STEREOTYPES as an ACTOR Approach Towards Solving the Problem of Procedural Attachment in FRAME Theories

Carl Hewitt

Abstract

This paper is a spin-off of our work on actors. We have worked out a dictionary for translating between what Minsky et. al. are saying about frames and what we are saying about actors. Using **PLASMA [PLANNER-like System Modeled on Actors]** we can demonstrate important relationships between the Minsky-frames and the PLANNER-like formalisms. PLASMA does not use the QA-4 context mechanism (Rulifson et. al. 1972): instead it uses explicit tags in assertions to keep track of the state of affairs in various situations. One problem with the QA-4 context mechanism is that the problem solver is forced to attempt to propagate all changes in the situation immediately on a frame shift since otherwise inconsistent information will be inherited from the previous situation. Another problem with QA-4 context mechanism is that it is sometimes difficult to reason explicitly about various situations using it because situations [frames] are not explicitly part of the assertions and goals. Events that are viewed from several different viewpoints [as in a murder mystery] are difficult to handle. Also it is difficult to retrieve the appropriate prior situations from memory to aid in recognition tasks using QA-4 contexts. However, without the example of the QA-4 contexts to guide us, we could never have realized how to deal with these problems using tagged assertions. The context mechanism in CONNIVER was modeled on the one in QA-4.

Actors make a contribution to the "declarative-procedure" controversy in that they subsume both the behavior of pure procedures [functions] and pure declaratives [data structures] as special cases. In this paper we use actors to investigate the question of how to do procedural attachment to frames [McCarthy 1969, Minsky et. al. 1974]. Actors provide an approach to solving to the problem of how to attach procedures to frames in such a way that the appropriate procedure is invoked with the inherited knowledge of the frame within which the procedure is incorporated, as well as newly introduced knowledge from other frames. Our approach also incorporates the insight gained from PLANNER-like formalisms [Hewitt 1969, 1971; Rulifson 1972; Davies 1972; Sussman and McDermott 1972, Hewitt et. al. 1973] for the procedural embedding of knowledge. We introduce **stereotypes** as an actor version of a frame theory. A stereotype consists of the following parts:

a collection of **characteristic objects**

characteristic relations for those objects

plans invoked by world directed invocation
for transforming the objects and relations

Overview

Apology! The schemes proposed herein are incomplete in many respects. First, I often propose representations without specifying the processes that will use them. Sometimes I only describe properties the structures should exhibit. I talk about markers and assignments as though it were obvious how they are attached and linked; it is not.

Marvin Minsky 1974

FRAMES like ACTORS are difficult to define and motivate. Through the dint of much effort [Greif and Hewitt 1975] actors are now becoming quite well defined [even axiomatized!]. In this paper we make a stab at similarly starting to pin down and unify various notions of "frames". We have observed a tendency for people to get hung up on extraneous details as they attempt to understand these difficult concepts. The notions of default values and expectations in particular are a cause of difficulty. For example expectation is often confused with scientific prediction. In attempting to sort all this out, we have found the methodology of CONJECTURES and REFUTATIONS as expounded by Karl Popper to be of great value.

The relationships between the assertional frames of McCarthy and the object frames of Minsky are explored in this paper using PLASMA. PLASMA allows us to demonstrate a kind of isomorphism between assertional frames and object frames. Using PLASMA accomplishes procedural attachment for both kinds of frames and shows how to move between object and assertional frames.

Assertional Frames and Semantic Nets

The example given below was originally used by Terry Winograd to illustrate the properties of Semantic Nets in his informal survey entitled "Five Lectures on Artificial Intelligence". Scott Fahlman is currently considering how to build a special purpose machine to solve such problems. We first give a formulation in terms of assertional frames. The addition of assertions of the form (... *never-are* ...) was inspired by Grossman [1975]. It is interesting to speculate whether the formulation in terms of assertional frames is as plausible a psychological model for humans as the formulation in terms of semantic nets.

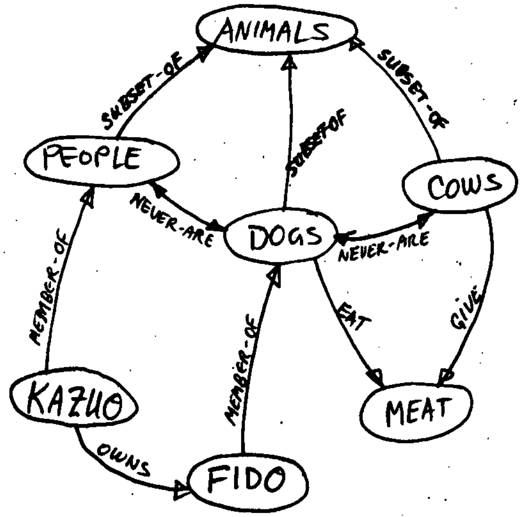
Formulated in Assertional Frames

```
(Kazuo ∈ people)                                   ;Kazuo is a person
((Kazuo owns Fido) in s0)            ;Kazuo owns Fido in situation s0
(Fido ∈ dogs)                                          ;Fido is a dog
(dogs eat meat)                                       ;dogs eat meat
(cows give meat)                                     ;cows give meat
(dogs never-are people)
(cows never-are people)
(dogs never-are cows)
(people ⊂ animals)                               ;people are animals
(dogs ⊂ animals)                                   ;dogs are animals
(cows ⊂ animals)                                   ;cows are animals

[to-demonstrate-animal-eats-sort-of-food ≡
          ;define a procedure to show that an animal eats a sort of food
   (to (demonstrate (=animal eats =sort-of-food)) consider-trying
      (demonstrate (animal ∈ =kind-of-animal)
         (then:
            (demonstrate (kind-of-animal eat sort-of-food)))))]
```

Formulated in Semantic Nets

Next we formulate the example in semantic networks.



Relationship of Worlds to Contexts

We conceive of **worlds** {Hewitt et. al. 1973} as actors that organize a body of knowledge for efficient use. Worlds are generalizations of the tree-structured contexts developed for QA-4 (Rulifson et. al. 1971) as a generalization of the data bases of PLANNER-69.

> *"A Frame is a collection of questions to be asked about a hypothetical situation; it specifies issues to be raised and methods to be used in dealing with them."*
>
> *"In fact we shall consider the idea that the frame terminals [for a scenario structure] are exactly those questions [commonly associated with it]."*
>
> *Marvin Minsky 1974*

Worlds have many of the behaviors attibuted to object frames {Minsky 1974} and assertional frames {McCarthy and Hayes 1969}. In particular worlds exhibit a very flexible form of inheritance of attributes which is realized through the mechanism of message passing.

Progressive Refinement of Plans

Part of Minsky's birthday party frame includes a PLANNER-style general plan for getting ready for a birthday party. It illustrates how PLANNER-style plans are a natural component of frame systems. In PLASMA the general plan can be expressed as follows:

```
[achieve-guest-attends-birthday-party-for-guest-of-honor ≡
   (to (achieve (=guest attends birthday-party for =guest-of-honor))
      (choose
         (=present is-suitable-present-for guest-of-honor from guest)
         (then:
            (achieve
               (∧
                  (guest has present)
                  (guest is-dressed-for birthday-party))
               (then:
                  (achieve
                     (guest is-present-at birthday-party
                        for guest-of-honor)))))))]
```

We use the above general plan as a template to write more particular plans tailored to special circumstance by progressive **refinement of plans** [Hewitt IJCAI-71] of the plan. This technique is an incremental glorified version of inline substitution of procedure bodies for their invocations which has been generalized to deal with pattern directed invocation. Cheatham and Wegbreit [1973] and Burstall and Darlington [1975] have studied the problems involved in the absence of pattern directed invocation.

We will suppose that early one morning that Marvin has been told that there will be a birthday party for Seymour the next evening. The above plan becomes specialized as follows:

```
[achieve-Marvin-attends-birthday-party-for-Seymour ≡
   (to
      (achieve (Marvin attends birthday-party for Seymour))
      (choose
         (=present is-suitable-present-for Seymour from Marvin)
         (then:
            (achieve
               (∧
                  (Marvin has present)
                  (Marvin is-dressed-for birthday-party))
               (then:
                  (achieve
                     (Marvin is-present-at birthday-party
                        for Seymour)))))))]
```

As Marvin rehearses his planned actions for the day his plan becomes further specialized. Eva, who is a friend of Seymour, probably will have suggestions as to what might be a suitable present for Seymour. Also, a party for Seymour is not the kind of affair that guests dress up for. These considerations result in the following refinement of the plan:

```
[Marvin-attends-birthday-party-for-Seymour ≡
      (ask (Eva which =present is-a-suitable-present-for Seymour
                  from Marvin)
         (then:
            (Marvin-gets-money (sufficient-to-buy present)
               (then:
                  (Marvin-buys present
                     (then:
                        (Marvin-goes-to (home-of Seymour)
                           at (time-of
                              (birthday-party-of Seymour)))))))))]
```

The above plans were formulated between yawns as Marvin awakened the morning of the party.

Procedural Attachment in Assertional Frames

> "*A situation* s is the complete state of the universe at an instant of time. We denote by Sit the set of all situations. Since the universe is too large for complete description, we shall never completely describe a situation; we shall only give facts about situations. These facts will be used to deduce further facts about that situation, about future situations and about situations that persons can bring about from that situation.
>
> ...
>
> *We shall further assume that the laws of motion determine from a situation all future situations. {This assumption is difficult to square with quantum mechanics, and relatively tells us that any assignment of simultaneity to events in different places is arbitrary. However, we are proceeding on the basis that modern physics is irrelevant to common sense deciding what to do, and in particular is irrelevant to solving the 'free will problem'.}"*
>
> *John McCarthy 1969*

In addition to making the assumptions listed above McCarthy and Hayes also assume the existence of a functional which maps a global state in to the "next" global state. Although we shall use tags on assertions and goals to relativize them to particular situations, hypotheses, and contexts we do not want to make any of the global assumptions of McCarthy and Hayes. All of our situations will be local and completely consistent with relativity and quantum mechanics. For a discussion of the issues involved in this decision for the actor model of computation see Greif and Hewitt [1975].

The Pattern Directed Invocation [Hewitt 1969] incorporated in PLANNER-like systems accomplishes procedural attachment for assertional frames [McCarthy 1969]. This procedural attachment is carried down to the level of the quantificational calculus where we have shown [Hewitt 1975a] how the logical operators of the quantificational calculus {∀, ∃, *implies*, ∧, ∨, ¬, etc.} can be behaviorally defined as actors. We use the following kinds of tags in assertions to relativize the assertions in the desired manner:

physical states
mental states
logical hypotheticals
hypotheses
view points
goal states
predictions
defaults

Stereotypes

We introduce **stereotypes** as an actor version of a frame theory. Our notion of a stereotype incorporates ideas from assertional frames [McCarthy 1969], actor worlds [Hewitt 1969, 1973], world-directed invocation [Hewitt 1969; 1971, 1973; Stansfield 1975]; social frames [Goffman 1974], and object frames [Minsky et. al. 1974]. A stereotype consists of a set of the following parts:

a collection of characteristic objects

characteristic relations for those objects

a set of plans invoked by world directed invocation
for transforming the objects and relations

It seems that many of the behaviors attributed to frames by Minsky can be realized by stereotypes. The characteristic objects of a stereotype correspond closely to the slots of a Minsky frame and the characteristic relations of a stereotype correspond to the constraints of a Minsky frame. Minsky calls simple unary characteristic relations **markers**. We instantiate stereotypes somewhat differently from the way in which Minsky instantiates object frames. Our approach is to plug in definite candidates for all the characteristic objects of a stereotype that we can and use anonymous objects for the rest. Defaults are done as assertions tagged to indicate that they are defaulted so that they can be easily displaced if an anomaly develops. Stereotypes communicate by making assertions in the data base and by world directed invocation which is a generalization of pattern directed invocation in which the invocation is done on the basis of a fragment of a micro-world instead of a single assertion. Inheritance of attributes is done using the message passing of actor semantics. If a questions cannot be answered directly then parts of the job are delegated.

Procedural Attachment for Stereotypes

We have found it useful to incorporate two types of worlds [called **UTOPIA** and **REALITY**] in every problem solving situation in order to incorporate goal orientation. The UTOPIA-REALITY machinery also enables us to incorporate **PLANNING** into problem solving using "islands" [Minsky 1963] as stepping stones. The utopia of one problem solving situation is taken as the reality of another.

> *"It will be worth a relatively enormous effort to find such 'islands' in the solution of complex problems. Note that even if one encountered, say, 106 failures of such procedures before success, one would still have gained a factor of perhaps 10^{10} in over-all trial reduction! Thus practically any ability at all to 'plan,' or 'analyze,' a problem will be profitable, if the problem is difficult."*
>
> *Marvin Minsky 1963*

Nested Continuation Control Structure Instead of CONNIVER-style Possibility Lists

Simple retrieval can be done using fragments of micro-worlds that consist of single patterns [pattern-directed retrieval][Hewitt 1969]. For example

```
(find (=apt is-an-apartment-in Cambridge)
                                    ;find an apartment in Cambridge
  (then:                                                      ;then
    (refute (apt is-an-acceptable-apartment)
                       ;find something wrong with the proposed apartment
      (else:                                                  ;else
        (move-into apt))))             ;move into the apartment
  (else:                                                      ;else
    (move-to Arlington)))                        ;move to Arlington
```

The use of "*then:*" and "*else:*" continuations seems to solve the scoping control problem which had been plaguing PLANNER-like languages for some time. CONNIVER attempted to solving the scoping control problem by introducing possibility lists and Landin-style non-hierarchical gotos. However possibility lists proved to have several deficiencies. They introduced side-effects into the basic communication mechanisms in CONNIVER which made it difficult for users to debug their programs since doing a try-next operation to print the next possibility destructively interferes with the operation of the programs being debugged. The other basic communication mechanisms of CONNIVER similarly have intrinsic side-effects built into their very structure.

> *"Their [Sussman and McDermott] solution, to give the user access to the implementation primitives of PLANNER, is however, something of a retrograde step (what are CONNIVER's semantics?), although pragmatically useful and important in the short term. A better solution is to give the user access to a meaningful set of primitive control abilities in an explicit representational scheme concerned with deductive control."*
> *Pat Hayes 1974*

Nested continuation control structure gives us the ability to influence or control any decision to the extent we desire. For example, in case of the apartment finder above, we can explicitly communicate complaints as to why a particular apartment is unacceptable in order to try to influence the selection of further proposed apartments.

```
[unacceptable-apartment-if-too-expensive ≡
   (to
      (refute (=apt is-an-acceptable-apartment)
            (with-complaint-dept: =the-complain-dept))
      (if ((rent apt) > $300)
         (then:
            (the-complaint-dept <= (rent-of apt is-too-high)))))]
```

```
[unacceptable-apartment-if-doesn't-have-dishwasher ≡
   (to
      (refute (=apt is-an-acceptable-apartment)
            (with-complaint-dept: =the-complain-dept))
      (if (apt has-a dishwasher)
         (else:
            (the-complaint-dept <= (apt doesn't-have-a dishwasher)))))]
```

Use of nested continuation control structure enables us to have the ability to control all of the decisions made while still retaining the high level goal oriented nature of PLANNER. PLASMA is able to accomplish this by basing its semantics on actor message passing and slightly changing the syntax of PLANNER-71. The change in syntax provides us with natural places to incorporate the control information and enables us to avoid the gratuitous side effects in PLANNER-71.

Unification of Pattern-Directed Invocation

We do not want to have to explicitly store every piece of knowledge which we have but would like to be able to derive conclusions from what is already known using procedures. Using the distinguished symbol *when* with the syntax

(*when* trigger *consider-trying* body)

or completely equivalently using the distinguished symbol *to* with the syntax

(*to* trigger *consider-trying* body)

creates a **plan** [high level goal-oriented procedure] that can be invoked by pattern directed invocation by a trigger which matches trigger. The following are all special case plans which are defined in terms of the above general pattern directed invocation machinery.

```
(to (demonstrate hypothesis) consider-trying body)
(when (assert statement) consider-trying body)
(to (disachieve condition) consider-trying body)
(when (deny statement) consider-trying body)
(to (refute hypothesis) consider-trying body)
(to (find description) consider-trying body)
(to (achieve condition) consider-trying body)
```

Additional kinds of plans can be defined as they are needed.

The process of invoking plans in worlds is controlled by recommendations made to the world when the plan is put in the world and by recommendations made at the site where a request is made of the world to achieve a particular goal. We envisage that problem solving would begin in a world with an initial class of plans. In many cases most of the plans that are used in the ultimate solution of the problem need to be constructed by other plans during the problem solving process. This is illustrated in a limited way by the domain of logic which is given as an example in Hewitt [IJCAI-75].

Pattern Directed Invocation Using Anonymous Objects

One important way in which plans can communicate is through making assertions, erasures, denials using worlds which they share in common. Another important means of communication is through pattern directed invocation. An important technical problem in implementing pattern directed invocation is how to solve the problem of matching the invoking pattern with the pattern in the trigger of the pattern of the plan being invoked. PLASMA uses anonymous objects to solve the problem. We assume the existence of a generator capable of generating new anonymous individuals anon$_1$, anon$_2$, etc. which have never before been encountered. To show the utility of such a generator consider the problem of proving $(x \subseteq z)$ [x is a subset of z] where we have a world which contains:

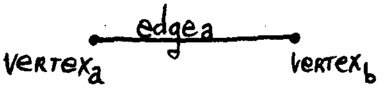
$(x \subseteq w)$
$(x \subseteq y)$
$(y \subseteq z)$

```
[demonstrate-⊆ ≡                        ;define the plan demonstrate-⊆ to be
   (to (demonstrate (=a ⊆ =c)) try
                                   ;to demonstrate that a is a subset of c try
      (demonstrate (a ⊆ =b)
            ;to demonstrate that a is a subset of another set [call it b]
         (then:                                                         ;then
            (demonstrate (b ⊆ c)
                                        ;demonstrate that b is a subset of c
               (using: set-theory)))
         (using: set-theory)))]
```

The problem is solved by "wishful thinking" i.e. reasoning within a hypothetical world. Unfortunately we do not have the fact $(x \subseteq z)$ explicitly given to us and so must do some computation. We note that we have a plan whose trigger $(=a \subseteq =c)$ matches what we are trying to achieve and so turn control over to it to see what it can do. In order to find b such that $(x \subseteq b)$ we let b be anon$_1$ which is a never before encountered individual which we wish to have certain properties. Then we note that anon$_1$ might be w. But we are unable to demonstrate $(w \subseteq z)$ so we reconsider and see that anon$_1$ might be y. We successfully demonstrate $(y \subseteq z)$ and so the problem is solved.

A Line Stereotype

Our first example of a stereotype is a very simple one. It is an ordinary line. A line-segment has three characteristic objects: one edge and two vertices.



A Line Segment

(vertex$_a$ *is-a-vertex-of* edge$_a$)
(vertex$_b$ *is-a-vertex-of* edge$_a$)
(edge$_a$ *is-the-edge-of* the-line)

There is not much that can be done with a line segment. However given one vertex, we ofthen need to be able to get the opposite vertex. A plan for doing this is given below:

```
[find-opposite-vertex ≡
   (to (find =v1 is-opposite =v2 along =e)
      (find
           (∧
              (v1 is-a-vertex-of e)
              (v1 ≠ v2))]
```

Face Stereotypes

The following are two common stereotypes of faces of blocks:

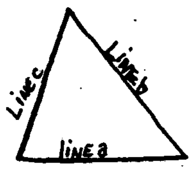
Rectangle **Triangle**

Two Common Face Stereotypes

In the remainder of this paper, we shall frequently refer back to the following stereotypes.

A Triangle Stereotype

A triangle stereotype contains three objects: all of which are lines.

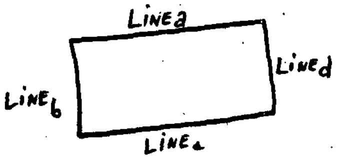


The important characteristic relation in the triangle stereotype is the relationship of that all three lines join each other.

(the-triangle ∈ convex-polygons)
i∈{a ... c} ***implies*** (line$_i$ *is-an-line-of* the-triangle)
(line$_a$ ***joints*** line$_b$)
(line$_a$ ***joints*** line$_c$)
(line$_b$ ***joints*** line$_b$)

A Rectangle Stereotype

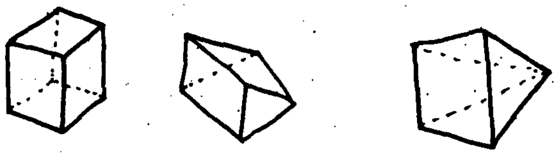
A rectangle stereotype contains four objects: all of which are lines.



(the-rectangle ∈ convex-polygons)
i∈{a ... d} ***implies*** (line$_i$ *is-a-line-of* the-rectangle)
(line$_a$ ***joins*** line$_b$)
(line$_b$ ***joins*** line$_c$)
(line$_c$ ***joins*** line$_d$)
(line$_d$ ***joins*** line$_a$)
(line$_a$ ***is-parallel-to*** line$_c$)
(line$_b$ ***is-parallel-to*** line$_d$)
(line$_a$ ***is-opposite*** line$_c$)
(line$_b$ ***is-opposite*** line$_d$)

Block Stereotypes

The following are three common stereotypes of blocks:

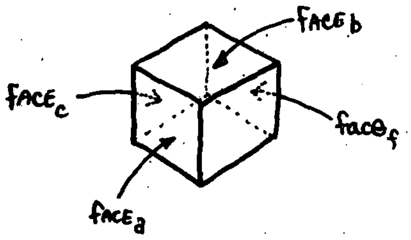


Cube **Wedge** **Square-Pyramid**

Three Common Block Stereotypes

A Cube Stereotype

A cube stereotype contains six objects [called faces] which are squares.



Two important characteristic relations in the cube stereotype are the relationship of two faces being adjacent two each other and the relationship of two faces being opposite each other.

(the-cube $\in$ convex-polyhedra)

$i \in \{a \ldots f\}$ *implies* (face$_i$ *is-a-face-of* the-cube)

$i \in \{a \ldots f\}$ *implies* (face$_i$ *is-square*)

(face$_a$ *is-opposite* face$_f$)

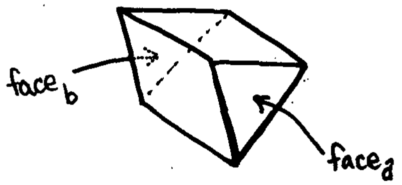
(face$_a$ *is-adjacent-to* face$_b$)

(face$_a$ *is-adjacent-to* face$_c$)

(face$_b$ *is-adjacent-to* face$_c$)

A Wedge Stereotype

A wedge stereotype contains five objects [called faces], three of which are rectangles and two of which are triangles. The two triangular faces are opposite each other.



(the-wedge $\in$ convex-polyhedra)

$i \in \{a \ldots e\}$ *implies* (face$_i$ *is-a-face-of* the-wedge)

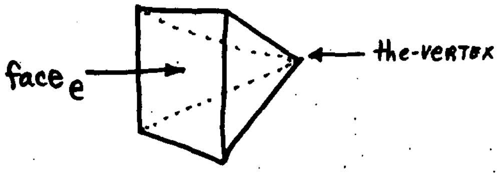
(face$_a$ *is-triangular*)

(face$_b$ *is-triangular*)

(face$_a$ *is-opposite* face$_b$)

A Square-Pyramid Stereotype

A square-pyramid stereotype contains six objects: four of which are triangular faces, one is a square face, and one is a vertex.



(the-square-pyramid $\in$ convex-polyhedra)

$i \in \{a \ldots e\}$ *implies* (face$_i$ *is-a-face-of* the-square-pyramid)

$i \in \{a \ldots d\}$ *implies* (face$_i$ *is-triangular*)

(face$_e$ *is-the-base*)

(the-vertex *is-opposite* face$_e$)

Stereotyped Views of Models of Blocks

A stereotype for a typical view of a cube stereotype contains three sides, one Y vertex, three arrow vertices, and three L vertices.

((side$_a$ *is-a-view-of* face$_a$ *of* block$_1$) *in* the-view)

((side$_b$ *is-a-view-of* face$_b$ *of* block$_1$) *in* the-view)

((side$_c$ *is-a-view-of* face$_c$ *of* block$_1$) *in* the-view)

((side$_a$ *is-left-below* side$_c$) *in* the-view)

((side$_b$ *is-right-below* side$_c$) *in* the-view)

((v$_a$ *is-a-Y-vertex*) *in* the-view)

((v$_b$ *is-an-arrow-vertex*) *in* the-view)

((v$_c$ *is-an-L-vertex*) *in* the-view)

((v$_d$ *is-an-arrow-vertex*) *in* the-view)

((v$_e$ *is-an-L-vertex*) *in* the-view)

((v$_f$ *is-an-arrow-vertex*) *in* the-view)

((v$_g$ *is-an-L-vertex*) *in* the-view)

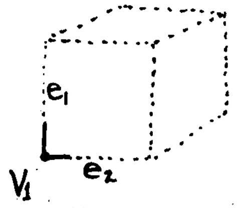
Stereotype for Typical View of a Cube

Recognition of Stereotypes

We make progress if, and only if, we are prepared to learn from our mistakes.
Karl R. Popper

Philosophically, this section builds on some work by Karl Popper on Conjectures and Refutations as a scientific methodology. More concretely it builds on a fine piece of work Ben Kuipers did on recognition of kinds of blocks [Kuipers 1974]. The reader is referred to Kuipers' paper for various kinds of discussion and background which will not be repeated here. The recognition problem is to incrementally recognize the above three kinds of blocks. This kind of problem has been extensively investigated by robot projects at the M.I.T. Artificial Intelligence Laboratory, the Stanford Artificial Intelligence Laboratory, and the S.R.I. Robot Project. In order to facilitate comparison, we have fairly closely followed Kuiper's scenario. Our motivation in re-exploring the block recognition problem is to investigate some techniques that have been developed by Marilyn McLennan for the very difficult domain of understanding plant pictures. We wanted to investigate a technique that she is developing for resolving conflicts between conflicting hypotheses recognition problems in another domain.

We have modified Kuipers scenario of the recognition of a block using stereotypes by using three-dimension stereotypes instead of the two-dimensional views which he used. These three-dimensional stereotypes are reminiscent of some of the three-dimensional "models" used in the robot vision programs. We start the recognition with an initial vertex v$_1$ (see figure below), which in this case happens to be an L-vertex. Our initial hypothesis h is that the drawing represents a cube as indicated by the dotted lines.

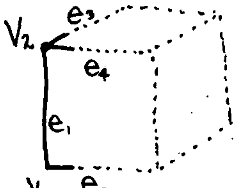


Vertex 1

We have the following situation:

(v_1 *is-a-vertex-of* e_1)
(v_1 *is-a-vertex-of* e_2)
(v_1 *is-an-L-vertex*)

((block$_1$ *is-a* cube) *in* b)
((v_1 *corresponds-to* v_g) *in* h)

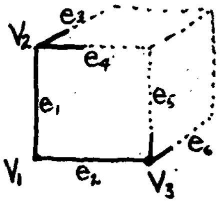


Vertex 2

We now have the following information:

(v_2 *is-an-arrow-vertex*)
(v_2 *is-a-vertex-of* e_1)
(v_2 *is-a-vertex-of* e_3)
(v_2 *is-a-vertex-of* e_4)
((v_2 *corresponds-to* v_b) *in* h)

The second vertex observed fits in completely with the hypothesis h.

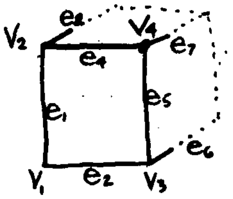


Vertex 3

(v_3 *is-an-arrow-vertex*)
(v_3 *is-a-vertex-of* e_2)
(v_3 *is-a-vertex-of* e_5)
(v_3 *is-a-vertex-of* e_6)

((v_3 *corresponds-to* v_f) *in* h)

This vertex fits the hypothesis h of the cube stereotype, since it is the anticipated arrow vertex.



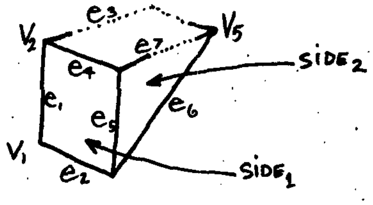
Vertex 4

We assume that the visual primitives that we are using are region-oriented as well as being line-oriented so that we can directly detect and recognized the shape of certain kinds of regions.

(side$_1$ *is-a-parallelogram*)
(v_4 *is-a-Y-vertex*)
(v_4 *is-a-vertex-of* e_4)
(v_4 *is-a-vertex-of* e_5)
(v_4 *is-a-vertex-of* e_7)

((v_4 *corresponds-to* v_a) *in* h)
((side$_1$ *is-a-view-of* face$_a$ *of* cube$_1$) *in* h)

The Y-vertex at the center of the figure also corresponds completely with the cube hypothesis h. A complete parallelogram face has now been observed and confirmed.



Vertex 5

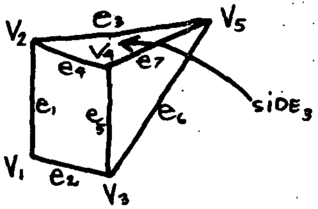
(side$_2$ *is-a-triangle*)
(v_5 *is-an-arrow-vertex*)
(v_5 *is-a-vertex-of* e_6)
(v_5 *is-a-vertex-of* e_7)

Confrontation and Refutation

With this observation, the cube hypothesis h finally breaks down. The anomaly occurs because the assertion that side$_2$ is a triangle conflicts with the characteristic relation of the cube stereotype that all of its faces are squares. Thus there is no easy to resolve the anomaly by simply rotating the cube. However, the cube stereotype is reluctant to give up completely so it imagines one of its sides shrunk down to an edge to fit the data. This reminds cube stereotype of the the wedge stereotype, which it suggests. The triangular side, in effect, caused the recognition system to do a "double-take" creating hypothesis h' that the block is a wedge since the wedge stereotype is happy with the situation.

(h' *obtained-by-refutation-of* h)
((block$_1$ *is-a* wedge) *in* h')
((side$_1$ *is-a-view-of* face$_c$ *of* wedge$_1$) *in* h')
((side$_2$ *is-a-view-of* face$_a$ *of* wedge$_1$) *in* h')

At this point, with the wedge stereotype directing the exploration, there is only one remaining edge. Unfortunately it refutes the hypothesis h'.



Vertex 5 Continued

(v_5 *is-a-vertex-of* e_3)
(side$_3$ *is-a-triangle*)

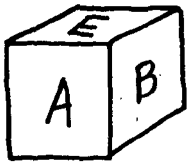
The observation that side$_3$ is a triangle conflicts with the characteristic relation of the **wedge** stereotype that a wedge does not have two triangular faces which are adjacent. Again no amount of rotation will help at all. However the triangular face side$_3$ suggests squeezing the edge of the wedge down to a point suggesting a pyramid. The pyramid stereotype takes a look at side$_1$ and decides that the square-pyramid stereotype should be invoked.

(h" *obtained-by-refutation-of* h')
((block$_1$ *is-a* square-pyramid) *in* h")
((side$_1$ *is-n-view-of* face$_e$ *of* square-pyramid$_1$) *in* h")
((side$_2$ *is-a-view-of* face$_a$ *of* square-pyramid$_1$) *in* h")
((side$_3$ *is-a-view-of* face$_b$ *of* square-pyramid$_1$) *in* h")

Since there is no further input data to be considered and thus further processing yields no refutations, hypothesis h' temporarily survives.

Tracking the Image of a Cube

In this section we shall consider an example due to Minsky from the point of view of stereotypes. We assume that the result of looking at the cube pictured below



view$_1$

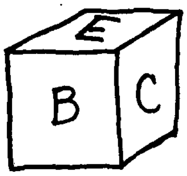
is the following symbolic description:

((A *has-color* red) *in* view$_1$)
((B *has-color* white) *in* view$_1$)
((E *has-color* white) *in* view$_1$)
((A *is-left-below* E) *in* view$_1$)
((B *is-right-below* E) *in* view$_1$)
((A *has-shape-of* vertical-parallelogram) *in* view$_1$)
((B *has-shape-of* vertical-parallelogram) *in* view$_1$)
((E *has-shape-of* parallelogram) *in* view$_1$)

The above view represents the situation below:

(view$_1$ *is-a-view-of* situation$_1$)
(A *is-a-view-of* face$_a$ *of* cube$_1$)
(B *is-a-view-of* face$_b$ *of* cube$_1$)
(E *is-a-view-of* face$_c$ *of* cube$_1$)

When we move to the right, face A disappears from view, while the new face decorated with C is now seen.



view$_2$

The new view has the following symbolic description:

(view$_2$ = (*move-to-right-around* cube$_1$ *from* view$_1$))
((C *has-color* white) *in* view$_1$)

Of course the situation of the cube has not changed but we have the following additional information:

((C *has-color* white) *in* situation$_1$)
(C *is-a-view-of* face$_f$ *of* cube$_1$)

Now if we imagine moving back to the left, we can hypothesize the view without any perceptual computation at all. We simply imagine that the new view view$_3$ is the same as view$_1$.

((view$_3$ = view$_1$) *in* hypothesis$_1$)

However, moving back to the left we are surprised to find that in the new view [view$_3$] that A has changed color to white! Thus hypothesis$_1$ is rejected. We also notice that

((floor *has* flecks-of-white-paint) *in* view$_3$)

which causes us to construct another hypothesis:

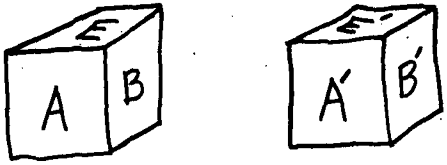
(situation$_2$ = (*paint* A white *in* situation$_1$))
((view$_3$ *is-a-view-of* situation$_2$) *in* hypothesis$_2$)

Further careful observation and testing does not refute hypothesis$_2$, so situation$_2$ can inherit suitably transformed attributes from situation$_1$ using the hypothesized paint transition.

More on Inheritance of Attributes

The "frame" problem of McCarthy for assertional frames corresponds closely to the problems of "inheritance of attributes" and "default values" for object frames. For example the fable quoted by Minsky about the wolf and the lamb is very close to the frame problems of McCarthy.

Recently Keith Nishihara has reconsidered the problem of gluing blocks together from the point of inheritance of attributes. Suppose that there actually were two cubes in the first situation considered above



view$_4$ of situation$_1$

so we have the following additional assertions:

(view$_4$ *is-a-view-of* situation$_1$)
(population$_{\text{situation}_1}$ blocks) = 2
((A' *has-color* red) *in* situation$_1$)
((B' *has-color* white) *in* situation$_1$)
((E' *has-color* white) *in* situation$_1$)
(A' *is-a-view-of* face$_a$ *of* cube$_2$)
(B' *is-a-view-of* face$_b$ *of* cube$_2$)
(E' *is-a-view-of* face$_c$ *of* cube$_2$)

Now we can create situation$_4$ by gluing side C of cube$_1$ to side A' of cube$_2$ to give block$_1$.

(population$_{\text{situation}_4}$ blocks) = 1

(situation$_4$ *results-from* (*glue* C *to* A' *in* situation$_1$))

((block$_1$ ∈ blocks) *in* situation$_4$)

((mass block$_1$) = ((mass cube$_1$) + (mass cube$_2$)))

(((face$_a$ *of* cube$_1$) *is-a-face-of* block$_1$) *in* situation$_4$)

(((face$_a$ *of* cube$_1$) *has-color* red) *in* situation$_4$)

Note that some attributes in situation$_4$ such as the color of A are inherited directly from one of the cubes in situation$_1$. On the other hand the mass of block$_1$ is inherited as the sum of the masses of cube$_1$ and cube$_2$.

Real World Recognition

The principal reason why vision programs at present perform so poorly is that the amount of knowledge they can bring to bear on the seeing process is so limited. For example, Waltz's program [Waltz 1972] is the latest in a line of development called scene analys, which was originated by Guzman [1968], and pursued by Huffman [1970] and Clowes [1971]. The usefulness of this approach is called into question by the difficulty of extracting, from information about intensity and color, the near perfect line drawing that such programs require; and by the restricted nature of the line drawing representation itself. Waltz has demonstrated that when a more detailed categorization of the kinds of labels is made that the number of ambiguous line drawings decreases dramatically.

We greatly admire Waltz's program, but we feel that the approach that it embodies is open to several criticisms. The first is that the knowledge that it uses is in a certain sense not explicit enough. Although it contains a great deal of information about the appearance of line drawings, this information is essentially in a compiled form: one reflection of this is that the structure of Waltz's program makes it inherently unable to use either explicit information about the three-dimensional form of what is being viewed, or the many pieces of special and general knowledge that we surely bring to bear on the process of seeing. Essentially, the only way one can attempt to influence the program is by adding or deleting junctions from a large table of "legal" labellings. There is no way in which pieces of its knowledge can be pulled out and examined while it tries to create an interpretation of, for example, a scene in which several lines are missing. Unless such knowledge, suitably embedded in a hypothetico-deductive system, can play a large part in the operation of a vision program, we see no prospect of such a program being able to interpret the incomplete information that is the diet of daily life.

The basic trouble with the labelling approach of scene analysis is that it is too limiting and stultifying a paradigm for vision, in much the same way that resolution is for deduction. The fundamental principle of resolution, that (¬ A) and (A ∨ B) together imply B, is occasionally useful. But attempting to make a uniform resolution proof procedure, to mechanize deduction in a way that cannot be very sensitive to hints, hunches, and a wide variety of higher level knowledge about the particular domain in question, is a cul-de-sac. Similarly, the line and vertex labels are local predicates that are occasionally useful, and are of some mathematical interest in their own right: but the problem of creating a uniform procedure to label arbitrary line drawing is not a central one for vision. Hence, we believe that the kind of knowledge contained in Waltz's program is probably relatively unimportant; and that the way in which it is made available there is certainly too restricting.

The proper endeavor of vision research is to decide what knowledge should be used to help a vision system to see, and to discover methods that make it possible to use such knowledge. How can one pursue this goal more effectively? There are several kinds of answers. The first is to abandon the restrictive format of line drawings, so that programs can use information about visual features that are not coded in this form. There is a large gain to be had by loosening up our formalisms to incorporate the many pieces of special and general knowledge that are necessary for purposeful vision in the real world. Vision is primarily a utilitarian function. In order to see properly it is necessary to know what kind of information is sought.

We believe that progress in particular domains of recognition on the following problems will pay large dividends:

> Getting Started: How to proceed from a set of feature clues to more global hypotheses. How to incorporate parallelism into this process.
>
> Keeping Going: How to recognize and mechanize confrontations between conflicting hypotheses. How to retain and use valid information that was incorporated in hypotheses that have been refuted.

Further Work

The PLASMA system described in this paper is currently being implemented at the MIT Artificial Intelligence Laboratory. The best version which currently runs was coded by Howie Shrobe. A better humanly engineered version has been designed and coded by Carl Hewitt in PLASMA with the extensive aid of Marilyn McLennan. At the time of this writing [March 1975] the new implementation is being translated into LISP by the MIT laboratory course 6.893: "Implementation and Application of Actor Systems". A rough draft of a primer by Brian Smith and Carl Hewitt for the new implementation exists.

Acknowledgements

The research reported in this paper was sponsored by the MIT Artificial Intelligence and Laboratory and Project MAC under the sponsorship of the Office of Naval Research.

The main example in this paper is adapted from a very niece piece of work by Ben Kuipers. The major change we made in his scenario was to introduce three-dimensional models and to try to reason as much as possible in three-dimensional terms. Marilyn McLennan, Ben Kuipers, and Candy Bullwinkle made helpful comments and criticisms which considerably improved the intelligibility and content of this paper. The progress we have made on actors would have been completely impossible without the contributions and questions of numerous MIT students. Ben Kuipers, Howie Shrobe, Keith Nishihara, Brian Smith, Aki Yonezawa, Richard Steiger, and Peter Bishop, and Irene Greif have done much of the work in making actors intelligible and relevant to the problems of constructing knowledge-based systems. Irene Greif is doing exciting work in extending behavioral semantics to the area of inter-process communication and putting semantics on a more solid foundation. Howie Shrobe, Brian Smith, Todd Matson, Roger Hale, and Peter Bishop have contributed to the multitude of "throw-away" implementations through which have we debugged many of our ideas.

This paper builds on extensive previous work on frames, paradigms, and hypotheses done by Fahlman, Hewitt, Kuhn, Kuipers, McCarthy, Minsky, Nishihara, Polya, Popper, and Winograd. Our notion of a stereotype is strongly reminiscent of the the logical notion of a formal system. However, stereotypes are used quite differently. Putting tags on assertions as advocated in this paper to record the state of affairs in various local situations seems more powerful and flexible than the context mechanism invented by Rulifson for QA-4 and later adopted by CONNIVER. Tags on assertions have previously been used to a limited extent in PLANNER-69 [MICRO-PLANNER] by Charniak and Biss to record situations. They are closely related to the global situations [frames] of McCarthy. The criticism of the line-labelling approach to vision comes from Marr and Hewitt 1973. Many of the ideas have emerged from the MIT course 6.893: "Implementation and Application of Actor Systems" given in the spring of 1975 with the following participants: Russ Atkinson, Mike Freiling, Kenneth Kahn, Keith Nishihara, Marilyn McLennan, Howie Shrobe, Kathy Van Sant, and Aki Yonezawa.

Bibliography

Biss, K.; Chien, R.; Stahl, F.; and Weissman, S. "Semantic Modeling for Deductive Question-Answering" Proceedings of IJCAI-73. August 1973. Stanford. pp. 356-363.

Burstall, R. M. and Darlington, J. "Some Transformations for Developing Recursive Programs" International Conference on Reliable Software. Los Angeles. April, 1975.

Cheatham, T. and B. Wegbreit, "A Laboratory for the Study of Automating Programming" SIGSAM Bulletin, Vol. 21, January, 1972.

Davies, D. J. M. "POPLER: A POP-2 PLANNER" MIP-89. School of A-I. University of Edinburgh.

Davies, D. J. M. "Representing Negation in a PLANNER System" Proceedings of AISB Conference. July, 1974. University of Sussex.

Greif, I. and Hewitt, C. "Actor Semantics of PLANNER-73" Proceedings of ACM SIGPLAN-SIGACT Conference. Palo Alto, California. January, 1975.

Grossman, R. W. "Representing the Semantics of Natural Language as Constraint Expressions" MIT Artificial Intelligence Working Paper 87. January, 1975.

Hayes, P. J. "Some Problems and Non-Problems in Representation Theory" Proceedings of AISB. Sussex. July, 1974.

Hewitt, C. "PLANNER: A Language for Manipulating Models and Proving Theorems in a Robot" IJCAI-69, Washington, D. C. May 1969.

Hewitt, C. "Procedural Embedding of Knowledge in PLANNER" IJCAI-71. London. Sept, 1971.

Hewitt, C. "Procedural Semantics" in Natural Language Processing Courant Computer Science Symposium 8. Randall Rustin, editor. Algorithmics Press. 1971.

Hewitt, Carl et. al. "Actor Induction and Meta-evaluation" Conference Record of ACM Symposium on Principles of Programming Languages. Boston. Oct, 1973.

Hewitt, C. "Protection and Synchronization in Actor Systems" M.I.T. working paper. November, 1974.

Hewitt, C. "The Semantics of ACTIONS and the Semantics of TRUTH" IJCAI-75. Tiblisi, USSR. Sept. 3-8, 1975.

Kuhn, T. The Structure of Scientific Revolutions Univ of Chicago Press. (2nd. Ed.) 1970.

Kuipers, Ben. "An Hypothesis-Driven Recognition System for the Blocks World" MIT-AI Working Paper 83. March, 1974.

Marr, D. and Hewitt, C. "VIDEO ERGO SCIO: An essay on some thing we would like a vision system to know" M.I.T. A.I. Lab Working Paper 60. November, 1973.

McCarthy, J. "Programs with Common Sense" Proceedings of the Symposium on the Mechanization of Thought Processes. Teddington. HMSO. London.

McCarthy, J. and P. Hayes, "Some Philosophical Problems From the Standpoint of Artificial Intelligence" Machine Intelligence Vol. 4. American Elsevier Publishing Company. New York. 1969.

McDermott, D. "Assimilation of New Information by a Natural Language Understanding System". MIT AI Memo 298. March, 1974.

McLennan, M. "Hypothesis Formation as a Method of Understanding Plant Pictures" Thesis Progress Report for Fall 1974. Private Communication.

Minsky, Marvin. "A Framework for Representing Knowledge" MIT AI Memo 306. June 1974.

Moore, J. and Newell, A. "How can MERLIN understand?" Department of Computer Science. Carnegie-Mellon University. 1973.

Popper, K. Conjectures and Refutations Basic Books. 1962.

Rulifson Johns F., Derksen J. A., and Waldinger R. J. "QA4: A Procedural Calculus for Intuitive Reasoning" Ph. D. Stanford. November 1972.

Smith, B. C. and Hewitt, C. "Towards a Programming Apprentice" Proceedings of AISB. Sussex. July, 1974.

Stansfield, J. L. "Programming a Dialogue Teaching Situation" Ph.D. Thesis, Edinburgh: University of Edinburgh. January, 1975.

Waltz, D. L. "Generating Semantic Descriptions from Drawings of Scenes with Shadows" M.I.T. Technical Report TR-271. November, 1972.

Winograd, T. "Five Lectures on Artificial Intelligence" Stanford A.I. Memo. September, 1974.