From Frames to Inference

Nancy Chang, Srini Narayanan, and Miriam R.L. Petruck
International Computer Science Institute
1947 Center St., Suite 600, Berkeley, CA 94704
{nchang,miriamp,snarayan}@icsi.berkeley.edu

Abstract

This paper describes a computational formalism that captures structural relationships among participants in a dynamic scenario. This representation is used to describe the internal structure of FrameNet frames in terms of parameters for active event simulations. We apply our formalism to the commerce domain and show how it provides a flexible means of handling linguistic perspective and other challenges of semantic representation.

1 Introduction

The development of lexical semantic resources is widely recognized as a prerequisite to progress in scalable natural language understanding. One of the most semantically sophisticated efforts in this direction is FrameNet (Baker et al., 1998; Fillmore et al., 2001), an online lexical resource[1] designed according to the principles of frame semantics (Fillmore, 1985; Petruck, 1996). FrameNet takes as foundational the assumptions that (1) lexical items draw on rich conceptual structures, or **frames**, for their meaning and function; and (2) conceptually related lexical items may foreground different aspects of the same background frame. Verbs involved with commercial events serve as canonical examples:

(1) a. Chuck bought a car from Jerry for $1000.
  b. Jerry sold a car to Chuck for $1000.
  c. Chuck paid Jerry $1000 for a car.
  d. Jerry charged Chuck $1000 for a car.
  e. Chuck spent $1000 on a car.

The sentences in (1) might describe the same interaction – in which one individual (Chuck) transfers money ($1000) to another (Jerry) in exchange for some goods (a car) – but differ in the **perspective** they impose on the scene.

[1] http://www.icsi.berkeley.edu/framenet/

The shared inferential structure of verbs like *buy* and *sell* is captured in FrameNet by the COMMERCE frame, which is associated with a set of situational roles, or **frame elements** (FEs), corresponding to event participants and props. These FEs are used as annotation tags for sentences like those in (1), yielding, for example:

(2) a. [Chuck]$^{\text{Buyer}}$ **bought** [a car]$^{\text{Goods}}$ [from Jerry]$^{\text{Seller}}$ [for $1000]$^{\text{Payment}}$.
  b. [Jerry]$^{\text{Seller}}$ **sold** [a car]$^{\text{Goods}}$ [to Chuck]$^{\text{Buyer}}$ [for $1000]$^{\text{Payment}}$.

FE tags act as a shorthand that allows diverse verbs to tap into a common subset of encyclopedic knowledge. Moreover, regularities in the set of FEs realized with specific lexical items can be taken as correlated with their favored perspective.

A significant gap remains, however, between the unstructured and intuitively chosen tag sets used in FrameNet and a formal characterization of the interrelated actions and relations holding among them. The explicit representation of such frame semantic information is needed for FrameNet's potential use in text understanding and inference (Fillmore and Baker, 2001) to be fully realized. In this paper we attempt to bridge the gap by defining a formalism that unpacks the shorthand of frames into structured event representations. These dynamic representations allow annotated FrameNet data to parameterize event simulations (Narayanan, 1999b) that produce fine-grained, context-sensitive inferences. We illustrate our formalism for the COMMERCE frame and show how it can account for some of the wide-ranging consequences of perspective-taking.

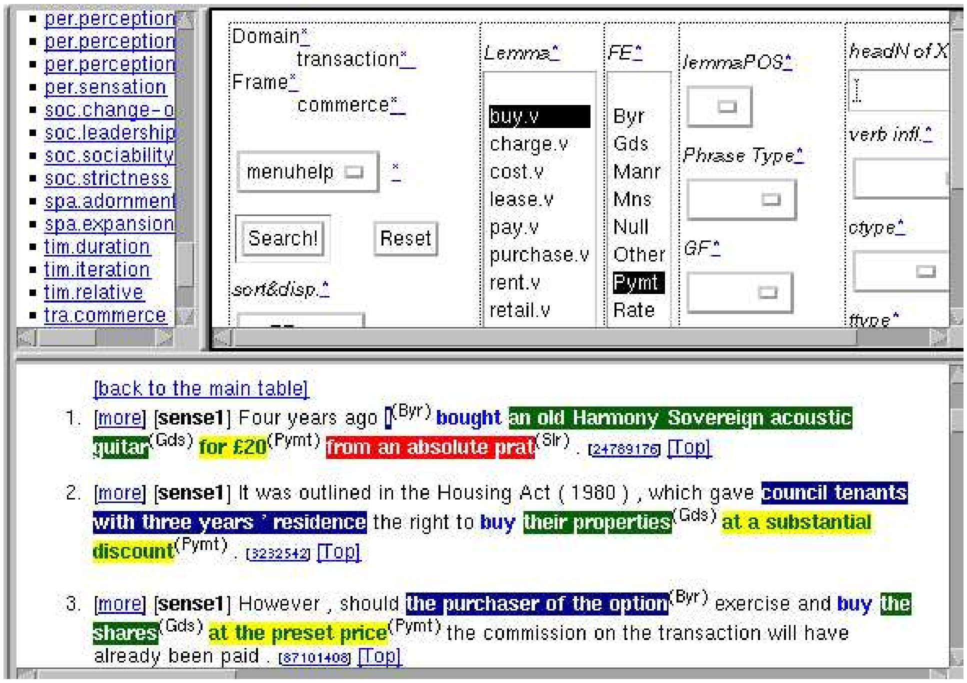

Figure 1: Results of a query on the FrameNet COMMERCE frame, showing annotated data for the verb *buy*.

2 The FrameNet COMMERCE frame

The FrameNet project has thus far produced two databases: a collection of approximately 80 frames with frame descriptions, chosen to cover a broad range of semantic domains; and a hand-annotated dataset of about 50,000 sentences from the British National Corpus (Baker et al., 1998). The databases document both syntactic and semantic behavior of a wide variety of lexical items (or **lemmas**) and thus have the potential to allow corpus-based techniques to be applied to semantically oriented tasks.[2]

The current release of the FrameNet databases[3] defines a COMMERCE frame with frame elements including the familiar Buyer, Seller, Payment and Goods, along with several other FEs needed to cover the data. The frame includes 10 verbs relevant to commercial transactions, for a total of 575 annotated sentences. Figure 1 shows a sampling of data annotated with respect to the COMMERCE frame.

The COMMERCE frame verbs exhibit relatively greater variety in argument structure possibilities (e.g., many but not all appear in the ditransitive construction), and not all of them license the expression of all FEs (e.g., *spend* rarely appears with the Seller). Also, some FEs are realized in different ways for different verbs: the pattern of appearance with *from-* or *to-* headed PPs differs for *buy* and *sell*, and *for-* headed PPs may mark either the Payment (for *buy* and *sell*) or the Goods (for *pay*).

Much of this diversity has been attributed to differences in perspective, but the complex connections among perspective, argument structure and FE realization have been difficult to explicate. One approach is to distinguish the COMMERCE frame from other **perspectivized** frames exhibiting more consistent behavior; proposals along these lines differ in their criteria for making these divisions and the relationships among the resulting frames.[4]

Our proposal likewise imposes additional structure on the COMMERCE frame to reflect both perspective-neutral and perspectivized situations. But for current purposes, we concentrate on capturing perspectival effects on *inference* in discourse

[2]See (Gildea and Jurafsky, 2000) for some promising initial work in applying statistical techniques to the FrameNet database to automatically label frame elements.

[3]Throughout the paper we refer to data from FrameNet I; an interim release of FrameNet II data is expected soon.

[4]FrameNet II is currently considering a reorganization scheme along these lines. Gawron (ms.) and Hudson (in press) address these and related issues, both arguing against handling perspective solely with multiple inheritance.

interpretation. Our structures are thus designed to support dynamic inferences about interrelated events and actions, and not specifically to reflect FE realization patterns.

We take the original COMMERCE frame as our starting point and define the interrelationships present among its FEs; the resulting event representation can license inferences for sentences like those in the annotated FrameNet data. We assume that any frame reorganization scheme (as well as any method for labeling roles in novel text) will define a mapping to the original FEs and thus could be integrated with our dynamic representations within a larger language understanding enterprise.

3 Structured event representations

In this section, we present a formal specification used for mapping the flat set of FEs in COMMERCE onto explicitly structured event representations that characterize commercial transactions. Our representation is based on the Embodied Construction Grammar (ECG) formalism (Bergen and Chang, 2002). As in that work, we assume that understanding an utterance involves the evocation of a complex network of conceptual **schemas** and the mental **simulation** of these schemas in context to produce a rich set of inferences.[5]

Among the features of the ECG schema formalism that we exploit to represent commercial transactions are (1) the ability to flexibly evoke and relate multiple schemas; and (2) the ability to assert dynamic conditions that apply to specific event stages. We rely heavily on the fact that schemas can be interdefined in our eventual representation of differences in perspective.

We briefly describe the schema definition language shown in Figure 2, deferring additional details for the examples. Keywords are shown in **bold**; a left square bracket ([) marks optional blocks; and curly braces ({}) enclose a set of options for the statements that may appear (possibly multiple times) in the block. Angle brackets (<>) denote a reference to an accessible structure or role, or a predicate on accessible structures.[6] The keyword **self** refers to the structure being defined, which we can consider a special kind of role. We use '//' to introduce italicized comments.

```
schema <name>
[subcase of <schema>
[evokes
   { <schema> as <local-name> }
 roles
   { <local-role>
     <local-role> : <restriction>
     <local-role> ⟷ <role>
     <local-role> ⟷ <role> : <restriction> }
[constraints
   { <role> ⟷ <role>
     <phase> :: <condition> }
```

Figure 2: Schema definition formalism.

The first line names the schema being defined. The next few lines (**subcase of** and **evokes**) indicate how this schema relates to other schemas. The **subcase** relation defines a lattice of schemas, with the local schema inheriting all roles and constraints. The **evokes** relation also makes the specified structures and constraints of the evoked schema (referred to within the current definition using a <local-name>) locally accessible, but it implies neither full inheritance of the evoked schema's roles nor containment in either direction.

The indented block labeled **roles** lists and constrains the schema's local **roles**, which are equivalent to frame FEs. Roles are declared with a local name (<local-role>) and may be accompanied by type restrictions (indicated with ':') and/or **identification** (or binding) constraints ('⟷'). The latter (which may also appear in the **constraints** block) causes roles and constraints to be shared between its arguments, similar to unification or coindexation in other frameworks (Pollard and Sag, 1994).

The final constraint type allowed is a **simulation constraint**, which uses the '::' notation to assert that some condition must hold (or not) at a particular phase of simulation (referred to as <phase>). These simulation phases correspond to event stages and serve as the bridging connection to previous work

[5]ECG includes formalisms for both **schema** definitions (conceptual representations) and **construction** definitions (conventionalized pairings of form and meaning). Since we focus here on translating frames to conceptual representations, we refer only to (a version of) the schema formalism.

[6]Accessible structures include locally declared roles and evoked or inherited structures, as well as any structures available through their roles. Standard slot-chain notation is used to refer to role y of a structure x as x.y.

on modeling event structure and linguistic aspect (Narayanan, 1997; Chang et al., 1998). An event may be viewed as having complex internal structure, including a start and finish and a period during which it is ongoing; at a coarser granularity it may also be viewed as a discrete temporal chunk that takes place between two time slices. The schemas defined below refer to the before, after and transition phases. (See Section 5 for more details.) The notation <condition> specifies a relation that holds or an event or action that takes place during the specified phase (see below).

We now show how the formalism just defined can be used to build up a set of progressively more complex schemas, culminating in an Exchange schema that provides much of the underlying structure we need to tackle the COMMERCE frame. For expository reasons, we omit details not relevant to the example. The schemas in Figure 3, for example, cast the relations of possession and causation as essentially unstructured. Each schema is declared as a subcase of Relation (a general frame not shown here), relating two entities (for the Possession schema) or two events (for the Cause-Effect schema).

```
schema Possession
  subcase of Relation
  roles
    possessor : Entity
    possesion : Entity
```

```
schema Cause-Effect
  subcase of Relation
  roles
    cause : Event
    effect : Event
```

Figure 3: Relational schemas.

```
schema Event
  roles
    before : Phase
    transition : Phase
    after : Phase
    nucleus
  constraints
    transition :: nucleus
```

```
schema Action
  evokes Event as e
  roles
    actor : Entity
    undergoer : Entity
    self ⟷ e.nucleus
```

Figure 4: The Event and Action schemas.

The schemas in Figure 4 define basic dynamic schema types and thus pave the way for dynamic constraints to be specified. The Event schema includes roles that refer to simulation phases, as described above, thus anchoring the event to the passage of time; its underspecified nucleus role is constrained to hold or take place during the transition phase. The Action schema corresponds to a prototypical situation in which an actor entity affects or manipulates an undergoer entity.[7] This schema is linked to its backgrounded temporal structure through an evoked Event. The explicit binding of **self** (the Action defined) to the nucleus role of the Event serves as the crucial link between events and actions. This link will be exploited in Section 4 to capture the relation between commercial transaction events and its various associated actions.

```
schema Receive
  subcase of Action
  evokes Possession as p
  roles
    receiver ⟷ actor ⟷ p.possessor
    received ⟷ undergoer ⟷ p.possession
  constraints
    e.before :: p.holds ⟷ false
    e.before :: p.holds ⟷ true
```

```
schema Transfer
  subcase of Event
  evokes Cause-Effect as c
  roles
    act-schema : Action
    rec-schema : Receive
    agent ⟷ act-schema.actor
    source : Entity
    theme ⟷ rec-schema.received
    recipient ⟷ rec-schema.receiver
  constraints
    transition :: act-schema
    transition :: rec-schema
    c.cause ⟷ act-schema
    c.effect ⟷ rec-schema
```

Figure 5: The Receive and Transfer schemas.

The relational and active schemas described above serve as building blocks for the more complex schemas in Figure 5. We have chosen to define a Receive schema – foregrounding an action defined by a change of possession state – and a Transfer event – a neutral rendering of an event in which one entity causes another to Receive something. Note that since Event and Action are linked as shown in Fig-

[7] This definition applies to all schemas relevant for our example, though it might be more precisely called Transitive-Action. Also, our role names coincide with those in Van Valin (1993) and are likewise associated with an array of prototypical features. We make no theoretical claims about their grammatical status, using them here simply as simulation parameters.

ure 4, we could also easily define an event-based, perspective-neutral version of Receive or an action-based, perspectivized version of Transfer. The formalism is flexible enough to represent any of these.

The Receive schema asserts that the evoked Possession relation holds (or does not hold) at the simulation phases specified, and it identifies the local receiver with both the actor (of the Action schema) and the possessor (of the Possession schema). The Transfer schema is defined as an event in which some action (act-schema) causes a receiving action (rec-schema). Note that these actions are conceptually distinct from the nucleus role inherited from Event, although all are constrained to take place during the event's transition phase. The Transfer schema also has an agent role that is constrained to be the same entity as the actor of the act-schema. Importantly, the Transfer event schema makes no commitment as to whether its agent – the entity seen as causing the overall event – is the source, recipient or even theme. It is in this respect that the Transfer schema defined here can be considered neutral in perspective.

```
schema Exchange
 subcase of Event
 roles
   participant1 : Human
   participant2 : Human
   entity1 : Entity
   entity2 : Entity
   transfer1 : Transfer
   transfer2 : Transfer
   agent : Entity
 constraints
   transition :: transfer1
   transition :: transfer2
   // Constraints on transfer1:
   transfer1.source ⟷ participant1
   transfer1.theme ⟷ entity1
   transfer1.recipient ⟷ participant2
   // Constraints on transfer2:
   transfer2.source ⟷ participant2
   transfer2.theme ⟷ entity2
   transfer2.recipient ⟷ participant1
```

Figure 6: The Exchange schema.

Finally, the representational machinery we have developed can be used to define an Exchange schema (Figure 6), which provides most of the relevant constraints needed for commercial transactions. It resembles the Transfer schema in structure and is similarly perspective-neutral. It includes two transfer events that occur during the transition phase and are parameterized straightforwardly in the constraints block by two human participants and two entities. An additional agent role is not bound to any particular entity, reflecting the ability of either participant (or both) to be viewed as active.

4 Commercial transaction schemas

We are now in a position to return to the commerce domain and put our inventory of domain-general event and action schemas to use. We first define the Commercial-Transaction (CT) schema as a subcase of the Exchange schema with appropriate role identifications and an additional type restriction on entity1. The role names in this schema differ slightly from those in FrameNet's COMMERCE, reflecting its perspective-neutral status. But given the obvious mapping to the FrameNet FEs, the CT schema fulfills part of our original objective: based on its inherited and evoked schemas and constraints, it concisely and precisely states the conceptual underpinnings of the basic commercial transaction, including all entailments that are perspective-neutral.

```
schema Commercial-Transaction
 subcase of Exchange
 roles
   customer ⟷ participant1
   vendor ⟷ participant2
   money ⟷ entity1 : Money
   goods ⟷ entity2
   goods-transfer ⟷ transfer1
   money-transfer ⟷ transfer2
```

Figure 7: The Commercial-Transaction schema.

The CT schema provides the underlying infrastructure for specifying how various associated schemas highlight different participants and event stages. As shown in Figure 8, we treat both Buy and Sell as subcases of Action that evoke the CT schema. Both action schemas identify themselves with the ct.nucleus role (inherited from Event) and are thus (separately) constrained to take place during the evoked CT's transition phase. They also both identify the undergoer with ct.goods, and the actor with ct.agent. The schemas impose different views on the same situation by virtue of a single additional constraint on this latter role (which corresponds to the active participant in the overall CT), binding it to either the ct.customer (Buy) or the ct.vendor (Sell).

Pay also evokes the CT schema but is directly identified with ct.money-transfer.nucleus; its actor is the ct.customer (as well as the agent of the money-transfer). This definition allows paying to refer specifically to a subpart of the overall commercial transaction, such that its execution does not necessarily entail the execution of the goods-transfer in the event (i.e., you don't always get what you pay for).

```
schema Buy
  subcase of Action
  evokes Commercial-Transaction as ct
  roles
    self ⟷ ct.nucleus
    buyer ⟷ actor ⟷ ct.customer ⟷ ct.agent
    goods ⟷ undergoer ⟷ ct.goods
```

```
schema Sell
  subcase of Action
  evokes Commercial-Transaction as ct
  roles
    self ⟷ ct.nucleus
    seller ⟷ actor ⟷ ct.vendor ⟷ ct.agent
    goods ⟷ undergoer ⟷ ct.goods
```

```
schema Pay
  subcase of Action
  evokes Commercial-Transaction as ct
  roles
    self ⟷ ct.money-transfer.nucleus
    payer ⟷ actor ⟷ ct.customer
              ⟷ ct.money-transfer.agent
    payment ⟷ ct.money
    payee ⟷ ct.vendor
```

Figure 8: The Buy, Sell and Pay schemas.

Other schemas associated with the CT schema lend themselves to similar analyses, though they draw on additional schemas not defined here. For example, the Spend schema evokes a schema for resource consumption (as in (Hudson, 2002)); Charge involves the vendor's communication of the price to the customer as a prequisite to the overall exchange of goods and money. In general, the CT schema explicitly specifies the internal event structure of a commercial transaction but remains noncommittal about which of its participants is seen as active. This flexibility in representation allows other schemas to effect the bindings that make appropriate commitments on an individual basis.

5 Simulation semantics

The structured event formalism we have described allows us to translate FrameNet descriptions into a representation suitable for simulative inference. Central to the representation is an event model called **executing schemas** (or **x-schemas**), motivated by research in both sensorimotor control and cognitive semantics (Narayanan, 1997). X-schemas are active structures that cleanly capture sequentiality, concurrency and event-based asynchronous control. They thus provide a cognitively motivated basis for modeling diverse linguistic phenomena, including aspectual inference (Chang et al., 1998), metaphoric inference (Narayanan, 1999a) and event-based reasoning in narrative understanding (Narayanan, 1999b).

The model is based on the Petri net, which in its basic form is a weighted, bipartite graph consisting of *places* (shown as circles) and *transitions* (shown as rectangles) connected by directed input and output arcs (Murata, 1989; Narayanan, 1997). Places may contain *tokens* (i.e., they may be *marked*), and they typically represent states, resources or conditions that apply. Transitions typically represent actions or events. X-schemas extend the basic Petri net to include typed arcs, hierarchical control, durative transitions, parameterization, typed (individual) tokens and stochasticity.

The most relevant property of the x-schema for this paper is its well-specified execution semantics: a transition is *enabled* when all its input places are marked, such that it can *fire* by moving tokens from input to output places. The active execution semantics serves as the engine of context-sensitive inference in the simulation-based model of language understanding mentioned earlier.

The ECG formalism is designed to allow constraints on x-schema simulation to be expressed. In particular, the Event schema in Figure 4 has roles that refer to event phases; these correspond to x-schema places and transitions. Other schema roles specify x-schema parameters, which allow x-schemas to give rise to different execution traces through the network with different parameters.

The Commercial-Transaction schema has been implemented in the KarmaSim x-schema simulation environment (Narayanan, 1997); Figure 9 shows part of the network. The phase roles from the schemas in Section 3 have been mapped onto the fine-grained temporal structure of each event, corresponding to the various control nodes in the network (ready, ongoing, finish, done, etc.); the transition phase referenced in the schemas includes start, ongoing and

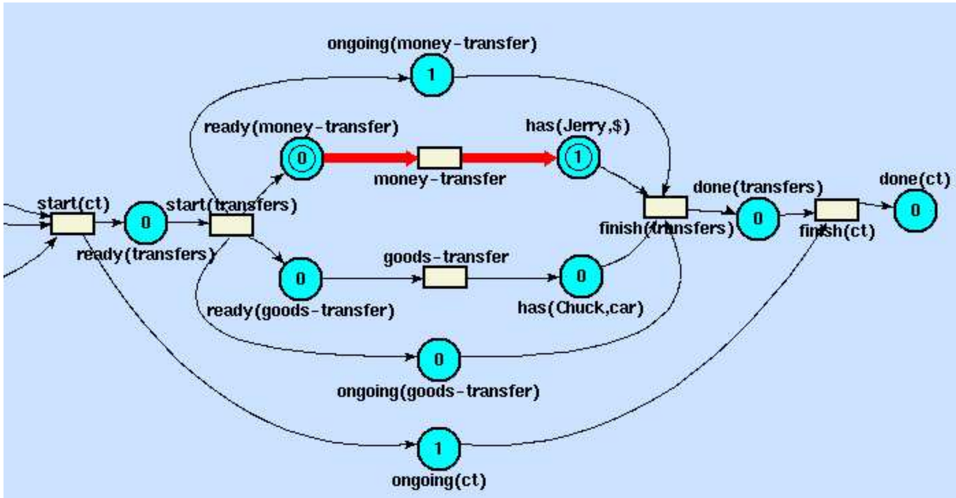

Figure 9: KarmaSIM simulation of the Commercial-Transaction schema. The highlighted execution is associated with the Pay schema, corresponding to the money-transfer event.

finish. As shown, execution of the overall CT schema comprises the execution of two subsidiary events, the goods-transfer and the money-transfer. These need not be synchronized, but both must complete for the overall commercial transaction to complete (enforced by the arcs from ongoing(money-transfer) and ongoing(goods-transfer) to finish(transfers)).

The highlighted money-transfer portion of the network corresponds to a simulation of the Pay schema. The token in ongoing(ct) shows that there is an ongoing transaction, but the finish(transfers) transition is not enabled. Technically, the done(ct) place is not *reachable* (absent other information), since the simulation of Pay does not provide direct evidence for the occurrence of a goods-transfer.[8] In contrast, both Buy and Sell involve simulating the entire transaction, include both transfers as well as the done(ct) node. (Figure 9 can be considered an expansion of the CT schema's transition phase.)

What about the perspectival difference between Buy and Sell? The simplest simulative counterpart of perspective is the specification of differing actor participants. The actor is defined as supplying tokens needed for simulation (not shown in the figure), which could be interpreted as providing the energy, initiative or control for the event.

Capturing the foregrounding effect of perspective is more challenging, and will require an account of linguistic focus. Such an account remains a topic of ongoing research; a preliminary suggestion is to allow simulation of different parts of the event at varying degrees of detail. For example, the simulation for Buy may execute the x-schemas in which the Buyer interacts with the Goods – such as the goods-transfer and its resulting possession (abbreviated as has(Chuck, car) in Figure 9) – at the default granularity, while other x-schemas are collapsed into less detailed simulations.

Although a detailed treatment of this topic is beyond the scope of this paper, these proposals illustrate how simulation semantics can offer elegant solutions to classic representational problems, or at least facilitate the articulation of factors affecting language understanding.

6 Discussion and conclusions

Frame semantics in general and FrameNet in particular show considerable promise for use in deep semantic analysis. FrameNet frames are intended to capture crucial generalizations not available in other lexical resources. WordNet (Fellebaum, 1998), for example, includes only simple taxonomic relations (*buy* and *sell* are listed as hyponyms of *get* and *give*, respectively, and as antonyms of each other). The PropBank project (Kingsbury and Palmer, 2002) is, like FrameNet, geared toward the creation of a

[8]Contextual or background knowledge could provide evidence for the other transfer or allow it to be inferred by default.

semantically annotated corpus (by adding general logical predicates to the Penn Treebank), though without any common background frame structures across lexical items.

While frames and FE tags are meaningful to human interpreters, they are not yet suitable for use in NLU. In this paper we have shown how FrameNet tags can be precisely defined in terms of structured event representations, which can support parameterize simulations that license active inferences. The formalism appears expressive enough for the COMMERCE frame and its associated perspectival effects, and we believe the methods we have used will scale well for other representational problems.

Work under way suggests that our schemas, together with formalisms developed for linguistic constructions, can be extended to account for some of the linguistic patterns associated with commercial transactions. For example, a countertheme argument could be added to each schema in Figure 8 and bound to the Payment for Buy and Sell and the Goods for Pay. A construction associating *for*-PPs with this role could have variable interpretation (relative to the overall CT) for the relevant verbs. Related work would show how the model could address more challenging perspective-related phenomena, including reference resolution (e.g., the differing interpretation of *He* in *Chuck bought a car from Jerry. He got a 10% discount/?commission*) and word sense disambiguation (e.g., whether *new* in *Chuck bought a new car from Jerry* refers to a car that has never been owned before or to one that has not previously been owned by Chuck).

Future work involves extending the representation to cover a broader subset of the FrameNet database and automating the process of mapping frame definitions to simulation parameterizations. The resulting representations should be useful for a variety of NLP applications, including question answering and information extraction.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of COLING-ACL*, Montreal, Canada.

Benjamin K. Bergen and Nancy C. Chang. 2002. Embodied Construction Grammar in simulation-based language understanding. Technical Report TR-02-004, International Computer Science Institute.

Nancy Chang, Daniel Gildea, and Srini Narayanan. 1998. A dynamic model of aspectual composition. In *Proceedings of the Twentieth Annual Meeting of the Cognitive Science Conference*, Madison, Wisconsin.

Christianne Fellebaum, editor. 1998. *WordNet: An Electronic Database*. MIT Press.

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

Charles J. Fillmore, Charles Wooters, and Collin Baker. 2001. Building a large lexical databank which provides deep semantics. In *Proceedings of the Pacific Asian Conference on Language, Information and Computation*, Hong Kong.

Charles J. Fillmore. 1985. Frames and the semantics of understanding. *Quaderni di Semantica*, IV(2).

Jean Mark Gawron. 2002. Frames, modifi ers, and argument structure. Manuscript, San Diego State University.

Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the ACL*, Hong Kong, October.

Richard Hudson. 2002. Buying and selling in Word Grammar. In press.

Paul Kingsbury and Martha Palmer. 2002. From Treebank to PropBank. Submitted to the 3rd International Conference on Language Resources and Evaluation (LREC-2002).

Tadao Murata. 1989. Petri nets: Properties, analysis, and applications. In *Proc. IEEE-89*, volume 77, pages 541–576.

Srini Narayanan. 1997. *Knowledge-based Action Representations for Metaphor and Aspect (KARMA)*. Ph.D. thesis, University of California at Berkeley.

Srini Narayanan. 1999a. Moving right along: A computational model of metaphoric reasoning about events. In *Proc. Sixteenth National Conference of Artificial Intelligence (AAAI-99)*. AAAI Press, Menlo Park.

Srini Narayanan. 1999b. Reasoning about actions in narrative understanding. In *Proc. Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*. Morgan Kaufmann Press.

Miriam R.L. Petruck. 1996. Frame semantics. In J. Verschueren, J. Östman, J. Blommaert, and C. Bulcaen, editors, *Handbook of Pragmatics*. John Benjamins, Philadelphia.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Robert D. Van Valin. 1993. A synopsis of role and reference grammar. In Robert D. Van Valin, editor, *Advances in Role and Reference Grammar*, pages 1–166. John Benjamins Publishing Company, Amsterdam.