*L*inguistic *I*ssues in *L*anguage *T*echnology – *LiLT*
Volume x, Issue y m 2017

Dynamic Argument Structure

Elisabetta Jezek
James Pustejovsky

Published by CSLI Publications

Dynamic Argument Structure

Elisabetta Jezek, *Università di Pavia,* James Pustejovsky, *Brandeis University*

This paper presents a new classification of verbs of change and modification, from a dynamic interpretation of the lexical semantics of the predicate and its arguments. Adopting the model of dynamic event structure proposed in Pustejovsky (2013), and extending the model of dynamic selection outlined in Pustejovsky and Jezek (2011), we define a verb class in terms of its *Dynamic Argument Structure* (DAS), a representation which encodes how the participants involved in the change behave as the event unfolds. We address how the logical resources and results of change predicates are realized syntactically, if at all, as well as how the exploitation of the resource results in the initiation or termination of a new object, i.e. the result. We show how DAS can be associated with a dynamically encoded event structure representation, which measures the change making reference to a scalar component, modelled in terms of *assignment* and/or *testing* of values of attributes of participants.

1.1 Introduction

In this paper we discuss the patterns in language that are exploited for expressing the semantics of change. We present a classification of change predicates based on a number of syntactic and semantic behaviors involving three semantic parameters: the nature of how the change is measured; the nature of the event capturing the change; and the nature of changed participants relative to the event itself.

LiLT Volume x, Issue y, m 2017.
Dynamic Argument Structure.

The kinds of change we will examine in this paper encompass those expressed by both creation and destruction predicates, i.e., predicates that denote the "coming into being" (*create* and *make*) and the "going out of being" of an entity (*disappear* and *die*).

We begin our discussion with perhaps the first systematic classification of change, what Aristotle calls "generation and corruption" in *de Generatione* (McKeon (1968)), outlined below.

(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*:
The destruction of an object into existing material
- disaggregation
- termination.

Both creation and destruction verbs have long posed a problem for linguistic classification (cf. Dowty (1991) and Tenny (1994), among others) and the interplay between their event structure, and the syntactic expressibility of their arguments is still not well understood (see Jezek (2013) for an overview).

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 role 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)) while still playing a role in the interpretation of the verb (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.

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

(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 (cf. Beavers (2008), Levin and Hovav (2010)) and dynamic event semantics (Pustejovsky (2000), Krifka (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 over the properties of the participants in each intermediate state during the event. However, they makes conflicting statements regarding the attribute available to construct the graduality scale for creation verbs, and no general consensus is 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 and Hovav (2010)). Finally, it does not take into account the contrastive data on argument variation introduced in (3) to (8) above.

In the following discussion, we propose a novel analysis of these predicates, based on a dynamic interpretation of their event and argument structure and its interplay with syntactic realization. We identify what we believe are the most essential parameters of behavior dictating the expression of change for creation and destruction predicates in language, what could be called a verb's "change profile". These parameters are the linguistic dimensions as defined by the answers to the following questions:

(9) a. Is the created/destroyed object realized in the syntax?
b. What is the semantic type of the resulting object in a creation?
c. What counts as a result of a destruction act?
d. Does the creation act make use of existing resources? How are these exploited in the event?
e. Are the resources that are used in the event themselves realized in the syntax?

The answers to these questions will help in creating a coherent classification of change predicates in language. To this end, the major contribution of this study is to characterize each class in terms of its *dynamic argument structure* (DAS), a representation, the features of which correspond to how the questions in (9) are answered (section 2). We will also show how the information encoded in the dynamic argument structure can be mapped onto a dynamic event representation, which tracks the temporal unfolding of the event. The theoretical underpinnings are introduced in section 1.2, while the analysis of members for each of the proposed class in provided in section 1.3 for creation classes and 1.4 for destruction ones. Section 1.5 examines the consequences of the proposed representation for DAS to the computational interpretation of linguistic expressions.

1.2 Dynamic Argument Structure

For the discussion that follows, we develop a model of dynamic selection, our goal being to motivate the minimal typology of change required to describe the linguistic behavior of creation and destruction predicates and their arguments. We will argue that the semantics of change, as expressed in verbal predicates, should include specification of how how the event participants behave throughout the event: that is, a participant can stay the same, be modified, come into existence, or be terminated. In the following, we will first review the dynamic model of events in which our proposal is couched; then we will outline Dynamic Argument Structure in full and show how is can be mapped onto the structure of the whole event.

1.2.1 Dynamic Event Structure

Let us then first examine how change can be semantically encoded in a verbal entry. For the present discussion, we start with the basic framework of event-based verbal semantics, as outlined in Generative Lexicon (Pustejovsky (1995)). On this view, event types are defined structurally in terms of a syntax of event and sub-event structure, capturing the conventional Vendlerian Aktionsart-based typology for events.

(10) 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

Event structure is but one component of the lexical semantic specification for a predicate P, as illustrated in (11).

(11)
$$\begin{bmatrix} P & \\ \text{AS} = & \begin{bmatrix} \text{A1} = x \\ \text{A2} = y \end{bmatrix} \\ \text{ES} = & \begin{bmatrix} \text{E1} = e_1 \\ \text{E2} = e_2 \end{bmatrix} \\ \text{QUALIA} = & \begin{bmatrix} \text{CONST} = \ldots \\ \text{FORMAL} = P_2(e_2, y) \\ \text{TELIC} = \ldots \\ \text{AGENTIVE} = P_1(e_1, x, y) \end{bmatrix} \end{bmatrix}$$

When integrated with the argument structure (AS) and qualia structure for a predicate as in (11), one can see that the qualia roles are temporally ordered, and the predicates associated with each subevent are interpreted as ordered frames of interpretation (Bouillon, 1997). For example, for a transitive verb denoting a transition (such as *kill*), the AGENTIVE role, introducing the causing act, precedes the FORMAL role (referencing the end state) in a complex event structure. Such predicates denote an event e, with a structure of $[e_1\ e_2]_e$, where distinct subpredicates occupy each frame (subevent).

Without an explicit representation of change of state, however, the representation in (11) does not adequately model change dynamically. For this reason, the concept of *opposition structure* was introduced in GL as an enrichment to the above representation (**?**), making explicit what predicate opposition is lexically encoded in a predicate. For example, the verbs *die* and *kill* are both encoded with the opposition structure $[\neg dead(x), dead(x)]$, (for their subject and object, respectively), and are associated as states holding during the sub-events e_1 and e_2. For example, the entry in (11) would be enriched to:

(12) $$\begin{bmatrix} P \\ \text{QUALIA} = \begin{bmatrix} \text{FORMAL} = [dead(e_2, y)] \\ \text{AGENTIVE} = [P_1(e_1, x, y) \wedge \neg dead(e_3, y)] \end{bmatrix} \end{bmatrix}$$

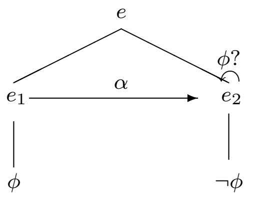
Incremental changes such as *climb* or *fall*, however, are still difficult to capture within this representation. In Pustejovsky and Moszkowicz (2011) a first proposal for a dynamically interpreted event structure is presented, which captures the scalar change associated with motion events. This is then extended to the class of incremental change predicates in Pustejovsky and Jezek (2011). Pustejovsky (2013) locates these insights within a global model of dynamic event structure representation in language.

All these studies accommodate change in the *assignment* of values to the relevant attributes of the participants in the event *being tracked* over time. Vendler classes are reinterpreted in terms of dynamic event structures. In a dynamic approach to modeling changes, there is a distinction made between formulas, ϕ, and programs, π (cf. Harel et al. (2000). A formula is interpreted as a proposition, evaluated relative to a specific state in the model. Programs are either atomic and 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 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 (13).

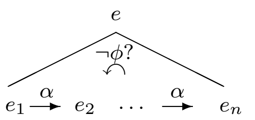
(13)



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 $\phi?$ is identical to the distinguished endpoint encoded in the event semantics ϕ, and stops when this point is reached. In our model, a test program references a *nominal scale*, a scale that introduces a dichotomy that did not exist before. In our use of the term *scale*, we depart from the conventional linguistic terminology, according to which a scale is usually understood as an ordinal structure (either two-point or multipoint, cf. Rappaport Hovav (2008)), and make reference in particular to the distinction between *nominal* and *ordinal* scales as discussed in Suppes and Tversky (1989). 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.

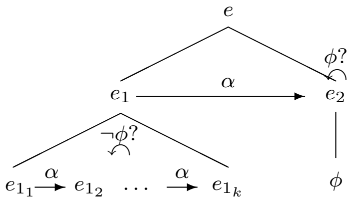
According to Pustejovsky (2013) we can define a *process*, in dynamic terms, as a sequence of transitions, where at each transition, the value of an attribute associated with an argument, is reassigned (Pustejovsky and Moszkowicz (2011)) (cf. 14). The incremental change in the attribute associated with the argument is encoded as an *assignment* program that references an *ordinal scale.* Verbs encoding processes are *assignment predicates.*

(14)



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* whether a proposition, ϕ, is true, i.e., ϕ?: the process continues while the test is not satisfied – $\neg\phi$? – until it finally is, i.e., ϕ? (Pustejovsky and Moszkowicz (2011)). This is represented in (15). Accomplishments verbs encode both an *assignment* and a *test* component.

(15)



1.2.2 Dynamic Argument Typing

As referenced at the very beginning of this section, in this paper we enrich and refine the analysis of change events reviewed in section 1.2.1 above by focusing on the participants of the event. That is, we take into account the argument structure of the verbs in questions, and examine and encode how each participant changes while the event unfolds over time. Particularly, we propose a new representational model for argument structure information encoded in verbs, called *Dynamic Argument Structure* (DAS). The Dynamic Argument Structure encodes how an argument to an event contributes to the change being expressed. DAS identifies the following properties for each argument to a predicate: the **mode** of change it undergoes; its semantic **role**; its semantic **type**; and whether it is **syntactically** realized.

We will distinguish four primary **modes of change** that a participant may assume, each of which refers to the role played by that specific argument as the event unfolds.[1] They are:

(16) a. Modification
b. Initiation
c. Termination
d. Merging

An argument undergoes *modification* if there is an identifiable attribute whose value is changed during the event. A modification may lead to a characteristic or distinguished change that is associated with a reified predication. In this case, an argument undergoes *initiation* if it is brought into existence as predicated by the verb, and undergoes *termination* if the converse is true. When two objects are brought together to bring about a new object, we say that they are *merged*. In this sense, merge can be seen as specific way to initiate a new object.

In the representations that follow, we also consider the *existence* mode. This is not a mode of change: it is a mode that qualifies the resource that exists prior 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).

As regards typing, we assume that the **type** of an argument is a value selected from an inventory of types in the language (Pustejovsky (1995), Asher and Pustejovsky (2006)). In addition to the Montague types, e and t, we assume a richer subtyping over the entity domain than is typically assumed in type theory , including complex types (Pustejovsky (1995)), that is, types which are composed by more than one type, such as the type associated with the word *book*, which include a physical and an informational component. For the present study, the relevant types are (17):

(17) a. physical object ($phys$), i.e.
b. informational object ($info$), i.e.
c. physical object and informational object ($phys \bullet info$), i.e. *book*, *letter*, etc.
e. event ($event$).

We will also distinguish collections ($group$s) from $individual$s, and

[1] A fourth 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.

use *entity* for the most general type.

Concerning the notion of **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. We will draw on the following semantic roles for characterizing the behavior of arguments of creation and destruction predicates during an event.

(18) 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: used for Dowty's source of representation;
d. Medium: the resource used by an animate object acting intentionally in an event that brings about a new object through the representation of an existing one.

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 and Moszkowicz (2011) for motion predicates and in Pustejovsky and 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 both the values of the attribute of the resource being modified - with change of state verbs, and the intermediate stages of the "result-to-be" - with creation or destruction predicates. It is annotated vith 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 $\overrightarrow{result}$.

Finally, the feature **syn** encodes the conditions under which an argument is expressed in the syntax. There are three possible values: *expressed*, *covert*, and *optional*. Both covert and optional arguments are subcategorized by the predicate and present at the level of semantic representation; but while the former can never be expressed, the latter are arguments whose expressibility is not mandatory. An example of covert argument is the dynamic program variable, which is present in the dynamic inspection of the event but not expressed syntactically.

Now let us consider how to encode these features of the arguments to a verb. For this purpose, we use a feature structure representation. The argument-based change profile DAS for a transitive verb, $\mathbf{verb}(x, y)$ is illustrated below. In the following, we will use the roles in (18) to label the variables, in order to have a better view of the dynamic behavior of the resource and the result through the change.

$$(19)\quad \begin{bmatrix} \textbf{verb(x,y)} & \\ \text{DAS} = & \begin{bmatrix} x : = [\textbf{type,mode,syn}] \\ y : = [\textbf{type,mode,syn}] \end{bmatrix} \end{bmatrix}$$

(19) can be seen as an abstraction of the phases of the event as encoded through the arguments. Hence, within this representation, there is no reference to the actual sub-predication in each event in the event structure. The question then becomes how one integrates the argument-based change profiles into a dynamic event structure representation. We suggest that annotating the arguments in the event structure with the modes of change introduced in DAS actually captures dynamics directly. That is, with this enriched notation, we are able to recover the dynamic aspects of the event from the annotation on the arguments to the verb.

In the remainder of the paper, we examine how the classification of creation and destruction predicates can be analyzed and represented in terms of the dynamic typing and event structure proposed above. For current purposes, we will simplify the representations in (13,) (14) and (15), and exclude the notation of the *test/assignment* components, assuming that they are present according to the model presented in section 1.2.1, namely: simple transitions as in (13) encode a *test* component, processes as in (14) encode an *assignment* component and accomplishments as in (15) encode both.

1.3 Creation Predicates

In this section we introduce our classification of creation predicates. For each class, we give a definition with examples, then we supply the formal analysis, and the compositional derivations, based on our model.

1.3.1 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.

Natural Growth

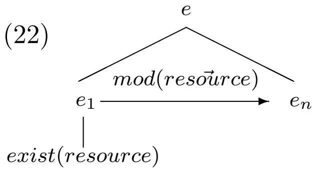
There are two aspects of natural growth. One aspect is the natural manifestation of an entity from another by virtue of the character of its inherent properties. Examples of such predicates can be found in Levin (1993) "grow verbs" class, which includes the verbs *develop*, *evolve*, *grow*, *hatch*, and *mature*. The other aspect relates to the same process, but focuses on the incremental change that the object undergoes. Both of these aspects are manifested in the two senses of the verb *grow.* To illustrate, consider the sentences below in (20).

(20) a. A tree is growing in the back yard.
b. A lump grew on the patient's arm.

In (20a), reference is made to the participant which undergoes the modification, i.e., an incremental theme argument. This property is reflected in the DAS and the dynamic event structure for the incremental sense of *grow*, as shown below in (21) and (22).

(21) $$\begin{bmatrix} \textit{grow} & \\ \text{DAS} = & \begin{bmatrix} \textit{resource} = & [\textbf{phys,exist,exp}] \\ \vec{\textit{resource}} = & [\textbf{phys,mod,cov}] \end{bmatrix} \end{bmatrix}$$

(22)

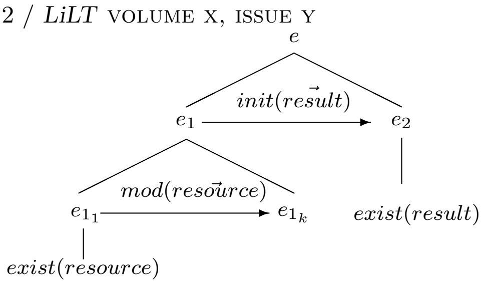


s The participant undergoing the change in (20a) is interpreted in (21) and (22) as the resource of the *grow* event. The modification mode (*mod*) applies to the resource program variable, i.e. the resource in its development. The program variable is a covert argument that captures the incremental changes the resource goes through as an argument of the *grow* function. 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 *mod.*

By contrast, in (20b), an object is brought into existence due to naturally occurring circumstances. The entity (a lump) is initiated through the modification of existing resources, which are left unexpressed. The DAS and the dynamic event structure for the creation sense of *grow* are represented in (23) and (24) respectively:

(23) $$\begin{bmatrix} \textit{grow} & \\ \text{DAS} = & \begin{bmatrix} \textit{resource} = & [\textbf{phys,exist,cov}] \\ \vec{\textit{resource}} = & [\textbf{phys,mod,cov}] \\ \vec{\textit{result}} = & [\textbf{phys,init,cov}] \\ \textit{result} = & [\textbf{phys,exist,exp}] \end{bmatrix} \end{bmatrix}$$

(24)



The difference in the two senses of *grow*, as used in (20), is captured by the distinct DAS representations above. The incremental theme reading of *grow* in (20a) is entailed by the sense of *grow* expressed in (24), where a new resulting object is identified by virtue of the process. There are two dynamic variables: one encodes the modification of the resource, the other one the coming into being of the resulting created object, in other words, $res\vec{o}urce = re\vec{s}ult$. The dynamic variables encode the modification of what is changing in the object being tracked, from both the perspective of the initial resource and of the resulting created object.

Compare this to the sentences in (25), where both resource and result objects are syntactically realized.

(25) a. The sapling *grew* into a beautiful tree.
b. The small boy *matured* into a healthy man.
c. The storm *developed* into a hurricane.

For the examples in (25), more so than for those in (20b), we can think of the resulting (created) object that appears in the PP complement as part of the "natural trajectory" of the resource object.[2] The representation of the DAS of the expression in (26) is given in (27).

(26) The sapling grew into a tree.

(27) $$\left[\text{DAS} = \begin{bmatrix} resource = [\textbf{phys,exist,exp}] \\ res\vec{o}urce = [\textbf{phys,mod,cov}] \\ re\vec{s}ult = [\textbf{phys,init,cov}] \\ result = [\textbf{phys,exist,exp}] \end{bmatrix}\right]$$

In (27) the sapling (as subject) is the syntactic manifestation of the resource. The dynamic variable $res\vec{o}urce$ 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 ($re\vec{s}ult$). While the resource

[2] The natural stages in a life cycle would be conceptualized as a large semantic structure. Possible model is the *projective structure* from Pustejovsky (1995).

continues on in the guise of $\vec{resource}$, reference to anything as the nominal classification of the resource (i.e., a sapling), is no longer possible, since a new type, i.e. $\vec{result}$ is initiated. The corresponding dynamic event structure mapping is the same as that in (24).

The lexical typing for this class of verbs will have to reflect the polyadicity demonstrated by both pure intransitive and PP-selecting intransitive forms. This can be accomplished by assuming that in the case of pure intransitives as in (20b), the last subevent is headed as a result of pragmatic factors, resulting in the realization of the corresponding result argument, while PP-selecting intransitive forms as in (25) the whole event is headed, resulting in the realization of both the resource and the result.

Aggregation

This class refers to the assembly or coming together of existing resources to create a distinct object, identified by a newly introduced nominal expression. Included in this class are the verbs *assemble*, *collect*, *gather*, and *congregate*. The resource encodes the referential objects available prior to the change, and can be either presupposed or directly selected as an argument. For those verbs that have intransitive forms (*assemble* and *gather*), this distinction can be clearly illustrated with the examples in (28).

(28) a. A crowd assembled / gathered in the lobby.
b. The students assembled / gathered in the lobby.

Notice that in (28a), 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 and gathering events. For this verb class, the key feature is, of course, the relationship between a set of individuals serving as the resource type, and a collection or group denoting the result type. Hence, even though the selectional restrictions on (28b) reflect the former constraint, we can interpret the NP in subject position (*the students*) as *coerced* into a group.[3]

This is reflected in the DAS and dynamic event structure representations below. Consider first (29) and the argument change profile in (30) (where ind[+pl] stands for a plurality of individuals).

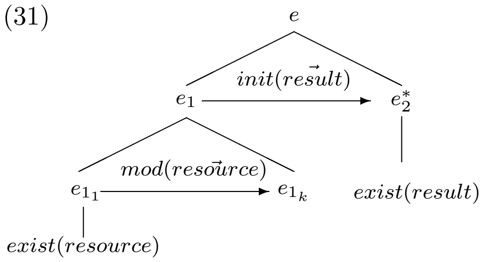
[3] Note the distinction between the typing restriction of [+*plural*] from the predicate *assemble* onto its subject, and the output condition (resulting type) from the computation, viz., a group or crowd. If this is a kind of coercion, then it is one that is captured only over a dynamic encoding of the domain and co-domain of the coercion function.

(29) A crowd assembled in the lobby.

(30)
$$\begin{bmatrix} \mathit{assemble} & \\ \mathrm{DAS} = & \begin{bmatrix} \mathit{resource} = [\textbf{ind}[\textbf{+pl}],\textbf{exist},\textbf{cov}] \\ \vec{\mathit{resource}} = [\textbf{ind}[\textbf{+pl}],\textbf{mod},\textbf{cov}] \\ \vec{\mathit{result}} = [\textbf{group},\textbf{init},\textbf{cov}] \\ \mathit{result} = [\textbf{group},\textbf{exist},\textbf{exp}] \end{bmatrix} \end{bmatrix}$$

The corresponding Dynamic Event Structure is in (31) (note that the headed subevent is e_2, i.e., the one associated with the result argument):

(31)



Now consider the sentence in (32), where the result is left covert, but the resource is expressed syntactically.

(32) The students assembled in the lobby.

The DAS of (32) is the same as (30), with the difference that the resource is expressed instead of the result:

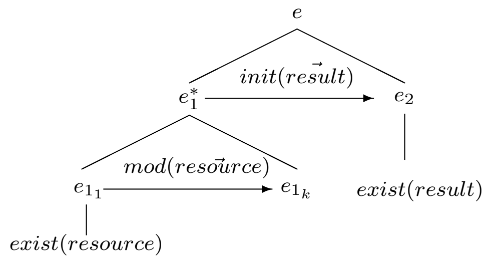
(33)
$$\begin{bmatrix} \mathit{assemble} & \\ \mathrm{DAS} = & \begin{bmatrix} \mathit{resource} = [\textbf{ind}[\textbf{+pl}],\textbf{exist},\textbf{exp}] \\ \vec{\mathit{resource}} = [\textbf{ind}[\textbf{+pl}],\textbf{mod},\textbf{cov}] \\ \vec{\mathit{result}} = [\textbf{group},\textbf{init},\textbf{cov}] \\ \mathit{result} = [\textbf{group},\textbf{exist},\textbf{cov}] \end{bmatrix} \end{bmatrix}$$

The dynamic event structure is also the same as (30), with the exception that the head of the event is in this case e_1, i.e., the subevent associated with the resource argument.[4]

(35)

[4] Notice that transitive forms for such verbs allow alternations on the direct object, as illustrated by the sentences in (34), where in (34a) the direct object encodes the result, whereas in (34b), it encodes the resource.

(34) a. Mary assembled the bookshelf.
b. Mary assembled the parts into a bookshelf.



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 (1992), Levin (1993), von Stechow (2001), Piñón (2008)), with verbs such as *build*, *knit*, *bake*, *create*, *produce*, *compose*, and so on (cf. Levin (1993)'s "create" verbs). 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 Bisetto and Melloni (2007) and Jezek (2013)). 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, as in *bake.*

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. 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) between transitive creation verbs which may exhibit an object-drop alternation (e.g., *draw*, *write*, *carve*, *knit*, *paint*) and those which typically do not exhibit this alternation (e.g., *build*, *construct*, *create*, *make*, *produce*). This distinction is illustrated below, where *build* and *construct* are contrasted with *knit* (see also Section 1).

(36) a. John built a wooden bookcase.
b. *John has been building for weeks.

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

(38) a. Mary knitted a sweater.
b. Mary knitted yesterday evening.

Consider first *knit* in the context below:

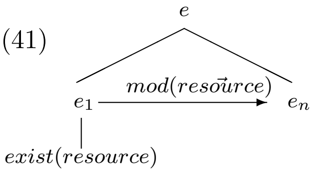
(39) Mary knitted yesterday evening.

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 object being knit. 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 DAS in (40) as a dynamic program variable, ($reso\vec{u}rce$), next to the resource argument available at the beginning of the event.[5]

(40)
$$\left[\begin{array}{l}\textbf{knit}\\ \text{DAS} = \left[\begin{array}{lll} resource & = & [\textbf{phys, exist, cov}]\\ reso\vec{u}rce & = & [\textbf{phys,mod,cov}]\end{array}\right]\end{array}\right]$$

The corresponding dynamic event structure is in (41):

(41)



The representation in (41) resembles the one proposed for *grow* incremental in (22), the difference being that with *grow*, the resource is realized in the syntax as subject.

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 the verb's semantics - namely as their selected direct object argument - which is typed as a quantized physical object. We contend that it is this reference to a test that makes object-drop typically impossible for such predicates.

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 program variable, as well as the argument identified with the test.

The DAS for this verb given in (42) introduces different modes for distinct arguments: first, an *initiation* mode will characterize the creation of the argument, expressed as the direct object; the resource is

[5] Recall that in our model, we factor out the Agent and focus on the Resource and the Result arguments, i.e. the participant undergoing change.

annotated as *existing* independently of the event, but *modified* via the program variable ($\overrightarrow{resource}$) that will become the resulting creation ($\overrightarrow{result}$). As in the case of *grow*, $\overrightarrow{resource} = \overrightarrow{result}$.

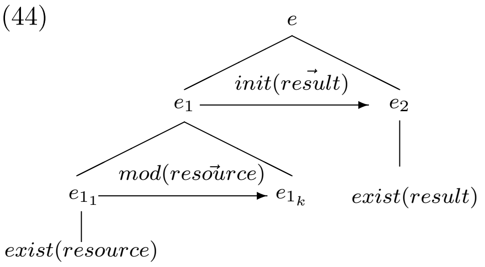
(42)
$$\begin{bmatrix} \textbf{build} & \\ \text{DAS} = & \begin{bmatrix} resource = [\textbf{phys,exist,opt}] \\ \overrightarrow{resource} = [\textbf{phys,mod,cov}] \\ \overrightarrow{result} = [\textbf{artefact,init,cov}] \\ result = [\textbf{artefact,exist,exp}] \end{bmatrix} \end{bmatrix}$$

Notice that *build* allows for the optional expression of the resource argument, y. This can be accomplished in one of two ways, illustrated in (43) below.

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

Let us now look how the DAS of *build* maps onto the dynamic event structure. We assume a lexically specified accomplishment structure for *build*; during the process component, the resource is modified, until the new object is initiated.

(44)


e
$init(\overrightarrow{result})$
e_1
e_2
$mod(\overrightarrow{resource})$
e_{1_1}
e_{1_k}
$exist(result)$
$exist(resource)$

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 (45).

(45) Mary knitted a sweater last night.

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

(46) 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 (45) 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.

(47) 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 created by the semantics of the assignment predicate; *knit until noon.*

Consider again the accomplishment interpretation available with the verb *knit* in (45) in the light of what we just said. In the example in (45), 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 DAS for the expression is as follow:

$$(48)\quad \begin{bmatrix} \textbf{knit a sweater} & \\ \text{DAS} = & \begin{bmatrix} \mathit{resource} = [\textbf{phys,exist,cov}] \\ \vec{\mathit{resource}} = [\textbf{phys,mod,cov}] \\ \vec{\mathit{result}} = [\textbf{phys,init,cov}] \\ \mathit{result} = [\textbf{phys,exist,exp}] \end{bmatrix} \end{bmatrix}$$

The corresponding dynamic event structure of the compositional expression *knit a sweater* is the same as that for *build*, as shown in (44). 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 and Jezek (2011).[6]

[6] As pointed out in (47), durative adverbials are a potential test expression in composition with assignment predicates. Hence, instead of argument introduction (option (47a) above), adjunction is possible. For example, in the sentence in (49),

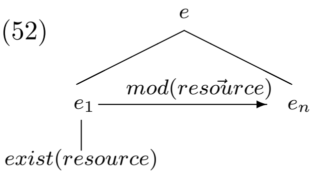
To conclude this section, we turn to the issue of how can we model verbs which exhibit an alternation between a change of state sense and a creation sense, namely, the polysemy exhibited by *bake* in (50a-b) and (50c) (cf. Atkins et al. (1988)):

(50) a. Mary baked the potato.
b. Mary baked the potato in 20 minutes.
c. Mary baked a cake in 20 minutes.

First, we must adequately model how the object in the change-of-state sense of *bake* in (50a) is being modified. We will treat the resource in the event as undergoing change, through a modification function. As with *build*, we model the modification function as being associated with the covert program variable ($res\vec{ou}rce$) introduced by the process. We can state the DAS and the dynamic event structure for the process reading of *bake* with object as follows, where the resource is expressed as direct object:

(51) $$\begin{bmatrix} \textbf{bake the potato} \\ \text{DAS} = \begin{bmatrix} resource = [\textbf{phys,exist,exp}] \\ res\vec{ou}rce = [\textbf{phys,mod,cov}] \end{bmatrix} \end{bmatrix}$$

(52)



Under the create interpretation for *bake* (50c), the resource (the ingredients being used) is modified but not expressed, while the output type is initiated, from the modified resource. The DAS for the create sense of *bake* reflects this observation. The co-composition reading that gives rise to the accomplishment reading in (50c) is shown below. This can refer back to the co-composition discussion from above.

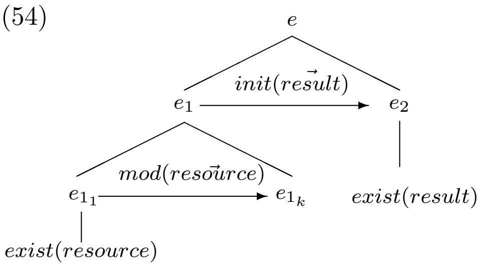
The creation sense is attained through introduction of an argument, particularly by co-occurrence with food artifacts, such as *bread*, *cake*, *cookies*, etc.

(49) Mary knitted for an hour.

the durative adverbial introduces a test over the assignment predication.

(53)
$$\left[\begin{array}{l}\textbf{bake a cake}\\ DAS = \left[\begin{array}{lcl} resource &=& [\textbf{phys,exist,cov}]\\ \vec{resource} &=& [\textbf{phys,mod,cov}]\\ \vec{result} &=& [\textbf{phys,mod,cov}]\\ result &=& [\textbf{phys,exist,exp}]\end{array}\right]\\ \end{array}\right]$$

(54)

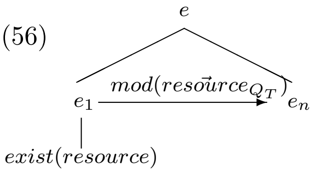


Now consider (50b) that we repeat below for convenience:

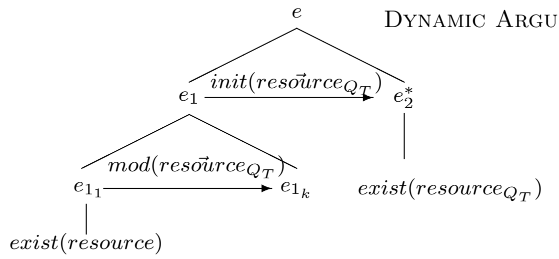
(55) Mary baked the potato in 20 minutes.

In this case, the frame adverbial “in 20 minutes” introduces the test which turns the whole expression into an accomplishment. This shift is captured below, where (56) is the representation of dynamic event structure encoded in *bake*, and (57) is the representation of the co-compositional interpretation. We claim that with “bake the potato in 20 minutes”, the modification function applied to $\vec{resource}$ triggers a sublexical feature of its semantics, namely its TELIC quale role, whose value is edibility. This is represented in (56) and (57) by the notation $resource_{Q_T}$. On the other end, the frame adverbial “in 20 minutes” introduces the attainment of the resulting state corresponding to the edibility of the potato.

(56)



(57)



1.3.2 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.[7]

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 (58a), the source of the representation (58b) as direct object, or can alternate between the two (58c):

(58) a. RESULT VERBS: *write.*
b. SOURCE VERBS: *photograph.*
c. ALTERNATING VERBS: *draw*, *paint*, *film.*

We first consider the subset of class (58a) 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 and (b) the created object is a representational artifact. Such verbs typically exhibit also a variant with no expressed created object, as repeated below.

(59) 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; a *medium* in our terms) that express 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 the paper. The basic DAS for (59a) is shown in (60): the *source* is optional (in the sense that it cannot

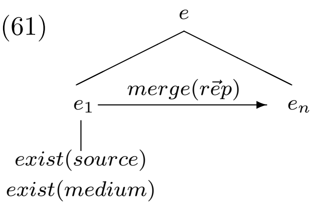
[7] See Dowty (1979), Dowty (1991) for early work on the interaction of "Representation" and "Source" in such verbs.

be expressed as a direct object but it can surface as indirect object: see "write about her mother") as well as the *medium* (see "write on paper"); the *representation* program variable tracks the creation of the complex object derived by merging *source* and *medium.*

(60)
$$\begin{bmatrix} \textbf{write} & \\ DAS = & \begin{bmatrix} source = [\textbf{info,exist,opt}] \\ medium = [\textbf{phys,exist,opt}] \\ \overrightarrow{representation} = [\textbf{phys}\bullet\textbf{info,merge,cov}] \end{bmatrix} \end{bmatrix}$$

The corresponding dynamic event structure representation is as follows:

(61)



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 (62) is the compositional introduction of a test through a quantized direct object.

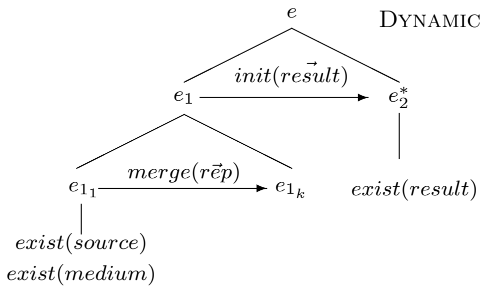
(62) Mary wrote a letter last night.

The DAS for the variant with introduced object of this verb type is shown below.

(63)
$$\begin{bmatrix} write & \\ \text{DAS} = & \begin{bmatrix} source = [\textbf{info,exist,opt}] \\ medium = [\textbf{phys,exist,opt}] \\ \overrightarrow{representation} = [\textbf{phys}\bullet\textbf{info,merge,cov}] \\ \overrightarrow{result} = [\textbf{phys}\bullet\textbf{info,init,cov}] \\ result = [\textbf{phys}\bullet\textbf{info,exist,exp}] \end{bmatrix} \end{bmatrix}$$

The dynamic event structure representation is as follows. Note that the head of the dynamic event structure is e_2, i.e. the subevent associated with the argument which is realized as direct object.

(64)



Consider now the second type of representational creation predicate from (58b) above, namely, the *representation source* verbs. In this class, the output object, i.e., the representational artifact cannot be expressed syntactically. 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.

(65) John photographed Mary.

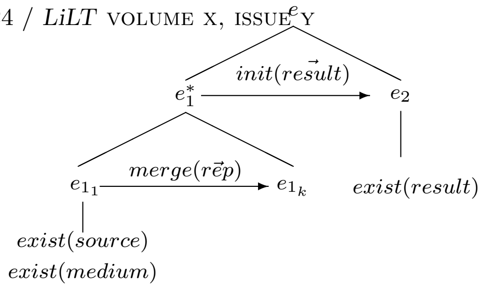
This subclass can be defined by the medium on which the information is captured. The medium is an argument that cannot be expressed because it is already incorporated in the root of the verb, i.e. it is a covert argument in our terminology.
The DAS of *photograph* can be given as follows. Note that the source ("Mary") is expressed, while the medium and the result remain covert, as well as the program variables. [8]

(66)
$$\begin{bmatrix} \mathit{photograph} & \\ \text{DAS} = & \begin{bmatrix} \mathit{source} = [\mathbf{info,exist,exp}] \\ \mathit{medium} = [\mathbf{phys,exist,cov}] \\ \mathit{\vec{representation}} = [\mathbf{phys{\bullet}info,merge,cov}] \\ \mathit{\vec{result}} = [\mathbf{phys{\bullet}info,init,cov}] \\ \mathit{result} = [\mathbf{phys{\bullet}info,exist,cov}] \end{bmatrix} \end{bmatrix}$$

The event representation is given below. Note that in this case the heading event is e_1, i.e., the subevent associated with the source, which is realized as direct object.

(67)

[8] The type of the source can be either a physical object or a situation or event. We will use $info$ as a cover term to indicate the information about the object or the event that is being reproduced. The important thing to realize here is that the source provides the information for the resulting representational artifact, the phys•info that is created by the photographing event.



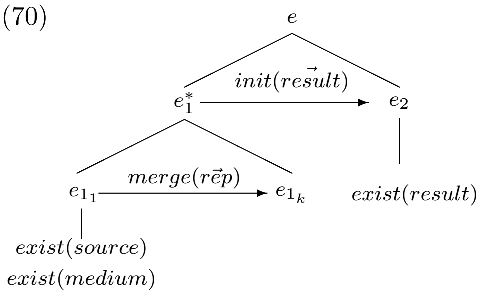
A third class is represented by verbs which permit reference to either the source or to the result of the creative act in direct object position, i.e. they alternate between a variant with the *source* as direct object, and a variant where the direct object realizes the created representational artifact, i.e. the *result*. An example is in (68) for *paint*, where the expressed argument in direct object position is the *source*:

(68) John painted the mountain.

The DAS for *paint* in (68) is illustrated below, where the *source* is typed $info$, i.e. information about the interpreted object or event, which might be real or imagined.

(69)
$$\begin{bmatrix} \textit{paint} \\ \text{DAS} = \begin{bmatrix} \textit{source} = [\textbf{info,exist,exp}] \\ \textit{medium} = [\textbf{phys,exist,opt}] \\ \overrightarrow{\textit{representation}} = [\textbf{phys}\bullet\textbf{info,merge,cov}] \\ \overrightarrow{\textit{result}} = [\textbf{phys}\bullet\textbf{info,init,cov}] \\ \textit{result} = [\textbf{phys}\bullet\textbf{info,exist,cov}] \end{bmatrix} \end{bmatrix}$$

(70)



In the case of alternating verbs, it is at the level of 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, and on e_2 in the case of the representation result variant, as in (71) below.

(71) Mary painted a portrait.[9]

[9] Notice that when the object position is occupied by the resulting representa-

Another verb within the class is *film*, as in (74):

(74) a. Mary filmed the wedding. (source)
b. Mary filmed a movie. (result)

Because in (74b), the initiated representation-artifact (the movie) is expressed, the DAS is distinct from that of *photograph* in (66) and is parallel to that of *write* in (63).[10]

$$(76)\quad \left[\begin{array}{l} \mathit{film} \\ \text{DAS} = \left[\begin{array}{l} \mathit{source} = [\textbf{info,exist,opt}] \\ \mathit{medium} = [\textbf{phys,exist,cov}] \\ \overset{\rightarrow}{\mathit{representation}} = [\textbf{phys}\bullet\textbf{info,merge,cov}] \\ \overset{\rightarrow}{\mathit{result}} = [\textbf{phys}\bullet\textbf{info,init,cov}] \\ \mathit{result} = [\textbf{phys}\bullet\textbf{info,exist,opt}] \end{array}\right] \end{array}\right]$$

Finally, the verb *copy* stands out as a special case. The difference between *copy* and the rest of the class of representation verbs, is that the former presupposes the existence of an object that is already typed as phys•info, where *phys* can be *media* of different kinds. In the case of *copy*, the *medium* is not specified in the argument structure of the verb as a separate participant, as in the case of *photograph*, *tape*, and *record*. What *copy* reports is the pure re-creation of an object, by way of representation. Consider (77):

(77) Mary copied the file.

The corresponding DAS is:

$$(78)\quad \left[\begin{array}{l} \mathit{copy} \\ \text{DAS} = \left[\begin{array}{l} \mathit{source} = [\textbf{phys}\bullet\textbf{info,exist,exp}] \\ \mathit{medium} = [\textbf{phys,exist,cov}] \\ \overset{\rightarrow}{\mathit{result}} = [\textbf{phys}\bullet\textbf{info,init,cov}] \\ \mathit{result} = [\textbf{phys}\bullet\textbf{info,exist,cov}] \end{array}\right] \end{array}\right]$$

The dynamic event structure representation is as follows:

tional artifact, the *source* may still be expressed as an adjunct or a complex NP, as in (72):

(72) Mary painted a portrait of a woman.

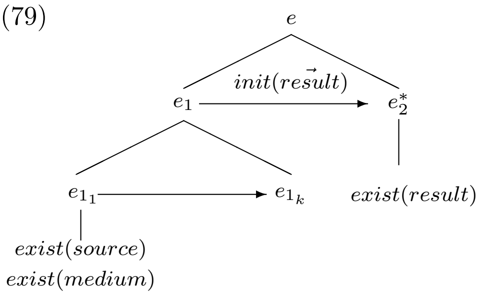
Finally, notice that the verb *paint* also allows reference to the *medium* being used in the activity, regardless of which argument is expressed in direct object position, i.e., the *source* or *result*.

(73) a. John painted the mountain in oil.
b. Mary painted a portrait in oil.

[10] Notice that the *source* may be expressed as an adjunct or a complex NP, as in (74):

(75) Mary filmed a movie of the wedding.

(79)



1.4 Destruction Predicates

In this section, we illustrate briefly how the dynamic argument structure and the associated typing assignment can be employed to differentiate the various destruction predicates. The purpose of this discussion is to demonstrate the application of the typing strategy developed in the previous sections to destruction predicates generally. Space does not permit a more detailed treatment of this class in the present paper.

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 exclusively on what is being done to the pre-existing 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 (the material constitution to what is output by the termination of the input object; this is modeled as a program variable in our framework). In this perspective, destruction predicates are also creation predicates, since the process of transformation can be viewed as creating a result (Jackendoff (1992), 118). 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) , or primarily on its ability to be used for its purpose (i.e. the Telic Quale) while still referring to modification over the formal aspect).

1.4.1 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.

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 (80a), or its component parts can be coerced (80b).

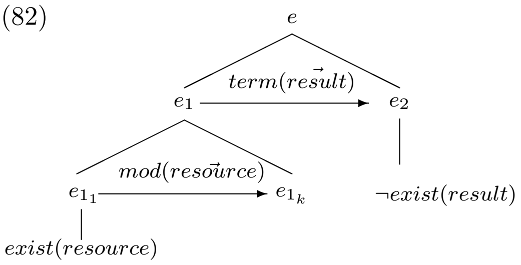
(80) a. The crowd dispersed suddenly.
b. The students dispersed.

The DAS for the sense in (80a) is shown in (81) below:[11]

(81)
$$\left[\begin{array}{ll} \textit{disperse} & \\ \text{DAS} = & \left[\begin{array}{l} \textit{resource} = [\textbf{group,exist,exp}] \\ \overrightarrow{\textit{resource}} = [\textbf{group,mod,cov}] \\ \overrightarrow{\textit{result}} = [\textbf{group,term,cov}] \\ \textit{result} = [\textbf{group, }\neg\textbf{exist,cov}] \end{array}\right] \end{array}\right]$$

The event structure representation is:

(82)



Termination

These verbs 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.

(83) 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.

[11] If we assume that **opt** is the value for syntactic expressibility (see section 1.2.2), giving rise to **cov** or **exp**, then we must mention that the subject in English must be expressed, and this is an independent principle, not expressed in the lexical entry for a verb such as *disperse*. What is expressed here is the underspecified nature of whether the subject is a resource or a result role.

With the predicates in (83), the expression of the resulting material (i.e., the output object) is usually odd (84), and remains hidden (Jackendoff 1990, 118).

(84) *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 DAS representation is as follows:

(85)
$$\begin{bmatrix} \textit{destroy} & \\ \text{DAS} = & \begin{bmatrix} \textit{resource} = [\textbf{phys,exist,exp}] \\ \overrightarrow{\textit{resource}} = [\textbf{phys,mod,cov}] \\ \overrightarrow{\textit{result}} = [\textbf{phys,term,cov}] \\ \textit{result} = [\textbf{phys,}\neg \textit{exist}\textbf{,cov}] \end{bmatrix} \end{bmatrix}$$

The dynamic event structure for *destroy* is the same as that given for *disperse* in (82) above.

As with *build*, verbs such as *destroy* denote a directed process which is measured against a defined *test*, introduced explicitly by the state obtained in the absence of the direct object argument.

Consider now *break* as in:

(86) Mary broke the cup.

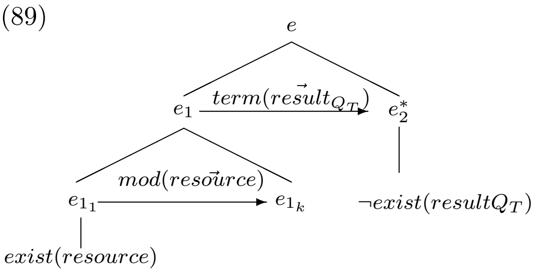
There appear to be two senses of *break*: (a) the physical integrity of the object is changed substantially; and (b) the ability to use the object for its purpose is no longer possible. Either sense can be considered a destruction predicate, but with different consequences. The representation of (a) is given in (87) below, whereas the one in (b) is given in (88), where Q_T stands for Telic Quale:

(87)
$$\begin{bmatrix} \textit{break} & \\ \text{DAS} = & \begin{bmatrix} \textit{resource} = [\textbf{phys,exist,exp}] \\ \overrightarrow{\textit{resource}} = [\textbf{phys,mod,cov}] \\ \overrightarrow{\textit{result}} = [\textbf{phys, term, cov}] \\ \textit{result} = [\textbf{phys,}\neg \textit{exist}\textbf{,cov}] \end{bmatrix} \end{bmatrix}$$

(88)
$$\begin{bmatrix} \textit{break} & \\ \text{DAS} = & \begin{bmatrix} \textit{resource} = [\textbf{phys,exist,exp}] \\ \overrightarrow{\textit{resource}} = [\textbf{phys,mod,cov}] \\ \overrightarrow{\textit{result}} = [\textbf{phys, } \textit{term}(_{Q_T})\textbf{, cov}] \\ \textit{result} = [\textbf{phys,}\neg \textit{exist}(_{Q_T})\textbf{,cov}] \end{bmatrix} \end{bmatrix}$$

The event structure representation of (88) is:

(89)



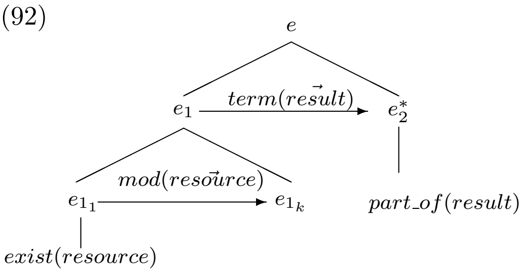
While *destroy* verbs do not allow the expression of the material entity resulting from the change as an optional prepositional phrase (see 84), *break* verbs (Dixon (1991), 119) appear to allow it, as illustrated in (90):

(90) a. Mary broke the table into small fragments.
b. The glass shattered into pieces.

The representation of the DAS and the event structure of the expression in (90a) are (91) and (92):

(91)
$$\left[\begin{array}{l} \textit{break} \\ \text{DAS} = \left[\begin{array}{lcl} \textit{resource} & = & [\textbf{phys,exist,exp}] \\ \vec{\textit{resource}} & = & [\textbf{phys,mod,cov}] \\ \vec{\textit{result}} & = & [\textbf{phys, term, cov}] \\ \textit{result} & = & [\textbf{phys,part\_of(res),exp}] \end{array}\right] \end{array}\right]$$

(92)



1.5 Using Dynamic Argument Structure for Computational Applications

In the discussion above, we have focused on the role that dynamic argument structure plays in mediating the mapping of a predicate's arguments to syntax. There are, however, some interesting consequences of this representation to the computational interpretation of linguistic expressions, in particular the identification of semantic roles associated with verbal predicates. If an argument is encoded as an array of features involving its type, syntactic expressiveness, as well as how it is acted on by the verb, we are distributing a richer set of aspects of the predicative core to each argument. This could have interesting consequences for training semantic role labeling and argument classification algorithms. We consider briefly how the dynamic argument structure (DAS), as developed in the previous sections, could be exploited in natural language engineering tasks. We will focus, in particular, on semantic role labeling (SRL) and event participant identification.

SRL is a computational task that involves identifying a verb's participant roles and labeling them correctly, regardless of their syntactic realization in a sentence. Since there are many different inventories and associated theories of semantic roles, the computational task is impacted by which approach is adopted for labeling. For example, if we adopt a conventional list of "thematic relations" or "case roles", we start with an inventory such as the following: AGENT, EXPERIENCER, THEME, INSTRUMENT, BENEFICIARY, SOURCE, GOAL, LOCATION. Because of long-acknowledged problems and ambiguities in role labels , one alternative strategy would be to adopt a more conservative list, as implemented in PropBank (Palmer et al. (2003)). PropBank has numbered arguments, ARG0, ARG1, ARG2, and non-numbered arguments, such as TMP, LOC, DIR, MNR, RESULT. Another strategy that has been explored is to label an entire linguistic expression as matching a frame, from an inventory of frames, rather than merely labeling a verb and its roles. This is what **FrameNet**-based approaches do (Fillmore and Baker (2001), Baker and Cronin. (2003)).

To illustrate the basic SRL task, in the sentence pairs below, not only is it important to recover the correct labeling of an argument's semantic function relative to the verb, but this must be performed in diverse syntactic contexts.

(93) a. The burglar [AGENT] **broke** the window [THEME].
b. The window [THEME] **broke** suddenly.

(94) a. Mary [AGENT] **gave** a cookie [THEME] to John [GOAL].
b. John [GOAL] was **given** a cookie [THEME] by Mary [AGENT].

Adopting the DAS strategy outlined above affords us the opportunity to encode dynamic aspects of how the argument changes over the course of the event within the context of a semantic role labeling exercise. For example, consider the aggregation-class verb, *assemble*, discussed in Section 1.3 above. From the DAS for this verb, we could (simplifying a bit), explicitly encode the argument as undergoing a specific change, directly in the SRL label, as illustrated in (95) below.

(95) A crowd [RESOURCE: ¬**exist** ↦ RESULT: **exist**] **assembled** in the lobby.

Similarly, for the sentence above in (93a), the change of state of the THEME is encoded through a dynamic argument structure value:

(96) a. The burglar [AGENT] **broke** the window [RESOURCE: **exist** ↦ RESULT: ¬**exist**].

Such a direct encoding of the dynamics of the event would also allow for a richer subevent interpretation from lexical resources, such as VerbNet (Pustejovsky et al. (2016)), a topic we are presently exploring.

1.6 Conclusion

Our purpose in this paper was to provide an enhanced linguistic representation of event structures, by adding information related to the changes that the participants undergo during the event. To this end, we have presented a classification of verbs of change and modification, that focuses on a dynamic interpretation of the change induced by the predicate on its arguments. We have looked at the major classes of creation predicates, as well as some major destruction classes, in terms of a *dynamic argument structure*, i.e. a change profile of the arguments, which tracks how the changed objects behave dynamically throughout the event. We have mapped this representation onto the event structure representation introduced in Pustejovsky and Moszkowicz (2011), which includes a scalar component conceived in terms of assignment and testing programs. This has allowed us to add that different kinds or scales can be referenced in a compositional process, and that the compositional shifts in the scale of interpretation can be analyzed as *scale-shiftings*. Finally, we have shown that the proposed representation has some potentially interesting consequences for the computational interpretation of linguistic expressions, and in particular we provided an example of an application focusing on how semantic role labeling could exploit richer argument structure information.

Acknowledgments

We would like to thank the participants of the "Scalarity in Verb-Based Constructions" Workshop, held at Heinrich-Heine-Universität Düsseldorf, April 7-8, 2011, where a preliminary version of this work was presented.

References

Asher, Nicholas and J. Pustejovsky. 2006. A type composition logic for generative lexicon. *Journal of Cognitive Science* 6:1–38.

Atkins, Beryl T, Judy Kegl, and Beth Levin. 1988. Anatomy of a verb entry: From linguistic theory to lexicographic practise. *International Journal of Lexicography* 1(2):84–126.

Baker, Charles J. Fillmore, Collin F. and Beau Cronin. 2003. The structure of the Framenet database. *International Journal of Lexicography* 16(3):281–296.

Beavers, John. 2008. Scalar complexity and the structure of events. *Event structures in linguistic form and interpretation* pages 245–265.

Bisetto, Antonietta and Chiara Melloni. 2007. Result nominals: a lexical-semantic investigation. In *Proceedings of the Fifth Mediterranean Morphology Meeting*.

Dixon, Robert M.W. 1991. *A new Approach to English Grammar, on Semantic Principles*. Oxford University Press.

Dowty, David R. 1979. *Word meaning and Montague grammar: The semantics of verbs and times in generative semantics and in Montague's PTQ*, vol. 7. Springer.

Dowty, David R. 1991. Thematic proto-roles and argument selection. *language* pages 547–619.

Fillmore, Charles J. and Collin F. Baker. 2001. Frame semantics for text understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*. Pittsburg, PA.

Harel, David, Dexter Kozen, and Jerzy Tiuyn. 2000. *Dynamic Logic*. The MIT Press, 1st edn.

Hay, Jennifer, Christopher Kennedy, and Beth Levin. 1999. Scalar structure underlies telicity in" degree achievements". In *Proceedings of SALT*, vol. 9, pages 127–144.

Jackendoff, Ray. 1992. *Semantic structures*, vol. 18. MIT press.

Jezek, Elisabetta. 2013. Classes of creation verbs. In R. Simone and F. Masini, eds., *Word Classes: Nature, Typology and Computational Representations*. Benjamins.

Kennedy, Chris. 2009. Aspectual composition and scalar change. In *Workshop on Scalarity and Event Structure*.

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

Levin, Beth. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

Levin, Beth and Malka Rappaport Hovav. 2010. *Proceddings of the 46th Annual Meeting of the CLS*.

McKeon, Richard. 1968. *The Basic Works of Aristotle*. Random House, New York.

Palmer, Martha, Dan Gildea, and Paul Kingsbury. 2003. The proposition bank: an annotated corpus of semantic roles. *Computational Linguistics* 31(1):71–106.

Piñón, Christopher. 2008. Verbs of creation. In T. H.-Z. Johannes Dölling and M. Schäfer, eds., *Event Structures in Linguistic Form and Interpretation*, pages 493–521. Berlin - New York: De Gruyter.

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

Pustejovsky, James. 2000. Events and the semantics of opposition. In C. Tenny and J. Pustejovsky, eds., *Events as Grammatical Objects*, pages 445–482. Stanford, CA: Center for the Study of Language and Information (CSLI).

Pustejovsky, James. 2013. Dymanic event structure and habitat theory. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon*.

Pustejovsky, James and Elisabetta Jezek. 2011. Scale-shifting and compositionality. In *Proceedings of the Workshop on Scalarity in Verb-Based Constructions*.

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

Pustejovsky, James, Martha Palmer, Annie Zaenen, and Susan Brown. 2016. Integrating verbnet and gl predicative structures. In *Proceedings of the 12th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*.

Rappaport Hovav, Malka. 2008. Lexicalized meaning and the internal temporal structure of events. In S. Rothstein, ed., *Crosslinguistic and Theoretical Approaches to the Semantics of Aspect*, pages 13–42. Benjamins.

Suppes, Krantz D. M. Luce R. D., P. and A. Tversky. 1989. *Foundations of Measurement (Vol. II: Geometrical threshold, and probabilistic representations*. Academic Press.

Tenny, Carol L. 1994. *Aspectual roles and the syntax-semantics interface*. Springer.

von Stechow, Arnim. 2001. *Temporally opaque arguments in verbs of creation*, pages 278–319. CLSI Publications.