

# Frame Activated Inferences in a Story Understanding Program

Peter Norvig

Division of Computer Science  
 Department of EECS  
 University of California, Berkeley  
 Berkeley, CA 94720

## ABSTRACT

An effective story understander must be able to reason about characters in the story, their affects, actions, plans, and goals, as well as the settings and important points of the story. In many systems this has been done with separate inference mechanisms for each class of knowledge structure. This paper proposes a story understander with a unified frame-based inference component used on each class of knowledge structure.

The frame-level inference mechanism is meant to be general enough to be applicable to a variety of tasks, not just story understanding. In fact, one implementation of the inference mechanism was built with Joe Faletti, and was used both by FAUSTUS and by Faletti's planning program, PANDORA. [Faletti82].

The story-understanding specific component is essentially an agenda system that decides what frame-based inferences to consider next at a given point. It can be characterized as a set of story understanding principles, as described in. [Wilensky83].

## Introduction

Early story-understanding programs were primarily concerned with understanding *events* in the real world, or the blocks world, or the restaurant world. Recently, researchers have shown the importance of *affect* [Dyer82] in the story understanding process. Other research addresses the problem of determining the *plot* units [Lehnert81] or main *point* [Wilensky82] of a story. Each of these systems introduces a new class of knowledge structure, and then introduces a new inference mechanism for that class. This research takes a different approach, proposing a single unified processing scheme which subsumes the need for specialized inference rules. The theory has been implemented in a program called FAUSTUS, (Frame Activated Unified STory Understanding System). FAUSTUS represents each class of knowledge structure as a frame, including objects such as chairs and people, plans such as ask and eat-at-restaurant, settings such as restaurant and supermarket, and points such as irony and goal-subsumption-state-termination. Of course, the frame for eat-at-restaurant is different *from* the frame for chair, but the same underlying processes are used to manipulate all types of frames.

The claim I am making is that many inferences that other systems have made through knowledge-specific inference rules are made "automatically" in FAUSTUS, due to the structure of frames in memory, and the primitives that manipulate the frames. Furthermore, more complex inferences needed for story understanding (or planning in Faletti's case) can be specified in terms of constraints on these primitives.

## The Frame Processing Mechanism

FAUSTUS's processing mechanism is summarized below. There are six basic processes which manipulate data in four different memory locations. Processing begins with the *story input*, which is parsed by PHRAN. The output from PHRAN is then used to index into the knowledge data base, which is a collection of assertions and associations organized into *frames* of related concepts. This is general world knowledge, independent of any particular story. Most concepts in the story will match parts of several frames. When this happens the concept is said to *invoke* these frames. Each invocation has a weight associated with it, and when a frame's accumulated invocation passes a threshold, the frame is *instantiated*. This means that an instance of the frame is created with default values replaced by those mentioned in the input. The set of instantiated frames is kept in the *active frame buffer*. Subsequent inputs are processed in the context defined by the frames in the active frame buffer; a new input may be found to fit in as a value of one of the active frames. This process of adding to an existing active frame is called *elaboration*.

## Components of the Model

FAUSTUS is composed of three main components, one for linguistic processing, one for frame-based inferences, and one for story-understanding specific inferences. This paper concentrates on the frame-based inferences. FAUSTUS does not address the problem of interaction between the first two levels; it calls directly on Arens' PHRAN [Wilensky80] program to generate the parse of each sentence. Thus, the program is *non-integrated*, in the sense used by Dyer. [Dyer81]. However it is *unified*, in the sense that one inference mechanism is used for all classes of knowledge structures.

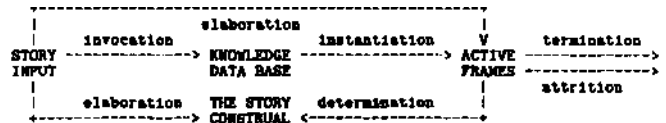


fig. 1

FAUSTUS's representation of the story is formed by choosing out of the active frame buffer those frames that it decides are appropriate to the story. This decision process is called *determination*, and frames that are so selected are called *determined* frames. The set of determined frames is FAUSTUS's *construal* of the story.

Because the content of the active frame buffer influences the processing of subsequent inputs, FAUSTUS has two processes that try to eliminate irrelevant frames from the buffer. *Termination* is the opposite of determination; it is the process of discarding an active frame because it is found to be inappropriate in some way. Finally, if a frame has not been determined after a sufficient passage of time, (where sufficient time *is* a function of activation) the process of *attrition* discards it. An important side-effect of this attrition is often the determination of some other frame. Often an input will invoke two or more frames, but no determination can be made among them. However, when all but the most highly activated frame is lost through attrition, that remaining frame becomes instantiated.

Thus, we have a two-step filter; while the inputs may invoke a large number of frames, only some of those are instantiated, and only some of the instantiated frames are determined to actually belong in the construal of the story. The invocation step is driven by activation and has an "analog" flavor, while the determination step relies on more "digital" evidence. There is no central control mechanism to decide what to do next. The instantiations and determinations are driven solely by associations within the frames themselves.

### Example

In the following example, I will show how the processes are applied to determine various classes of frames. Since the system hinges on the content of the frames in the knowledge data base, I will go into some detail in presenting these frames.

Input: Frank hated his job at the factory. He wanted a job where he wouldn't have to work so hard. He envied his friends who went to college, and didn't have to work. So Frank quit building cars and enrolled at the local University. However, as a student, he was soon working harder than he ever had in his life.

### 1. Object Frames

In the first sentence of the story, FAUSTUS determines an instance of the factory frame. Because FAUSTUS knows of several types of factories, it invokes each sub-type. At this point, there is not enough evidence to choose among them, so FAUSTUS makes no determination. When the fourth sentence is read, the phrase "building cars" matches the purpose "(manufacture (object car))" in the automobile-assembly-plant frame, and so invokes that frame. Because the frame was invoked earlier, there is now enough evidence to instantiate the frame. FAUSTUS has made the connection between the fairly abstract concept of factory and the more concrete concept of automobile assembly plant. Below are simplified versions of the factory and automobile-assembly-plant frame, and the elaborated instance representing the factory mentioned in the story.

```
(factory
  (ako institution)
  (kinds (list textile-mill automobile-assembly-plant ...))
  (instances (list factory-54 ...))
  (purpose (manufacture)))

(automobile-assembly-plant
  (ako factory)
  (instances (list factory-54 ...))
  (purpose (manufacture (object car))))

(automobile-assembly-plant
  (self factory-54)
  (employees (Frank)))
```

Of course, there are other possible construals of the story. Perhaps Frank worked at a textile mill, and he quit his hobby of building cars when he enrolled at the University. Such a construal is consistent both with the facts of the story, and with world knowledge. However, to arrive at such a construal would require spreading invocation to a larger number of sub-frames, and instantiating frames after a single weak invocation. FAUSTUS's instantiation and determination mechanisms guide it away from this and towards a simpler construal.

### 2. Plan and Goal Frames

The two frames below say that Frank's enrolling in school was a plan for having an easy job, but that the plan failed, he actually wound up with a difficult job. Of course, many story understanding programs are good at constructing frames just like these; what is interesting is the way they were arrived at. In the PAM (Plan Applying Mechanism) system for example (Wilensky 1983), there was an explicit explanation procedure that attempted to explain every action as a plan for a known goal. In FAUSTUS this same effect is accomplished by giving high priority to the tasks of elaborating the plan slot of each instance of a goal frame, and the goal slot of each instance of a plan frame. The same connection is found, but the explanation process has been demoted from the status of a procedure that was the main inner loop of the program to the simple assertion that plans and goals are important.

```
(goal
  (self goal-23)
  (actor Frank)
  (desired (occupation (actor Frank) (difficulty low)))
  (plan plan-37)
  (outcome (student (actor Frank) (difficulty highest)))
  (status failed))

(plan
  (self plan-37)
  (planner Frank)
  (action (enroll (actor Frank) (institution college-17)))
  (goal goal-23)
  (status executed))
```

### 3. Story Points Frames

So far we have seen that FAUSTUS can process a story, recognizing and elaborating frames of varying level of abstraction. What is needed is a mechanism for separating the important frames from the trivial ones. As mentioned in the introduction, several current programs have attacked this problem with a processing scheme separate from the normal inference process. FAUSTUS has integrated Wilensky's *story points* approach in an implementation that does not require new processing rules, just the same indication of importance that was used to find plan-goal relations.

For example, an important point in the story above is a goal-failure frame which says that Frank's goal was avoiding hard work, that he tried to achieve that goal, and that he failed. While this is true, and it is an important part of the story, it seems to miss the main point. More important is the irony in Frank's actions bringing him harder work when he was trying to avoid work. If the story were changed so that the last line read "However, as a student, he worked just about as hard as he did at the factory," then we would still have the same goal-failure point, but the story would not be ironic, and would not be as interesting.

An obvious solution is to introduce a new story point, the ironic-goal-failure frame. This frame would be a kind of goal-failure, with the provision that the outcome of the plan must be the opposite (in some way) of the desired goal state. While this approach would work, it would miss an important processing generalization; knowing about ironic-goal-failure would be of no help in detecting other instances of irony.

The approach I took was to try to detect instances of irony in general, rather than trying to enumerate special cases. Irony occurs when (1), there is a strong expectation, (2), there *is* a violation of that expectation, and (3), the concept which replaces the expected one is the opposite of the expectation. As it turned out, FAUSTUS was already capable of tracing expectations, so it was easy to have it create an instance of an expectation or expectation-violation frame. Note that these new frames are different in an important way from all the ones we have seen before; they are derived not from concepts mentioned explicitly in the story (like University and factory), nor from concepts inferred by FAUSTUS (like plans and goals), but from FAUSTUS' own internal processing behaviour.

FAUSTUS makes a distinction between two types of story points; *static* points, which are found by relating inferences to a stored frame marked as inherently interesting, and *dynamic* points, which are uncovered as the result of processing events like expectation violation. Irony, humour, and surprise would all be examples of dynamic points. This distinction is orthogonal to Wilensky's [Wilensky82] categorization of *external* and *content* points. Wilensky enumerates some content points, while Schank *tt ai* [Schank82] does the same for external points.

Below are two points detected by FAUSTUS, a static and a dynamic point. Either of them could be used to generate a summary of the story. For example, the first corresponds to "Frank wanted an every job. He enrolled in college. It was not easy. He didn't try anything. . else." If this were the main point of the story, the story would be incomplete; the subsequent-action slot of the goal-failure frame has yet to be elaborated. The story is incomplete from the point of view of a problem resolution episode. The second point corresponds to the summary "Frank enrolled in college. He thought being a student would be easy. Ironically, he ended up working harder than ever." This is a better summary because it is a completely elaborated frame; the story is complete from the point of view of an ironic episode.

(goal-failure

(actor Frank)  
(goal goal-23)  
(plan plan-37)  
(subsequent-action nil))

(ex pectation-violation

(expectation (student (actor Frank) (difficulty low)))  
(triggered-by plan-37)  
(violation (student (actor Frank) (difficulty highest))))

It was easy to add dynamic points because FAUSTUS's basic mechanism for processing expectations is so simple. Once expectation and expectation-violation frames were defined, they were easily handled by the standard frame manipulation processes. In a system with distinct processing schemes for each of several levels it would be more difficult to add expectations as manipulatable objects.

Of course, the analysis of irony is far from complete. There will surely be non-ironic episodes which fit the expectation-violation frame described above, and ironic episodes which do not. The concept of *opposite* is surprisingly complicated, and is another source of difficulty. FAUSTUS makes do with a very simple notion of opposite.

## Advantages of the Unified Approach

There are a number of reasons why this unified approach is advantageous, both as a cognitive model of story understanding, and as a methodology for developing a working program.

First of all, FAUSTUS has a flexible control structure. It is not constrained to making inferences in either a strictly top-down or bottom-up manner. This is important because certain instantiations and determinations can only be made from evidence acquired from several different levels. FAUSTUS is able to find this evidence, while a strictly top-down system, such as a text skimmer (e.g. [DeJong82]), must identify the correct script to process an input.

Unified systems have a certain economy that tends to make them easier to understand and to modify. Any improvements to the system immediately propagate to all types of inferencing. In a non-unified mechanism, an improvement to, say, the point-handling mechanism does nothing to improve the goal detection mechanism. Of course, the complexity has not disappeared; it has merely moved from the processor to the knowledge base. I feel that that is where it belongs, for the reasons stated here. The ease with which I was able to add dynamic points and use them to detect irony in stories supports this claim.

## References

DeJong82.

DeJong, Gerald, "An Overview of the FRUMP System," pp. 149-177 in *Strategies for Natural Language Processing*, ed. Martin H. Ringle, Lawrence Erlbaum Associates, Hillsdale, NJ (1982).

Dyer81.

Dyer, Michael G., "Integration, Unification, Reconstruction, Modification: An Eternal Parsing Braid," *Proceedings of the Seventh IJCAI*, pp. 37-42 (1981).

Dyer82.

Dyer, Michael G., "Affect Processing for Narratives," *Proceedings of the National Conference on Artificial Intelligence*, pp. 265-268 (1982).

Faletti82.

Faletti, Joseph, "PANDORA — A Program for Doing Commonsense Planning in Complex Situations," *Proceedings of the National Conference on Artificial Intelligence*, (1982).

Lehnert81.

Lehnert, Wendy, "Affect and Memory Representation," *Proceedings of the Third Annual Conference of the Cognitive Science Society*, pp. 78-83 (1981).

Schank82.

Schank, Roger C, Gregg C. Collins, Ernest Davis, Peter N. Johnson, Steve Lytinen, and Brian J. Reiser, "What's The Point," *Cognitive Science* 6(3) pp. 255-276 (1982).

Wilensky80

Wilensky, Robert and Yigal Arens, "A Knowledge-based Approach to Natural Language Processing," Memorandum No. UCB/ERL/M80/34, UC Berkeley ERL, Berkeley (1980).

Wilensky82.

Wilensky, Robert, "Points: A Theory of the Structure of Stories in Memory," pp. 345-374 in *Strategies for Natural Language Processing*, ed. Martin H. Ringle, Lawrence Erlbaum Associates, Hillsdale, NJ (1982).

Wilensky83.

Wilensky, Robert, "Memory and Inference," *Proceedings of the 8th IJCAI*, (1983).