INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • (510) 666-2900 • FAX (510) 666-2956

Image Schemas and Force-Dynamics in FrameNet*

Klinton Bicknell† and Ellen Dodge‡

TR-04-00X

December 2004

Abstract

This report presents the early results of the Frame+Schema project, a project to represent image and force-dynamic schemas in FrameNet. These structures are eventually intended to be used in leveraging the wealth of frame semantic knowledge available in the FrameNet database for situation-specific reasoning. We detail our image and force-dynamic schemas in the Embodied Construction Grammar formalism and give correspondence rules to represent these schemas in FrameNet. We also analyze much of the current force-dynamic structure in the FrameNet database and suggest changes to make it more compatible with image and force-dynamic schema-based inference. We note many of the issues raised by our representation as well as the unsolved problems and also make proposals for future work in the Frame+Schema project.

*This would be far worse if not for the many suggestions of the others at the Frame+Schema meetings: Jerry Feldman, Srini Narayanan, Michael Ellsworth, Collin Baker, and Josef Ruppenhofer. Johno Bryant was also more than helpful and generous with his time.

†klinton@icsi.berkeley.edu

‡edodge@socrates.berkeley.edu

0 Introduction

Schemas have been hypothesized to be central to the human capacity to reason (Johnson 1987; Lakoff & Johnson 1999), with image schemas and force-dynamic schemas (Talmy 1988) playing central roles. The goal of making inferences from language in a human-like way, then, would require at the least a force-dynamic/image schematic analysis. (Hereafter, image/force-dynamic schemas will be collectively referred to simply as schemas.) To allow for situation-specific inferences, these schemas must interface with real-world knowledge about the scenario under discussion. Frame semantics characterizes this real-world knowledge in units called *frames*, which are representations of *gestalt* scenarios defined by a set of entities and states of affairs which participate in the scenario, and relations between these frames (Fillmore 1975, 1985). The FrameNet project at the International Computer Science Institute has built a large database of such frame semantic knowledge and connects it to corpus text by a process of annotation (Fillmore *et al.* 2003; Baker *et al.* 2003). Because schemas can also be formally represented as a list of participating elements and relations to other schemas, it seems natural that the FrameNet project can be extended to represent both kinds of structures required for reasoning, the frames and the schemas, as well as the interactions between these two structures. As a first step toward using the extensive frame semantic knowledge available in the FrameNet database for reasoning, this paper proposes a means of representing image/force-dynamic schemas in the FrameNet database and a method of relating these structures to each other and to the frames already present in the database.

This basic proposal can be broken into two main pieces: (1) defining the relevant schemas and (2) representing these schemas in the FrameNet database. A third piece is (3) investigating how FrameNet currently represents force-dynamic and image schematic ideas such as causation, and changing these representations to use the newly represented schemas. For the first two pieces, we define our schemas in the Embodied Construction Grammar (ECG) formalism and give mapping rules to represent the ECG schemas in FrameNet. We use the ECG formalism to represent our schemas independently of the details of their FrameNet representation because a sizable body of literature exists showing that image schemas can be effectively represented in the formalism[1] (Bergen & Chang In Press; Chang *et al.* 2002; Feldman 2002b). This literature provided us with some initial ideas about how to represent schemas in terms of roles and relations, since there is no complete ontology of image schemas and force-dynamic schemas available for use.

This paper treats all three pieces of the task: Section 1 presents a short overview of the basic method of representation, i.e., of mapping from ECG schemas to FrameNet frames[2] and Section 2 (providing the bulk of the paper) details our schema ontology in ECG and highlights anything controversial about a schema definition, anything not straightforward about its FrameNet representation, or any relevant relation between that schema and the rest of the FrameNet database.

[1] In fact, the ECG literature works on the hypothesis that image schemas, force-dynamic schemas, action schemas, and frames can all be represented in a similar way, using roles and relations, calling them all just "schemas." It thus works very nicely with our current project of representing schemas as frames.

[2] This assumes basic familiarity with the ECG formalism, which may be gained from Bergen & Chang (In Press) or Chang *et al.* (2002).

1 The Basic Representation Strategy

Most of the mapping from ECG to FrameNet representation is straightforward. A schema is represented as a class of frame[3]. Roles in ECG map to frame elements in FrameNet. The **subcase of** operator corresponds to an Inheritance relation. Type-constraints on roles are translated into semantic types on frame elements if possible, and otherwise are just mentioned in the text of the frame element's description. ECG identity statements (i.e., unification) correspond to pairing frame elements in frame-to-frame relations (more on this below). All this seems to be quite straightforward, since structures are available in the FrameNet database basically equivalent to these ECG structures.

1.1 Representing evokes

The **evokes** operator in ECG should clearly specify some sort of frame-to-frame relation in FrameNet. In order to explain why we chose the "Chatty-Using" relation as its correspondent, it seems helpful to describe the other options.

FrameNet has four types of frame-to-frame relations: Inheritance, Subframing, "Chatty-Using," and "POV-Using." Inheritance is exactly what one would expect, a requirement that every element in the parent frame have a counterpart in the child frame. The Subframe relation is used for the event structure of frames within a larger scene. For instance, the Arresting frame would be a subframe of the larger Criminal_process frame, as would the Arraignment frame and the Trial frame. There is a only a single formal relation called *Using* in the FrameNet database, although it is used to represent two distinct relations, what I have called "POV-Uses" and "Chatty-Uses." "POV-Uses" is a quite rare relation, and is the relation between a point-of-view-independent representation of a scenario and the point-of-view-specified versions of it. So, for example, the Commerce_buy and Commerce_sell frames both have "POV-Uses" relations to Commerce_goods-transfer, since buying and selling describe the same event from two different perspectives. "Chatty-Uses" is the relation between a frame and a background frame needed to make sense of it. This is, for example, the relation between the Volubility frame (which is referred to by words such as *chatty*, *loudmouth*, and *mum*) and the Communication frame. The Volubility frame only represents a valid concept given the Communication frame.

	DEFINES PARENT	DEFINES CHILD ONLY
FORCED INHERITANCE	"POV-Uses"	Inheritance
FREE INHERITANCE	Subframing	"Chatty-Uses"

Table 1: The four frame-to-frame relations in FrameNet, separated by binary features

These four relations can be thought of in a matrix of binary features (Table 1). "POV-Uses" and Inheritance both require that every element in the parent be mapped to (i.e., unified with) an element of the child, while the Subframe relation and "Chatty-Uses" do not have such a requirement. "POV-Uses" and the Subframe relation also express relations in which the child helps to define the parent. For example, the Criminal_Process frame is defined mostly by the definitions of its constituent parts (such as Arresting or Arraignment). Similarly, the Commerce_goods-transfer frame is defined by being what is left after the point-of-view is removed from the Commerce_buy and Commerce_sell frames.

[3]The way this class is marked in the FrameNet database has not yet been specified.

For Inheritance and "Chatty-Uses" relations, this is not the case. Communication is defined just fine without reference to Volubility, and obviously, an inheritance relation does not add anything to the superclass.

Given these two formal features of the four relations, the final relation, "Chatty-Uses," seems to correspond most closely to the ECG **evokes** relation, since it is certainly not definitional for the parent, and there is no requirement that the child have the same elements as the parent. By representing **evokes** with "Chatty-Uses," we are extending the meaning of "Chatty-Uses." For our purposes, it will often not be a scenario in which a larger *context* frame must be present for the structure to be made sense of (as with the Volubility frame and Communication), but will often be a scenario in which a smaller frame is used as part of the frame's structure (as the Jailing frame might use the Container schema). Both of these types of relations (and a couple of others) can be represented by **evokes** in ECG. In practice, representing ECG **evokes** with the "Chatty-Uses" relation works quite well, with only one potential issue (mentioned below).

1.2 "Dependent Frames"

As mentioned above, one important part of the effort to incorporate schemas into the FrameNet database is to relate them to the preexisting frames. An Inheritance relation intuitively seems to be the right kind of link between a schema and frame, but it is not immediately obvious which direction this link should go. Intuition suggests that the schemas, being more 'basic,' should be the parents of the frames, which are more complex. This is not a viable solution, however, because a frame like Locative_relation is much broader in one regard than a schema such as In, since it covers many sorts of locative relations. The reverse idea, that schemas should inherit from frames does not work because schemas are common to 'metaphoric' and 'non-metaphoric' scenarios, while frames are more specified and are domain-specific. For example, both "I am in Austin" and "I am in mourning" would use the same In schema, while only the first case would use the Locative_relation frame.

Our solution to this problem is to add another mediating structure to the hierarchy called a "dependent frame." Every dependent frame inherits from both one schema and one frame; thus they the very bottom of the hierarchy, and all FrameNet annotation should be done relative to them. For example, one dependent frame might inherit from the In schema and the LocativeRelation frame; this dependent frame would be used in phrases such as "the ball in the box." Another dependent frame might inherit from In schema as well as the TakingTime frame; this dependent frame would be used in cases such as "Andy finished in ten minutes." Dependent frames, then, are domain-specific pairings of schematic structure with frames. They inherit their domain-specificity from the frame, which they unify with the schematic structure inherited from the schema. This provides a nice solution to the problem of the relation between frames and schemas because it gets around the problems with inheritance, and it seems to be an effective way of modeling this relation.

Despite their effectiveness, it is unclear that dependent frames are the ideal way to represent the interaction of frames and schemas, since they imply no intrinsic relation between frames and schemas. Dependent frames are thus unconstrained and theoretically could pair any schema with any frame. They would also not explain why some frames are always paired with certain image schemas. Despite these issues, however, dependent frames proved to be an effective way of representing schemas in FrameNet for the purposes of reasoning.

```
schema BoundedEntity
  subcase of Entity
  evokes
    ClosedBoundary as c
  roles
    interior ⟷ c.side1
    exterior ⟷ c.side2
    boundary ⟷ c.self
```

```
schema Container
  subcase of BoundedEntity
  roles
    portal
```

```
schema AxialEntity
  subcase of Entity
  roles
    PrimaryAxis
```

Figure 1: The Entity schemas

```
schema SPG
  roles
    source : Place
    path : Directed-Curve
    goal : Place
    trajector : Entity
```

Figure 2: The Source-Path-Goal (SPG) schema

2 The Ontology

Having established the mode of mapping from ECG to FrameNet, we present our ontology of schemas in ECG. As the ontology is detailed, any points of special interest, whether in the schema definitions, their representation, or relations to the rest of the database, will be highlighted.

2.1 Basic Entity and Source-Path-Goal schemas

2.1.1 Entity schemas

One of the most basic of our image schemas is that of an entity, something perceived as a unified whole. (schemas detailed in Figure 1). Entities are often seen as bounded (using an evokes relation to the ClosedBoundary schema not shown). Bounded entities with force-dynamic properties are referred to as Containers, and contain a means of entry and exit. The last type of entity here is one viewed as axial, meaning that it has a primary axis. Many of these entity schemas are evoked by trajector-landmark relation schemas.

2.1.2 The Source-Path-Goal schema

The Source-Path-Goal (SPG) schema describes a trajectory with a source, path, and goal, and a trajector which is (presumably) following this trajectory (schema detailed in Figure 2). One idea, which will be developed further in the trajector-landmark relations, is that the trajector may be able to be merely one's (moving) line of sight. This might account nicely for instances of this schema such as "I looked along the edge of the desk for my contact lens." Perhaps, then, a Vision-SPG schema is a **subcase of** the SPG schema.

2.2 Trajector-Landmark Relations

2.2.1 The Trajector-Landmark schema

Langacker (1982) first coined the distinction between *trajector* and *landmark* in this context, extending the earlier use of *landmark* by Miller & Johnson-Laird (1976). For Langacker (1982), the trajector/landmark distinction is just an application of more general figure/ground organization to any linguistic relation between two entities. Our use of the terms (schema detailed in Figure 3) is meant to express this figure/ground structuring as well as Langacker's (1991, 1993) target/reference-point distinction, equating the trajector with the target and the

```
schema Trajector-Landmark
  roles
    trajector : Entity
    landmark
    trajectorLocation
```

Figure 3: The Trajector-Landmark schema

landmark with the reference-point. Because these two categories always coincide only in prepositions in English, we use them solely for schemas represented by prepositions, giving information about one entity (the trajector) in terms of another (the landmark). Thus, e.g., the ContainerRelation schema given below, a **subcase of** the Trajector-Landmark schema, would not be used for an idea expressed without a preposition, such as "This office contains the desk," which not only has a different element of the containment relationship in focus, but also has more of a focus on force-dynamics. We represent the trajectorLocation as a separate role instead of merely a property of the trajector role so that it could unify in the In and Into constructions below, although this solution is not completely satisfying.

2.2.2 The Proximal/Distal schemas

These schemas represent concepts implicit in the English terms *near to* and *far (away) from*. The highest-level schema divides the space around a landmark into (at least) two zones, and the more specified versions unify the trajector's location to one of those zones (schemas detailed in Figure 4). Although not currently represented, these shemas should perhaps make reference to the Scale schemas and a Center-Periphery schema (not discussed here).

```
schema ProximalDistal
  subcase of Trajector-Landmark
  roles
    origo
    proximalSpace : BoundedEntity
    distalSpace : BoundedEntity
```

```
schema Proximal
  subcase of Proximal-Distal
  constraints
    trajectorLocation ⟷ proximalSpace
```

```
schema Distal
  subcase of Proximal-Distal
  constraints
    trajectorLocation ⟷ distalSpace
```

Figure 4: The Proximal/Distal schemas

2.2.3 The Axial Relation schemas

These schemas represent the concepts expressed by English *across* and *along* (schemas detailed in Figures 5–6). The similarity between the Across and Along schemas—that they each require that the landmark be axial and that an axis related to the trajector have some relation to the landmark's primary axis—is captured by the highest level schema. Complicating this set of schemas, both *across* and *along* can be used both in a 'static' way, in which the trajector itself is axial, and in a 'dynamic' way, in which the trajector's path serves as its axis. To capture the generalities between *across* and *along*, one could define the static idea as central and specify that the dynamic idea was an instance of the static one (cf. Talmy (2000:189ff)), perhaps through metaphor or some type-coercion process. However, one could just as easily define the dynamic idea as central and specify that the static was an instance of the dynamic by similar means, as others have done. Because there is no clear benefit or motivation to either approach, the representation presented here captures the com-

```
schema AxialRelation
  subcase of TrLm
  evokes
    Axis as a1
    Axis as a2
  constraints
    lm : AxialEntity
    lm.PrimaryAxis ⟷ a2
```

```
schema AxialRelationStatic
  subcase of AxialRelation
  constraints
    tr : AxialEntity
    tr.PrimaryAxis ⟷ a1
```

```
schema AxialRelationSPG
  subcase of AxialRelation
  evokes
    SPG as s
  constraints
    tr ⟷ s.tr
    s.path ⟷ a1
```

```
schema Across
  subcase of AxialRelation
  constraints
    Crosses(a1,a2)
```

```
schema Along
  subcase of AxialRelation
  constraints
    Parallel(a1, a2)
```

Figure 5: The higher-level Axial Relations schemas

```
schema AcrossStatic
  subcase of Across2, AxialRelationStatic
```

```
schema AcrossSPG
  subcase of Across, AxialRelationSPG
```

```
schema AlongStatic
  subcase of Along, AxialRelationStatic
```

```
schema AlongSPG
  subcase of Along, AxialRelationSPG
```

Figure 6: The lower-level Axial Relations schemas

monalities with multiple inheritance, such that each of the four bottom-level schemas inherits from either Across or Along and from either the static or dynamic axial relation. This multiple inheritance gives no special status to either of the two meanings, and seems to be a good way of computationally modeling the cognitive phenomenon of radial category structure (Lakoff 1987) in general, having each member of a radial category inherit some of a group of qualities that, in some combinations, define the category. In addition to this, there is neural motivation for giving the static and the dynamic versions independent statuses, because the brain has different structures in different areas for processing motion and processing object shape. The last note here is that for the dynamic axial relation schema AxialRelationSPG, it intuitively seems that s.source and s.goal should be defined somewhat. The current representation ignores that fact because it is unclear exactly how to formalize it, and it is unclear that it would really be a useful part of the schema for reasoning.

2.2.4 The Container Relation schemas

These schemas represent imposing a trajector-landmark scheme onto a container/contained relation[4] (schemas detailed in Figure 7). For these relations, the container is always equated to the landmark (so these would not be used for, e.g., “the moat around the castle”).

This set of schemas highlights two important points about representation. Both points concern certain features added to these schemas to make them suitable for a FrameNet representation. The first is the addition of the

[4]There is some question whether relations as specific as In and Into should properly be called image schemas, or if they are instead just simple lexical constructions. In the ECG literature, there is no consensus on which way they should be represented. FrameNet, however, has no functionality to represent constructions, so we adopt the image schema approach here.

```
schema ContainerRelation
  subcase of tr-lm
  evokes
    Container as c
  constraints
    landmark ⟷ c.self
```

```
schema In
  subcase of ContainerRelation
  constraints
    trajectorLocation ⟷ c.interior
```

```
schema Into
  subcase of ContainerRelation
  evokes
    SPG as s
  roles
    Source-Exterior ⟷ c.exterior
    Goal-Interior ⟷ c.interior
  constraints
    trajector ⟷ s.trajector
    Source-Exterior ⟷ s.source
    Goal-Interior ⟷ s.goal
```

Figure 7: The Container Relation schemas

self role[5] to the Container schema (given in Figure 1 above), and the reference to it in the ContainerRelation schema. In standard ECG, there would be no **evokes** block in the ContainerRelation schema, and the landmark would simply be constrained to be of type Container (instead of being unified with the self role of an evoked Container schema). This must be done because FrameNet does not link semantic types (used in type constraints) to actual frames; thus, even if the landmark was constrained to be of type Container in FrameNet, there would be no way to access its frame elements as a Container (as needs to be done for the In and Into schemas). Instead, we just **evoke** that frame and then bind its self role to the landmark. The X.self role, then, just refers to the frame X itself. This is just a current work-around, but it seems to do its job nicely. An alternate (and largely equivalent) method of getting around this limitation is to define these frames, which need to be unified with an element of another frame, more broadly; for example, the Container schema could be recast as a Containment schema, having Container as one of its roles, in addition to the other roles already there. The disadvantage of that method of representation is that the schema now has internal part-whole relations, since, e.g., both Container and Interior (of the container) are independent elements. This solution seems less elegant and certainly less intuitive. Our current representation is also not perfect, however, and some subtle computational issues have been raised regarding it. The problem is that ECG is not redundant, and there really is a distinction between constraining a role to be of a certain semantic type, and unifying a role with an evoked structure. Though this problem's existence should be noted, our current thoughts are that collapsing this distinction should not cause problems for reasoning with the FrameNet database.

The Into schema raises the second point about representation. In the standard ECG representation, the **roles** block would not exist, and the **constraints** would be to unify structures from the two evoked structures (i.e., c.exterior with s.source and c.interior with s.goal). Because the "Chatty-Uses" relation in FrameNet can only link frame elements from the Used frame to frame elements in the Using frame, this unification has to be accomplished by creating new frame elements in the Using frame and linking those elements to elements from both of the Used frames. The elements in the Used frames would thus be linked indirectly through the elements of the Using frame. Again, there is a subtlety that this representation is ignoring, since these elements really aren't elements of the Using frame, but unless FrameNet changes the way its Using relation works, this representation seems to be

[5] this should not be confused with the **self** keyword in standard ECG. In fact, because of this similarity, it may be wise to change the name of this role.

```
schema SPG-Relation
  subcase of tr-lm
  evokes
    SPG as s
  constraints
    trajector ⟷ s.trajector
```

```
schema From
  subcase of SPG-Relation
  constraints
    landmark ⟷ s.source
```

```
schema To
  subcase of SPG-Relation
  constraints
    landmark ⟷ s.goal
```

Figure 8: The Source-Path-Goal Relation schemas

both necessary and perfectly usable.

2.2.5 The Source-Path-Goal Relation schemas

The source-path-goal relation schemas represent imposing trajector-landmark structuring onto a source-path-goal schema (schemas detailed in Figure 8). As before, the commonalities in the lexical-level schemas (From and To) are captured in a higher-level schema.

2.3 The Linear Scale schemas

The linear scale schemas represent another basic part of our set of schemas (schemas detailed in Figure 9). They are required for a multitude of reasoning processes and show up linguistically in items as diverse as prepositions, nouns, and verbs. Our representation is just a formalization of the ideas articulated in Feldman 2002a. Open issues with this representation include whether inheritance is the correct type of link between LinearScale and PositionOnAScale. At an intuitive level, it is unclear

```
schema LinearScale
  roles
    bottom
    top
    category
    property
```

```
schema PositionOnAScale
  subcase of LinearScale
  roles
    center
    spread
```

Figure 9: The Linear Scale schemas

```
schema WeightScale
  subcase of LinearScale
  evokes
    Weight as w
  constraints
    property ⟷ w
```

Figure 10: The WeightScale schema

that a position on a scale is really a subtype of scale, especially since every **subcase of** LinearScale would presumably be associated with its own **subcase of** the PositionOnAScale schema. For lack of a more effective alternative, however, our representation tentatively adopts the inheritance relation.

The larger issue introduced by these scale schemas stems from the fact that they are—more than any of the rest of our schemas—*parameterized*. That is, any of their roles can be further specified to create a more specific schema/frame. (This continuum of specification is also a key area where the schema/frame distinction blurs.) For example, the WeightScale schema (Figure 10) further specifies the property role. The WeightScale schema could be further specified by providing more of the parameters, as in CowWeight (Figure 11), which specifies the domain of weight being referred to and

```
schema CowWeight
  subcase of WeightScale
  constraints
    category ⟵ cow
    bottom ⟵ 100 (lbs)
    top ⟵ 2000 (lbs)
```

Figure 11: The CowWeight schema

```
schema Heavy
  subcase of WeightScale, PositionOnAScale
  constraints
    center ⟵ .6
    spread ⟵ .3
```

Figure 12: The Heavy schema

```
schema AquityScale
  subcase of LinearScale
  constraints
    property ⟵ Aquity
```

```
schema PositionOnAquityScale
  subcase of AquityScale, PositionOnAScale
```

Figure 13: The AquityScale and PositionOnAquityScale schemas

```
schema BeingWetBeingDryFrame
  subcase of PositionOnAquityScale
  roles
    item
    liquid
    cause
    sub-location
```

Figure 14: The BeingWet/BeingDry frame

gives end-points to the scale. Alternatively, the WeightScale could be further specified another way by providing a Center and Spread, approximating a concept such as Heavy (Figure 12). A HeavyCow schema (not given) would merely inherit from both these schemas (CowWeight and Heavy). Certainly, this sort of gradual specification and nearly boundless combinatory possibility is not seen in the rest of the FrameNet database, and the FrameNet database is really not properly equipped to handle it fully. There is currently, for example, no way to say that a frame itself actually specifies one of its inherited frame elements (as WeightScale, etc. does)[6].

It also remains an open question how many of these combinations would actually exist in the FrameNet database, and how many would merely be specified through annotation.

The scale schemas also suggest another kind of relation between frames and schemas. Instead of creating dependent frames, which inherit both from one frame and one schema, the relation suggested here is that a frame would itself inherit from a schema. For example, given an AquityScale (where 'aquity' is intended to be a cover term for wetness and dryness), and a PositionOnAquityScale schema (given in Figure 13), a frame for BeingWet/BeingDry (Figure 14) could just inherit from the PositionOnAquityScale schema. The frame would also have extra roles for other parts of the *gestalt* scenario, such as the item that is wet/dry or the cause of the wetness/dryness, but could make use of the center and spread roles to specify the degree of wetness/dryness (i.e., the position

[6] The closest approximation to this functionality is the ability to *incorporate* a frame element into a lexical item (below the frame level). Of course, a frame could be made to effectively incorporate a frame element by giving that property to every lexical item within it. This solution is not very elegant. Further, the incorporation ability does not provide any semantic information about what is incorporated, but only notes that this information is provided by the lexical item. For example, a WeightScale and HeightScale would be identical from the database's standpoint, since both would merely incorporate the property frame element. This behavior would also need to be changed or worked around somehow to perform reasoning.

on the scale). This seems like a tenable relation between frame and schema, for this domain at least, but the details of how this interaction would work exactly are as yet unresolved.

2.4 Force Dynamic Schemas and Mental Spaces

Force-dynamic schemas include such notions as causation, prevention, and enabling[7]. Because these notions already exist in the FrameNet database to some extent, our proposal for the representation of force-dynamic schemas includes a critical examination of the usefulness of their current status in FrameNet from the perspective of reasoning in addition to the representation of the force-dynamic schemas as frames.

2.4.1 The current state of force-dynamics in FrameNet

Most of the force-dynamic-related structure currently in the database is encoded by two frame-to-frame relations, Inchoative_of and Causative_of. Both relations are ultimately shorthand for a relation that could be formally specified by Inheritance and Subframe relations (see Section 1.1 for more information on these relations). The exact relations that Inchoative_of and Causative_of encode is expressed pictorially in Figure 15.

As seen from the figure, these two relations encode what would otherwise have to be specified by creating an extra frame and quite a number of frame-to-frame relations. They imply some *Scenario* in which a *Previous State* becomes a *Post State*. One or two events may occur which have a hand in bringing this change about. The subframes of the Scenario are ordered in the FrameNet database, as shown by the 'O' arrows connecting them in the diagram. In the two-event version (pictured), a *Causative Event* may happen, which is Causative_of an *Inchoative Event*, which is itself Inchoative_of the *Post State*. For example, take the triplet of frames Cause_temperature_change, Inchoative_change_of_temperature, and Temperature. The scenes encoded by these frames are demonstrated in (1) to (3).

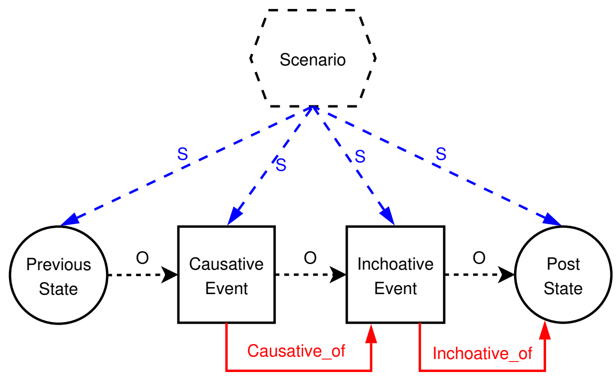

Figure 15: The meaning of the Inchoative_of and Causative_of frame-to-frame relations in FrameNet. Dashed lines indicate implied structure not actually present in the database. 'S' lines indicate Subframe relations; 'O' lines show Subframe ordering.

(1) The sailor *heated* the iron. (Cause_temperature_change)

(2) The iron *heated up* nicely in the fire. (Inchoative_change_of_temperature)

(3) The iron is now *hot*. (Temperature)

The Cause_temperature_change frame, which concerns a cause of the iron's becoming

[7] Often, notions such as *confining* or *jailing* are also said to be force-dynamic concepts. Here, we assume that the force-dynamics of these scenarios is based not on anything inherent to the verbs themselves, but on real-world knowledge-based inference, so these concepts will not be discussed.

hot, would be Causative_of the Inchoative_change_of_temperature frame. The Inchoative_change_of_temperature frame, which really just concerns the inception of the state in which the iron is hot, would itself be Inchoative_of the Temperature frame. The Temperature frame would of course represent the Post State, and, it should be noted, would also represent the Previous State of not being hot, a fact discussed further below.

In practice, there are many times when either the Causative Event or the Inchoative Event is not conventionally conceptualized, and thus that frame is not in the FrameNet database. Each of these cases would change the diagram slightly: if the Causative Event is not present, its frame and the Causative_of link would also not be present, and if the Inchoative Event is not present, its frame and the Inchoative_of link would not be present *and* the Causative_of link would go from the Causative Event to the Post State. Thus, the Causative_of link can have two meanings; it can connect a Causative Event to an Inchoative or can connect a Causative Event to a Post State if no Inchoative exists.

Although the Inchoative_of and Causative_of relations are nicely defined, they are actually only represented in about 20 frames in the FrameNet database[8] (given in Table 2). As seen in the table, 3 of the 9 scenario sets do not have a Stative frame linked. In each case, this is just because that frame does not (yet) exist in the database. In 2 of the sets, it is unclear whether the non-Causative frame is Inchoative or Stative, and these cases seem to provide motivation for the Causative_of relation remaining ambiguous. The other noteworthy item from this table is that the naming schemes are relatively inconsistent, as only 6 of 9 of the Causatives are named as such and a mere 2 of the Inchoatives are named Inchoative. This is apparently a reflection of natural language, which assigns centrality to different members of the set depending on the scenario; however, when using the FrameNet database for reasoning purposes, it may become necessary at some point to impose a stricter naming scheme.

In addition to these two frame-to-frame relations, the FrameNet database also has frames pertaining to force-dynamics, e.g., the Causation and Causation_scenario frames; these frames are an independent way to represent these ideas and are in no way connected to the Causative_of or Inchoative_of relations. Some force-dynamic structure is also just specified in the (text) descriptions of some frames. None of this force-dynamic structure, whether frames, frame-to-frame relations, or frame descriptions, relates concepts such as causation and prevention, which are intrinsically and closely related. An optimal representation would reflect the fact that individual force-dynamic schemas, e.g., causation and prevention, are closely related to each other.

2.4.2 Proposals to make FrameNet's force-dynamic structure more suitable for reasoning

As mentioned above, the force-dynamic structures implicit in the Causative_of and Inchoative_of relations are unsuitable for reasoning in their current form. The main reason is that the Previous State and the Post State (in Figure 15) are indistinguishable instances of the same frame. There is thus no force-dynamic difference between, e.g., the Apply_heat frame and the Apply_Cold[9] frame, since each would just be the Causative frame in a scenario set with the Stative Heat.

The immediately intuitive solution would be to split the Stative Heat frame, perhaps into a Heat frame and a Coldness frame; then the Apply_heat would only be a Causative in the

[8] Thanks to Carol Hays for getting this list of frames from the database and Branimir Ćirić for modifying the inheritanceGraph program to visualize these relations.

[9] Again, this frame is an explanatory fiction

CAUSATIVE	INCHOATIVE	STATIVE
Attaching	Inchoative attaching	Being attached
Apply heat	Absorb heat	
Cause change	Undergo change	
Cause change of consistency	Change of consistency	
Cause change of scalar position	Change position on a scale	Position on a scale
Cause temperature change	Inchoative change of temperature	Temperature
Cause to move in place	Moving in place (*arguable*)	
Cause to start	Process start	Process continue
Emitting		Emanating (*arguable*)

Table 2: The frames utilizing the Inchoative_of and Causative_of relations in the FrameNet database

scenario where Heat is the Stative and Apply_cold would be a Causative in the scenario where Coldness is the Stative.[10] This solution of splitting the frames, however, would completely obscure the fact that these two scenarios were inherently related, thus losing important information for reasoning.

The solution we propose instead is to use our scale schemas (see Section 2.3) to give semantics to the frames in these scenarios. If, as proposed above, the Stative frame just inherited from the relevant PositionOnAScale schema, then the causative/inchoative frames could specify directions of change on that scale. For scenarios that involve binary instead of scalar change (such as Attaching), the semantics would of course not use the scale schemas, but another sort of schema should work to the same effect.

Much more work still needs to be done on this front. All these details still need to be worked out. Further, there are distinctions that even this scalar semantics cannot capture, such as the distinction between reversible and irreversible changes. At this stage, though, it is already apparent that this is a case where the addition of schemas to the database can be directly used to facilitate reasoning.

[10]Strangely, although this is not standard FrameNet practice, this splitting appears to have already been done with the Being_wet and Being_dry frames.

2.4.3 The Compressed Mental Space schemas

Now we finally turn to the force-dynamic schemas we created. Our representation does not use the fairly common style of force-dynamic analyses introduced by Talmy (1988) and extended by e.g., Wolff *et al.* (2002), which analyze force-dynamic concepts in terms of primitives such as the tendency of the agonist and the relative strengths of the agonist versus antagonist. For many reasons not detailed here, we instead opt for a conditional analysis, in which X causes Y if, under the hypothetical condition that X is absent, then Y does not occur. The conditional is represented as a pair of mental spaces, cognitive partition structures introduced by Fauconnier (1985), that are "compressed" into schemas in a manner described in Mok *et al.* (2004). The schemas for these mental spaces (modeled very closely on Mok *et al.*) are given in Figure 16.

The CompressedMentalSpace schema is quite skeletal here. The description of most of its structure is not relevant here (see Mok *et al.* 2004 for explanation), except to note that two of its roles are type-constrained to be (uncompressed) mental spaces. Uncompressed men-

```
schema CompressedMentalSpace
  roles
    ums : MentalSpace
    parentSpace : MentalSpace
    status
```

```
schema ConditionalSchema
  subcase of CompressedMentalSpace
  roles
    epistemicStance
    condition : Predication
    premise : Predication
    conclusion : Predication
    ums : ConditionalSpace
```

Figure 16: The Compressed Mental Space schemas

tal spaces are not very schema- or frame-like, and thus cannot be represented in FrameNet, so these roles could only be used if they were somehow linked to something external to the database.

The structure of the ConditionalSchema is worth describing here. Essentially, this is a mental space in which the condition is assumed to be true and it follows that the conclusion is true. The role epistemicStance specifies the likelihood of this scenario. The premise role is useful only for counterfactuals, so can be ignored here, and the ums role can again be ignored, since a ConditionalSpace is just a **subcase of** a MentalSpace, which cannot be represented in FrameNet.

2.4.4 The Force-Dynamic schemas

Only two of the standard force-dynamic schemas are represented here: Causation and Prevention. To allow for the eventual addition of more schemas, however, there are two levels of schema structure higher than these two (given in Figure 17). All force-dynamic schemas involve three elements. The antagonist is an entity, event, or state of affairs that is interpreted

```
schema ForceDynamics
  roles
    agonist
    antagonist
    effect : Predication
  evokes
    ConditionalSchema as actual
    ConditionalSchema as hypothe
```

```
schema CausationPrevention
  subcase of ForceDynamics
  constraints
    hypothe.epistemicStance ⟵ very negative
    hypothe.condition ⟵ absent(antagonist)
    actual.epistemicStance ⟵ positive
    actual.condition ⟵ not(hypothe.condition)
    hypothe.premise ⟷ hypothe.condition
    actual.premise ⟷ actual.condition
```

Figure 17: The high-level force-dynamic schemas

as e.g., causing or preventing the effect. The agonist is an entity, event, or state of affairs that is e.g., caused or prevented from doing the effect or being the effect. The effect is the proposition that was either caused to obtain or prevented from obtaining. All force-dynamic schemas will also require at least a couple of conditional spaces.

On our analysis, Causation and Prevention have nearly all their structure in common. As shown in the CausationPrevention schema, both set up a hypothetical space (which is not really believed) and an actual space (which is believed). The only difference between the two spaces is that in the hypothetical space, the antagonist is absent.

The differences between Causation and Prevention (given in Figure 18) are simply in the spaces' conclusions. For Causation, the effect *does* obtain in the actual space and does *not* in the hypothetical space, when the antagonist is absent. For Prevention, it's just the opposite: the effect does *not* obtain in the actual space and *does*

```
schema Causation
  subcase of CausationPrevention
  constraints
    actual.conclusion ⟵ effect
    hypothe.conclusion ⟵ not(effect)
```

```
schema Prevention
  subcase of CausationPrevention
  constraints
    actual.conclusion ⟵ not(effect)
    hypothe.conclusion ⟵ effect
```

Figure 18: The low-level force-dynamic schemas

in the hypothetical space, when the antagonist is absent.

This analysis seems to do quite an elegant job of capturing causation and prevention. It, of course, does raise some questions, though. For example, it's not entirely clear that absent(antagonist) is the right predication for the hypothetical spaces.

The question of how to relate these force-dynamic schemas to the larger FrameNet force-dynamic structure is also open. The Causation and Causation_scenario frames should be able to at least inherit from the Causation schema, and the Prevention frames should do similarly, etc. The force-dynamic structure represented by the Causative_of and Inchoative_of relations, however, is currently far too simple for a mental space style analysis. For use in reasoning, these relations would probably need to bind their semantics to parts of the force-dynamic schemas. Although this sounds like a tenable solution, the details have not been worked out.

3 Conclusion

As expected, FrameNet can mirror most of the structure present in schemas defined in Embodied Construction Grammar. This paper perhaps posed more "open questions" than it gave details of representation, but has shown that FrameNet promises to be a very useful tool for situation-specific reasoning.

A few of the sticky issues surrounding representation of the ECG schemas in FrameNet could potentially be resolved by changes in the nature of the database. We would like to propose a few here, realizing that any changes will have to be made carefully:

1. The ability to bind a frame to a frame element of another frame would remove the need for the postulation of the self role and help create a more elegant schema representation. This ability has been independently shown to be useful on purely frame-specific grounds.

2. The ability to link semantic typing of frame elements to frames in the database would make for more complete semantics in the FrameNet project and would aid the reasoning effort. Further, if the roles of these semantic types could be linked to roles of the current frame, it would allow for a still-more complete semantic representation of schemas (and possibly frames too) by removing the need to replace many ECG type-constraints with separate **evokes** statements and identity statements for each of the called structures.

3. The ability to specify the semantics of a filled frame element (e.g., one that is inherited but specified by the current frame) is necessary for reasoning. If this functionality could be added to FrameNet, it would add another layer of semantics, assuming this functionality fit in somewhere else for reasoning (perhaps in a parallel database?).

Other than those general FrameNet functionality issues, there still exists a large amount of work before the Image+Schema project can be considered finished.

Regarding the schemas, of course, there are huge classes of schemas required for our reasoning tasks that this paper did not even consider. E.g., orientation, possession, visibility, accessibility, and verticality would all be directions in which the project should proceed. There are also many schemas detailed in this paper that are incomplete, or should be extended, e.g., Container, AxialRelationSPG, etc.

It may also be helpful to more systematically relate frames to schemas. As mentioned above, our general idea for the relation between the two was "Dependent Frames." The scale schemas, however, suggested a different (and arguably more intuitive) relation: inheritance. The problems mitigating against a systematic inheritance link that were mentioned above primarily concerned metaphor, and may be resolved by the emerging MetaNet project.

The force-dynamics side of things also needs much more work. The connections postulated between the FrameNet schemas and mental spaces (which must exist somewhere outside the FrameNet database) need to be worked out for the conditionals to be functional for reasoning. Further, the relation between the force-dynamic schemas and related frames such as Causation, CausationScenario, Thwarting, and Preventing needs to be explored. The connection between scales and the force-dynamic frame-to-frame relations still needs to be precisely specified. The interplay of that story and non-scalar (i.e., binary) causation scenarios also must be resolved. It may be solved by the ability to semantically fill roles if that is ever enabled.

The relation of this work to the FrameNet project's annotations also remains to be worked out. Many of the schemas only have one or two annotatable elements (often just trajector and landmark). Every element of a schema is, however, by definition conceptually present, so it is unclear how to assign coreness statuses.

Many questions still exist for the project, but most of the initial questions have been answered, including the method of representing a wide variety of ECG schemas in FrameNet and a method of linking much of FrameNet's force-dynamic structure to reasoning.

References

BAKER, COLLIN F., CHARLES J. FILLMORE, & BEAU CRONIN. 2003. The structure of the FrameNet database. *International Journal of Lexicography* 16.281–296.

BERGEN, BENJAMIN, & NANCY CHANG. In Press. Embodied construction grammar in simulation-based language understanding. In *Construction Grammar(s): Cognitive and Cross-Language Dimensions*, ed. by Jan-Ola Östman & Mirjam Fried, p. 147ff. John Benjamins.

CHANG, NANCY, JEROME FELDMAN, ROBERT PORZEL, & KEITH SANDERS. 2002. Scaling cognitive linguistics: formalisms for language understanding. In *Proceedings of the 1st International Workshop on Scalable Natural Language Understanding (ScaNaLU)*. Heidelberg.

FAUCONNIER, GILLES. 1985. *Mental spaces: Aspects of meaning construction in natural language*. Cambridge: MIT Press.

FELDMAN, JEROME. 2002a. The meaning of reference in embodied construction grammar. Technical Report TR-02-011, International Computer Science Institute. Also available `ftp://ftp.icsi.berkeley.edu/pub/techreports/2002/tr-02-011.pdf`.

——. 2002b. A proposed formalism for ECG schemas, constructions, mental spaces, and maps. Technical Report TR-02-010, International Computer Science Institute. Also available `ftp://ftp.icsi.berkeley.edu/pub/techreports/2002/tr-02-010.pdf`.

FILLMORE, CHARLES. 1975. An alternative to checklist theories of meaning. In *Proceedings of the First Annual Meeting of the Berkeley Linguistics Society*, 123–131. Berkeley: Berkeley Linguistics Society.

——. 1985. Frames and the semantics of understanding. *Quaderni di Semantica* 6.222–253.

FILLMORE, CHARLES J., CHRISTOPHER R. JOHNSON, & MIRIAM R.L. PETRUCK. 2003. Background to FrameNet. *International Journal of Lexicography* 16.235–250.

JOHNSON, MARK. 1987. *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. Chicago: University of Chicago Press.

LAKOFF, GEORGE. 1987. *Women, fire, and dangerous things*. Chicago: University of Chicago Press.

——, & MARK JOHNSON. 1999. *Philosophy in the flesh: the embodied mind and its challenge to western thought*. New York: Basic Books.

LANGACKER, RONALD W. 1982. Space grammar, analysability, and the English passive. *Language* 58.22–80. Revised version: *Concept, Image, and Symbol*, ch. 4.

—— 1991. *Foundations of Cognitive Grammar. Vol. 2. Descriptive Application*. Stanford: Stanford University Press.

—— 1993. Reference-point constructions. *Cognitive Linguistics* 4.1–38.

MILLER, GEORGE A., & PHILIP N. JOHNSON-LAIRD. 1976. *Language and perception*. Cambridge, MA: Harvard University Press.

MOK, EVA, JOHN BRYANT, & JEROME FELDMAN, 2004. Scaling understanding up to mental spaces. Presented at the 2nd International Workshop on Scalable Natural Language Understanding (ScaNaLU-2004).

TALMY, LEONARD. 1988. Force dynamics in language and cognition. *Cognitive Science* 12.49–100.

——. 2000. *Towards a Cognitive Semantics*. MIT Press.

WOLFF, PHILLIP, GRACE SONG, & DAVID DRISCOLL. 2002. Models of causation and causal verbs. In *Papers from the 37th Meeting of the Chicago Linguistics Society, Main Session*, ed. by Mary Andronis, Christopher Ball, Heidi Elston, & Sylvain Neuvel, volume 1, 607–622. Chicago: Chicago Linguistics Society.