is the dynamic program variable introduced above, which is present in the dynamic inspection of the event but is never expressed syntactically.

Now let us consider how to enrich the dynamic event structure representations illustrated above with the modes of change and the features of the arguments, in order to attain a fully-fledged dynamic predicate-argument structure. We propose to annotate the modes of change as predicates that make up the meaning of the selecting verb (i.e. its *Qualia*, cf. Section 2), and to notate each arguments with its semantic role and argument type. As an example, for

[11] The operation of *defaulting* is subject to specific conditions, which we cannot discuss here. See however Jezek 2018 for a thorough discussion.

a verb encoding an incremental Process involving a single argument that constitutes the resource of the incremental change, such as *grow* in "A tree is growing", we notate is as follows:[12]

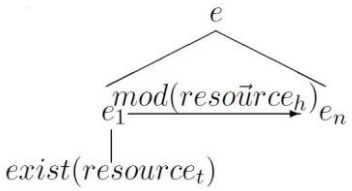

FIGURE 5: DYNAMIC PREDICATE-ARGUMENT STRUCTURE.

The representation in Figure 5 illustrates an incremental change, during which the value of one or more attributes of an existing resource classified as true argument ($resource_t$) undergoes modification. The program variable (the hidden argument $\vec{resource}_h$) keeps track of the value of the attribute(s) of the resource being modified during the process.

In the following section, we apply the notation proposed in Figure 5 to the analysis of selected subclasses of creation and destruction predicates.

4. CREATION PREDICATES

This section introduces our classification of creation predicates. For each class, we give a definition with examples, then we supply the formal analysis, and the compositional operations, based on the model presented in section 2 and 3. Our overall classification of creation predicates is outlined in (15). Classes are identified based on the interplay of the syntactic and semantic behavior of verbs and their arguments, as will become clear in the following sections.

(15) Classes of Creation Predicates
- Creation through Exploitation of Resource
  - Natural Growth
  - Aggregation
  - Resource Manipulation
- Creation through Representation
  - Result Predicates
  - Source Predicates
  - Alternating Predicates

[12] For the purposes of our current discussion, in the representations that follow, we simplify the dynamic notation presented at the beginning of the section (Figure 1, 2 and 3) – while still continuing to assume it as present – and focus on the features of the arguments.

As shown in (15) we identified two main classes of creation predicates, *creation through exploitation* of the resource argument, and *creation through representation* of a source object. Moreover, for each class, we identify three distinct subclasses, which are discussed in separate sections below.

4.1 Creation through Exploitation of Resource

The verbs associated with this class denote any transformation of an existing object or objects, through: (a) natural growth (generation), (b) aggregation or (c) resource manipulation.

4.1.1 Natural Growth

There are two aspects of natural growth. One aspect is the natural "manifestation" of an entity out of another by virtue of the character of its inherent properties. Examples of such predicates can be found in Levin (1993) "grow verbs" class, which includes the verbs *develop*, *evolve*, *grow*, *hatch*, and *mature*. The other aspect relates to the same process, but focuses on the incremental change that an object undergoes. Both of these aspects are manifested in the two senses of the verb *grow* in (16) below:

(16) a. The sapling grew into a beautiful tree.
b. A tree is growing in the back yard.

We focus here on (16a), where the resulting (created) object that appears in the PP complement can be thought of as part of the "natural trajectory" of the resource object: that is, the characterization of this individual as a *sapling* is more accurately replaced by the sortal characterization brought about by *tree*. The proposed dynamic representation of the change described in (16a) is given below:

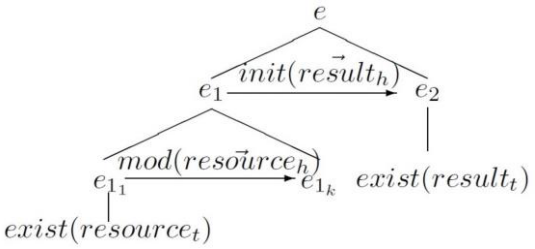

FIGURE 5: THE SAPLING GREW INTO A BEAUTIFUL TREE.

In Figure 5, the *sapling* (as subject) is the syntactic manifestation of the *resource*, which is mandatory, hence a true argument, notated as $_t$ in the figure. The dynamic variable $\vec{resource}$ is a hidden argument (notation $_h$); it is not

expressed in the syntax but it is the argument that keeps track of the accumulated changes in value of the resource, as viewed from the predication of the result, i.e., the initiation of the tree ($res\vec{u}lt_{h}$). While the *resource* continues on in the guise of the hidden argument $reso\vec{u}rce_{h}$, reference to the nominal classification of it is no longer possible as soon as the new type, i.e. $res\vec{u}lt$, is initiated. Note that only when the initiation is completed, the *result* (*the tree* in the PP) can be said to exist and can be expressed as a true argument in the syntax (*exist*(*result*$_{t}$)).

4.1.2 Aggregation

This class refers to the coming together of existing resources to create a distinct object, identified by a newly introduced expression. Included in this class are the verbs *assemble*, *collect*, *gather*, and *congregate*. The *resource* encodes the referential objects available prior to the change, whereas the *result* encodes the object formed by the event. Both participants can be expressed, as is illustrated with the examples in (17a) (*result*) and (17b) (*resource*).

(17) a. *A crowd assembled in the lobby*.
b. *The students assembled in the lobby*.

In (17a), the subject directly denotes the object formed by the event, i.e. the *result*. That is, the crowd is brought into existence by the assembling event. In (17b), on the other hand, the subject denotes the *resource*. For this verb class, the key feature is, of course, the relationship between a set of individuals serving as the resource type, and a group denoting the result type. Even though the selectional restrictions on (17b) reflect the former constraint, it is possible to interpret the NP in subject position (*the students*) as being *coerced* into a group interpretation by the verb,[13] i.e., we assume that *assemble* in its intransitive use selects a group as subject and that there exists a distinction between the typing restriction of individual[+plural] from the predicate *assemble* onto its resource, and the output condition (resulting type) from the computation, viz., a group. Taking *event headedness* into account, the representation of (17a) is presented below, where the headed subevent is e_2 i.e., the one associated with the expressed result argument *result*$_{t}$. *Event headedness* (Pustejovsky 1995) is the property of a subevent to act as the most prominent part of the matrix event from a structural point of view, contributing to the focus of the interpretation and determining the realization of the arguments associated to it. It is marked with e*. We propose here that headedness is not only assigned lexically but can also be assigned compositionally and pragmatically.

[13] For a definition of the compositional mechanism of coercion, see section 2.

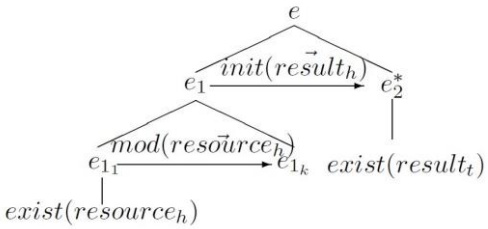

FIGURE 6: A CROWD ASSEMBLED IN THE LOBBY.

Now consider the sentence in (17b), where the *result* is defaulted (it may be expressed through a PP), while the *resource* is expressed syntactically. The representation of Figure 7 is the same as Figure 6, with the difference that the head of the event is in this case e_1, i.e., the subevent associated with the resource argument, and that the *resource* is expressed ($resource_t$) instead of the *result*.

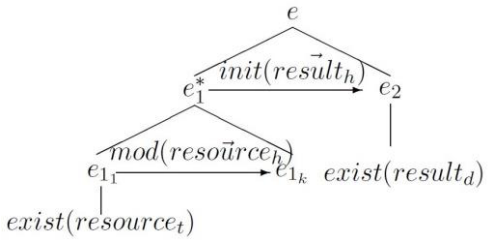

FIGURE 7: THE STUDENTS ASSEMBLED IN THE LOBBY.

An alternative account to the coercion analysis we propose is to assume that the event structure of *assemble* is underspecified lexically, and allows for two distinct syntactic realizations, depending on which subevent is under focus and acts as head. This analysis could be preferable to account for examples of use such as "Bits and pieces are waiting to be assembled into a whole", in which both the resource (bits and pieces) and the result (a whole) are expressed, and the whole event would then be analyzed as headed. However, corpus analysis performed on the *enTenTen15* corpus, an English web corpus of about 16 billion words, queried through the Sketch Engine tool using the *Word Sketch* function (Kilgarriff *et al.* 2004, 2014), has shown that *assemble* has a high number of collective nouns such as *congress*, *parliament*, *crew*, *congregation*, *army*, *fleet*, *team* among its most frequent subjects, leading us towards the coercion account.[14]

[14] The Sketch Engine (https://www.sketchengine.eu) is an online tool to query corpora in very sophisticated ways. The *Word Sketch* function take as input a word and returns its combinatorial profile organized per syntactic relation of the collocates (in the case of a verb, subject of, object of, etc.), ranked by frequency and mutual information score based on the *logDice* measure. For example in the case of *assemble*, it returns the list of subjects (40.738 hits in the *enTenTen15* corpus) among which we selected the ones listed above among the top frequent types.

4.1.3 Resource Manipulation

We turn now to the major group within the class, namely resource manipulation. This includes what are conventionally viewed as creation predicates in the literature (Dowty 1979; Jackendoff 1990; Levin 1993; von Stechow 2001; Piñón 2008), with verbs such as *build*, *knit*, *bake*, *create*, *produce*, *compose*, and so on (cf. the "create" verbs in Levin 1993). Obviously, this class is not a homogeneous one and there are many semantic and syntactic distinctions to be made within it (for an overview, see Jezek 2014). In this section, we examine how the model developed here can characterize the common elements of the verbs within this class, as well as how they differ amongst themselves. In particular, the major distinctions within this class that we wish to characterize formally are the difference in behavior in object selection/dropping, and the alternation between a creation sense and a change of state sense of *bake*, as in *bake the cake* (creation) vs. *bake the potato* (change of state).

Traditional analyses of these verbs view them as differing from regular accomplishments in that they take a participant that measures out the event, i.e., an incremental theme argument (Dowty 1991). More recent work on scalar change offers a new way to look at the inherent temporal properties of these verbs, but none of these analyses, however, can account for the long recognized distinction (Dowty 1979; Resnik 1996) between transitive creation verbs that frequently exhibit a so-called object-drop alternation (e.g., *draw*, *write*, *knit*, *paint*) and transitive creation verbs that typically do not (e.g., *build*, *construct*, *create*, *make*, *produce*). This distinction is illustrated below, where *build* and *construct* are contrasted with *knit* (see also section 1).

(18) a. *John built a wooden house in 2003.*
b. *John built in 2003.*[15]

(19) a. *The city constructed a new fence around the reservoir.*
b. **The city constructed in the spring.*

(20) a. *Mary knitted a sweater.*
b. *Mary knitted yesterday evening.*

Consider first *knit* in the context below:

(21) *Mary knitted yesterday evening.*

[15] As noted in section 1, it is not impossible to make up a context in which *build* and *construct* are constructed without direct object, but differently from *knit* below, the referent of the argument must in this case be retrieved from the immediate situational or discourse context.

In our model, predicates like *knit* denote a process lexically, i.e. they are assignment predicates which leave a trail of the process, namely the knitted material. This argument is unexpressed in the syntax but present in the inspection of any state of the process. As referenced above, this argument is encoded in the representation in Figure 8 as a dynamic program variable, thus hidden in the syntax, i.e. $\vec{resource}_{h}$.[16] The *resource* exists before the event, and is coded as default argument (it might be expressed as in "It's the first time I've knitted with cotton"), thus notated as $_{d}$; the modification mode (mod) from the verb's semantics applies to its program variable, i.e. the hidden argument that captures the incremental changes the resource goes through as an argument of the *knit* function.[17]

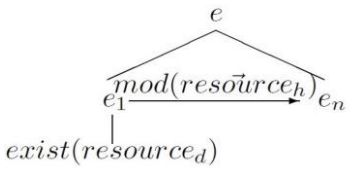

FIGURE 8: MARY KNITTED YESTERDAY EVENING.

Now consider *build*. The chief characteristic of predicates such as *build* that distinguishes them from assignment predicates such as *knit* is the explicit encoding of a *test* – in addition to the *assignment* component – in the verb's semantics – namely as their selected direct object argument – which is typed as a quantized physical object and has the status of true argument. We claim that it is this reference to a *test* that makes object-drop typically not possible for such predicates (expect for pragmatic deletions).

Thus, a verb such as *build* leaves a *trail*, generated by the assignment component associated with the process subevent, while testing the value of this trail against the semantics of the distinguished value denoted by the direct object. Therefore, it has both an incrementally created (trail) argument, denoted by a hidden program variable, as well as the true argument identified with the *test*, i.e. the resulting created object.

Let us now see how the dynamic semantics of *build* and its arguments can be represented. We assume that *build* is lexically an Accomplishment (head on e_2); during the process component (e_1), the *resource* is modified, until the new object is initiated.

[16] Recall from section 3 that in this contribution, we factor out the persistent Agent and focus on the Resource and the Result arguments, i.e. the participants undergoing change.

[17] Because of this, we do not annotate the subsequent events following e_1 with content indicating *exist* ($\vec{resource}$), as it is entailed by the program variable undergoing change through modification (mod).

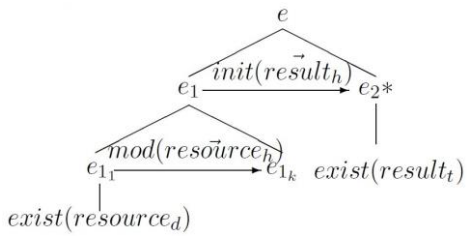

FIGURE 9: JOHN BUILT A HOUSE.

The representation in Figure 9 introduces different modes for distinct arguments: first, the *resource* is annotated as existing independently of the event, and encoded as a defaulted argument ($\text{resource}_{\text{d}}$), because *build* allows not to express the resource argument, as in Figure 9; the defaulted resource is modified via the program variable $\vec{resource}_{\text{h}}$ that initiates the program variable $\vec{result}_{\text{h}}$; the latter becomes the resulting creation, expressed as direct object of the verb, $\textit{result}_{\text{t}}$.

Note that when the resource is expressed with build, this can be accomplished in one of two ways, illustrated in (22) below.

(22) a. *John built a wooden bookcase.*
b. *John built a bookcase out of wood.*

Now that we have introduced a formal distinction between *assignment* predicates such as *knit* and *test* predicates such as *build*, it is necessary to explain why verbs in the former class can typically take NP direct objects and, in case these are quantized, acquire an accomplishment reading, as shown in (23).

(23) *Mary knitted a sweater last night.*

To answer this question, we argue that there are two semantic interpretations to constructional transitivity in creation predicates:

(24) a. Selection of a *test* (as a quantized NP) as an argument to a transition predicate;
b. Expression of a *test* as the quantification resulting from an *assignment* predicate.

The first interpretation is employed by direct argument selection in transitive creation predicates, e.g., *build*. The second is what we see at play in (23) above, and for this interpretation, there are at least two distinct grammatical strategies that introduce a *test* over the activity denoted by the matrix predicate, listed below.

(25) a. ARGUMENT INTRODUCTION; this creates a *test* by making reference to a nominal scale denoted by the NP in object position; *knit a sweater*.

b. ADJUNCT INTRODUCTION; this creates a *test* with a measure phrase relevant to the trail left by the assignment predicate, namely the object being knit; *knit until noon*.

Consider again the accomplishment interpretation available with the verb *knit* in (23) in the light of what we just said. In the example in (23), we see the composition of the process reading of *knit* with its argument, resulting in an interpretation as an accomplishment, where the result argument, *the sweater*, is brought into existence, i.e., initiated.

The dynamic predicate argument structure of the compositional expression *knit a sweater* is the same as that for *build*, as shown in Figure 9. In other words, with predicates such as *knit*, argument introduction creates a *test*, which determines when the incremental directed Process should terminate. This shifts the scale of interpretation for these predicates from ordinal to nominal, something referred to as *scale shifting* in Pustejovsky & Jezek (2011).

To conclude this section, we turn to the issue of how our representation allows to model verbs which notoriously exhibit an alternation between a change of state sense and a creation sense, namely, the polysemy exhibited by *bake* in (26a-b): while in (26a) the object already exists before baking and changes its state during baking, this is not the case with the object of (26b), that does not exist prior to the event and is created during its unfolding (cf. Atkins *et al.*, 1988):

(26) a. *John baked the potato.* (change of state sense)
b. *John baked a cake.* (creation sense)

First, we must adequately model how the object in the change-of-state sense of *bake* in (26a) is being modified. We will treat the resource in the event as a true argument ($resource_t$) undergoing change, through a modification function. As with *build*, we model the modification function as being associated with the hidden program variable ($\vec{resource}_h$) introduced by the Process. We can state the dynamic representation for the process reading of *bake* with object as follows, where the resource is expressed as direct object:

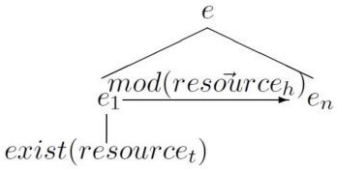

FIGURE 10: JOHN BAKED THE POTATO.

Turning to (26b), under the create interpretation for *bake* in this context, the *resource* (the ingredients being used) is modified but not expressed in this context, thus typed as defaulted ($resource_d$), while the output type is initiated

$\vec{result}_{\text{h}}$, from the modified resource ($\vec{resource}_{\text{h}}$). The resulting created object is types as a true argument ($result_{\text{t}}$). The accomplishment reading of (26b) is shown in Figure 11 below. The creation sense is attained compositionally through the introduction of an argument denoting a food artifact, *cake*. Following Pustejovsky (1995), we use the term *co-composition* (cf. section 2) to characterize compositional operations of this kind, where arguments behave as functors in composition: in (26b) the argument *cake* acts functionally and co-determines the creation meaning of the verb in the context of use.[18]

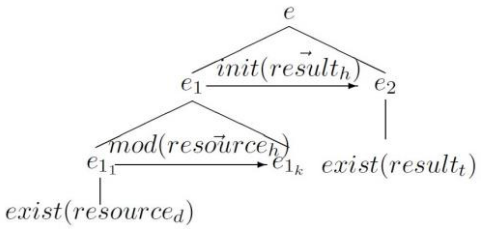

FIGURE 11: JOHN BAKED A CAKE.

4.2 Creation through Representation

In this section, we discuss the class of predicates that bring about representational artifacts. This class of predicates involves the act of creating a representation of a source object, either real or imagined.[19] In the model proposed here, the creation of a representation of an object acting as *source* involves the transfer of information from the *source* to the resulting created object (*result*) through a *medium*.

For our present discussion, we will distinguish three classes of representational creation predicates, mostly based on their behavior at the syntax-semantics interface, i.e. whether their direct object expresses the representational created artifact (27a), the source of the representation (27b) or can alternate between the two (27c):

(27) a. *Result predicates*: write
b. *Source predicates*: photograph
c. *Alternating verbs*: paint

[18] Notice that *bake* also allows uses where both the *resource* and the *result* are expressed, as in "Kim baked the expensive ingredients into a fine cake". According to our approach, in this case the *resource* would be typed as true arguments, instead of defaulted as in Figure 11. The representation with both *resource* and *result* expressed would be identical to the representation in Figure 5, corresponding to the change described by "The sapling grew into a beautiful tree".

[19] See Dowty (1979, 1991) for early work on the interaction of "Representation" and "Source" in such verbs.

We first consider the class (27a) involving incremental themes of representation (cf. Dowty 1991), namely verbs such as *write*. This class is very similar to the object incremental theme verbs studied above, such as *knit*, except for two parameters: (a) the *source* is not necessarily a physical object, as in the case of *knit*, and (b) the created object is a representational artifact, that is, the representation of a source object. Such verbs typically exhibit also a variant with no expressed created object, as repeated below.

(28) a. *Mary wrote all morning.*
b. *Mary wrote a letter last night.*

When we talk about *write*, we assume that we are utilizing a script (syllabary, characters, alphabet, semaphores) that expresses the interpreted signs in a presupposed language. The act of writing is to create the physical representational manifestation of the information that was present or being thought up (mentally), and that now is "made manifest" on paper. Both the script and the physical object on which the script is written make up the *medium* in our terms. The dynamic representation for (28a) is shown in Figure 12: the *source* (i.e. the information being represented) is not expressed as a direct object but it can surface as a prepositional object ("write about her mother", where "mother" stands for "the information about the mother") as well as the *medium* ("write on paper"); both arguments can thus be defaulted and are notated as $_d$. The resource program variable $\vec{resource}$ tracks the transfer of information (trans, cf. (12d) in section 3) from a *source* through a *medium*.

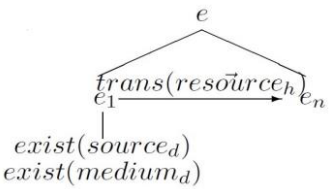

Figure 12: MARY WROTE ALL MORNING.

Also the compositional behavior associated with object selection for this class is similar to what we saw for *knit*, namely, object selection as seen in (28b) is the compositional introduction of a *test* through a quantized direct object. The dynamic predicate-argument structure representation for the variant with introduced object of this verb type is presented in Figure 13. Note that the head of the structure is e_2, i.e. the subevent associated with the argument, which is realized as direct object.

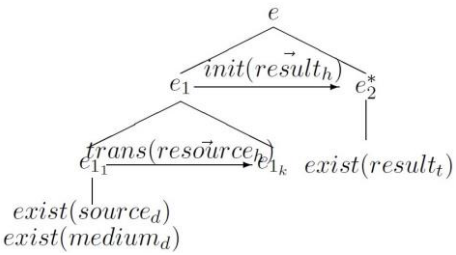

FIGURE 13: MARY WROTE A LETTER LAST NIGHT.

Consider now the second type of representational creation predicates from (27b) above, namely, the *representation source* verbs. In this class, the output object, i.e., the representational artifact, cannot be expressed syntactically, hence it qualifies as a hidden argument. This is a hallmark of *representation source* predicates, where the syntactic focus is on the source of information rather than the resulting object. Consider the following examples illustrating this well-known property.

(29) *John photographed Mary.*

This subclass can be defined by the *medium* on which the information is captured. The *medium* is an argument that cannot be expressed because it is incorporated in the root of the verb, i.e. it is a shadow argument in our terminology.

The dynamic predicate-argument structure representation is given below. Note that the *source* ("Mary") is obligatorily expressed (a true argument, notated as t), while the *medium* and the *result* remain unexpressed, as well as the program variables. We classify the program variables, as well as the resulting object, as hidden arguments, and the *medium* (the camera) as a shadow argument incorporate in the verb.[20] Note that the heading event is e_1, i.e., the subevent associated with the *source*, which is realized as direct object.

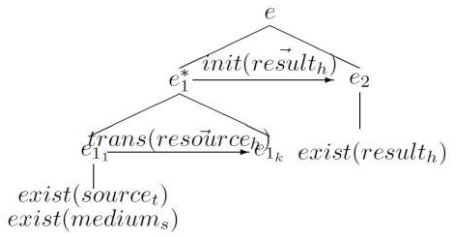

FIGURE 14: JOHN PHOTOGRAPHED MARY.

[20] The *source* can be either a physical object or a situation or event. What is relevant in this context, however, is its informational component. The source provides the information for the resulting representational artifact that is created by the photographing event.

A third class is represented by verbs which permit reference to either the *source* or the *result* of the creative act in direct object position, i.e. they alternate between a variant with the *source* as direct object, and a variant where the direct object realizes the created representational artifact, i.e. the *result*. An example is in (30) for *paint*, where the expressed argument in direct object position (the woman) is the *source*:[21]

(30) *John painted the woman sitting next to the table lamp.*

The representation for *paint* in (30) is illustrated below, where the *source* is the true argument that provides the information about the object or event to be represented, the *medium* is a shadow argument incorporated in the verb (see the unacceptability of *"John painted with paint"), and the *result* is a hidden argument. The head of the event is on e_1, which is the event associated with the *source*.

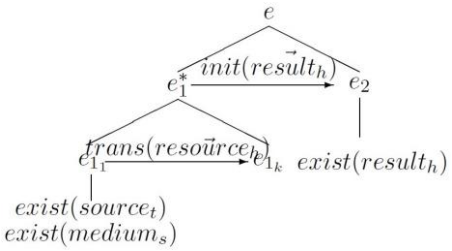

FIGURE 15: JOHN PAINTED A WOMAN SITTING NEXT TO THE TABLE LAMP.

In the case of alternating verbs, it is at the level of the event structure that argument realization is defined, i.e. the head of the event will be on e_1 in the case of the representation source variant (as in Figure 15) and on e_2 in the case of the representation result variant, as in "John painted a portrait", represented in Figure 16 below.

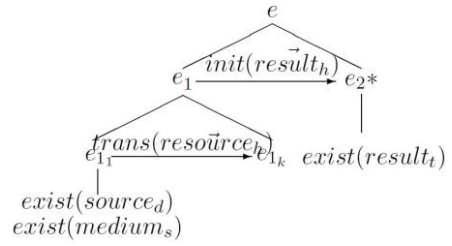

FIGURE 16: *John painted a portrait*.

[21] Note that as with *write*, *paint* is frequently found with no object, suggesting that it lexically encodes a Process (see "He painted all morning"). Differently from *write*, however, *paint* allows the direct object to express either the *source* or the *result*, when realized, whereas *write* only allows the *result* as direct object.

Notice that when the object position is occupied by the resulting representational artifact, the *source* may still be expressed as a complex NP, as in (31):

(31) *John painted a portrait of the woman sitting next to the table lamp.*

5. DESTRUCTION PREDICATES

In this section, we illustrate briefly how the dynamic predicate argument structure can be employed to differentiate the various types of destruction predicates. The purpose of this discussion is to demonstrate the application of the representation developed in the previous sections to destruction predicates generally.

Traditionally, the destruction of an object is seen as the act which takes that object out of existence (terminates it, in our terms). In our model, we examine destruction acts in more detail, and distinguish different kinds of destruction predicates depending on the type of *result* they encode. We claim that destruction predicates may either focus on what is being done to the preexisting object (the object being destroyed, identified as the input/*resource* variable in our model), or focus also on what is being brought about by the activity (what is the output/result of the termination of the input object). In this perspective, as observed in Jackendoff (1990: 118), destruction predicates are also creation predicates, since the process of transformation can be viewed as creating a result, although these verbs cannot express a created "product" (see also Levin 1993). The issue is of course one of whether the language makes any explicit or implicit reference to the new object resulting from the destruction of the old, if there is one.

Another point we focus on in our classification is whether the predicate focuses on the physical integrity of the object undergoing the change (i.e. its *Formal quale*, according to Pustejovsky's qualia theory), or primarily on its ability to be used for its purpose (i.e. its *Telic quale*).

5.1 Destruction through Modification

This class will include any process resulting in the termination of an object being classified as a given sortal type, through: (a) dis-aggregation; or (b) termination.

5.1.1 Dis-aggregation

This is the disassembly of an existing object, into its component parts. There are two grammaticalizations associated with verbs such as *disperse*, just as

there are with *assemble*, its inverse. The object being transformed and terminated can be selected (32a), or its component parts can be coerced (32b).

(32) a. *The crowd dispersed suddenly.*
b. *The student dispersed.*

The dynamic predicate-argument structure for the sense in (32a) is shown in Figure 17 below:

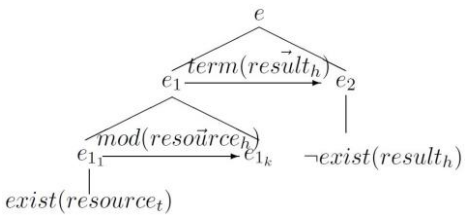

FIGURE 17: THE CROWD DISPERSED SUDDENLY.

It should be pointed out that, while the original group interpretation for the resource argument *the crowd* is terminated by the result of the event of dispersing, the component parts (i.e., the individual members of the crowd) will still exist.

5.1.2 Termination

Predicates of termination report events in which an entity is taken out of existence. Characteristic verbs in this class include the predicates *destroy*, *demolish*, *topple*, and *dissolve*, as shown below.

(33) a. *The earthquake destroyed Mary's house.*
b. *The fire demolished the church.*
c. *The winds toppled the tower.*
d. *Mary dissolved the tablet in water.*

With the predicates in (33), the expression of the resulting material (i.e., the output object) is usually odd (34), and remains hidden (Jackendoff 1990: 118).

(34) **The earthquake destroyed Mary's house into pieces.*

The syntactic focus is on the *resource*, while the semantic focus is on the result state of the transformation of this resource: i.e., non-existence, which acts as a *test* to the expression. When this result state is reached, that is, when the object is no longer classifiable as the nominal it was typed as the *resource*, the *test* is passed. The object is now any number of things, should there exist nominal classes to describe it, but it is identified by the predicate by what it no longer is. The dynamic event structure for *destroy* is the same as that given

for *disperse* in Figure 17 above. As with *build*, verbs such as *destroy* denote a directed process that is measured against a defined *test*, corresponding to the state of absence of the direct object argument.

Consider now inchoative forms of the verb *break*. There appear to be at least two senses of *break*: (a) the physical integrity of the object is changed substantially, as in (35a) below; and (b) the ability to use the object for its purpose is no longer possible, its functional integrity is compromised, as in (35b).[22]

(35) a. *The tree broke in two.*
b. *My computer broke back in July.*

Either sense can be considered a destruction predicate, but with different consequences. For the present discussion, we will focus on the sense in (35b), represented in Figure 18, where $result_{QT}$ stands for the *Telic Quale* of the *result* argument (i.e. its purpose or function), which is terminated as the output of the event:

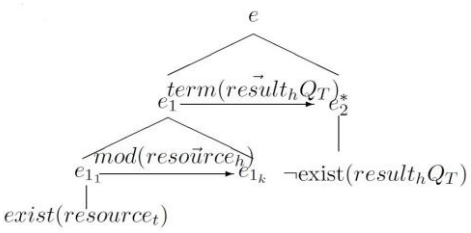

FIGURE 18: MY COMPUTER BROKE BACK IN JULY.

Consider now the following example:

(36) *The jar crashed on the floor and broke into fragments.*[23]

While *destroy* verbs do not allow the expression of the material entity resulting from the change as an optional prepositional phrase (see example (34)), *break* verbs (see Dixon 1991: 119) appear to allow it.

The representation of the dynamic predicate-argument structure of the expression in (36) is in Figure 19, where $result_{QC}$ stands for the *Constitutive Quale* of the *result*, which makes reference to the pieces (object parts) resulting from event, namely the fragments in (36):

[22] There also seems to be cases where both entailments are simultaneously present, as in "Susan broke her glasses".

[23] Note that when the resulting parts are expressed, as in the example above, *break* may also acquire the meaning of 'divide', as in "Mary broke the chocolate into small pieces"; we will not discuss these intentional uses of *break* here.

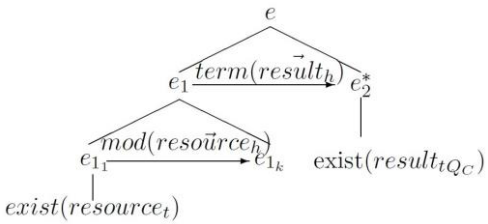

FIGURE 19: THE JAR CRASHED ON THE fIOOR AND BROKE INTO FRAGMENTS.

6. CONCLUDING OBSERVATIONS

In this paper we have provided a new representation of predicate-argument structures for verbs, in which the changes that the participants undergo during the event are encoded directly in the representation. As a case study, we have examined the major classes of creation and destruction predicates, in terms of *a dynamic predicate-argument structure*, i.e. a change profile of the predicate and its arguments, which tracks how the changed objects behave dynamically throughout the event.

Adopting the dynamic event structure first proposed for motion verbs in Pustejovsky & Moszkowicz (2011), enriched with roles for the arguments and extended to change of state verbs in Jezek & Pustejovsky (2017), we have provided a fully integrated representation that includes the argument roles (*resource*, *result*, *source*, *medium*), the argument types that are relevant for the syntactic realization (*true*, *defaulted*, *shadow*, *hidden*), and the modes inherent in the verb semantics that cause the changes in the arguments (*modification*, *initiation*, *termination*, *transfer*).

Moreover, by introducing the components of *assignment* and *testing* in the aspectual profile of the verbs and its arguments, and by associating them with an *ordinal* and a *nominal* scale respectively, we have offered a viable solution to the long standing problem of determining the exact contribution of each member of a predicate-argument structure to the measurement of the change.

Recall from section 1 that various scholars have observed that with expressions that denote a scalar change, it can be difficult to identify the exact source(s) of the measure of change. Rappaport Hovav 2008 for example has claimed that "the scale which occurs with incremental theme verbs (extent scale) is not directly encoded in the verb, but rather provided by the referent of the direct object". This has lead scholars to the assumption that when nominal reference plays a role in measuring the change, the verb is not associated with a scale (denoting a non-scalar change). We have claimed and showed instead that this latter conclusion is not necessary and that different kinds of scale can be referenced in a compositional process. According to Pustejovsky

& Jezek (2011) we have called compositional shifts in the scale of interpretation from *assignment* (*ordinal* scale) to *test* (*nominal* scale) *scale-shiftings*.

The dynamic predicate-argument structure proposed here can be easily transferred to other change of state verbs, for which different or additional modes of change and argument roles with respect to the ones listed in (12) and (13) might be necessary to represents the semantics of the verb and how it acts on its arguments dynamically.

REFERENCES

Atkins, B.T., J. Kegl & B. Levin (1988). Anatomy of a verb entry: From linguistic theory to lexicographic practice. *International Journal of Lexicography* 1(2). 84–126.

Beavers, J. (2008). Scalar complexity and the structure of events. In J. Dölling, T. Heyde Zybatow & M. Schäfer (eds.), *Event Structures in Linguistic Form and Interpretation*, 245–265. Berlin: Mouton de Gruyter.

Bisetto, A. & C. Melloni (2007). Result nominals: a lexical-semantic investigation. In G. Booij, B. Fradin, E. Guevara, S. Scalise & A. Ralli (eds.), *On-line Proceedings of the Fifth Mediterranean Morphology Meeting* (MMM5), Frjus 15-18 September 2005. Bologna: University of Bologna.

Dixon, R.M.W. (1991). *A New Approach to English Grammar, on Semantic Principles*. Oxford: Oxford University Press.

Dowty, D.R. (1979). *Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague's PTQ*. Berlin: Springer.

Dowty, D.R. (1991). Thematic proto-roles and argument selection. *Language* 67(3). 547–619.

Harel, D., D. Kozen & J. Tiuyn (2000). *Dynamic Logic*. Cambridge, MA: The MIT Press.

Hay, J., C. Kennedy & B. Levin. (1999). Scalar structure underlies telicity in "degree achievements". In T. Mathews & D. Strolovitch (eds.) *Proceedings of* SALT IX, Ithaca. 127–144.

Jackendoff, R. (1990). *Semantic Structures*. Cambridge, MA: The MIT press.

Jezek, E. (2014). Classes of creation verbs. In R. Simone & F. Masini (eds.) *Word Classes: Nature, Typology and Computational Representations*. Amsterdam: John Benjamins.

Jezek E. & C. Melloni (2011). Nominals, Polysemy and Co-predication. *Journal of Cognitive Science* 12. 1-31.

Jezek, E. & J. Pustejovsky. (2017). Dynamic Argument Structure. *Linguistic Issues in Language Technology* 15(3). 1–35.

Jezek, E. (2018). Partecipanti impliciti nella struttura argomentale dei verbi. In S. Dallabrida & P. Cordin (eds.) *La grammatica delle valenze*. Firenze: Franco Cesati.

Kennedy, C. (2009). Aspectual composition and scalar change. In Workshop on Scalarity and Event Structure. Paris.

Kilgarriff, A., P. Rychlý, P. Smrz & D. Tugwell (2004). The Sketch Engine. *Information Technology Research Institute* (ITRI) *Technical Report Series*. 105–116.

Kilgarriff, A., V. Baisa, J. Bušta, M. Jakubíček, V. Kovář, J. Michelfeit, P. Rychlý & V. Suchomel. (2014). The Sketch Engine: Ten Years on. *Lexicography* 1(1). 7–36.

Krantz, D., R. Luce, P. Suppes & A. Tversky (1971). *Foundations of Measurement* (vol. I: Additive and Polynomial Representations). Academic Press.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago: University of Chicago Press.

Levin, B. & M. Rappaport Hovav. (2010). Lexicalized scales and verbs of scalar change. In *Proceedings of the 46th Annual Meeting of the Chicago Linguistics Society*. 8–10.

McKeon, R. (1968). *The Basic Works of Aristotle*. New York: Random House.

Naumann, R. (2001). Aspects of changes: a dynamic event semantics. *Journal of Semantics* (18). 27–81.

Piñón, C. (2008). Verbs of creation. In J. Dölling, T. Heyde-Zybatow & M. Schäfer. (eds.). *Event Structures in Linguistic Form and Interpretation*, 493–521. Berlin: Mouton de Gruyter.

Pustejovsky, J. (1995). *The Generative Lexicon*. Cambridge, MA: The MIT Press.

Pustejovsky, J. (2013). Dynamic event structure and habitat theory. In N. Calzolari *et al.* (eds.), *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon*, Instituto di Linguistica Computazionale, CNR, Pisa.

Pustejovsky, J. & E. Jezek. (2011). Scale-shifting and Compositionality. In J. Fleischhauer, T. Gamerschlag & R. Osswald (eds.), *Proceedings of the Workshop on Scalarity in Verb-Based Constructions*, 45–48. Heinrich Heine Universität, Düsseldorf.

Pustejovsky, J. & J. Moszkowicz. (2011). The qualitative spatial dynamics of motion. *Spatial Cognition and Computation* 11(1). 15–44.

Rappaport Hovav, M. (2008). Lexicalized meaning and the internal temporal structure of events. In S. Rothstein (ed.), *Crosslinguistic and Theoretical Approaches to the Semantics of Aspect*, 13–42. Amsterdam: John Benjamins.

Resnik, P. (1996). Selectional Constraints: An information-theoretic Model and its Computational Realization. *Cognition* 61(1). 127–159.

Tenny, C.L. (1994). Aspectual Roles and the Syntax-semantics Interface. Berlin: Springer.

von Stechow, A. (2001). Temporally opaque arguments in verbs of creation. In C. Cecchetto (ed.), *Semantic Interfaces: Reference, Anaphora, and Aspect*, 278–319. Stanford, CA: CSLI Publications.

Elisabetta Jezek
Università degli Studi di Pavia
Dipartimento di Studi Umanistici
Strada Nuova 65, Pavia - Italy
email: jezek@unipv.it

James Pustejovsky
Brandeis University
Volen National Centre for Complex Systems
415 South Street
Waltham 02453, Massachusetts
U.S.A.
email: jamesp@brandeis.edu