THESIS PROPOSAL

One more step: An on-line model of metaphoric interpretation

by

Srini Narayanan
University of California at Berkeley
Professor Jerome A. Feldman, Advisor

January 30, 1995

Abstract

On-line narrative interpretation can be viewed as the simultaneous satisfaction of two sets of constraints. First, the linguistic structure of the input governs how linguistic constructions are accessed and selected. Second, the agent's mental model of plausible world situations constrains the set of allowable interpretations. My work proposes a uniform processing model where both types of constraints can be represented and their joint effects on interpretation studied. In the model, constructions and conventional metaphors are integrated if this best reduces the uncertainty of the resulting interpretation. Initially, the agent makes a selection choice between competing interpretations based on the degree to which they activate an existing mental model. Once activated the executing mental model sets up a dynamic context which conditions future interpretations of the linguistic input. Preliminary results indicate that the approach proposed here shows promise in solving some well known problems in disambiguation and automatic inference.

Contents

1 Introduction

This project attempts to build an on-line computational model of the semantic structure of causal narratives. The input is taken from newspaper stories in the domains of Politics and Economics, and the proposed system should represent the semantics in a way that generates commonsense inferences about the stories.

In the proposed model, linguistic structures map the textual input onto the interpreter's active mental world model. The mapping process is probabilistic and utilizes relevant grammatical and pragmatic knowledge (constructions and conventional metaphors). I will refer to individual elements of the grounding mental model as *schemas*. Schemas encode dynamic knowledge. When activated, schemas execute and allow for certain types of automatic inference. Executing schemas also set up a dynamic context which participates in the selection of new linguistic structures in the developing interpretation.

Two observations motivate the approach outlined above. First, causal narratives from abstract domains acquire a significant portion of their semantics from metaphoric mappings to specific schemas in the concrete spatial and temporal domains. I postulate therefore, that a computational framework for the representation and integration of metaphoric knowledge into the on-line interpretation process is crucial to model the semantics of these narratives. Recent work in Cognitive Semantics provides good evidence for this view. In particular, there are proposals for schema-based semantics, as in the case of *Image Schemas* [Lakoff1987]. The work in Cognitive Semantics, however, lacks a computational model for such theories.

Second, I note that the deep semantics of the causal narratives are *dynamic* and arise from a *continuous interaction* between input and memory. This enables the high degree of context sensitivity required since changing input context can dramatically affect the correlation between input and memory and thereby the set of possible expectations, goals, and inferences. Any representation of the semantics should therefore be of a fine granularity, be flexible and adaptive in the way context affects interpretation, and be able to support concurrent activities in the memory storage, retrieval, and indexing process.

The first observation suggests using a uniform framework where metaphoric knowledge should be integrated into the developing interpretation if its usage can allow the listener (reader) to better map from the linguistic input to the appropriate mental model(s). It is also obvious that in cooperative communication, the speaker (writer) can also count on the conventional metaphor being used by the listener in interpreting the narrative.

The second observation suggests the use of *process semantics* [Nielsen1981] to characterize the deep semantics of causal narratives. Of special interest to us is the possibility of modeling such processes in structured connectionist models [Feldman1989].

We first illustrate these observations by describing their usage in interpreting a recent newspaper story. We then describe the status of this effort and the problems remaining. Finally the scope of the project is summarized.

1.1 Example

Consider the following paragraph that appeared in the New York Times on Sunday, August 1, 1993.

> Britain was deep in recession while Germany was booming three years ago. France kept booming long after Germany sunk into recession. Now France is plunging deeper while Germany hopes it has bottomed out. Britain, which was able to cut interest rates after leaving the exchange rate mechanism, has been able to stimulate its economy and has started to emerge from recession.

In interpreting the narrative above, there are several aspects to be noted.

- Germany, France, and Britain are all conceptualized as persons. This is a fairly common occurrence and is based on a conventional metaphoric mapping *Country Is Mapped As Person.*
- Economic recession is treated as a feature in a terrain, namely a *hole* that you can sink into, plunge into, be deep in, emerge from, and a *hole* which can bottom out. The mapping from an abstract economic state (such as Recession) to a concrete spatial location/region is an instance of the metaphoric mapping *Abstract State Is Mapped As Spatial Region.* This particular metaphor belongs to a set of inter-related metaphors called The Event Structure Metaphor [Lakoff1992].
- There is an implied inference about the *negative* aspects of sinking into holes, and the *positive* aspects of booming or emerging out of them. This kind of directional preference is an instance of a class of *orientational* metaphors. Notice that this preference generates an *implicit goal* for any agent, that constrains the possible interpretations of the input text.
- From the fact that the countries are conceptualized as persons, we note the specific causal events that are used in the article, such as sink into, emerge from, plunge into, stimulate, leave, etc. These are specific *agent motions in space* whose semantics is based on the *manner* of motion, the implied or explicit *path* (such as *into*), the *resultant state* (such as increased energy in stimulate) and *aspect* (*kept booming* versus *was booming*, *started to emerge*, *has bottomed out*, *is plunging*, etc).

1.2 Summary

From the example above, it is clear that various kinds of knowledge have to be brought to bear simultaneously in narrative interpretation. This thesis proposes to investigate a computational model where conventional metaphors *unify* linguistic knowledge with world knowledge to provide context-sensitive disambiguation and automatic inference. The proposed solution makes use of ideas and techniques from process theory, structured connectionist systems and belief nets, and from recent research in linguistics.

The expected contributions of the proposed research include:

1. A computational architecture to unify linguistic knowledge (grammar) with cognitive models *schemas*. If successful, this has the potential of bringing together research from the currently separate fields of *cognitive linguistics* and *computational linguistics* together in a computational model.

2. A probabilisitic model of narrative understanding by metaphoric mapping from abstract domains, such as politics and economics to the concrete spatial domain. I claim that such a mapping process constrains acceptable parses of an input sentence thereby impacting the disambiguation process.

3. The translation of the metaphoric interpretation process into multiple, executing models or schemas. This allows for a single system to be capable of representing context and be capable of modeling certain types of automatic inference.

1.3 Road map

Section 2 describes the proposed implementation strategy. This should give the reader an understanding of the specific task, the previous work that this proposal draws from, an an idea of the current status of the effort. Sections 3 and 4 describe the representation and use of linguistic and metaphoric knowledge. This section sets up the theoretical framework for one part of the thesis, namely mapping from linguistic input to schemas. Section 5 illustrates an example of the mapping process. Sections 6 and 7 describe the representation and use of world knowledge and the overall interpretation process. These sections describe the role of schemas, their representation, use, and capabilities. Section 8 outlines some issues and possible extensions to the proposed effort. Section 9 concludes by identifying some areas of potential contribution. Finally, Appendix A describes briefly some of the common metaphors found in my database of simple newspaper stories. Appendix B shows a trace of the proposed program interpreting a simple narrative.

2 Implementation Approach

Here I will outline the task I propose to solve and a solution approach. The proposed implementation approach connects up closely to previous work in on-line interpretation. I will try to make this connection explicit in this section. I will then present an abstract version of the overall algorithm and describe the current status of the project. Details of individual aspects of the project can be found in future sections.

2.1 The Task

Consider the following narratives.

1. France had fallen into recession. Germany pulled France out.
2. India removed import tariffs. Japanese goods flooded the market.
3. Stock prices rose today. They had fallen last week.
4. Britain entered the common market. Britain tried to leave the common market.[1].

The first narrative only mentions two events. But the interpreter can immediately assume several other events/facts about the situation being described. For instance, the interpreter can conclude that France is out of the recession at the end of the narrative; that Germany must have expended some effort and resources in pulling France out; that Germany was most likely *not* in recession when the incidents described in the narrative occurred, etc.

In the second narrative, the interpreter felicitously makes the causal inference that the presence of import tariffs is responsible for keeping goods *out* of the market, and that absence of import tariffs leads to goods *coming in* to the market, etc. If asked to paraphrase, the interpreter may return something like:

1. Import tariffs keep goods from coming into a market.
2. India had *high* import tariffs.
3. This kept goods from flowing in to India's markets.
4. India removed import tariffs.
5. This allowed Japanese goods to *flow into* the market.
6. Unrestricted and fast flow of goods resulted in flooding the market.

[1] taken from Wilks [Wilks1978]

The interpreter uses background knowledge that treats import tariffs as if they were *barriers* that keep *inflow* to a minimum. Removing import tariffs is thus viewed as removing an inflow barrier. The interpreter treats goods as if they were a fluid that can flow in and out of markets. This allows goods to flow into, accumulate and flood the market. Clearly, this logic extends to other predicates and implicatures such as *lower tariffs* ⇒ *lower barriers*, and *weaker regulations* ⇒ *greater flow*.

In the third narrative, the interpreter makes the inference that today the prices of the stock are *more*, and that last week they were *less*.

In the fourth narrative, mapping Britain as a person who has entered a community (common market) allows the interpreter to access facts about *entering*, *leaving* and *being in* communities. The second sentence in the narrative is felicitously interpreted since mapping the first sentence sets up a context where the person (Britain) ends up *inside* the community (common market). This allows for the person (Britain) to try to leave the community (common market).

How should the interpreter be making these inferences? There are two possible answers to this question.

1. The lexicon of the interpreter contains a separate lexical "sense" for events such as fall, pull, rise, and flow as they are used in the domain of economics. Whenever the interpreter sees these events associated with the words recession, tariffs, interest rates, markets, or other terms from economics, it selects this word sense. This implementation strategy will work fine, but will not have generative capacity. For instance new sentences the "stocks plummeted" or "India stumbled into recession" cannot be accounted for. The problem is even more acute since these events can be used in multiple domains (The Chargers' popularity rose after their Super-bowl victory (not!)).

2. The second possibility, and the one I intend to explore in this thesis, is that the interpreter makes use of explicitly represented common metaphors, and uses the metaphoric interpretations to generate the inferences shown. For instance, the metaphoric interpretation of the first narrative uses the mappings "Economy is a traveler in a landscape","Country is a Person", and "Recession is a Hole". This allows the interpreter to metaphorically interpret the first sentence as being similar to the familiar experience of people falling into holes. Of course, not all the inferences can be obtained this way, and an expert economist will infer several other things (such as GDP decrease over last two quarters, etc.). However, non-experts read the newspaper and using metaphors helps non-experts comprehend the narrative.

The "familiar experience" of falling into holes mentioned earlier is encoded by a **schema** in the proposed model. Interpreting the input sentence corresponds to selecting and instantiating a schema. This selection and instantiation process will be referred to as **binding** to a schema. Metaphors and other common linguistic usage patterns (constructions) aid in the schema binding process. For example, the "Economy is a traveler in a landscape" metaphor has many source domain schemas corresponding to people moving ahead, falling into holes, climbing out of holes, being pulled out, being sidetracked, racing ahead, stumbling into holes, crawling out of holes, etc. Notice how all the predicates project from the spatio-temporal domain (the source) to structure narratives about economics (the target). The use of this metaphoric mapping thus generates a context for the story being understood. Since schemas are **active** memory elements (my current design uses Petri nets), they generate automatic inferences. Thus the context used in interpretation is dynamic and depends both on the metaphor being used and the specific schema being bound to.

2.2 Solution Approach

Schemas encode dynamic knowledge. When activated, schemas execute by simulating the effects of the linguistic input on the agent's mental model. Figure 1 shows a simple finite-state machine like schema for simulating the relevant components of falling into holes.

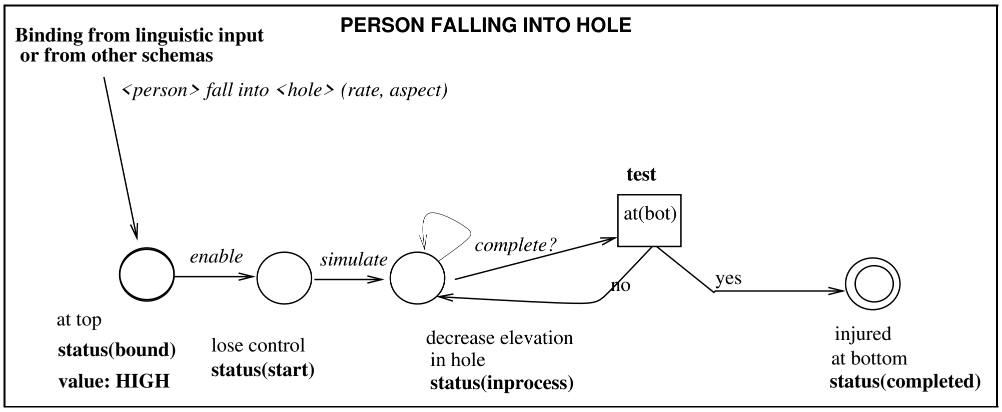


Figure 1: Schemas encode dynamic knowledge, and are *active* memory elements. The figure shows a simplified representation of a person falling into a hole. Details can be found in Sections 6 and 7.

For example, when activated, the schema shown in Figure 1 generates the spatial inference that in the beginning of the simulation, the person falling was outside the hole (at the top), and at the end of the all he is in the hole (at the bottom). Other inferences like the person loses control of his location during the fall, etc. are also generated. In the proposed model, multiple schemas may be active at the same time, and schema execution may lead to chaining and threading into other schemas. These considerations and other control and binding issues resulted in the current proposal for schema representation that uses augmented Petri nets. Detailed discussion of the issues involved and the representation chosen will have to wait until Section 6.

The central idea that will recur throughout the discussion to follow is **Linguistic input binds to schemas**. This casts the disambiguation problem into one where the interpreting agent is selecting between **bindings** to different schemas. Constructions, metaphors and other knowledge sources are brought to bear on the input to converge onto a schema-binding. Binding to a schema involves selection of the schema and supplying fillers to specific slots in the schema. Schema-slots are a superset of the theta-roles [Fillmore1989] in construction grammar or frame-semantics. This schema-specific role-filler representation is called the **binding structure** of the relevant schema.

The problem of disambiguation is a problem of *binding* at two levels. First, the interpreter has to be able to recognize the linguistic input as referring to a schema. Second, the interpreter has to bind the appropriate theta-roles to possible fillers in the input. Usually, the matrix verb in a thematic grid, or the verb-construction in a verb sub-categorization frame explicitly refers to the schema type. In cooperative communication, the speaker usually supplies the name of the relevant schema in the input, so the hearer does not have to deduce it through some search process. For instance, the input "Brazil fell into recession" refers to a *fall into* type[2] of schema. Selecting the appropriate fillers for the theta-roles is similar to the job of unifying roles and fillers in unification grammars. This leads to the well known problems of disambiguation.

Consider the metaphoric interpretation of the input sentence "France fell into recession". The best interpretation consists of the following bindings[3]:

Schema name/type	**fall into**
Schema instance	fallinto1
Trajector	France:person1
Tr.location	
Landmark	Recession:hole1
Source	
Destination	
Rate	
Status	Active
Aspect	Perfective
Tense	Past

Bindings to schemas are maintained in a schema specific feature vector representation called the **binding structure**. The table above shows shows the binding structure for the fall schema with the input bindings filled in after the linguistic input has been disambiguated. Only part of the structure has fillers for slots. In this case, only the theta-role (verb valence) slots have fillers. The rest of the structure is unfilled or has default values for its fillers. At this stage the underlying schema is *bound* and can be *activated.*

Many binding sequences may partially satisfy the constraints imposed by the theta roles. The problem is thus one of *choosing* among the various possibilities. In many cases metaphoric interpretation provides a preferred binding choice. As an example of this process consider the two possible binding sequences in the sentence "Prices rose". One sequence results in the literal interpretation *prices rose in altitude* (where prices are movable substances). However, using the *more is up* metaphor we get the new choice *prices increased in quantity* (where prices are quantities) In this case, the metaphoric interpretation fits the theta-role constraints better and is the preferred interpretation.

Selecting an interpretation can be viewed as the problem of *diagnosis.* Here the interpreter uses linguistic knowledge to select the most likely schema to bind to *and* the best theta-role fillers to select. The selection process can be modeled by formal probabilistic inference techniques. In short,

[2]Schemas have the verb and path collapsed, whereas traditionally constructions are indexed by root verb with a path as a parameter. This schema design is currently to ease implementation of schemas. However, as [Talmy1985] points out, many languages only allow constructions where the path is a suffix to the root verb, so there is some linguistic justification for this choice.

[3]I will denote the metaphoric binding France is interpreted as a person by France:person, where the colon stands for the the Metaphor Country ISA person

constructions explain linguistic input. Competing constructions accessed by the input become competing explanations for the presence of the input in the given context. For example, both the *fall* verb-construction and the *fell swoop* constructions can be viewed as competing explanations for the orthographic input *fell* in "France fell into recession". One is much more probable than the other, and given the various constraints, the verb construction is selected as the better *cause* that explains the presence of the input evidence.

This formulation allows us to investigate the use of **belief network** construction and inference techniques [Pear1988] to model disambiguation. In the current formulation, as the input utterance comes in, a belief net is constructed where competing constructions accessed by the input are parents of the equivalent input node in the belief net. Fillers to theta roles in constructions are selected based on how well a role construction associates with the filler. The measure used here is a variant of an information measure called Selection Association (SA) used by [Resnick1993] to quantify how a type of role selects for a filler. This allows us to use standard belief net inference algorithms combined with the SA measure to generate the best interpretation for the input. This process returns a pointer to the schema to bind to, as well as the fillers for the various theta-roles of the selected schema. Section 3 and 4 outline the current formulation in detail.

Schemas are *active* memory elements that execute when **bound**. As they execute, they generate new events and bindings. For instance, the falling into a hole schema executes and supplies a filler to the location slot of the binding structure. In this case, the interpreter infers that the faller is at the bottom of the hole when the fall is completed. The table below shows the binding structure after schema execution. The main thing to notice here is that the execution of the schema fills in most of the binding structure. Some slots are still unfilled (such as the rate slot) or may have defaults. At this stage the bound schema has finished execution.

Schema name/type	**fall into**
Schema instance	fallinto1
Trajector	France:person1
Tr.location	Lm.bottom
Landmark	Recession:hole1
Source	Lm.top
Destination	Lm.bottom
Rate	
Status	Completed
Aspect	Perfective
Tense	Past

When the next input comes in, there is a context which consists of the agent being located at the bottom of the hole. In the proposed system, this context plays a role in the explanation of future input. The mechanism proposed allows the fact of being in a hole to prefetch certain constructions that pertain to ways of getting out (such as being pulled out). This allows for context sensitive interpretation of Germany pulled France out. Specifically, it allows for the anaphoric binding that binds the *implicit* object x in the prepositional phrase "out-of (x)" to the recession:hole object. Also implicit at the end of the narrative is the implication France is out of recession. This comes about because of the use of the perfective aspect [Comrie1976] in the narrative.[4]

Schema name/type	**pull out**
Schema instance	pullout1
Trajector1	Germany:person2
Tr1.location	
Trajector2	France:person1
Tr2.location	Lm.bottom
Landmark	Recession:hole1
Rate	
Status	Active
Manner	
Aspect	Perfective
Tense	Past

The binding structure shown when the second utterance comes in. The schema (someone pulling someone out of a hole) has made the simple anaphoric binding that the agent in the first sentence is at the bottom of the hole, and the pull landmark is the same hole mentioned in the first sentence. This resulted in prefetching the pullout schema (and the corresponding pull-event construction) with these slots already bound (so the speaker does not have to specify them). When the rest of the input comes in, the pullout1 schema is **bound** and ready to execute. In fact, once the source domain mapping has been accomplished, the binding structures could be a part of a large Script [Schank1977] where the anaphor resolution involves no search. One common Script is:

1. Person1 is moving ahead.
2. Person1 falls into hole.
3. Person1 is pulled out by Person2 out of the hole.
4. Person1 is moving ahead again.

When the pullout1 schema executes, the rest of the structure gets filled in, resulting in a new binding structure shown below. The execution of the schema asserts that the person pulled is out of the hole, an inference that comes from the semantics of pull out and the perfective aspect used. When projected back onto the target domain, this asserts that France is *out of recession* at the end of the narrative.

[4]Compare this with the sentence "Germany was pulling France out."

Schema name/type	**pull out**
Schema instance	pullout1
Trajector1	Germany:person2
Tr1.location	Lm.top
Trajector2	France:person1
Tr2.location	Lm.top
Landmark	Recession:hole1
Rate	
Status	Completed
Manner	
Aspect	Perfective
Tense	Past

Scripts allow for the use of a single binding structure for the entire narrative. Scripts illustrate one kind of inference source domain schema execution provides. However, as is well recognized in the plan recognition community [Carberry1991], the process of indexing to world knowledge requires much finer-grained representation and dynamic assemblages than is provided by Scripts. My own study modeling aspectual inferences confirms this hypothesis in metaphoric interpretation, and suggests requirements for schema representation. The topic is discussed in detail in Section 6.

Clearly, not all characteristics of falling into holes (or other schemas) will necessarily be projected metaphorically onto other domains. However, specific *types* of characteristics, such as the fact that *falling into a hole* leaves you *in the hole* and other *spatial inferences* are projected (if a country falls into recession, it is in recession). One of the contributions of this thesis is to propose which kinds of inferences are projected between domains in metaphoric mappings. A first cut proposal can be found in Section 6.3.

In general, the both the target domain and the source domain of a metaphor map generate inferences. The example *McEnroe killed Connors* [Martin1990] illustrates this. The proposed schema representation can potentially model both target and source domain inferences. In this case, we have parallel contexts being maintained, one in the source and another in the target domain. The resulting interpretation is a **blending** of the two simultaneous inference processes.

However, in the proposed project, I intend to concentrate on the set of inferences that can be obtained from the source domain. While this still leaves a significant portion of the inference problem unsolved, I will try to illustrate that modeling the source domain is a crucial part of any model of inference. In terms of implementation, my architecture for disambiguation is capable of selecting the correct interpretation for the case above, but the problem of inference with **blending** shall await future work. Some preliminary ideas for the kinds of operators required is discussed in Section 8.

Given this overall approach, the specific problems I intend to address are are how to use common metaphors and constructions to interpret simple narratives such as the ones above. This involves representing metaphors in an on-line parser, using metaphors for disambiguation and representing and using relevant source domain schemas for certain types of automatic inference.

For most of the discussion, I will concentrate on those aspects of parsing, disambiguation and inference that deal with metaphors. However, the solution I propose should deal well with other kinds of ambiguity, such as syntactic ambiguity, lexical ambiguity and semantic ambiguity. In fact the general approach has already shown promise in solving these other disambiguation problems. A detailed exposition of this work can be found in [JN1995].

2.3 Background

Modeling online metaphoric interpretation is not new. [Martin1990] studied this problem in his thesis, and in many respects the work proposed here draws on his effort. However, there are many points of difference between the two efforts. The main difference is that Martin concentrated on using metaphors to aid in *learning*, whereas I concentrate on the use of metaphors in disambiguation and inference. This makes the modeling of the source domain of the metaphor an important component of my thesis, while it was not a central issue of focus in Martin's research. There are several other differences in our approaches, and I will point them out in the appropriate sections.

Probabilistic network techniques to model the interpretation process have been proposed by Dekai Wu [Wu1993] and Charniak [Charniak1988] [Charniak1989]. My proposal is different from these efforts in that I show how conventional metaphors and dynamic context can be brought to bear during interpretation. Also, as Norvig and Wilensky observe, [Norvig1990], the proposals of Charniak and his students did not take into account linguistic usage patterns (such as constructions). In more recent work [JN1995], we have demonstrated the utility of incorporating such knowledge. This allows us to study the problem of disambiguation in greater detail than was previously possible.

Apriori probabilities of constructions are less important for disambiguation than conditional probabilities, given a context. For instance the words *ball, bat, pitch* when seen together may result in the selection of either the *cricket* or the *baseball* sense of these words (perhaps depending on which country of origin of the interpreter). I propose to model such contextual effects in disambiguation using an active memory model called *schemas*. The degree to which an interpretation binds to a schema and the prior frequency of use of the schema affect the disambiguation process.

Extending metaphoric interpretation to generate dynamic context and source domain inferences is entirely new and is the significant open computational and linguistic problem addressed in this thesis. In designing the representation of schemas, I intend to borrow methods and techniques from robotics [Brooks1986], [Drescher1991], from computational neuroscience [Arbib1992], and from research in situated automata [Rosenshein1985].

The proposed implementation approach also relies on previous work in on-line interpretation, and can be viewed as an extension of Dan Jurafsky's thesis [Jurafsky1992] in several areas. Jurafsky's thesis used the theory of *Construction Grammars* [Fillmore1989]. The basic idea behind this theory is that knowledge of language is represented as a collection of uniform structures or *constructions* which contain lexical, syntactic, and semantic usage information.

Interpretation was divided into the three processes of *access*, *integration* and *selection*. In Jurafsky's implementation, the interpreter worked by accessing grammatical constructions from the knowledge-base or *long-term store* which holds the linguistic knowledge (grammar) of the interpreter. The access function combines both top-down and bottom-up evidence from syntactic and semantic sources. When the evidence for a construction exceeds a *threshold*, the construction is said to be accessed. Accessed constructions are part of the *working store* of the interpreter. The interpretation for the overall sentence is constructed by *integrating* elements of the working store into candidate interpretations. In my proposal, I will assume the presence of the *access* function, and be concerned mainly about the *integration* and *selection* processes.

Described below are the inputs and outputs of the proposed model followed a brief description of the overall algorithm and a summary of the current status of the effort. Details of the various parts of the project can be found in subsequent sections.

2.4 Input/Output

- **Input**.

  I will *assume* that input sentences are parsed using a probabilistic Construction Grammar Interpreter (such as *SAL+* [JN1995]). I propose to enhance the grammar to include metaphoric knowledge.

  The proposed system will also include a lexicon of the relevant action constructions and the associated theta-role constraints. I hope to test the theory on examples of action terms such as carry, lead, climb, plunge, push, touch, back, cripple, fall, and stumble, flow, etc.

- **Output**.

  The output of the proposed system will be the set of bindings that result from interpreting the abstract narrative as a series of mappings to schemas. The mappings make use of metaphors (such as the Event Structure Metaphor) as well as the specific process characteristics resulting from schema execution.

2.5 Overall Algorithm

At any time during the processing of a narrative, the interpreting agent maintains a *working store* that consists of a partial interpretation I, a set of constructions and metaphors that are yet to be integrated into I, and a set of *bound schemas* S.

For ease of implementation, I will use a local *working store*. In a connectionist implementation, the working store would be distributed over the entire knowledge-base and would represent the overall pattern of activity at any stage of the interpretation.

The procedure *interpreting-agent* illustrates *one cycle* of the overall algorithm. The input text is first interpreted in the *context* of the current state. The result of the interpretation may activate new schemas. The schemas execute and may add to the local state by changing the degree of belief of different constructions, or modifying the set S. When the input is exhausted, the contents of the working store is the final representation of the story.

- **procedure:** interpreting-agent(input)

  working store: A interpretation I, a set of bound Schemas S, and a set of unintegrated constructions and metaphors).

- **Steps**
  1. Interpret new input into I using the working store contents and the input.
  2. Bind to schemas in S by filling slots of the relevant binding structure or to new schemas s where $S = S \cup s$.
  3. Activate bound schemas in S.
  4. Extract new bindings from schema execution.
  5. Modify working store.

The procedure *Interpret* is discussed in Section 4. Interpret returns a set of bindings to schemas. Schemas are discussed in Section 6. The activation and extraction processes are discussed in Section 7.

At the end of the cycle, the interpreting-agent has a new local state (working store contents) that results from processing the current input. When all the input is exhausted, the contents of the working store corresponds to the final interpretation of the input narrative.

2.6 An illustrative example of the interpretation algorithm

Consider the narrative

> France fell into recession. Germany pulled France out.[5]

Figure 2 shows how the algorithm works for this example. It is expected that the interpretation will work in the following manner. A trace of the expected processing sequence is reproduced in Appendix B. The main points in processing the example are:

1. The first input sentence triggers the *interpret* procedure with the initial context that consists of the accessed *Fall* event verb construction shown in Figure 3. The valence slots of the construction specify the typical slot fillers of the various roles. Also accessed are the frequently used metaphors Country ISA person and Recession ISA hole. The metaphoric interpretation is a better fit to the input given the slot constraints. The result is to bind the schema that corresponds to a person falling into a hole. At this stage, the interpret procedure returns and the bound schema begins to execute.

2. Active schemas execute and return more complete binding structures. The *fall into* schema encodes the fact that the faller is in the hole during and at the end of the fall. This results in the bound schema that corresponds to the faller being in the hole. The parent construction corresponding to the *be* verb is also accessed. Since it is already bound, it is added to the interpretation.

3. During execution, schemas may **thread** into other schemas. Threading allows inter-schema activation. This may bring the threaded schema into the working store. The threaded schema

[5] Of course, this happened in the distant past.

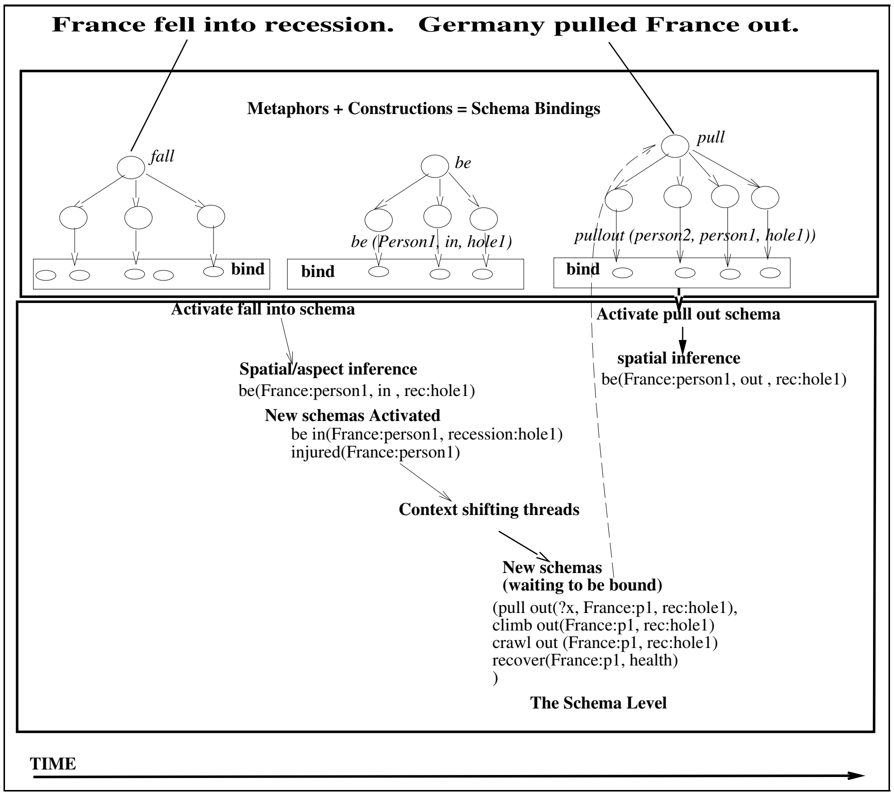


Figure 2: An illustrative example of the interpretation algorithm

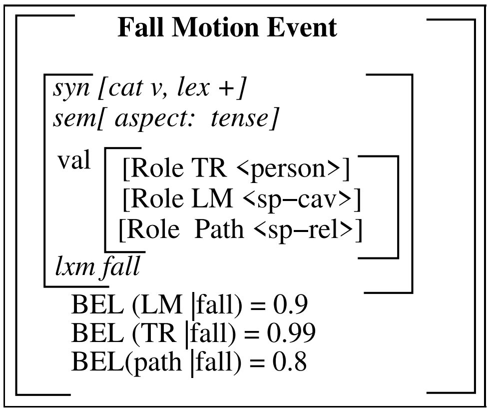


Figure 3: The **fall** event construction.

may or may not access the parent construction depending on strength of activation and prior frequency of access. If the construction is accessed, it is present in the working store and can be integrated with new input. In this case, the three partially bound schemas, climbout, crawlout, and pullout are accessed and added to the working store. The contents of the working store at this stage represent the interpretation after the first input.

4. When the second input sentence comes in, there is no need to run the access function, since there is already a match in the working store. The construction *pull* is now integrated and selected. The anaphoric inference is automatic, since the relevant slot is already bound. The **pull out** schema is bound and active. The returned binding structure asserts that the faller in the first sentence is not out of the hole. Projected onto the target, this implies France is out of recession. Schemas that are inactive timeout and suspend execution. At this stage, the contents of the working store contain the interpretation:

```
fall-event1(France:person1, into1, recession:hole1),
be-1(France:person1, in1, resession:hole1)
pull-event1(Germany:person2,France:person1, out1, resession:hole1)
be-2(France:person1, out1, resession:hole1)
```

At this stage the narrative interpretation is complete.

Future sections will discuss the details of each of the issues above.

2.7 Current Status

The proposed approach to modeling the on-line narrative interpretation system is proceeding along the following dimensions. Wherever a specific section contains additional information on a topic, I indicate the section number while describing the topic.

- **Identification and representation of the set of conventional metaphors**. I am currently designing the interpretation system using the *event-structure* and the *fluid flow* models as these are well motivated and fit a large portion of the data.[6] A few examples of different metaphors that I have identified from the data can be found in Appendix A.
- **Investigation of the use of metaphors to guide on-line interpretation**. I am now implementing metaphoric integration and selection into a probabilistic version of SAL. To do so, I propose a theory of metaphor integration and selection. Details can be found in Section 3. The interpretation algorithm s described in Section 4. An example is described in Section 5.
- **Representation of schemas**. Apart from the general requirement of being able to model states (with real variables) and transitions (including hierarchical transitions) we have identified some specific requirements that pertain to modeling concurrent activation and execution of multiple schemas. These specific features are outlined below.

[6]Our database consists of English language newspaper stories from around the world. We have about 2000 2-3 sentence stories in our database.

1. Should be able to distinguish *enabling* of actions from their *execution*.
2. Should support specific process primitives such as *temporal order*, *enabling and disabling*, *conflict*, *choice/nondeterminism*, and *covarying actions*.
3. Should be implementable as a connectionist style of computation (ie. should be based on finite state machines, and should use a generalized spreading activation mechanism for inference).

The requirements outlined above suggest a net based model of process semantics. The current model uses a variant of high-level Petri Nets [Nielsen1981]). I have done preliminary investigation of such a model to model aspectual distinctions made by different languages. The work on modeling aspect can be found in [Narayanan1995]. Schemas are discussed in Section 6 along with a proposed model. Section 7 discusses the modeling capabilities of Schemas.

- **The Schema Communication**. This includes schema activation and extraction. Section 7 discusses the proposed approach. This is the main area of current research.

3 Metaphoric Unification

3.1 Introduction

Modeling the access and selection of grammatical constructions formed the nucleus of Dan Jurafsky's [Jurafsky1992] Ph.D. thesis work. While incorporation of such linguistic knowledge is a necessary component of any interpreter, it tends to underconstrain the set of possible interpretations, especially those that require a world model for disambiguation. On the other hand, researchers who have focussed solely on "real-world knowledge" representation [GL1990] have yet to demonstrate a system capable of tractably expressing the linguistic richness and regularity observed in human language.

The model of interpretation proposed here attempts to provide a uniform processing model where both types of constraints can be represented and their joint effects on interpretation studied. In recent work on modeling construction access and selection [JN1995], we viewed the human interpreter as a decision maker who decides to unify a constituent (filler value) to a *role* or *site* (a slot in an instantiated construction) if the binding results in a *reduction in uncertainty* of the resulting interpretation. Here the agent makes a choice between two interpretations based on the degree to which an interpretation converges to an predictable world model.

In the current proposal, I extend the basic idea in the following ways:

1. I claim that metaphors are intermediate structures that may *integrate* linguistic constructions to an agent's world model. The proposal to follow outlines an architecture where we can quantify the utility of a metaphor in a specific input context as well as be able to integrate metaphoric knowledge in an on-line interpreter.

2. I claim that an agent's model of the world is *active* and arises from a continuous interaction between input and memory. In our work we have referred to this dynamic model as an *schema*. One could view the successful indexing into a known schema as a plausible interpretation.

This section examines the theoretical basis and modeling approach to investigate the first idea. Subsequent sections focus on schema representation and usage and propose a model of the interaction between linguistic knowledge and commonsense knowledge in narrative interpretation.

Before proceeding further, it may be helpful to state some basic assumptions behind this proposal.

3.2 Basic assumptions

1. Metaphors can be selected by linguistic constructions.

2. Metaphors facilitate the unification of different constructions.

3. A class of metaphors (which we will refer to as *basic metaphors*) map static constructions to dynamic schemas.

The model we will be using is a probabilistic version of the construction grammar interpreter. The interpreter is described in detail in [Jurafsky1992]. The probabilistic version is described in [JN1995].

The basic ideas that carry over to the current discussion are outlined below.

1. Constructions are accessed, integrated and selected probabilistically. The access process retrieves constructions from long-term store, and the selection process integrates the accessed constructions dynamically where roles are bound to fillers if the result reduces the uncertainty in explaining the input sentence.

2. The current implementation of the interpretation algorithm dynamically constructs and evaluates an *Influence diagram* [Howard1981]. The orthographic input is the *evidence*, explanatory constructions are modeled as **belief nodes**, and potential *role-filler* binding sites are modeled as **decision nodes** in the developing network.

3. The decision to bind two constituents is determined by a value function. When all decision nodes are instantiated, the the highest valued (Maximum Expected Value (MEV)) structure is the selected interpretation.

The value function mentioned earlier uses the idea of mutual information and is similar to previous work in quantifying selectional restrictions. I will give a brief overview of the background work that I draw from.

3.3 Quantifying role-filler selectivity

There are many ways to quantify the idea of the selectivity of a role for a filler. In this work, I propose to use a measure based on computing the additional information that different *role-filler* pairs provide about the situation being described.

Mutual information is a standard measure used to model this idea [Cover1991]. A variant of the mutual information measure was used by Resnick [Resnick1993] to quantify the selectivity of different *predicates* for the type of their arguments. He used his technique to predict possible arguments given the occurrence of a specific predicate. Using Wordnet [Word], Resnick demonstrated how this idea could be used to quantify the idea of *Selectional Restrictions* by measuring the degree of *fit* between predicates and their arguments from predefined Wordnet groups.

Resnick called his measure the *Selectional Association* (SA) between a predicate and its argument. Before proceeding further, we need to define these quantities.

Definition : 1: Average Mutual Information

The average mutual information $I(X;Y)$ between two constructions X and Y with a joint probability mass function $p(x,y)$ and marginal probability mass functions $p(x)$ and $p(y)$ is the relative entropy between the joint distribution and the product distribution $p(x)p(y)$.

$$I(X;Y) \stackrel{\text{def}}{=} \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{1}$$

Definition : 2: Selectional Association

The Selectional Association [Resnick1993], $A(p_i, c)$ between a predicate p_i and an element of the class c is the *conditional mutual information* provided about the class c by the occurrence of the predicate p_i.

$$SA(p_i, c) = \frac{1}{\beta_i} P(c|p_i) \log \frac{P(c|p_i)}{P(c)} \tag{2}$$

where the normalization is based on the overall selectivity of the predicate p_i. This ensures that the predicate-argument selection measure is insensitive to the specific strength with which the predicate selects overall.

$$\beta_i = \sum_c P(c|p_i) \log \frac{P(c|p_i)}{P(c)} \tag{3}$$

3.4 An illustrative example

Let us use Resnick's measure to measure the association between the action-verb construction *open* and the construction *door*, a possible direct object.

Using definition 2 and ignoring normalization, the SA value is

$$SA(open, door) = P(door|open) \log \frac{P(door|open)}{P(door)} \tag{4}$$

Consider the following boundary conditions.

1. If *door* only occurs when *open* does, then it is completely predicted by *open*. The mutual information then becomes

$$SA(open, door) = \log \frac{1}{P(door)} \tag{5}$$

   Since $P(door)$ is a very small value $SA(open, door)$ is a very high value.

2. If *door* occurs completely independently of *open* (open cannot predict door), the mutual information then becomes

$$SA(open, door) = \log(1) = 0. \tag{6}$$

3. If the occurrence of *open* inhibits the occurrence of *door*, we get a small negative value for the mutual information statistic.

3.5 Issues with using the SA measure in on-line interpretation

One problem with using the SA statistic is its tendency to overfit the data. For instance, if the highest class for open is door, then the measure would tend to restrict the type of object argument of the predicate open to be of type door. This is clearly unsatisfactory since open can potentially be used with numerous other arguments (such as open a lecture) which clearly do not satisfy this restriction.

Our approach to solving this problem is based on the following observations.

1. While the overfitting problem is severe where trying to characterize selectional restrictions for a role, it could still be used to *rank* different possible bindings in a specific context.

2. Several of the possible predicates or arguments to predicates are linked by conventional metaphors [Lakoff1987], [LJ1980]. For instance, there are conventional metaphors that link the start of a process to the opening of a container [Martin1990]. Once these metaphors (or static correspondences) are accessed, the common inference, namely, *allow access*, can apply to both situations. Including such metaphors would reduce the number of possible word sense clusters and thereby impact the overfitting problem.

The work outlined in this report attempts to implement these solutions. In summary, since our job is restricted to return the *best* interpretation *given* the input constraints, our solution implicitly addresses issue 1. Our approach to implement issue 2 is dealt with in the next section.

A second problem with the SA statistic is that it is context free. The preference of a role for a filler is usually dependent on the dynamic context in which the binding takes place. For instance, the filler or role in question typically explains some other part of the input.

Our solution to this problem is to weight the SA statistic by a context sensitive prior that captures the plausibility that a specific argument may be present in order to explain some other part of the input. This allows the selection measure to depend on the overall interpretation, rather than be statically computed and set for a predicate. This requires us to construct a *joint probability distribution* of several variables. A compact way of representing such a joint distribution is to use **belief networks** [Pearl1988]. Belief networks are compact because their topology reflects the independence relationships amongst the different variables. Belief networks also allow distributed inference and an asynchronous control. This makes them an attractive representation for modeling narrative interpretation.

Our technique of combining external constraints (such as resource constraints and the role-filler SA measure) with standard belief updating (such as revision of beliefs in the presence of new evidence) uses *decision networks* [Howard1981]. The main intuition is that the interpreter considers all the input evidence as well as its own SA measure and any resource constraints in making a *decision* to bind to a schema. This results in the most informative binding given the input *and* the background context (set of unintegrated constructions).

3.6 Decision networks

As mentioned above, the proposed approach to the dynamic construction and evaluation of interpretations involves the use of *Influence Diagrams* [Howard1981], [Schacter1990]. Influence diagrams are also called *Decision networks* [Russell1995]. I will be using the terms interchangeably. A decision net is a directed acyclic graph with three types of nodes: namely chance nodes, decision nodes, and value nodes, drawn as circles, squares and diamonds respectively. Nodes correspond to propositions and arcs correspond to interaction between nodes. Arcs between chance nodes capture probabilistic relationships. Decision nodes represent a choice among a set of alternatives. Arcs into decision nodes determine time precedence, so that information at the source of incoming arcs is available at the time a decision is made. Given the state of information at the time of a decision, the alternatives selected maximize a utility function normally associated with a value node. Influence diagrams without value nodes or decision nodes are called **Belief Nets**[Pear1988]. Belief nets are used for probabilistic inference in the presence of changes in information. Figure 4 shows an influence diagram with three belief nodes (r, f_1, and f_2), one decision node (representing binding alternatives) and one value node.

3.7 Probabilistic binding

The expected value of the binding (V) is the Selectional Association (SA) between the role and the filler weighted by the prior belief of the role and filler being in the *working store*.

The priors on different constructions in the *working store* captures the agent's degree of belief in these constructions given the input sentence and the context.

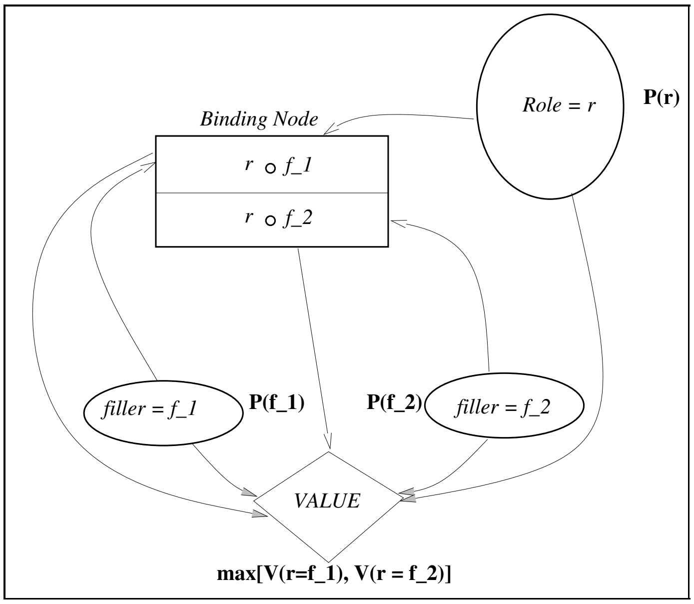


Figure 4: Probabilistic Binding: A Simple Example

The expected value of binding a role r to a filler f is thus:

$$V(r \circ f) \stackrel{\text{def}}{=} BEL(f)BEL(r) \times SA(r, f) \tag{7}$$

BEL(f) corresponds to the agent's degree of belief in the presence of the filler, BEL(r) to the belief in the presence of the role.

Now let us expand this idea to selecting between many possible fillers for a given role. Figure 4 shows the relevant portion of the network that pertains to the binding of a role r to fillers f_1 or f_2. The binding node is a *decision node* whose domain consists of the two possible *role-filler* combinations, namely $[(r \circ f_1), (r \circ f_2)]$.[7]

The prior probabilities of the various fillers are indicated alongside the nodes in the network. These will weight the value computation. In general, BEL(f) and BEL(r) will be computed by standard belief net algorithms, given the input. The value node computes the Information value of a *role-filler* pair. The two possible bindings have the value

$$V_i = V(r \circ f_i) = BEL(r)BEL(f_i) \times SA(r, f_i) \tag{8}$$

where i varies over the possible filler nodes.

In selecting a binding, the following options are available.

- The value node returns the highest value binding (corresponding to the maximum V_i). This is the *Maximally Informative Binding* (MIB) for the *role-filler* pair, assuming that different role-filler pairs are conditionally independent.
- As we shall see, this assumption that the different *role-filler* pairs are conditionally independent is flawed. In language, the integration and selection of role-filler bindings is likely to depend on previous bindings and consequently will affect the value of past and future bindings.
- The specific calculation of the expected value in case of dependencies between bindings is the *degree of fit* between the input and the interpreter's world model or *schema*. The overall value of a specific set of bindings is the degree to which making those bindings can instantiate a schema. This returns the *Maximally Informative Interpretation* for the input sentence.

3.8 Probabilistic bindings with metaphoric unification

Metaphors are intermediate structures that help generate maximally informative interpretations of the input. Our theory of probabilistic unification naturally allows metaphors to be integrated if the resulting interpretation is a better *fit* to the world model. The degree of fit with and without

[7] We also consider a third possibility that the role is left unbound in the final interpretation. This is expressed by a special variable, *pending* that is in the domain of all decision nodes. Unless otherwise mentioned, we will assume this possibility for the rest of the discussion.

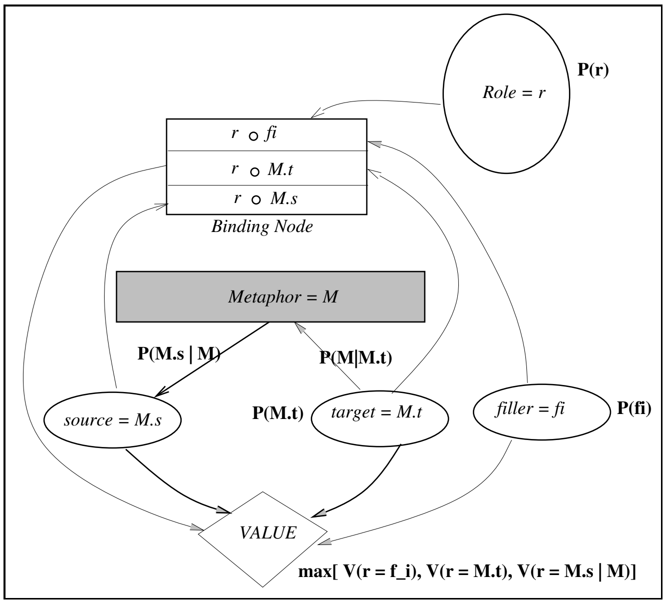


Figure 5: Probabilistic Bindings With Metaphoric Unification

metaphoric integration determines whether the metaphor is selected in the final interpretation. In this section, we will lay the theoretical framework for this claim.

Figure 5 depicts the general case of a single r, with three potential fillers, one of which results from the selection of a metaphor.

For the particular unbound role in question (r), there are three possible fillers ($M.t$, $M.s$, and f_i). $M.t$ refers to the target concept that has been accessed from the input. $M.s$ refers to the source type of the Metaphor M. If $M.t$ accessed the Metaphor M (with a prior belief, $BEL(M|M.t)$), the source concept is available for selection with a prior belief $BEL(M.s|M)BEL(M|M.t)$. We will refer to the metaphor as *non leaky* if $BEL(M.s|M) = BEL(M.t|M) = 1$.

The filler f_i *could be* the direct binding of the target type $M.t$ to the role r, the metaphoric binding that results from the selection of M, namely the binding of r to the source $M.s$, or could be any other construction (such as f_i).

In Figure 5, metaphor M has been accessed from the target domain. While this is the default case, in general the architecture proposed here can easily deal with the case where the metaphor is accessed from the source domain. This will happen when the ongoing context suggests the metaphor M. Here $BEL(M|M.s)$ will be different then that from the target $BEL(M|M.t)$. So while metaphor access is bi-directional, the actual belief value depends on the direction of access.

The expected value of binding the different fillers to the role in question are:

$$V(role \circ M.t) = \beta_f^{-1}[BEL(M.t) \times [BEL(M.t|r) \log \frac{BEL(M.t|r)}{BEL(M.t)}]] \tag{9}$$

$$V(role \circ M.s) = \gamma_m^{-1}\beta_f^{-1}[BEL(M.t)BEL(M|M.t) \times [BEL(M.s|r)\log\frac{BEL(M.s|r)}{BEL(M.s)}]] \tag{10}$$

$$V(role \circ f_i) = \beta_f^{-1}[BEL(f_i) \times [BEL(f_i|r)\log\frac{BEL(f_i|r)}{BEL(f_i)}]] \tag{11}$$

where:

$$\gamma_m = \sum_m BEL(M_m|M.t) \wedge \beta_f = \sum_i BEL(f_i|r)\log\frac{BEL(f_i|r)}{BEL(r)} \tag{12}$$

γ_m normalizes the set of metaphors for a specific construction (here $M.t$), making the measure insensitive to the overall metaphor access strength of that construction. β_f is the same normalization measure used in the definition of SA (definition 3). To make the exposition more readable, we will use these normalization constants in future equations only when it is relevant to the discussion.

Equation 10 has an additional factor that $BEL(M|M.t)$ that is the conditional belief in the metaphor M given the target concept $M.t$. This is obvious from Figure 5 since the degree of belief for the source $M.s$, is

$$BEL(M.s) = BEL(M.t)BEL(M|M.t)BEL(M.s|M) \tag{13}$$

If we assume a *non-leaky* metaphor, $BEL(M.s|M) = 1$ and we get equation 10.

If we assume that the only possible bindings are with and without the use of a Metaphor M (assuming only possible fillers are $M.s$ and $M.t$), we can compare the additional information that a metaphor provides (the utility of using the metaphor) in the situation of interest (interpreting the specific input sentence).

Added Information Value Of the Metaphor M is (Metaphoric Utility (MU))

$$MU_M = \frac{V(role \circ M.s|M)}{V(role \circ M.t)} = \frac{BEL(M|M.t) \times [P(M.s|r)\log\frac{BEL(M.s|r)}{BEL(M.s)}]}{[BEL(M.t|r)\log\frac{BEL(M.t|r)}{BEL(M.t)}]} \tag{14}$$

Rewriting this in terms of the definition of Selectional Association (SA) (definition 3), we get:

$$MU_M = \frac{V(role \circ M.s|M)}{V(role \circ M.t)} = BEL(M|M.t) \times \frac{SA(r, M.s)}{SA(r, M.t)} \tag{15}$$

Let us look at the various boundary conditions to understand how the MU value determines the bindings.

- **Case 1:** $MU_M = 0$

Interpretation: *Sentence can be interpreted only in the target domain.*

In this case, *either* $SA(r, M.s) = 0$ in which case, the role r does not select the type of $M.s$, *or* $BEL(M|M.t) \rightarrow 0$ in which case, the metaphor is not selected.

Hence the value $V(role \circ M.s)$ is 0. The only possible binding is $V(role \circ M.t)$. This is the standard non-metaphoric or literal interpretation that is modeled by most computational theories of on-line interpretation.

- **Case 2:** $MU_M \rightarrow \infty$

  Interpretation: *Only source can be selected.*

  In this case, the target concept has no association to the construction and all meaning is acquired from using the metaphor map to produce the filler ($M.s$) for the role r. In this case, $SA(r, M.t) = 0$. So the only possible binding is one that results after the application of the metaphor. If this condition is true for all the predicates in a target domain, then we could conclude that the target is completely structured by the source domain. Some of Burstein's [Burs1983] examples exhibit this character and may be good test examples for our hypothesis.

- **Case 3:** $MU_M = BEL(M|M.t)$

  Interpretation: *Target Domain and Source Domain may fill role with equal felicity.*

  In this case we have the following relations.

$$SA(r, M.s) = SA(r, M.t) \tag{16}$$

  Given this the preferred binding is $r \circ M.t$, since $BEL(M|M.t) \leq 1$, hence $MU \leq 1$ and $V(r \circ M.t) \geq V(r \circ M.s)$.

  Hence, *cetirus paribus*, the model gives more weight to the target domain and if a target domain interpretation is felicitous, it will be selected. This extends the *Specificity principle* [Wilensky1983] (which states that all other things being equal, select the more specific interpretation) to metaphoric unification as well.

  For instance, we expect our interpreter to return the correct interpretation for the sentence:

  Example 3: McEnroe killed Connors.[8]

  Here the tennis domain bindings *tennis-player tennis:kill tennis-player* will be preferred over *person kills person.* [9] In order for the latter interpretation to work, the metaphor *Game ISA War* has to invoked with the appropriate sub-categories

  1. *Tennis:kill ISA Kill,*
  2. *Victor ISA Killer,*
  3. *Defeated ISA Killed.*

  This results in a lower valued binding since each probability ≤ 1.

[8] Taken from [Martin1990].

[9] We have not yet seen how the two different senses of kill become active given the domain and the argument types of the *kill* schema. This is part of future work, and is assumed in the current discussion.

4 The Interpretation Process

For the purposes of this section, we shall restrict our consideration of the Interpretation process to single sentences. Our input will be the set of accessed constructions and metaphors (the contents of the working store).

4.1 The Interpretation algorithm

Given the contents of the working store, the interpretation algorithm is responsible for the dynamic construction and evaluation of an influence diagram that corresponds to the developing interpretation. Outlined below is a functional description of the algorithm.

- **Procedure**: *Interpret (constructions, metaphors)*
  - **Steps:**
    1. Construct decision net.
    2. Evaluate net, with the input as evidence, sequentially instantiating the binding nodes and updating probabilities of all parents to the value node.
    3. Select the set of assignments that results in the maximum expected value for the binding choices.
  - **Output**: The best set of bindings.

4.1.1 The construction procedure

1. For every construction in the working store, create a binary valued **belief node** that represents the belief that the construction is present in the interpretation.

2. For slots in accessed constructions, create a link from the construction to the slot node (a binary valued **belief node**). Set the link probability to be the conditional belief that the slot is present given the construction. Figure 6 shows the translation from constructions to the equivalent decision net for the *fall motion event* construction.

3. For every unbound slot in a construction, add a decision node and a link from the relevant slot to the newly created decision node. Set the domain of the decision node to the special value **pending**. Figure 6 illustrates this construction process.

4. If the construction is accessed directly by the input then it provides a direct explanation for the input. Correspondingly, there is a causal link from the explanatory construction to the input. The conditional probability table of the input node reflects the *explanatory model* of the accessed constructions. Note that while the network is constructed as a *causal model*, inference goes the other way, evidence arrives and constructions are selected based on their ability to explain the evidence.

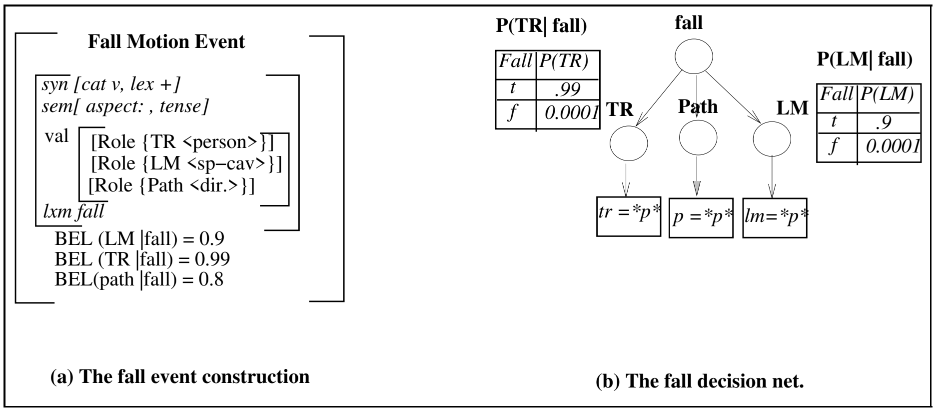


Figure 6: A construction and its associated decision net representation

5. For every metaphor that is accessed by a target (source) construction construct the corresponding source (target) node, with the a prior belief given by $BEL(M.t)BEL(M|M.t)$ (in case of a target-based access). Figure 7 illustrates the metaphor *Country ISA Person* and the corresponding decision net representation.

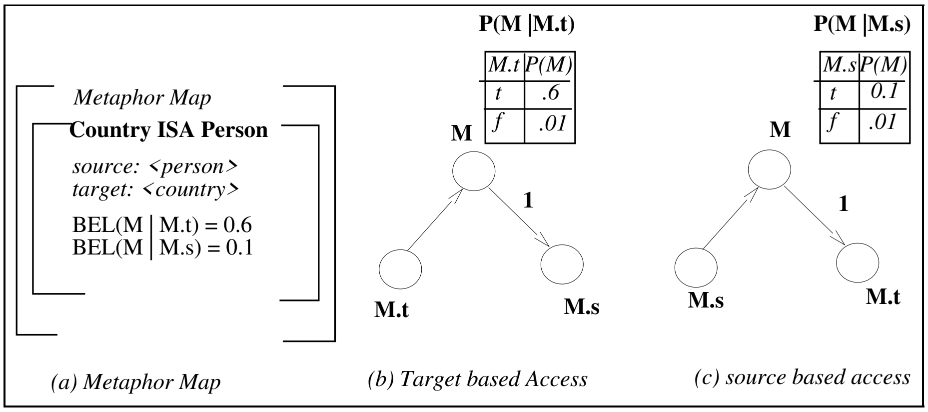


Figure 7: The metaphor *Country ISA Person* and the corresponding decision net for both target and source based access.

6. For every instantiated construction (including those created in step 2) in the working store, create an arc directed to the newly created decision node (in step 1). The domain of the decision node is now incremented to allow for the binding $R \circ c_i$ where R is the unbound slot responsible for the creation of the decision node, and c_i are the set of possible filler constructions.

7. Create a *Value* node with incoming links from all decision nodes and from all *role-filler* nodes identified in steps 1 and 2.

4.1.2 Evaluation

1. Set the input nodes to be the evidence.
2. For every instantiation of the binding nodes
   (a) Calculate the posterior probabilities of all the role and filler nodes in the network.
   (b) Calculate the MEV for the binding choice.
3. Return the bindings $r_i \circ f_i$ with the highest value.

4.1.3 Discussion

Steps 1 to 4 of the network construction process similar in spirit to the work of Charniak and his students [Charniak1988], [Charniak1993], [Charniak1994] in dynamic belief net based story understanding. However, the addition of metaphors and constructions, and the use of the decision net make the algorithm quite different from these attempts. Our interpretation algorithm also takes a step toward formalizing the concept of interpretation as abduction that has characterized much of Jerry Hobbs's work [Hobbs1990].

Our algorithm has been implemented for the non-metaphoric case representing bindings as a **belief node** and not as a decision node. The need to include resources and other external constraints resulted in the current proposal of viewing bindings as decisions made by the interpreting agent. [10] We finesse the issue of exactly how often the procedure *interpret* is called, by assuming that the procedure is called every time a set of constructions corresponding to an input phrase is accessed. The exact details of the control mechanism (when to update the dynamic net construction algorithm) are yet to be worked out and any serious model of timing details with metaphoric integration awaits psycholinguistic research on on-line access and selection of metaphoric knowledge.[11]

In the evaluation phase, we use standard belief net inference algorithms for step 2 [Pear1988]. Note that once the decision nodes are instantiated, all cycles are blocked, leaving a virtually singly connected network. This allows for the inference algorithms to be local, and linear in the size of the network. We also note that in the presence of multiple binding nodes, the number of polytrees is exponential in the number of bindings (each binding node is a cutset). This suggests that accumulating unbound slots makes the evaluation phase intractable. Such a result is compatible with research in reflexive reasoning [Shastri1990], and provides an intriguing avenue for further exploration.

While returning the maximum valued interpretation requires us to exactly evaluate all possible interpretations, preliminary results suggest that evaluating a few of the most promising interpretations and setting fairly loose tolerances for the actual probability estimation is sufficient. This

[10] The current implementation makes use of the IDEAL package, which consists of a set of Common-lisp routines for influence diagram evaluation.

[11] In the non-metaphoric case, there is extensive psycholinguistic data that can be used to model the timing of various interpretation process. We included some of the resource constraints in our non-metaphoric version of the probabilistic interpreter. However there is very little psycholinguistic data in the metaphoric case and hence the simplification here.

suggests that stochastic methods such as maximum likelihood sampling, or approximate methods such as bounded conditioning may work fine. Stochastic sampling has the additional attractive property that continuous variables (such as binding the force parameter in a *push* construction) are felicitously handled.

We believe that our architectural framework allows us to investigate the the computational impact of cognitive constraints such as working memory resources, garden-path effects and delayed binding in a systematic and formal way that makes our approach more attractive than previous attempts at modeling on-line narrative interpretation.

5 An Example Of the Interpretation Process

To illustrate the interpretation process, we will use the example

Example 1: *Brazil fell into recession.*

We will then look at the interpretation generated by the proposed version which incorporates Metaphoric Unification and integrates it with the probabilistic construction unification process. In the example, I will concentrate on the qualitative aspects of the construction and evaluation process, since the process of updating CPT entries and the inference algorithm used can be found in [Pear1988].

5.1 Input

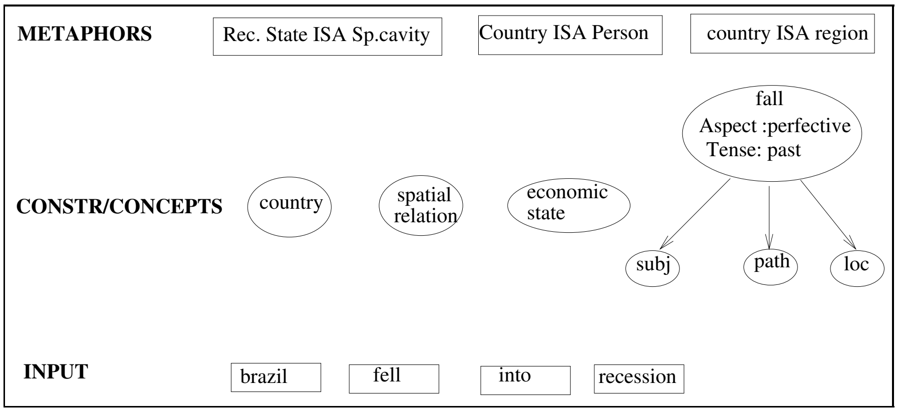


Figure 8: Initial Contents of the Working Store for the input "Brazil fell into recession".

Figure 8 shows the initial contents of the working store which forms the input to the interpretation algorithm. As mentioned earlier, we will assume that the entire phrase has been accessed before selection starts. This may not be the case and the exact timing details of the access and selection process are not modeled.

1. The **fall** construction is shown with three slots/roles (*subject*, *path*, and *location*). The construction and corresponding roles chosen here are for illustrative purposes only, and the actual grammatical construction may have a different structure. The overall algorithm or the arguments here do not depend on this fact.

2. The **fall** construction has additional slots that pertain to the Aspect and Tense of the input. These features affect the dynamic inferencing process. A detailed model of Aspect can be found in [Narayanan1995]. For our purposes in this section, we will simply assume that these are static features that are bound to the values indicated.

3. The inputs "Brazil", "into", and "recession" unambiguously access the *country*, *spatial relation* and *economic state* concepts.[12]

4. Constructions access conventional metaphors. The three accessed metaphors *Country ISA person*, *Recession state ISA Spatial cavity* and *Country ISA region* shown in Figure 8 are some of the conventional metaphors in the target domain (Economics). Each domain has a store of such conventional metaphors that can be accessed by specific constructions.

5.2 Dynamic Network Construction: Details For The Example

Figure 9 shows the interpretation after the application of step 6 of the *construction* procedure to the input in Figure 8.

Regarding the network in Figure 9, the following points should be noted.

1. An accessed metaphor allows **both** the source and the target to be bound as constituents to other constructions in the working store (as mentioned in step 4).

2. In general, there will be several competing metaphors ($M_1 \cdots M_n$) that will be accessed. An example of such a competing set in this case is *Country ISA Region* and *Country ISA Person*. Each such metaphor provides a a potential new binding choice (the source domain value). For instance, the subject of the *Fall-action* construction could be bound to the type *region* or to the type *person* that result from the selection of the two metaphors shown.

3. In general, a filler could potentially attach to multiple roles (the actor and patient in a *give* scenario). This is illustrated by having multiple links from a single filler constituent to the binding nodes for multiple role constituents. In our example the construction *region* could be both a location and possible subject of the fall construction.

5.3 Selecting an interpretation: The simple case

One strategy for selecting bindings is to select the maximally informative value pair for each individual binding node. In the simple case, the maximum value is calculated for each binding independently. The possible binding choices are the following

[12] For the purposes of this discussion, we assume that the lexicon contains the knowledge that Brazil is a Country. In general, this knowledge is handled by a type hierarchy in the domain(s) of interest. Similarly, we will assume that the input word token "recession" has been identified with the type "Economic state".

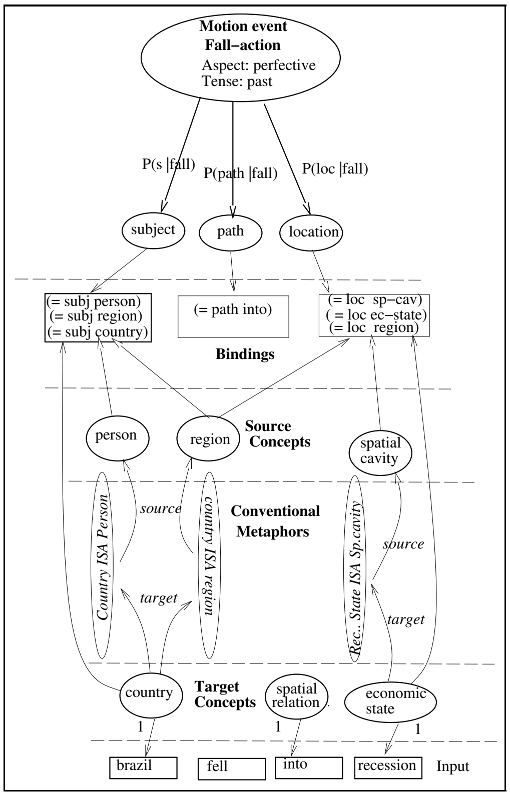


Figure 9: Evaluation Stage (after step 6) for the Input "Brazil fell into recession". Value node is not shown for clarity.

Role	Filler1	Filler2	Filler3
Subject	person	country	region
Path	into		
Location	economic state	spatial cavity	region

Any combination of choices is an acceptable binding. One of the 9 possible binding sequences results in the maximally informative interpretation. Figure 10 shows a possible interpretation structure for the same example outlined earlier.

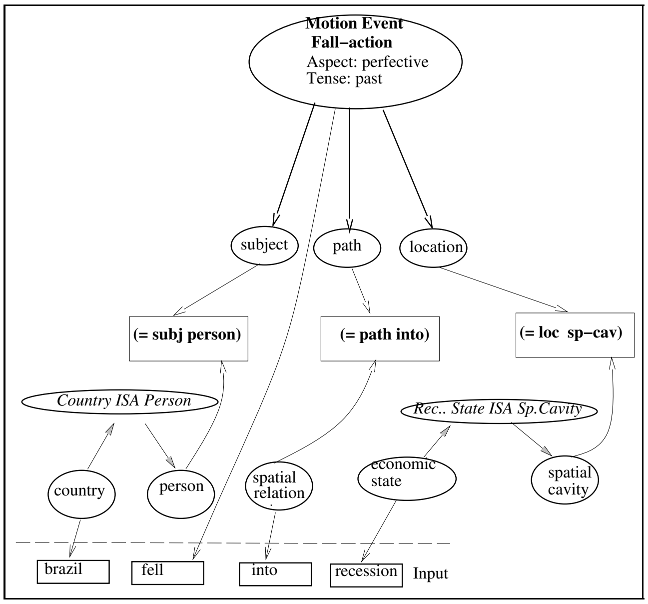


Figure 10: Final Interpretation of the input “Brazil fell into recession”.

With the simple strategy, the individual bindings are independent of each other and the *any* maximum value *role-filler* pair can be selected as being in the final interpretation. Thus for the example above this translates to an interpretation that satisfies the following three constraints:

$$\begin{aligned} fall_{subj} &= arg\max_f[V(fall_{subj} \circ f)] \\ fall_{path} &= spatial - rel(into1) \\ fall_{loc} &= arg\max_f[V(fall_{loc} \circ f)] \end{aligned} \tag{17}$$

The interpretation in Figure 10 will be returned if the MU measure for the metaphor *Country ISA Person* and that of the metaphor *Economic State ISA Spatial Cavity* is greater than the utility of other possible metaphors, such as *Country ISA region.*

5.4 Putting it together: considering how the interpretation plays

In general, the value is a function of the values for the individual bindings. Figure 11 shows how the value computation for the *maximally informative interpretation* may proceed. The basic idea is to have the overall value computation depend on the set of available schemas and the various individual binding constraints. I assume that each schema has a node for the appropriate theta-roles and other required slots for activation.

One possible design (shown in Figure 11) is to have the corresponding theta roles pose queries to the developing interpretation. An important class of such queries are *conjunctive queries* of the form:

$$q = \wedge_{r_i \in R, f_j \in F}(r_i \circ f_j) \tag{18}$$

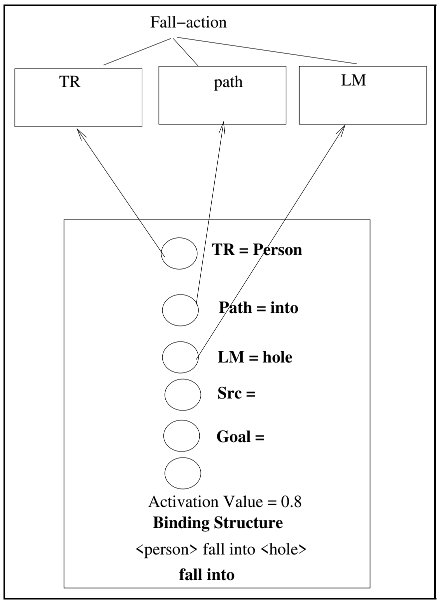


Figure 11: Binding constraints are conjunctive queries to the Decision net.

To compute the information value of a conjunctive query of length m, we can use the chain rule to expand the query to be the product of m belief measures, computing each such measure by evidence propagation.

$$q = (r_1 \circ f_1) \wedge (r_2 \circ f_2) \wedge (r_3 \circ f_3) \tag{19}$$

we can use the chain rule to expand this to:

$$V(q) = V((r_3 \circ f_3)|(r_2 \circ f_2), (r_1 \circ f_1)) \times$$

$$V((r_2 \circ f_2)|(r_1 \circ f_1)) \times V(r_1 \circ f_1) \tag{20}$$

Each of the factors is equivalent to the value of the belief in the proposition $r_i \circ f_i$ given that all the preceding bindings $(r_{i-1} \circ fi - 1 \cdots (r_1 \circ f_1))$ are evidence. Thus the product of the 3 belief measures yields the value of the overall conjunctive constraint. Note that in the evidence propagation step [Pear1988], the impact of each instantiation (the $\lambda - \pi$ messages) are transmitted only to the variables included in the query.

For the example, we had three bindings. Figure 11 shows how the schemas are indexed by the predicate *fall into*. Each such schema executes when a specific set of parameters get instantiated. For instance, the binding (subject = person, location = hole) starts executing the schema that simulates *person falls into hole*.[13]

In general, several bindings are possible, since any combination which can activate a schema is selected as a possible interpretation (some thresholding may be required). The total activation value of the relevant schema is the total value of the various relevant bindings (probably a weighted average, weighted by the prior frequency of execution of the relevant schema). Since many schemas may execute simultaneously and asynchronously, there are some special design considerations that govern how schemas can be represented.

[13]The exact interface and bi-directional interactions between schema space and construction space is currently being worked out.

6 Schemas

Schemas are activated by linguistic input and constrain the set of possible interpretations. Schemas encode dynamic knowledge. When activated, schemas execute by simulating the effects of the linguistic input on the agent's mental model of the world. Executing schemas set up a dynamic context which may condition expectations for the selection of specific constructions and metaphors in the developing interpretation.

In this section, I first try to justify the claim that modeling the source domain of the metaphor can generate useful inferences. I use some recent results from Cognitive Semantics as well as some of my own analysis of the database of newspaper stories. I then outline a possible scheme by which linguistic input maps into schemas and how schemas set up a dynamic context. This involves the use of a special interface structure called the *Binding structure*. Finally, I develop a set of computational requirements for schemas and a possible schema representation scheme.

6.1 Cognitive Semantic Results

In Cognitive semantics, Lakoff [Lakoff1992] has hypothesized that the structure of events is understood metaphorically through mappings to the force dynamic and spatio-temporal domain. The general name for this mapping is the Event Structure Metaphor.

The Event Structure Metaphor consists of the following mappings

1. Abstract States are mapped to Locations.
2. Changes are mapped to motions.
3. Causes are mapped to forces.
4. Actions are mapped to self-propelled motions.
5. Means are mapped to paths to destination locations.
6. Difficulties are mapped to impediments to motion.
7. Expected progress is mapped to a travel schedule.
8. External Events are mapped to large moving objects.
9. Long term, purposeful activities are mapped to journeys.

The Event Structure metaphor is a rich and complex metaphor with several sub-parts. If the main intuition is correct, [14] then by representing the metaphoric mapping and modeling motions in space, we should be able to capture much of the structure of stories dealing with political and economic causation. This requires representing metaphoric correspondences and making use of inferences in

[14] Including Lakoff's hypothesis that the Event Structure Metaphor is cross-linguistic.

the spatio-temporal domain. The last chapter addressed the issue of activating and integrating metaphors into on-line interpretation. In fact, the example we used involved activating a part of the Event Structure Metaphor, namely *Abstract States (Recession) are mapped to Spatial Regions (holes)*. Now that we can perform such mappings, this chapter explores the additional benefits to be obtained by simulating the spatio-temporal domain.

The Event Structure has as its source domain, agent motions in space. There are other domains that researchers in cognitive science [Langacker1987] have identified as basic domains. These include color, orientational and configuration domains, and also basic social behaviors. While the general process of mapping and inference proposed here could be extended to model some of these domains, this discussion will focus mainly on the spatio-temporal domain. Appendix A lists some of the metaphors that seem to recur in newspaper articles from around the world.

6.2 The Binding Structure

In the proposed model, input sentences **bind** into schemas. Bindings to schemas control much of schema execution. The basic mechanisms by which bindings propagate schemas and constructions is controlled by a frame-like structure called the Binding Structure.

The basic mechanism can be stated as follows:

1. Schemas can be **bound** by linguistic input. The binding into schemas is controlled by the posing of conjunctive queries to the developing interpretation as part of the *Interpret* procedure.
2. When bound, schemas may execute. As schemas execute, they interact with the binding structure and may modify it. This allows for certain types of automatic inferences, and generates a dynamic context for the narrative.

In order to start execution, the relevant *schema* must be **bound**. Bindings to a schema occurs when the slots (theta roles) of the corresponding grammatical construction have fillers. In our example, the *Interpret* procedure described in the last section returned the bindings:

Schema name/type	**fall into**
Schema instance	fallinto1
Trajector	Brazil:person1
Landmark	Recession:hole1
Aspect	Perfective
Tense	past

Figure 12 shows the binding structure that is the interface between schemas and constructions. When the construction-specific theta roles have fillers, the corresponding schema is said to be **bound**. In general, the speaker needs to generate only sufficient linguistic input so that the hearer can successfully bind to a schema using constructions and conventional metaphors. The structure

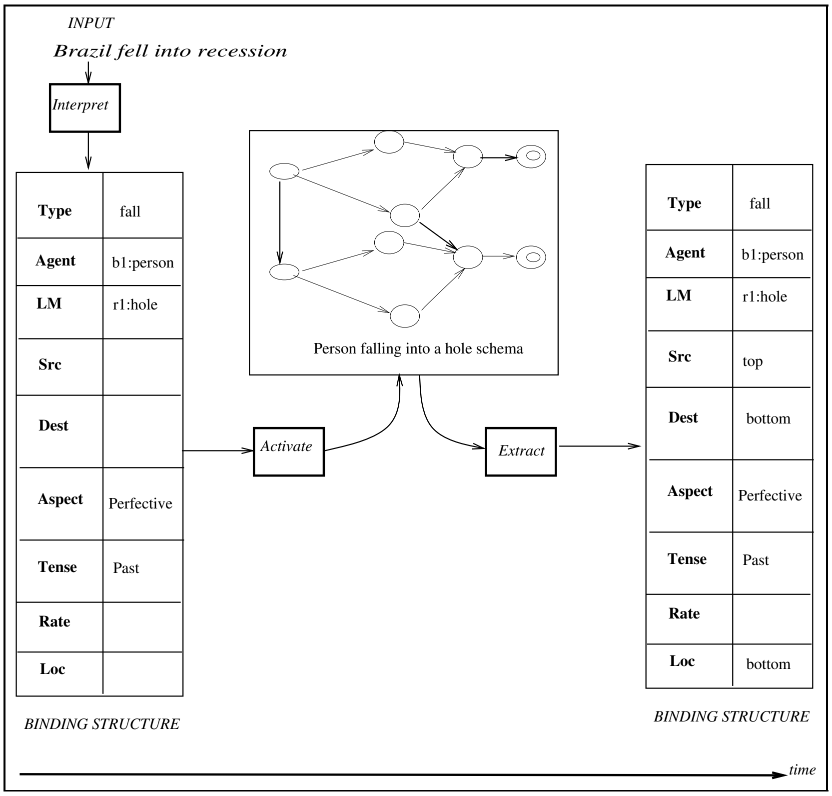


Figure 12: Binding to schemas. The Binding structure before and after schema activation

of the executing schema then allows the hearer to generate additional facts and inferences. For instance, executing the bound *fall into* schema for the input sentence will allow the hearer to conclude that "Brazil was *in* a recession" at the end of the input sentence. This inference is a normal consequence of someone falling into a hole. It is a source domain inference that the speaker is counting on the hearer to make. Hence the hearer is expected to felicitously comprehend the possible next sentence "Brazil started a slow climb out." In the proposed implementation, the inference that falling into a hole implies being in the hole is encoded in the *fall into* schema whose binding structure is shown in Figure 12.

The binding structure is *schema specific*. The binding pattern depends on schema requirements. For instance, in Figure 12, the binding structure for the *fall into* type of schema may contain slots for the rate of fall, the current location of the agent, and the source and goal locations of the agent. Linguistic input need only supply a few fillers (those that are sufficient to bind to a schema). The rest of the structure may get *filled in* during schema execution. In the example shown in Figure 12, executing the schema with the initial set of bindings generates fillers for the source, goal, and location slots of the binding structure. Some slots of the binding structure may not get bound at all or may have a default value (such as the rate slot in this example).

Of course, if the speaker considers it relevant, he may choose to supply values to specific slots of the binding structure. For instance, the speaker may use an adverb like *slowly* to specify the rate of fall, as in "Brazil slowly fell into recession". Note that this information affects the execution of the schema as well as the subsequent instantiation of the binding structure. In interpreting the sentence above, the hearer may well conclude that the location of the faller at the end of the situation described was *in* the but *not* at the bottom. In fact, he may expect the speaker to explicitly specify the location in a possible next sentence such as "World Bank aid cushioned the fall".

The mechanism described above, where inferences are made in the source domain allows for a limited automatic inference capability. The next section describes the set of source domain inferences that seem to project onto the target domain during metaphoric interpretation.

6.3 Inferences that are preserved by metaphoric unification

The following types of inferences are preserved when metaphors map concepts and predicates between domains. My hypothesis is based on analysis of about 2000 newspaper stories. These predictions are considerably more specific (and hence testable) than those of Carbonell [Carbonell1982] who also attempts to categorize the inferential structure provided by the source domain of a metaphoric map.

- **Aspectual inferences** (has fallen versus was falling versus fell versus used to fall, etc.) are preserved. For instance the different entailments of *Brazil was falling into recession* and *Brazil fell into recession* are preserved by the mapping.

- **Spatial Inferences** (through → out - in - out) are preserved. For example the sentence *India went through socialism* allows for the inference that India is now a non-socialist country.

- **Context shifting inferences.** If we assume Griceian postulates of informativeness and relevance, we would expect the speaker/text to explicitly signal any changes in context. In the concrete domain of the event structure metaphor, the standard story may change the default context (moving forward) by an explicit *failure* (stumble, fall, obstacle) and then a recovery (return to default mode) or a continuing failure. In other domains the story patterns assume different context chains which are projected onto the target domain.
- **Force dynamic** inferences (high-force/low-force, effort, intention) are preserved. For example the difference in the sentences *attracted jobs away* versus *sucked out jobs*[15] is projected onto the target domain.
- **Manner** inferences (stumble vs. fall, etc.) are preserved. Consider the difference between *Gold prices rose* and *Gold prices soared.*

6.4 An example of source domain inference: Aspect

Consider the following examples (all in the *past tense*):

1. India was falling into recession.
2. India fell into recession.
3. India had fallen into recession.
4. India used to fall into recession.
5. India was on the verge of falling into recession.
6. India fell back into recession.
7. India almost fell into recession.
8. India started falling into recession.
9. India kept falling into recession.
10. India continued falling into recession.
11. India used to be falling into recession.

It seems that the speaker has a bewildering array of choices to describe the same type of situation to the hearer. Is there any systematicity to these choices, and if how does this transfer to meaning and inference differences for the interpreter? Here are some example inferences we would want the hearer to draw.

[15] As in the famous gaint sucking sound of jobs leaving

1. In the first sentence, the hearer/interpreter *cannot* conclude that India's recession has *bottomed out.* In fact, it is almost as if the speaker *intends* to disallow this inference. Consider how easy it is to interpret the next sentence, *World Bank aid cushioned the fall.*

2. In the case of the third sentence (which uses the perfect aspect), the hearer can confidently conclude that (a) India is in recession, and (b) that the results of falling into recession (such as an injured or weak economy) are available and will be used in future input. So the sentence earlier *World Bank cushioned the fall* fails to be properly interpreted, but the sentence *World bank aid strengthened the fallen economy* works fine.

3. If India started falling into recession, then India must have been falling at the end of the story. So the sentence *World Bank aid cushioned the fall* is acceptable, but leaves the interpreter slightly puzzled[16] as to why the *started* frame was used. However, if the sentence had been *Quick World Bank aid prevented further slide*, the interpreter would be more satisfied.

4. If India was on the verge of falling into recession, the interpreter would be foolish to conclude that India was in recession, since the next sentence in the narrative is *World Bank aid prevented the fall.* The same is true in spades for the sentence, India almost fell into recession.

5. If India used to fall into recession, it fell into recession more than once. In fact, it used to fall into recession *habitually*, maybe every year. So the sentence *World Bank aid cushioned its fall* would have to now be *extended* in time *beyond* its ability to stretch. However, an aspectual word like *always* is sufficient for the interpreter to make sense of the utterance. Consider the next sentence to be *World Bank aid always cushioned the fall.*

The inferences outlined above and other similar inferences have been studied from the days of Panini and Aristotle. Clearly there seems to be some shared strategy for the representation of situations in the world between speaker and hearer that allows the speaker to point to specific temporal *aspects* of the situation and control how much he wishes the interpreter to infer. The examples above demonstrate that the general aspectual structure works across domains, and in the presence of metaphors and other usage constraints. Also, there is a vast amount of literature that shows that a more or less similar structure (with different language distinctions) exists across cultures and languages.

These considerations made Aspect the first problem I looked at in designing schema representations. It turns out that a relatively simple and elegant computational structure can capture much of the inferential capability and entailments described above. The central intuition is that different stages of situations in the world could be different ways of **binding** into schemas. Furthermore, the structure of these bindings reflect the structure of the how situations and processes evolve. By allowing linguistic means to refer to these bindings, the speaker implicitly indicates what kinds of inferences are sanctioned. The resulting structure is a simple *controller* for schemas. The interesting thing is that this simple structure works for all the languages I looked at so far.

[16] At least this one

6.5 Modeling Aspectual Inferences: The control graph binding structure

Controlling schema execution is done from a binding structure called the *control graph*. Executing schemas follow a control graph that consists of specific control statuses. The control graph is shown in shown in Figure 13. The various schema control statuses are nodes in the control graph.

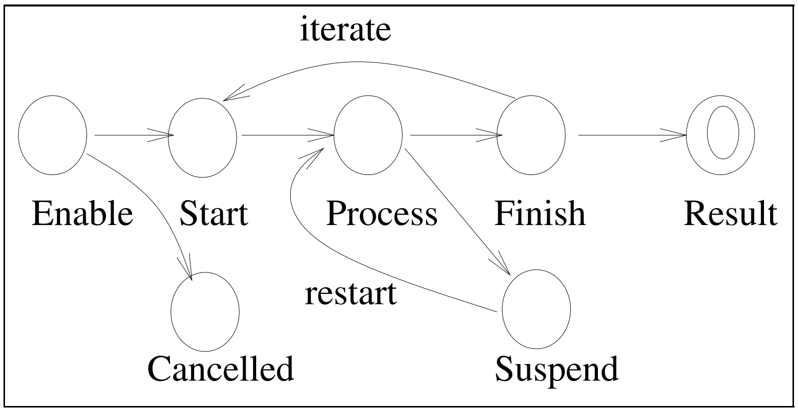


Figure 13: Basic control graph for a schema

The control graph is a reasonable ontology of the various stages in the evolution of situations. Most of the stages are self-explanatory. A behavior evolves in stages that follows some path in the control graph.

We represent the control graph explicitly since it allows us to model linguistic aspect markers (such as the progressive *-ing*) which refer directly to specific stages or sequence of stages of situations. Such markings allow the speaker (text) to *profile* or *focus* on a specific stage or stages of a behavior. This in turn enables the listener (reader) to make certain inferences and to set certain future expectations. Consider the difference between the following two sentences.

1. India had fallen into recession.
2. India was falling into recession.

In the first case the speaker expects that the user would *bind* to the situation described at the *result* stage of the *fall into* schema. This would involve facts like the faller was injured (economy was injured), that the faller was at in the hole (in recession), etc. By specifying through aspect that the hearer is to bind to the *result* stage of the situation is question, the speaker can indicate the kinds of automatic inferences he wants the hearer to make.

Note that following the causal graph, we would conclude that it is inappropriate to say next that *World Bank cushioned the fall.* In order to interpret the new input, the hearer would have to *undo* some already allowed inference, and re-simulate the fall from the previous stage to get the new effects. The control graph does not directly encode this and so for the speaker to say this is just *bad form* since it would involve too much work for the hearer.

In the second case, the expected interpretation would *bind* to the *process* stage of the fall situation. This would automatically allow the hearer to conclude that the economy was in a downfall, the depth of the recession was increasing, that the recession had not bottomed out - though the distance to the bottom was decreasing, etc. By specifying through the imperfective aspect, which part of the *control graph*, the hearer should select, the speaker can very felicitously *slip in* the fact that *World Bank supplied a soft bed of money to pleasantly break the fall.*

The speaker can also select *paths* in the control graph, informing the user to bind to a specific sequence. For instance, the sentence *India used to fall into recession* selects the path *enable-start-process-finish-iterate-start.* The difference between *habitual* aspect like *used to* and a merely iterative one such as India fell into recession again/three times is that the number of loop traversals is unbounded in the first case and bounded in the second.

Constructions like *on the verge of*, *on the edge of*, *ready to*, etc. *bind* to the *enable* stage of the underlying situation. This allows for the inference that the situation was prevented from occurring in some way (if the speaker is cooperating with the hearer).

In the implementation of the causal graph, it must be noted that there is a bi-directional asynchronous interaction (from linguistic input) as well as from schema execution results. Also, the wiring of the control graph by itself is insufficient to generate the inferences, it must interact with the execution of the underlying schema. Since more that one schema may be active, more that one controller needs to be active as well.

The use of the causal graph in making aspectual inferences in different languages has been studied in detail in [Narayanan1995]. The results of that model are directly applicable to the work proposed here.

6.6 Computational characteristics of schemas

Now that we have seen how source domain inferences can establish context and enable limited inference, let us look at some of the basic computational characteristics of schemas that meet the kinds of requirements outlined by the cognitive semantic data as well as the inference requirements.

1. Multiple schemas may execute concurrently. This is because multiple metaphors may be active, or both metaphoric and non-metaphoric interpretations may be active (as in *Stud tires out*).

2. Schema execution proceeds with distributed control. The activation of different schemas is *asynchronous* and the flow of activation within a single schema is locally controlled. Since the activation of schemas is controlled from both the input and other schemas it is asynchronous. We have seen how aspect controls a) at what phase a schema is bound and b) how the bindings to the underlying schemas change over time. Hence control within a schema follows the control graph.

3. Schemas need to be able to model source domain behaviors. This involves the necessity to model sequences of actions, alternative actions, and recursive plans.

4. Schemas must be active if a certain minimal set of binding constraints, usually the theta-roles of the corresponding construction are *bound*. This requires schemas to be able to model *applicability constraints*.

5. When enabled, schemas may or may not execute, depending on if specific *resources* are present (such as *strength* to *pull* someone out of a hole) or if specific preconditions or triggers are present (as in the case of climb, where the person has to be supported before beginning to climb).

Having established the general characteristics of schemas, we now turn to our *working model* of their representation.

6.7 Representation Of Schemas

We represent schemas as a modified class of Petri Nets [Nielsen1981], where the modifications allow for the representation of stochastic transitions, as well as special compositional states and transitions (decomposable to a an ordered collection of primitive states and transitions).

A hierarchical network structures that represent an active schema by simulating execution of the component actions. Before delving into schemas, we define the basic Petri Net.

Definition : Petri Nets

A Petri net consists of *places*(P) and *transitions* (T) connected by weighted directed arcs $\mathcal{A}$ ($\mathcal{A} \in (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$). Each arc $a_{ij} \in \mathcal{A}$ has weight w_{ij} (($w_{ij} \in \mathcal{N}$.). Input Arcs $*\mathcal{T}$ ($*\mathcal{T} \in (\mathcal{P} \times \mathcal{T})$) connect Input Places to Transitions. Output Arcs $\mathcal{T}*$ ($\mathcal{T}* \in (\mathcal{T} \times \mathcal{P})$) connect Transitions to Output Places.

The underlying model of the Process net is a weighted, bipartite graph that consists of **Places** and **Transitions**. Arcs are either from Places to Transitions, in which case the Places are *Input Places* of the Transitions, or from Transitions to Places, in which case the Places are *Output Places* of the relevant transition.

Figure 14 shows a very simple net with a single transition $T1$, two Input Places $I1$ and $I2$, and a single Output Place $O1$.

In the graphical representation, Places are drawn as circles, and Transitions as boxes. Arcs are labeled with an *integer* weight. An arc weight of k, can be interpreted as k parallel arcs. Labels for unity weights are usually omitted. A *marking* assigns to each Place a non-negative integer. If a Place p is marked with the integer k, we say that the "*place* p is marked with k *tokens*." Graphically, this is illustrated by the presence of a black dot with the integer k alongside the relevant place p. A marking is algebraically described by an M-*vector*, where the i_{th} element of the vector is $M(i)$, denoting the number of tokens in the i_{th} Place. Clearly, the M-vector ranges over the number of Places in the net.

In order to simulate the dynamic behavior of a system, a Marking of the Process Net is changed according to the following rule, which is the **firing rule** of the transition.

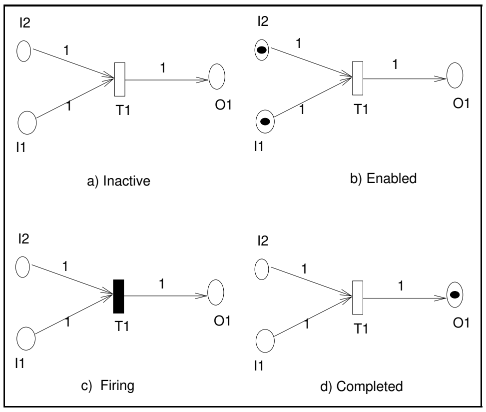


Figure 14: Basic Execution Statuses of a Petri Net(PN)

1. A transition t is said to be enabled if each input place p of t is marked with at least w_{pt} tokens, where w_{tp} is the weight of the arc from p to t.
2. The firing of an enabled transition t, removes w_{pt} tokens from each input place p and places w_{pt} tokens in each output place of t.

6.8 Modifications to the Basic Petri Net

- **High Level Nets**

  High level nets are Petri nets where tokens carry information (ie. they are individuated and typed) and transitions are augmented with predicates and actions. This modification allows for predicates to select tokens from input places based on the token type, as well as relate the type of the tokens produced by the firing to the types of tokens removed from the input. This allows us to enforce equality and uniqueness constraints.

  The extension to individual tokens preserves the *liveness* and *reachability* properties of the underlying ordinary net. Any net with a finite number of types can always be *unrolled* into an ordinary net by adding special places for type combinations. A general treatment of such nets can be found in [Murata1988].

- **Transition Types**

  A Process Net consists of three types of transitions, namely *deterministic* transitions, *stochastic* transitions, and *hierarchical* transitions.

  Figure 15 shows the various transition types and their graphical representations.

  1. Deterministic Transitions fire with a fixed delay after being *enabled*. The actual delay is an integer which varies depending on the nature of the transition. For logical conditions, for instance, the delay is very small, and the transition can be considered instantaneous.
  2. Stochastic Transitions are associated with an exponentially distributed random variable that expresses the delay between enabling and firing. Due to the memoryless feature of

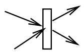
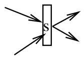
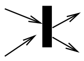
Transition Type	*Description*	*Graphical Representation*
Determinstic	Fires after a fixed delay after being enabled.	
Stochastic	Fires after an exponentially dist. delay after being enabled	
Hierarchical	Activates a subnet and is instantaneous	

Figure 15: A description of Basic Transition Types.

the exponential transition, PN's with stochastic transitions are isomorphic to Markov Chains [Molloy1982].

3. Hierarchical Transitions correspond to the *activation* and *execution* of a **subnet**. Hierarchical transitions are instantaneous.

6.9 Why Petri Nets?

There are several reasons for choosing Petri Nets are our schema representation language. They are outlined below.

1. Petri nets are a graphical and mathematical tool for modeling asynchronous and concurrent systems. We believe that the our mental models inherently have these characteristics. While Petri Nets allow tokens to be used to simulate dynamic and concurrent activities of such systems, they also allow for the setting up of state equations and other mathematical models to theoretically study the behavior of such systems.

2. Petri Nets can be used to model resource constraints. A resource that is consumed during the operation of an action is present in the input place (in(t)) where t corresponds to the transition that represents the action. Similarly, a resource that is created or produced by the action, or a resource that is returned at the completion of an action (a shared lock, for instance) is modeled by an output place (out(t)) of the corresponding transition t.

3. Petri Nets allow for the same mechanism to be used for cognitive simulation and for monitoring the execution of motor control of action. This makes extended Petri nets a potential formalism for monitoring the execution of plans. The next section deals with this issue in greater detail.

4. Petri nets have been shown capable of simultaneously describing the logical schema that underlies a database and its execution semantics. This makes it of potential use in semantic network research. The earliest attempts to use Petri nets to model semantic networks seems

to have been done by Sowa [Sowa1984] who modeled data flow in *conceptual graphs* using a variant of Petri nets. Petri nets also show promise in formalizing aspects of marker passing architectures.

5. One of our interests in this project is to stick as closely as possible to a structured connectionist implementation while retaining a formal specification and analysis process. While the exact correspondence between Petri Nets and Connectionist Networks is yet to be worked out (though Linear Logic [Girard1987] may prove to be a unifying semantics), the connection between places and nodes, markings and activation, arc weights and connection weights, token flow and activation flow is obvious. If the correspondence could be worked out, Petri Nets have attractive mathematical properties that could be used as a connectionist specification language.

7 Meeting the modeling requirements

Clearly, my schema representation meets the requirement of asynchronous and concurrent operation of multiple schema instances. The question gets a little trickier for the other requirements.

7.1 Modeling Aspectual Inferences

Different schemas can execute concurrently, and are activated in an asynchronous fashion. Each schema has a local controller. This allows for a distributed control architecture between schemas, and a local controller that synchronizes actions within a schema.

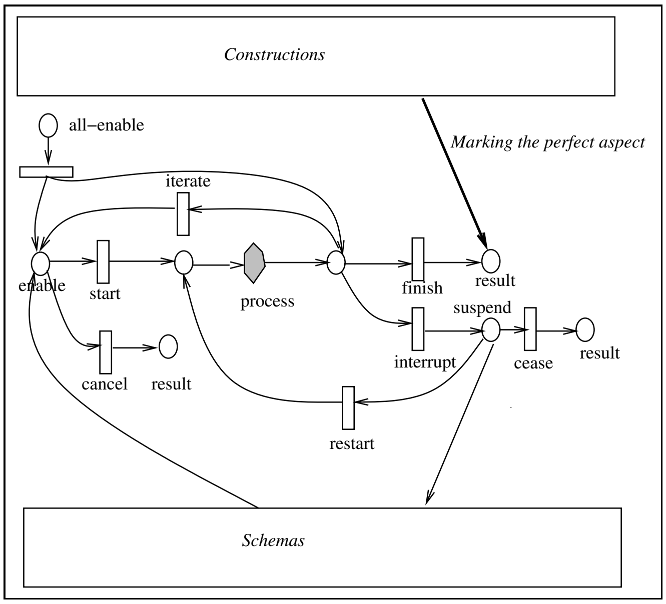


Figure 16: The Schema Controller.

The controller described in Figure 16 follows the natural stages of situations shown in the transition graph of Figure 13. The various transitions (*start*, *process*, *finish*, *interrupt*, *cancel*, *iterate*, and *restart*) all refer to links or states in the *control graph.* There is an additional *all-enable* state that marks both the *enable* and a *process* state. The *process* transition is usually a hierarchical transition in that it is composed of interacting subnets.

In several places of the controller we have multiple outgoing arcs leading to multiple transitions. Such a situation is a referred to as a **decision** or **conflict** point and leads to ambiguity in interpretation. The solution to the **conflict** is provided by the specific combination of the relevant Linguistic Construction (denoting an *aspect* marking, for instance, and specific constraints provided by the schema.

To illustrate the disambiguation of a **conflict** consider the Output of the *Process* transition. The Output place could potentially enable three transitions, namely *iterate*, *finish*, and *interrupt.* Let us look at possible selection of transitions.

1. The construction used chooses a specific transition-sequence such as *interrupt-suspend-cease-result* in *World Bank aid ended India's slide into recession.*
2. The interpretation uses a path such as *The middle-east economies stalled (stopped moving forward).* only the *interrupt-suspend* is selected.
3. The linguistic input may choose specific paths, as in the case of the *used to* construction. In this case, the selected transition is *iterate*.

Grammatical Constructions and other ways of expressing aspect interact with the schema controller to jointly produce an interpretation. An example of some ways linguistic input interacts with the controller through aspect is illustrated in Figure 17. Figure 17 graphically illustrates my model for the **cessive** (ex. *stop*), **cumulative** (ex. *one by one*), and **Protracted** (ex. *keep on, continue*) aspects.

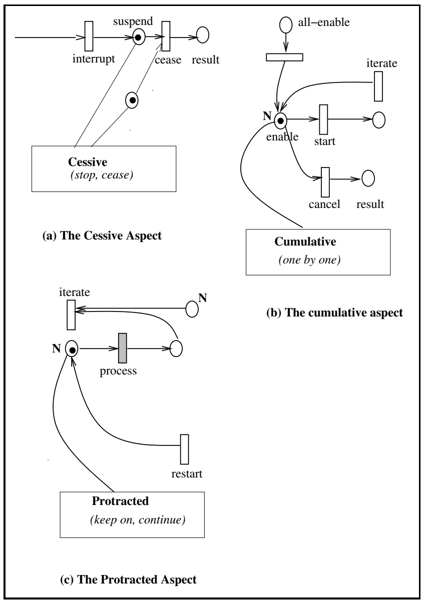


Figure 17: Extended Aspect: Example Of the (a) cessive, (b) cumulative, and (c) protracted aspects.

The following observations can be made about the model.

1. In the case of the **cessive** form, we see that the *cease* transition is enabled. This allows all inferences that follow from the resulting stage of the cease transition automatically. For instance, in the interpretation of the sentence *US stemmed immigration tide*, the automatic inference *US expended effort* would be available upon inspection of the underlying schema.

2. In the case of the *cumulative* aspect (and any other instance of expressing countable instances of a situation), the placing of multiple tokens allows for the same transition to fire multiple times. For instance, to model the sentence *The stock market plumetted twice*, the *start* transition of the *plummet* situation is enabled with 2 sets of tokens.

3. In some cases, the supply of tokens to a place may be unlimited, as in the case of a *protracted* situation. For example, in the case of *Prices kept soaring*, the *process* transition characterizing the *quantity increase* schema is continuously enabled. Other constructions such as *until* (as in *Prices kept soaring until inflation slowed down.*) may inhibit the *process* transition when a specific condition (inflation slowed) obtains.

7.2 Schemas are plan/behavior interpreters that can guide execution

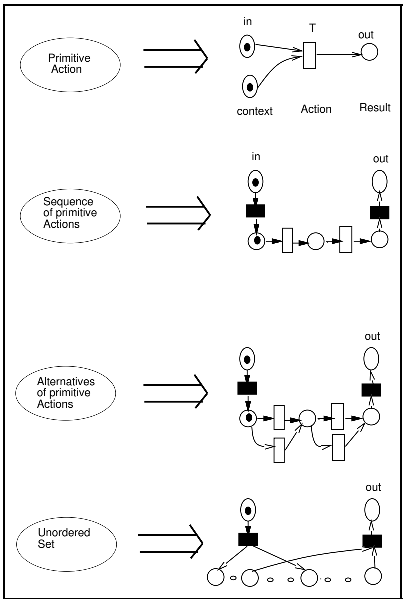


Figure 18: Schemas are plan interpreters

Figure 18 shows the net based representation of schemas. Various arrangements of actions are shown with the corresponding schema realizations. When schemas are used to perform simulations, they are "timed ". Hierarchy transitions are instantaneous, and action transition rules have a stochastic transition duration.

7.3 Falling into holes: A schema description

Here I describe in a STRIPS-like manner the fall into hole schema. Since the description is for illustrative purposes only, I will only describe the main path (*enable-start-process-complete*).

```
ENABLE:

Context                  Action                     Effects
-------                  ------                     -------
appear(faller)          atomic                      support(faller, ground)
appear(hole)                                        control(faller, location)
                                                    Status: start

START

Context                  Action                     Effects
--------                 ------                     -------
support(faller, gr)      atomic                     Status: Process
control(faller, loc)                                in(faller, hole)
Status: Start

PROCESS

Context                  Action                      Effects
--------                 ------                      -------
~support(faller, gr)     (concurrent                 support(faller, ground)
~control(faller, loc)      (decrease elevation)      control(faller, location)
Status: Process            (decrease dist to bottom) Status: Finshed
                         )                           location(faller, bottom)
                                                     injured(faller)
                                                     THREAD pullout
                                                     (VALUE: medium)
                                                     (tr2(faller)
                                                     (loc(tr2, loc(faller))))
                                                     THREAD climbout
                                                     (VALUE: medium)
                                                     (tr(faller)
                                                     (loc(tr, loc(faller))))
```

All transitions are atomic, except for the *process* transition. The process transition consists of the sub-actions that correspond to simulating the fall (elevation decreasing, moving toward bottom of hole) The effects consist of *threading* to another schema that corresponds to a being pulled out of a hole. This creates a new instance of the threaded schema (pullout) with the following bindings.

Fall into	**Pull out**
Faller	Person being pulled out
Location of faller	Initial location of person pulled out

Clearly, a number of schemas that correspond to getting out of holes (like climb out, crawl out, etc.) will also have threads from the fall schema. All such accessed schemas will be in the working store to be bound to by future input. Obviously, the branching factor will be an important issue in the design of schemas. However, since my hypothesis is that *spatio-temporal domain* schemas structure many abstract domains, and it is likely that many source domain schemas would be readily accessible.

7.4 Falling into hole: The proposed representation

Figure 19 shows the schema implementation of the description above. For ease of description, I have collapsed the context and effects into a single input or output *place*. Since tokens are *individuated*, each token may represent a predicate-binding in the domain. Tokens carry this information as they flow around the net. When a token that is an instance of a particular type appears at a place that represents a predicate, it binds to the argument of the predicate in question.

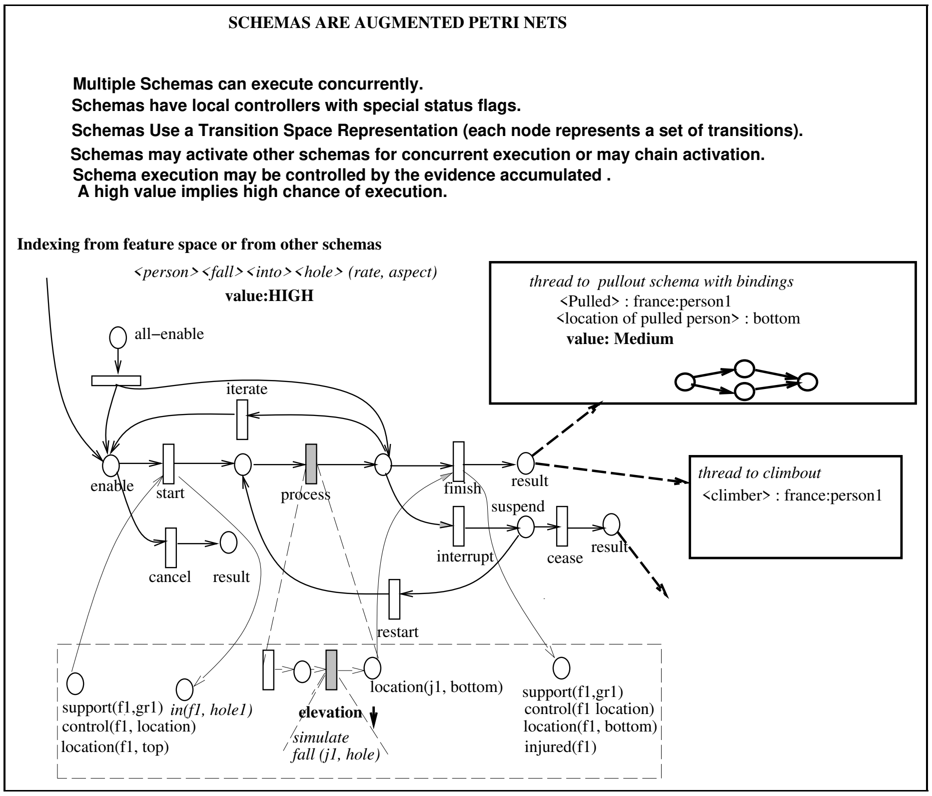


Figure 19: Schema Characteristics: Falling into holes

Threads are forked off at the various output places of the net. Threads may or may not activate another schema, depending on the value of activation, and the prior frequency of activation of the threaded schema, and the set of unbound slots. For instance, the pullout schema **cannot** execute since the theta roles of its binding structure (specifically the puller slot) is still unbound. It will also usually be the case that unless a specific linguistic input directly indexes to the schema in question, the activation level will not be high enough to bind the schema. The exact design of the various parameters outlined above is still being worked out.

If a schema is threaded into, its binding structure, and corresponding construction (with the bound slots) are now in the working store. For instance, the result of activating the *fall into* schema will add the climbout and pullout schemas into the working store. Also added will be the corresponding constructions as shown in Figure 20. Thatched nodes are nodes that have already been bound.

These nodes may be viewed as evidence nodes. If the particular pull construction is selected by future input, these nodes will already be bound and once the puller is identified, the underlying pullout schema can start execution.

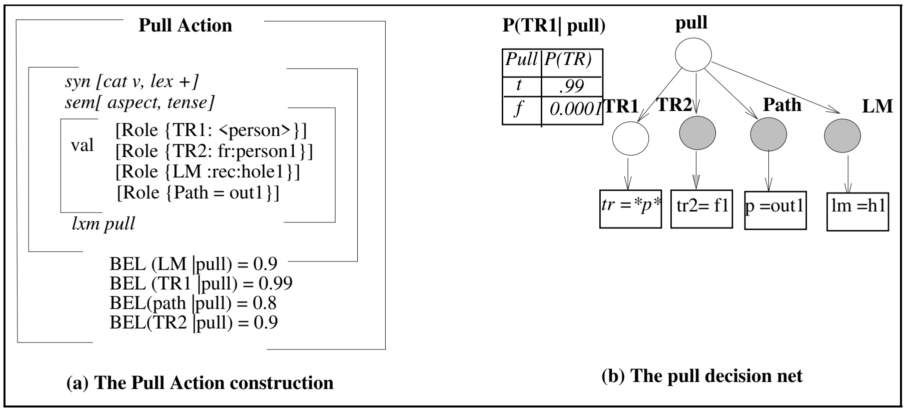


Figure 20: Threading brings new construction into working store

---

These constructions can be referenced by future input, and hence lead to filling the unfilled slots. For instance, the next input "Germany pulled France out" will reference the pull construction already in working store, and will bind the metaphoric interpretation Germany:person2 to the unfilled slot of the construction and this completes the binding to the the pullout schema, which will execute and assert that France is out of the recession. An interesting thing to note is that this allows for interpreting **elliptical** narratives like:

- France fell into recession. Pulled out by Germany.

In some cases, the schema that is threaded into may already result in all application constraints being satisfied. Clearly in this case, if the schema executes, it may lead to irrelevant inferences. The times that a threaded schema can execute without being indexed by the linguistic input will depend on the value of activation, and the degree to which the schema is bound. Further implementation and data analysis is required before these parameters can be instantiated.

7.5 Activating Schemas

The following steps occur when a schema is activated by being bound from a construction or by threading.

- name/type of schema places a token at the enable place of the schema.
- Aspect: place the appropriate token at the schema controller place.
- LM: creates individual token corresponding to the binding.

- TR: creates individual token corresponding to the binding.
- For parameters, the value generates the appropriate type of parameter. The actual translation is schema dependent.

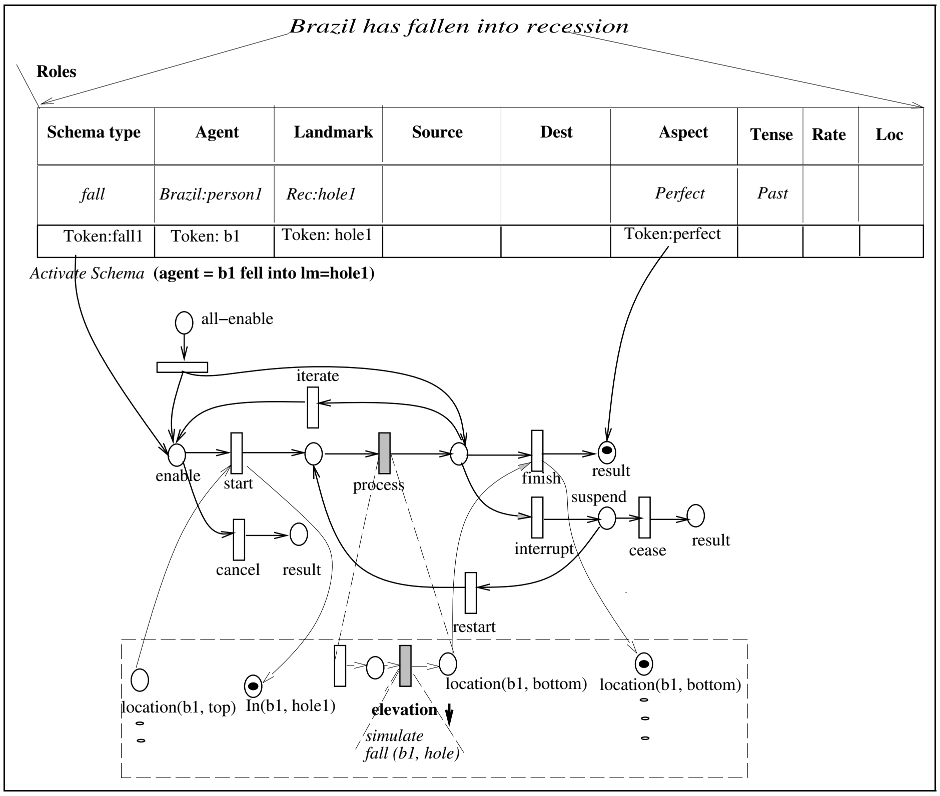


Figure 21: Activating schemas by *marking* the underlying net

Figure 21 shows the activation of the schema corresponding to the input "Brazil has fallen into recession". Note how the Aspect (perfect) maps to the specific stage of the controller of the fall into schema. Only relevant parts of the schema are shown.[17] The network shown has parameterized places. Tokens circulating correspond to the individual tokens that maintain unique binding constraints. In this case the marking vector shows the places corresponding to the result of fall to be active. The set of tokens are

- Schema: fallinto1 (enables the schema fallinto1)
- status: *perfect.* Place a token at the *result* stage of the schema controller.
- TR: b1: create an individual token of type *person* for the TR.
- LM: hole1: create an individual token of type *hole1* for the LM.

[17]The overall schema allows for other inferences, and can be found in [Narayanan1995].

7.6 Extracting from Schemas

When a schema executes, it modifies the binding structure by creating new bindings. We saw how in the example, the slots for the location, goal and destination became bound as a result of schema execution. In essence, a schema can be viewed as taking a partially bound binding structure, and returning a more bound structure. In the case of the input discussed above, extraction of fillers for specific unfilled slots is easy, since the mapping is direct.

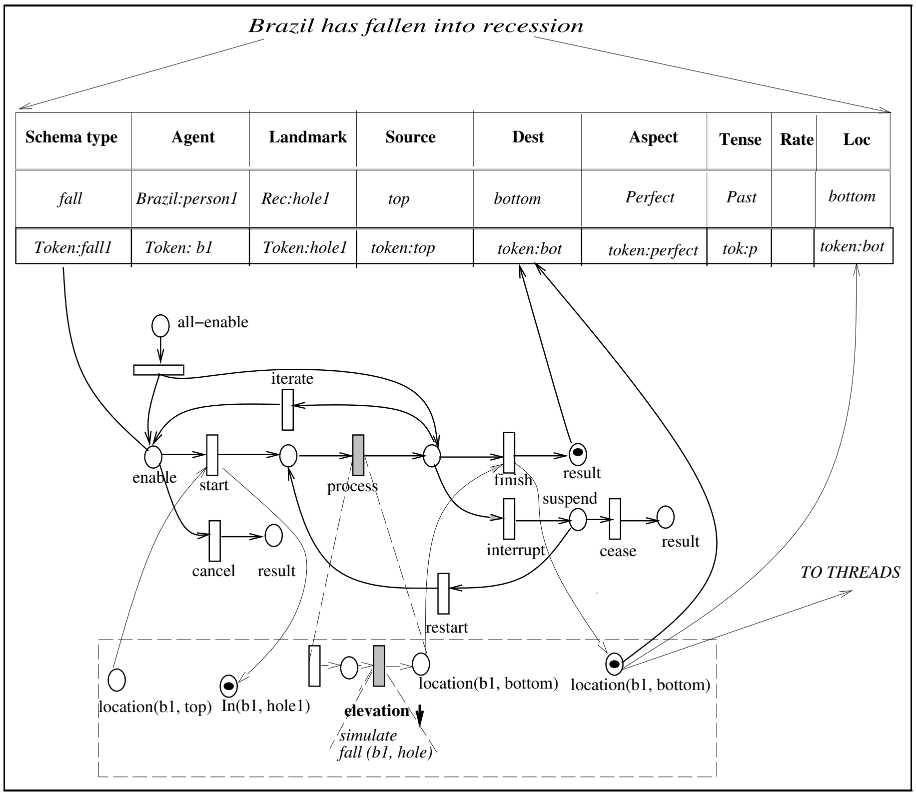


Figure 22: Extracting bindings from executing schemas

Figure 22 shows the activation record of the schema corresponding to the input "Brazil has fallen into recession"after schema activation has occurred. Here, no further execution is possible, since the tokens are all at output places. Consider, the difference if the sentence were instead "Brazil was falling into recession". The process transition would be active, and the location of the agent (while in the hole) would be continuously decreasing and the location slot would be left unbound.

The values for the unfilled slots are directly wired into the binding structure. All the values, except the location slot are statically wired in. In the location case, the high-level token that marks the location of the agent as being at the bottom of the hole should be extracted.

In general, extracting information from executing schemas requires inspection of the marking vector and reachability graph of the underlying Petri net. The issues of designing tractable extraction mechanisms is part of ongoing research.

7.7 Modifying Working Store

The new binding structure that result from schema execution may result in the following changes to the Working Store. Two basic ways in which this can happen are:

1. Activate a schema which is completely bound. This results in a fully integrated construction in the interpretation. For instance the spatial inference *fall into hole* $\Rightarrow$ *being in hole* results in the corresponding structure to be added to the interpretation.

2. Access a partially bound construction, and add it to the working store. For instance, the result of activating the *fall into* schema allows the *pull* construction to be selected by future input. The corresponding schema is not *bound* and is waiting for additional input to get bound.[18] If it does not get bound, the schema may get cancelled and never execute.

Appendix B shows a complete trace of the expected interpretation of the narrative "France fell into recession. Germany pulled France out."

[18] Of course, in general there has to be some search. Having a single underlying domain *may* allow hardwiring the various combinations.

8 Discussion

8.1 Inference in NLP: Using World Knowledge

Using world knowledge in discourse interpretation was first studied extensively by Chuck Rieger [Rieger1975] under Roger Schank. In this seminal work, Rieger implemented a system where sentences were mapped to a semantic representation that generated inferences. All possible inferences that could be drawn were drawn. In the presence of ambiguity, the interpretation that generated the most number of inferences was selected. While his proposal showed the close connection between world knowledge and interpretation, the uncontrolled nature of forward inferences led to severe intractability and counter-intuitive results[Charniak1976].

Later, Schank's group scaled back the effort and used the idea of Scripts [Schank1977], where inferences were limited to narrow domains and text interpretation relied exclusively on stereotyped scenarios. This limitation proved too severe, and led to extensions that pertained to recognizing and using goal-based strategies [Wilensky1983], and dynamic memory assemblages [Kolodner1984].

In more recent philosophical work, Sperber and Wilson [Sperber1986] have made an argument in many ways similar to that of Rieger. They argue that the contextual implication of a new utterance is the conclusion that can be drawn from the currently *active* background knowledge and the input information. In their argument an utterance is *relevant* to the extent that it allows easy inferences in the current context. In the presence of ambiguity, the best interpretation is that which is most relevant.

In other work, Hobbs [Hobbs1990] and others have argued for a backward chaining mode of inference called abduction. As I argued earlier, searching for the best explanation given the input is an in instance of abduction. The correspondence between weighted abduction and probabilistic inference was shown by Charniak and Shimony [Charniak1994].

In this thesis, I take the view that interpretation is a combination of *abduction* and *prediction*. Binding to schemas is primarily an abductive process, while schema execution is a mainly a predictive process. The hope is that this combination will allow for the main intuitions behind both approaches, that language interpretation is simultaneously *diagnostic* and involves limited *automatic inference*.

8.2 Where do the numbers come from

This is a big question for any probabilistic estimation method. Since no corpus is currently tagged with metaphoric information, there is no easy way to estimate the frequency of access and selection of different metaphors. In fact, one of the potential contributions of the research proposed here would be to spur attempts to tag corpora with metaphoric information. Regarding correct estimates, preliminary results indicate that exact numerical values are not required for the disambiguation problem that we are trying to solve. Also, [Russell1994] has recently shown how gradient descent can be used in combination with the standard probabilistic inference algorithms to update the CPT adaptively. Such a scheme has several advantages including localized learning and ability to use prior knowledge in the learning process. So the crude estimates I propose to use in my thesis could later be refined when future work produces better estimates.

8.3 Lexicalized metaphors

Metaphors may be lexicalized. This may occur through learning. Lexicalized metaphors can be treated as *regular* concepts in the lexicon.

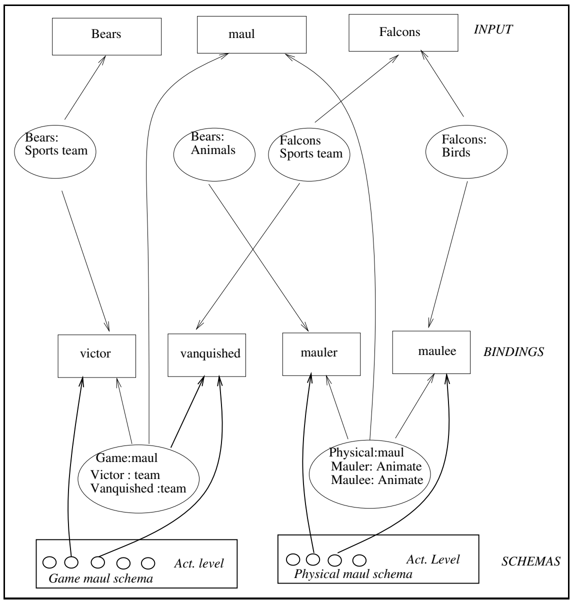


Figure 23: Metaphors can be lexicalized

Figure 23 shows an example of the interpretation structure for the utterance *Bears maul Falcons*. Notice, in this same a lexicalized entry for the concept *maul* is treated as another competing explanation of the input. Now in the target domain (sports), the prior beliefs for the sports team sense would be much higher that for the animal sense of the concepts Bears and Falcons. Even if the SA measure for both predicates are the same, the target domain schema would be a much more likely candidate for selection.

8.4 Metaphors are structured objects

[Martin1990] makes a convincing case that metaphors are structured objects that can be arranged in dominance hierarchies. In the model of interpretation proposed here, this is a natural extension. Figure 24 shows how the metaphor *Recession is a Hole* gets activation from both the Event structure metaphor (which is activated by the fall event construction) as well as from the presence of the target construction (recession economic state).

The standard belief net inference algorithms will work for this case as well. The exact details of the construction algorithm in the presence of metaphor hierarchies is still to be worked out.

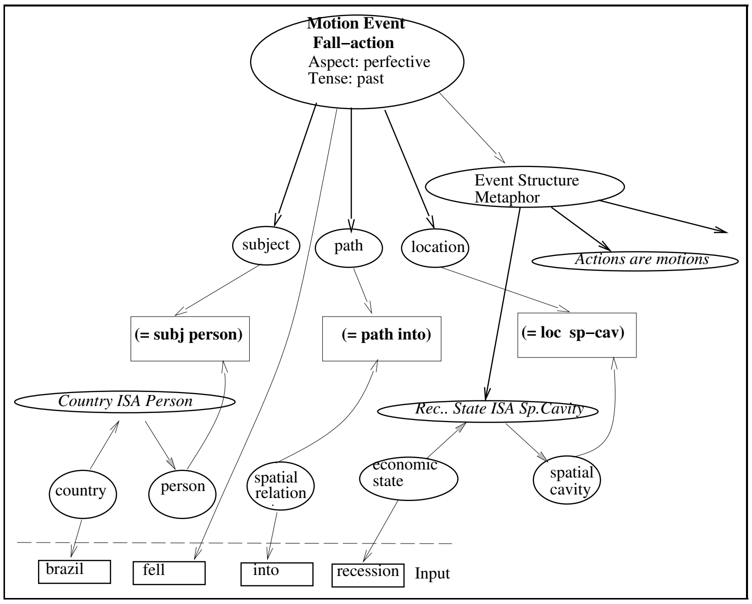


Figure 24: Metaphors can activate other metaphors

8.5 Other connectors

Consider the sentence *The mushroom omelet left a large tip* [Fauconnier1985]. In this case a domain specific metonymy maps the input mushroom omelet to the customer who ordered the eggs. This helps the overall convergence to the schema *customer leaves tip*. Such connectors can also be felicitously handled in the proposed framework. Figure 25 shows the decision net that satisfies this example.

8.6 What Level Do We Simulate?

Clearly, this is an important question to be answered. While we have narrowed our domain considerably by using metaphoric mappings, we are left with simulating the source domain at multiple levels of resolution. We expect the answer here will come from cognitive constraints and their computational implications. Whenever there are timing studies of metaphor processing such as [Gibbs1993], we can include them. For the vast majority of the stories in our database, metaphoric mappings are used mainly for disambiguation and for fairly shallow inferences (mainly aspectual and context setting inferences for anaphora resolution). So detailed world simulation may not be necessary to generate the inferences required. [19]

[19] But this may also be an artifact of the newspaper type of domain.

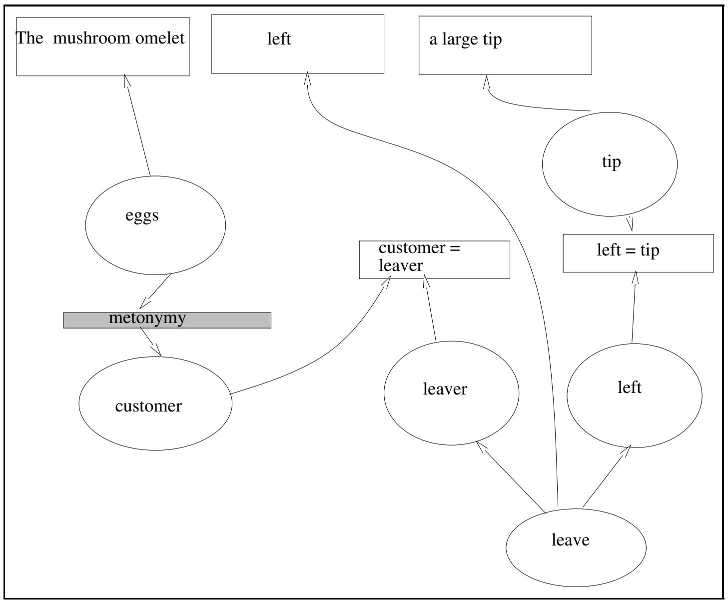


Figure 25: Constructions may activate metonymies and other pragmatic connectors.

8.7 Goal-based Enabling

Schemas can be indexed in at least two ways. One is by enabling the goal (called *goal-enabling*). Another is by *direct enabling*, as is usually the case when linguistic input activates new schemas. This allows for both forward and goal-based inferences in schemas. While this report only describes schemas for Script-like automatic inferences, answering queries requires *reconstruction* and goal-based enabling. I intend to only model *direct enabling* intitially.

There are two possible methods to implement goal-based enabling in schemas as currently defined.

1. Use the declarative form of the Petri Net (as a set of plan operators), and do goal-regression.
2. Define a new kind of marker that moves *backward* on the net. Then standard reachability analysis can determine the required action sequences.

The second possibility will make the current representation resemble very closely the dataflow graph representation using in Conceptual Graphs [Sowa1984]. In Sowa's representation scheme, actor nodes attached to a conceptual graph are also joined to form a dataflow graph. These dataflow graphs resemble Petri nets with two kinds of tokens, namely *assertion marks* which resemble the forward propagating Petri net tokens, and *request marks* which are propagated backwards. This connection between conceptual graphs and the schema representation scheme proposed here bears detailed investigation.

8.8 Threading

Schemas can thread into each other. As an example, consider the case where the fall-into hole schema activates the more general linear scale schema with the following mapping.

Fall Into Hole	**Linear Scale**
Top Of Hole	Top Of Scale
Bottom Of Hole	Bottom Of Scale
Person in hole	Entity on Scale
Location Of Person	Level Of Entity on Scale

This is a dynamic thread that links the motion of a falling person to the vertical motion of an entity on a scale. This kind of threading allows for the interesting reconstruction operator, *dynamic bootstrapping*. Such an operator may help explain the common occurrence of certain types of mixed metaphors, such as the one in the example below.

- France fell into a recession. It has now recovered and is healthy again.

One possible explanation is that both falling and being sick share the linear scale schema, where in the one case it is a person's health on the scale, and in another the person's location. This allows the cross domain inference felicitously.

An immediate question is if metaphoric mappings are onto such abstract schemas (such as linear scales) or are they to actual motions in space (as we have been arguing). The answer seems to be both. In this case, the context changing inference (default -fail -recover) is all that is used. However, the sentence from New York Times (August 1994) which was "The yen crawled higher sluggishly on the back of the dollar" seems to invoke much more that just linear scales.

A possible answer comes from Section 5.4 where I summarized the kinds of inferences that are made in the concrete domain. In the example above, what is *relevant* is that the speaker is indicating the change in context (from default to failure in the first sentence and from failure to recovery in the second). Since this is the only information conveyed, the interpreter can make sense of this input because both chains

1. healthy $\Rightarrow$ sick $\Rightarrow$ healthy
2. moving-forward $\Rightarrow$ fall $\Rightarrow$ recover

satisfy the context shifting constraint and hence even with the mixing of metaphors, the interpreter is able to felicitously comprehend the input narrative.

Other reconstruction operators include constructing causal chains and identifying other aggregations (object-composition, attribute composition, etc.) The exact implementation of threading is still being worked out.

9 Conclusions

This project proposes to investigate some new ideas for modeling the comprehension of narratives such as newspaper stories involving political or economic causation. If the project is successful, the following novel ideas would have been implemented in a computational model of narrative interpretation.

1. Narrative understanding by metaphoric mapping from abstract domains, such as politics and economics to the concrete spatial domain. The claim that such a mapping process constrains acceptable parses of an input sentence would be demonstrated in an implemented model that enhances an existing construction grammar parser. We believe that such a framework should provide a sound computational testbed for other theories of on-line narrative interpretation, especially theories of non-literal interpretation.

2. The translation of the metaphoric interpretation process into multiple, executing models or schemas. Context created by schema execution conditions the listener/reader's expectations of future linguistic input. This models the cooperative interaction between speaker/hearer or text/reader. Section 7 outlined some ideas for implementing schemas. Some work has already demonstrated the use of schemas in modeling aspectual distinctions of the worlds languages. The exact nature of the construction/schema interaction is still being researched and will form a substantial part of the proposed work. The results of this effort should be of interest to both cognitive and computational linguists.

3. The grounding of the deep semantics of abstract concepts in active concrete domain models. Initially, I propose to implement the *Economy is a traveler* model to demonstrate the overall idea and to flesh out the issues pertaining to the bi-directional mapping process. Once this is completed, the use of a formal specification language (the extended Petri nets discussed in Section 7) should allow systematic extensions to the other domain models and *open the door* for computational models of cognitive linguistics.

10 Appendix A

This section describes a few frequent metaphors that have been culled from newspaper stories in the domain of Economics and Politics accessed from the LEXIS/NEXUS service.

10.1 Conventional Metaphors For Economics: The flow Of commerce

Flowing Things	**Dir**	**Barriers**
Investment	high to low wages	tariffs, investment barriers
Labor	low to high wages	regulations, labor law
Jobs	high to low wages	regulations, insurance

- Example: China to stem migrant labour flood at New Year (Reuters, Dec 7, 94).
- Exchange rates *control* flow, other regulations*modulate* flow.
- labor is *attracted* by the *force* of high wages, investment by low.

10.2 Conventional Metaphors For Economics: Agent motion in a landscape

Economy is a traveler in a landscape:

- By default economies are *moving ahead* in the landscape.
- Any change in economic status corresponds to changes in the agent's motion or in the terrain characteristics.
- Examples of actions include climb, crawl, sidestep, step, stride, leap ahead, change direction, stand still etc.
- Any actions are self-propelled motions of the agent.
- Forces may push, propel, pull, stimulate, reorient, impede, weaken, or injure the economy.
- As a consequence of these forces, economies may fall, stumble, slide, slip, plummet, be stuck, etc.

10.3 Conventional Metaphors For Economics: Vehicle in a landscape

- By default economies are *driving ahead* in the landscape.
- Economies may stall, overheat, or run out of gas.
- Economies may have to be jump-started.

10.4 Conventional Metaphors For Economics: Markets are communities

- Common Market, EEC are such communities.
- Individual trading units (nations, firms) are members in the community.
- Members can enter and leave markets as they can enter and leave communities.

11 Appendix B

Putting all elements of the system together for the simple illustrative example used in the introductory section, the following is the expected processing trace for the following narrative.

- France fell into recession. Germany pulled France out.

```
INPUT > France fell into recession.

RUNNING INTERPRET
-- working store contents
input
Accessed constructions (fall-event1)
Accessed metaphors (Country ISA person, Recession State ISA hole)
Accessed concepts (Country, Recession State, spatial relation)

Constructing net

Evaluating net
-- Instantiating Interpretations (I1, I2)
          -- I1
Fall.Trajector = France1,
Fall.Landmark = Recession1
Fall.path = into1.
VALUE: 0 -- No schema
          -- I2
Fall.Trajector = France:person1 (Metaphor Country ISA Person),
Fall.Landmark = Recession:hole1 (Metaphor Recession State ISA hole)
Fall.path= spatial relation(into1)
VALUE: 0.9

BINDING to (fall into) with bindings
(instance= fallinto1, TR=France:person1, LM = Recession:hole1, Aspect...)

Schema fallinto1 bound.

ACTIVATING fallinto1
Schema fallinto1 active.
Schema fallinto1 completed execution.

EXTRACTING BINDINGS fallinto1
New bindings:
(instance= fallinto1, TR=France:person1, LM = Recession:hole1, Aspect....)

MODIFYING WORKING STORE
THREAD to schema (be in) with bindings
(TR=France:person1, LM= recession:hole1)
VALUE: 1.0
Accessed construction: be, PRIOR = 0.8
(France:person1, in1, recession:hole1)

THREAD to schema (pull out) with bindings
(instance= pullout1, TR1 = ?x, TR2=France:person1, LM = Recession:hole1, Aspect...)
VALUE: 0.3
Accessed construction pull, PRIOR: 0.3 with bindings
Pull.Trajector1 = *pending*
Pull.Trajector2 = France:person1 (Metaphor Country ISA Person),
Pull.Landmark = Recession:hole1 (Metaphor Recession State ISA hole)
Pull.path= spatial relation(out)
```

```
WAITING pullout1.

THREAD to schema (crawl out) with bindings
(instance= crawlout1, TR=France:person1, LM = Recession:hole1, Aspect...)
VALUE: 0.05
Accessed constructions : none (threshold not met)
WAITING crawlout1.

THREAD to schema (climb out) with bindings
(instance= climbout1, TR = France:person1, LM = Recession:hole1, Aspect...)
VALUE: 0.1
Accessed constructions: none (threshold not met)
WAITING climbout1.

END interpretation-cycle1

> INPUT: Germany pulled France out.

RUNNING INTERPRET
-- working store contents
Interpretation (fall-event1(bindings ..), be1(bindings ...))
accessed constructions (pull-action1)
accessed metaphors (Country ISA person, Recession State ISA hole)
accessed concepts (Country, Recession State, spatial relation)
schemas (pullout1, climbout1, crawlout1, fallinto1 be-in1)

Building net

Evaluating net

Instantiating Interpretations (I1, I2)
-- I1
Pull.Trajector1 = Germany:person2,
Pull.Trajector2 = France:person1
Pall.Landmark = Recession:hole1
Pull.path = out1.
VALUE: .9 -- pullout schema bound
--I2
Pull.Trajector1 = Germany1,
Pull.Trajector2 = France1
Pall.Landmark = Recession:hole1
Pull.path = out1.
VALUE: 0 -- No schema to bind to

BINDING (pull out) with bindings
(instance= pullout1, TR1 = Germany:person, TR2=France:person1,
LM = Recession:hole1, Aspect...)

ACTIVATING SCHEMA pullout1

Schema pullout1 active.
Schema pullout1 completed execution.

EXTRACTING BINDINGS pullout1
(instance= pullout1
TR1= Germany:person1
TR2=France:person1
TR2.location= out of recession:hole1
...)

MODIFYING WORKING STORE

THREAD
climbout1 TIMEOUT SUSPEND
```

```
crawlout1 TIMEOUT SUSPEND

Working Store Contents:

Interpretation:
(fall-action(France, into, recession),
be(France, in, recession),
pull-action(Germany, France, out, recession)
be(France, out, recession))

Accessed metaphors (Country ISA person, Recession State ISA hole)
Accessed concepts (Country, Recession State, spatial relation)

INPUT > bye.
```

References

[Allen1995] J. A. ALLEN, 1995. *Natural Language Understanding.* The Benjamin Cummings Publishing Company, 1995.

[Arbib1992] M.A. ARBIB, 1992. *The Metaphorical Brain: 2.* Oxford University Press.

[Brooks1986] R. A. BROOKS, 1986. *A Robust-layered controlled system for a mobile robot.* IEEE Journal of Robotics and Automation, 2:14-23.

[Burs1983] M.H. BURSTEIN, 1983. *A model of learning by incremental analogical reasoning and debugging.* Proceedings of AAAI 83, 1983.

[Carberry1991] S. CARBERRY, 1991. *Plan Recognition in Natural Language Dialog.* Cambridge, MA:MIT Press, 1991.

[Carbonell1982] J. CARBONELL, 1982. *Metaphor Comrehension.* Strategies for natural language processing, pp 413-433, Lawrence Earlbaum:1982.

[Charniak1976] E. CHARNIAK, 1976. *Computational Semantics.*

[Charniak1988] E. CHARNIAK AND R. GOLDMAN, 1988. *A logic for semantic interpretation.* ACL 1988.

[Charniak1989] E. CHARNIAK AND R. GOLDMAN, 1989. *A semantics for probabilistic, quantifier-free, first order languages, with particular emphasis to story understanding.* Proceedings of IJCAI, 1989.

[Charniak1993] E. CHARNIAK AND R. GOLDMAN, 1993. *A Bayesian Model of Plan Recognition* Artificial Intelligence (64), 1993: 53-79.

[Charniak1994] E. CHERNIAK AND R. SHIMONY, 1994. *CBA and MAP Explanation.* AI Journal, Apr. 1994

[Comrie1976] B. COMRIE, 1976. *Aspect.* Cambridge University Press.

[Cover1991] T. COVER AND J.A. THOMAS, 1991. *Elements Of Information Theory.* Wiley Interscience Publications, 1991.

[Drescher1991] G.L. DRESCHER, 1991. Made-up minds: A constructivist approach to AI. MIT Press, 1991.

[Feldman1989] J. A. FELDMAN, 1989. NEURAL REPRESENTATION OF CONCEPTUAL KNOWLEDGE. Neural Connections, Mental Computation. MIT Press 1989.

[Feldman1994 etal] FELDMAN, LAKOFF, BAILEY, NARAYANAN, REGIER, STOLCKE, 1994. *Lzero: The first four years.* To be published in AI Review Journal, Kluwer Press, 1995.

[Fillmore1989] CHARLES FILLMORE, 1989. *On Grammatical Constructions.* Course notes draft.

[Fauconnier1985] G. FAUCONNIER, 1985. *Mental Spaces.* MIT Press, 1985.

[Gibbs1993] R. GIBBS, 1993 *Why idioms are not dead metaphors* Idioms: processing, Structure and Interpretation. Lawrence Earlbaum and Associates, pp. 55-77, 1993.

[GL1990] R.V. GUHA, AND D.B.LENAT, 1990. *Cyc: a mid-term report.* AI Magazine, 11 3 (1990), 32-59.

[Girard1987] JEAN-YVES GIRARD, 1987 *Linear Logic.* Theoretical Computer Science 50 (1987) 1-102.

[Hobbs1985] J. HOBBS AND B. MOORE, 1985. *Formal Theories Of The Commonsense World.* Ablex Publishing Corporation, NJ.

[Hobbs1990] J. HOBBS ET AL., 1990. *Interpretation as Abduction.* SRI Tech. Note 499, SRI International, Menlo Park, Ca, 1990.

[Hobbs1993] J. HOBBS, M. STICKEL, D. APPELT, AND P. MARTIN 1993. *Interpretation as Abduction.* AI Journal 63, 1-2:69-142.

[Howard1981] R.A. HOWARD AND J.E. MATHESON, 1981. *Influence Diagrams*, Strategic Decisions Group, 37, 1981.

[Jurafsky1992] D. J. JURAFSKY 1992. *An On-line Computational Model Of Human Sentence Interpretation.* UCB Ph.D. thesis 1992.

[JN1995] D. J. JURAFSKY AND S. NARAYANAN, 1994. *Probabilistic Unification.* ICSI TR, In Preparation.

[Kolodner1984] J. KOLODNER 1984. *Retrieval and Organization Strategies in Conceptual Memory: A computational model.* Hillsdale, NJ:Erlbaum 1984.

[LJ1980] G. LAKOFF AND M. JOHNSON 1980. *Metaphors we live by.* Chicago, University Of Chicago Press.

[Lakoff1987] G. LAKOFF 1987. *Women, Fire and Dangerous Things.* Chicago, University Of Chicago Press, 1987.

[Lakoff1992] G. LAKOFF, 1992. *What is Metaphor?.* Advances in Connectionist Theory. V2 : Analogical Connections, V2.

[Langacker1987] R. LANGACKER, 1987. *Foundations Of Cognitive grammar.* Stanford University Press.

[Martin1990] J. MARTIN 1990. *An A Computational Model of Metaphor Interpretation.* Academic Press, NY, 1990.

[Mcdermott1982] D. MCDERMOTT 1982. A Temporal Logic For Reasoning About Processes and Plans In *Cognitive Science 6: pp. 101-155* .

[Molloy1982] M.K. MOLLOY, ET AL. 1982. Performance Analysis Using Stochastic Petri Nets. *IEEE Transactions on Computers, C-31, No.9, pp.913-917* .

[Murata1988] T. Murata. 1989. Petri Nets: Properties, Analysis, and Applications. In *Proc. IEEE–89, V77, Number 4, April 1989, pp. 541-576.*

[Narayanan1995] S. Narayanan, 1995. *A computational semantics of Aspect.* In Preparation.

[Nakhimovsky1988] A. Nakhimovsky. 1988. Aspect, Aspectual Class, and the Temporal Structure Of Narrative. In *Proc. ACL–88, V4, Number 2, June 1988, pp. 29-44.*

[Nielsen1981] M. Nielsen and G. Plotkin and G. Winskel, 1981. *Petri Nets, Event Structures, and Domains, Part I".* Theoretical Computer Science, V13, 1981.

[Norvig1990] P. Norvig and R. Wilensky, 1990. *A critical evaluation of commensurable abduction models for semantic interpretation.* Proceedings of the Thirteenth Internation Conference on Computational Linguistics, 3, Helsinki, Finland(1990), 225-230.

[Pear1988] J. Pearl, 1988. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, CA, 1988.

[Pear1994] J. Pearl, 1994. *A Probabilistic calculus of actions.* Uncertainty in AI, 1994.

[Resnick1993] P. S. Resnick, 1993. *Selection and Information: A class based approach to lexical relationships (Ph.D. Dissertation).* Computer and Information Science, University Of Pennsylvania, 1993.

[Rieger1975] C. Rieger, 1975. *Language Comprehension.* Readings in KR,1975.

[Rosenshein1985] J.S. Rosenschein, 1985. *Formal theories of knowledge in AI and Robotics.* New Generation Computing 3(4):345-357.

[Russell1994] S.J. Russell, J. Binder, and D. Koller, 1994. *Adaptive Probabilistic Networks.* UCB Tech Report, CSD-94-824, 1994.

[Russell1995] S.J. Russell and P. Norvig, 1994. AI: A Modern Approach. Prentice Hall Series in AI, 1995.

[Schacter1990] R. Schacter 1990. *Evaluating Influence Diagrams.* Readings in Uncertain Reasoning, Morgan Kaufmann, CA, 1990, 79 -90.

[Schank1977] R. C. Schank and R.P. Abelson 1977. *Script, Plans, Goals, and Understanding: An inquiry into human knowledge structures.* Hillsdale, NJ:Erlbaum 1977.

[Shastri1990] L. Shastri and V. Ajjanagadde, 1990. From Structured Associations to Systematic Reasoning. Tech Report 90-05, CIS, U.Penn 1990.

[Sowa1984] John Sowa, 1984. *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley 1984.

[Sperber1986] D. Sperber and D. Wilson, 1986. *Relevance: Communication and Cognition.* Harvard University Press, 1986.

[Talmy1985] Len Talmy, 1985. Lexicalization Patterns:semantic structure in lexical forms In *Language Typology and Syntactic Description, V 3, pp. 59-138* Cambridge University Press.

[Wilensky1983] R. Wilensky, 1983. *Planning and Natural Language Understanding.* Adddison Wesley, 1983.

[Wilensky1991] R. Wilensky, 1991. *Etending the lexicon by exploring subregularities.* IJCAI Workshop on Computational Approaches to Non-literal language, Tech Report CU-CS-550-91, University of Colorado at Boulder: 161-169.

[Wilks1978] Y. Wilks, 1978. *Making Preferences more active.* AI Journal 11, 1978.

[Word] Miller et al, 1990. *Wordnet: An on-line lexical database.* International Journal of lexicography 3, 4 (1990) 235-245.

[Wu1992] D. Wu, 1992. *A Probablistic basis for Natural Language Interpretation.* Ph.D. Dissertation Dept. Of Computer Science, UCB, 1992.

[Wu1993] D. Wu, 1993. *Estimating Probability Distributions over Hypothesis with Variable Unification.* IJCAI-93 procedings 1993, pp. 790-795.