# LEVELS OF PATTERN DESCRIPTION IN LEARNING[1]

Elliot M. Soloway and Edward M. Riseman
Computer and Information Science Department
University of Massachusetts
Amherst, Massachusetts 01003

Abstract

A learning system in a complex, real-world domain will require a significant amount of knowledge to be used in order to (1) deal with large numbers of features, most of which are irrelevant, and (2) find similarities between the concepts that are inferred from the observed data. Use of knowledge-free, syntactic approaches to generalization in complex environments will result in a combinatorial explosion in the number of possible generalizations. Moreover, the important semantic features are not "in" the data; rather they must be hypothesized using prior knowledge.

The learning system described in this paper uses a multi-level knowledge-directed approach in order to cope with these problems. This paradigm is explored in the action-oriented game of baseball. The system attempts to interpret observed activity in terms of general knowledge provided about competitive games. This approach to learning can be viewed as a type of recognition, where the level of initial knowledge is general and where the specific observations mold a particular structure from the general knowledge. The system is organized into multiple levels of pattern descriptions, processing, and knowledge, reflecting the logical structure of the problem. In moving through those levels of description, the system filters out irrelevant features, hypothesizes additional semantic features (goals and relationships) and forms a hierarchy of generalized classes that extract the similarities in the descriptions. Examples of learning by a working computer program are presented.

## I. Introduction

### 1.1 Learning as Knowledge-Directed Interpretation or Recognition

Based on our experience with a working computer system, we shall discuss several important issues involved in the tasks of learning in a complex real-world problem. The approach we have taken to learning is eloquently expressed in the following quote taken from Jordan's [1968] commentary on Sir Aired North Whitehead's view of the nature of explanation and generalization.

> "Faith in reason should not totter in the face of incoherence. Observers on Mars, without our knowledge, have planted a 'probe' with television cameras and are now watching a game of rugby football being played in England. They want some explanation of what the cameras are recording which will cohere with their general theories of what happens on our planet. The ranges of the cameras are not powerful enough for the Martians to see the ball; it appears that a lot of men in patterned shirts are performing a dance or orgy. The Martians' attention is drawn to the goalposts. They connect these with similarly shaped objects to be seen on the roofs of some nearby houses. Now it is to be imagined that the Martians understand religious notions but have no sporting instincts. They conclude that the game is a religious dance rite and that the buildings with the H-shaped signs are temples.
>
> The roof signs, of course, are television aerials, and their resemblance to rugby goalposts are accidental. The Martians are wildly mistaken. But their guess illustrates cohesion. They are trying to find meanings in the things seen which will lie together in a harmony that excludes the merely arbitrary. This is precisely the nature of the philosopher's faith in reason, a faith widely asserted in spite of the frustrations to which the above fantasy points."

One interprets and thereby understands new situations in the world in terms of the frame of reference that one brings to the learning situation. For example, our curious Martian friends used their understanding of religion and religious ceremonies to focus on specific features in the environments under observation; they attended to the T.V. antennas and the goalposts rather than the thousands of other features in the "countryside scene" and the "athletic-contest scene." They interpreted these features in the context of religion and then classified both scenes as similar, i.e., as different aspects of a religious ceremony. The knowledge which the Martians used to perceive the world permitted this classification — religion wasn't "in" the observations, but rather, religion was "in" the heads of the Martians.

The form of learning discussed above can alternatively be described as a type of recognition. The Martians recognized various features in the observations as examples of a religious ceremony. The difference between recognition in a learning situation and recognition in speech understanding or scene analysis is the degree of detail in the knowledge which the system initially possesses. In the learning situation that knowledge often is very general and not tuned to the specific observations. In the perceptual tasks, knowledge of more specific details is usually

[1] This work was supported by ARI grant DAHC19-76-G-0013.

necessary in order to achieve minimal levels of performance. In both cases recognition of examples of stored concepts must occur. This view of learning as a form of recognition can be traced back to the Greek philosophers, e.g., Plato [Jowett, 1949].

### 1.2 Use of a Multi-Level Organization for Learning

In order to explore knowledge-directed learning, we have built a computer system that observes human activity in the domain of action-oriented games, namely, baseball. The system discovers some of the concepts and structure in that game. A hierarchical network is constructed which relates the similarities in the acquired concepts at various levels of abstraction. For example, the system moves from observing actions such as catch, run, and throw to inferring concepts such as infield single and groundout, to ultimately understanding those acquired concepts as examples of more general classes such as "hits" and "outs."

The organization of the knowledge, processing, and pattern description is decomposed into the levels shown in Figure 1. (Unless otherwise noted, all further references to level numbers address Figure 1). The logical structure of the problem is captured in the multiple levels of pattern descriptions, and in the corresponding knowledge and processing components which operate on those descriptions. Each level of description of the problem has some meaningful interpretation in the problem domain. For example, level 3 (Figure 1) represents the segmentation of the observations into episodes of high activity cvcles. Level 5 attempts to describe the goals and causally related interactions of the players in the game.

Correspondingly, there are levels of knowledge and processes provided to the system that facilitate the successive transformations in the behavior descriptions. The details of the description at each level serve to make explicit the input-output relationships of the processes which are to perform the transformation and the type of knowledge that must be employed. Some knowledge provides an understanding of spatio-temporal activity independent of a game context. Other more general knowledge about the types of goals and action relationships often found in competitive games, is used to make inferences (hypotheses) about the specific goals in the observed activity. Such hypotheses represent a description of the activity at a level far removed from the actual perceptions of the physical events at level 1. Finally, the processes which use the levels of knowledge in order to achieve the various levels of description are task independent and represent a general paradigm for knowledge discovery (see fCollins, 1976]).

This highly structured organization also facilitates the integration and subsequent use of

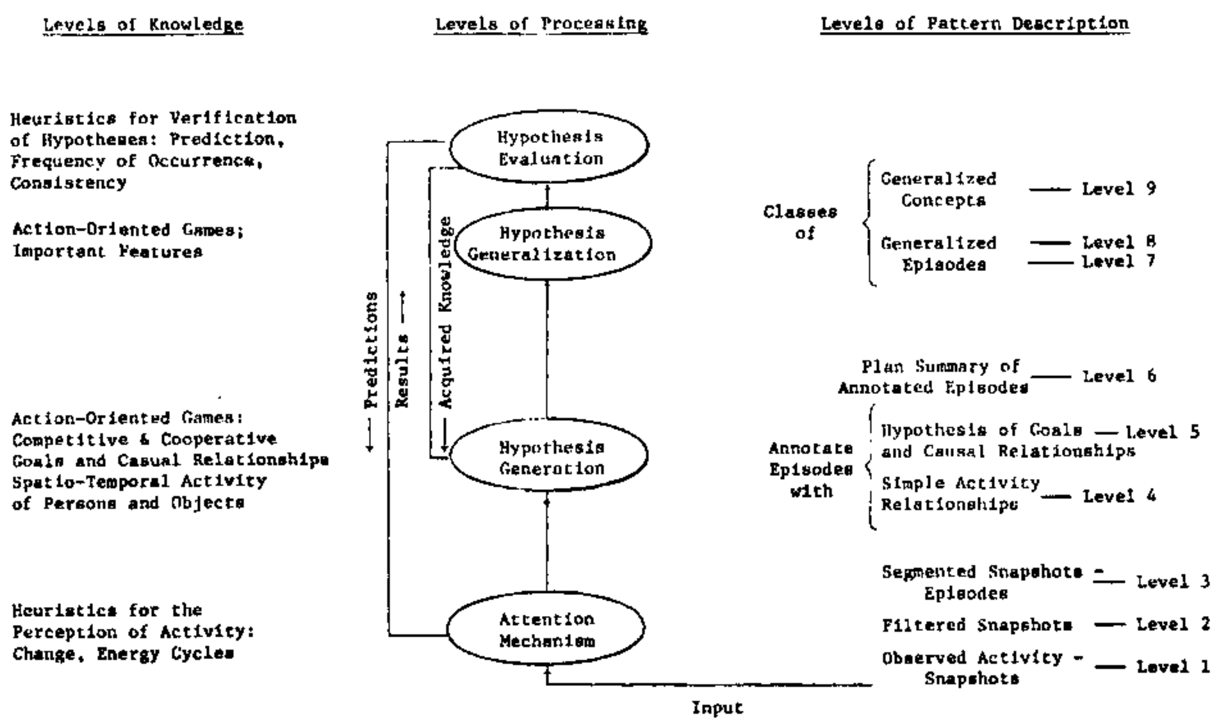


**Figure 1 - Organization of System**

**Levels serve to structure the knowledge, processing and pattern descriptions.**

acquired knowledge by the system. Since each level plays a specific role, the system implicitly knows what function the new information serves, where that new information should fit in, and how it should be used. Thus, acquired knowledge at a level is available for further use in the same format as the a priori knowledge which was used to acquire it; this aspect is developed in more detail elsewhere [Soloway, 1977],

The specific game under observation drives the system bottom-up to make the specific inferences and hypotheses. Thus, if the system observed a variant of baseball or even another game, the same general knowledge would be used but instantiated differently. For example, the same knowledge used to understand the timing relationship in an infield single in baseball could be used to understand the timing relationship in a "run" in cricket.

The approach to learning outlined above requires the integration of a large spectrum of issues. It resembles the work done on understanding/recognition systems; the multi-level architecture is similar in spirit to that of the HEARSAY 11 speech understanding system [Erman, 1975; Lesser, 1977], and the VISIONS scene interpretation system [Hanson, 1976], while the belief systems work done by Schmidt [1976] and Sridharan [1977] has influenced our approach to the inference of goals and causal relationships of humans. In this regard Schank's [1974] and Rieger's [1973] work is also relevant. To demonstrate the effectiveness of our systems, we require that it use the knowledge that it acquires [Soloway, 1977], Waterman [1970] has investigated this problem in the context of production rules, while Sussman's HACKER system [Sussman, 1973] can subsequently use LISP code that it has constructed and debugged. Unlike the uniform syntactic strategy employed by some formal systems for rule induction/concept formation [Vere, 1977; Michalski, 1974; Hayes-Roth, 1976; Riseman, 1969], our system uses a knowledge-directed strategy to perform generalizations over various subsets of features in the description of the acquired concepts. Recently, Hayes-Roth [1977] has surveyed the need for such a knowledge-directed approach to generalization, calling it the "partial-match problem." Lenat's [1976] AM and Buchanan, Feigenbaum and Lederberg's [1971] Meta-DENDRAL uses a knowledge-based heuristic search paradigm for concept formation; AM discovers new mathematical concepts while Meta-DENDRAL discovers rules for mass spectroscopy analysis.

The organization of the rest of this paper is as follows: Section II will discuss the problem of dealing with large numbers of features, while Section III will discuss the problem of detecting similarities among events and concepts. Section IV will present an overview of the computer system which embodies our knowledge-directed approach to learning. Subsequent sections will relate the various stages in processing (focus of attention, hypothesis generation, hypothesis generalization, and hypothesis evaluation) to the multiple levels of descriptions.

## II. Problem 1: Dealing with a Large Number of Features

Our system sees the continuous activity of a baseball game broken up into discrete "snapshots." A snapshot contains a description of the activity of each of the players and the state of the scoreboard markers at each moment in time. A behavior descriptor unit captures 4 dimensions of the situation: action, actor, location, and time (Figure 2). A snapshot contains about 100 features, a typical episode might contain about 18 snapshots, while a game might contain about 3,300 snapshots (Soloway, 1975; Soloway, 1976].

Contrast this with the "blocks-world" state description depicted in Figure 3 (from Vere [1977]) or the concept description in Figure 4 (from Hayes-Roth [1976]). The object is to learn the stack operator (Figure 3) or the most general description (Figure 4) by extracting out the

| | 102 | 103 | 104 |
|---|---|---|---|
| Level 1 | (THROW A1 PM BALL)<br>(AT A2 HP)<br>(AT A3 FB)<br>⋮<br>(AT A9 RF)<br>(AT B1 HP)<br>(AT B2 DUGOUTB)<br>(AT B3 DUGOUTB)<br>⋮<br>(AT B9 DUGOUTB)<br>(INNING 1) | (MOVING BALL PM (AIR FAST))<br>(AT A1 PM)<br>⋮<br>(AT A9 RF)<br>(AT B1 HP)<br>⋮<br>(AT B9 DUGOUTB)<br>(INNING 1) | (AT A1 PM)<br>(AT A2 HP)<br>⋮<br>(AT A9 RF)<br>(SWINGHIT B1 HP BALL)<br>(AT B2 DUGOUTB)<br>⋮<br>(AT B9 DUGOUTB)<br>(INNING 1) |
| | The pitcher throws the ball. | The ball moves through the air. | The batter hits the ball towards the shortstop |

Figure 2 - Unfiltered Snapshots: Taken from an Infield Single

Each snapshot describes the activity of each player at a moment in time. Time is encoded implicitly in the sequencing of the snapshots. "Homeplate" is used for the reader's convenience; the system knows this only as an X-Y location.

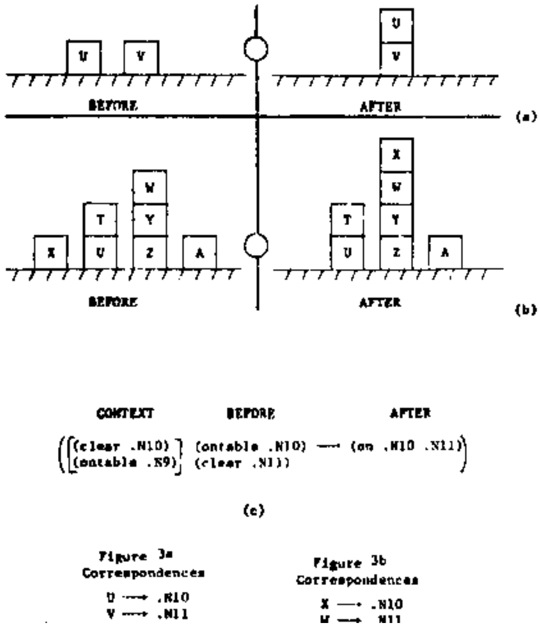


Figure 3 - Inducing the "Stacking. Operator"

The final pattern abstracted requires the block to be stacked („N10) to be on the table, while the blork (.NII) on top of which the first block is to be put must be initially clear oi any other blerks. Note, the . before a name indicatex a variable (from Vete. 1977).

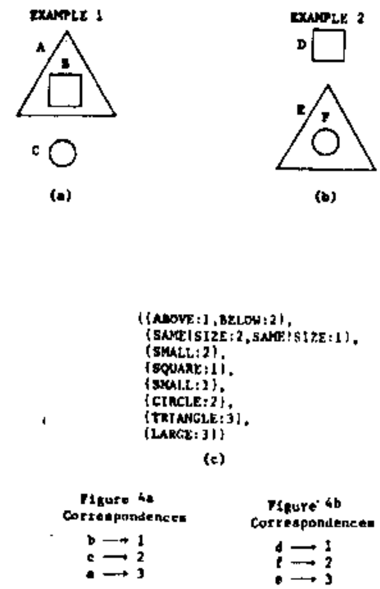


Figure 4 - Concept Formation Task

The resultant general concept captures the commonalltlet} in both geometric patterns (from Hayes-Roth, 1976).

commonalities and deleting the differences. Essentially, the before-after pair in Figure 3a is matched against the before-after pair in Figure 3b, while the description in Figure 4a is matched against the description in 4b. Where there are differences in items, variables are substituted and bindings noted. The inferred generalized "stacking operator" requires that the block to be stacked must be on the table (U in Figure 3a, X in Figure 3b, so variable .N10 in Figure 3c), and the block on which the first block will be put must not have any other block on top of it already. Similarly, in the generalized concept description described in Figure 4c only those features common to both figures remain, e.g., one block is above another block, where both those blocks are small.

While strict data-directed generalization may work in problems on the order of complexity of the above examples, as soon as the number of features increases, some direction must be supplied in order to find the significant features. Are features of color, size, and age of the blocks important to the stacking operator? If both blocks were 4 years old and green, should that be a property of the stacking operator? Knowledge of features important to the physical manipulation of objects might provide direction for this analysis.

If a syntactic matching technique for generalizations were employed in the domain of baseball, it would result in a combinatorial explosion. Matching of just one before-after pair of snapshots against just one other before-after pair yields 2 possible different generalizations! This problem is exacerbated because sequences of length far greater than two are needed to capture other meaningful "operators or rules" in baseball! Knowledge is clearly required in order to direct the search for the interesting generalizations.

## III. Problem 2: Finding Similarity at Different Levels of Description

Let us reiterate — the goal of our system is to construct a hierarchical network of generalized concepts (classes of events) that capture similarities in observations. The "hierarchy" represents our intuition that two examples may look quite distinct from one perspective (level), yet appear similar or identical from another (usually more abstract) point of view. In linguistics it has been argued that the notions of deep structure and surface structure [Ross, 1967; Lakoff, 1969] capture that intuition. For example, while the surface structures of corresponding active and passive sentences are different, their deep structure representation captures the equivalence of their meaning.

As we saw in the Martian scenario, what is considered similar in two situations is dependent on the frame of reference of the observer. In baseball a "walk" and a "single" have quite different activity sequences (surface structure); however, their meaning (deep structure) relative to this particular competitive game shows them both to be means to achieving the same goal: getting-on-firstbase.

In order to reflect a domain specific interpretation, additional features need to be added to the original description. The Martians added the feature descriptor, "religious item," to their description of the roof signs (television antennas). Similarly, as we shall see in Section IV.2, the process of hypothesis generation attempts to add features relevant to competitive games — namely, the goals of the actors and the causally related interactions between the actors. It is these features which capture the meaning — the deep structure — of the activity, and which form the basis of the subsequent generalization process.

Knowledge is used to form potential classes of events at various levels of generalization. However, the system does not know what classes should exist in baseball — if it did, that would beg the whole question of learning. Rather, general heuristics suggest subsets of features to serve as the basis for class formation. For example, the feature "competitive goal" (Section IV.3) will be used to form classes; those episodes with the same competitive goals will be grouped together. Distinct sets of features at the different levels permits the system to find various similarities among the observed events.

## IV. Overview of System

While the preceding discussion has focussed on the issues, the following sections outline how our learning system actually accomplishes multiple shifts in description and how those levels are used in the learning process. The following sections will mirror the flow diagram of the system processes depicted in Figure 1. In moving from level 1 to level 3 irrelevant features are filtered out while important ones are highlighted. Levels 4 - 6 annotate the output of level 3 (episodes) by hypothesizing additional feature descriptors that are relevant to action-oriented games. Levels 7 - 9 represent the discovered similarities in the events as generalized classes.

### IV.1 The Attention Mechanism

The attention mechanism uses general domain knowledge to focus on potentially "interesting" aspects of the observed behavior. This module accepts complete snapshots at level 1, and proceeds to filter them and partition them into logical groups (level 2 and level 3, Figure 1). Two heuristics are used in this process. First, the biologically-motivated notion that "change is important" guides the system to filtering out of the snapshots all those actions that remain constant. This process reduces the number of act descriptors in a snapshot from about 25 to an average of 2 or 3 per snapshot. Certainly, things that don't change can be important. On a first pass, we will miss such subtleties, but hopefully later processing can re-direct the attention mechanism to take note of such non-change, when necessary.

Next, snapshots are partitioned into episodes based on the following observation: in action-oriented games the amount of activity measured in terms of energy expended by the actors goes in cycles; a low amount of activity (e.g., pitcher holding the ball) is usually followed by a high amount of activity (e.g., players moving), which is usually followed by a lull In activity (e.g., the pitcher holding the ball again). Each episode contains on the average 30 act descriptors over a range of about 15 time units. The episode partitioning is crude and later stages provide more semantic analysis in order to punctuate the boundaries of the competitive episodes more clearly.

### IV.2 Hypothesis Generation

The hypothesis generation process uses a priori general knowledge to interpret the observed activity from the perspective of an action-oriented game. To this end, it makes hypotheses about competitive/cooperative goals and causal relationships of the players in the observations.

The inferences depend upon the system understanding the observations first in terms of spatio-temporal activity independent of a game context. For example, the system adds to the description of the actions Throw and Swinghit in Figure 5 that the throw AI, set up a condition that enabled the hitter B2, to execute his act; i.e., AI performed a physical action which enabled B2 to hit the ball. Knowledge about the various primitive actions in the system is represented as Act-Schemata. They are implemented as templates with constraints on slots, which represent the various aspects of an action (e.g., the physical enabling condition for Swinghit in Figure 5).

Hypotheses of goals and causal relationships of the players are added to the description of the observations at level 5. Causal-Link Schemata (CLS), which encode the general game knowledge of the system, are the agents in this process. Represented as production rules [Davis, 1976], CLSs draw on the output of the Act-Schemata as they test for competitive and cooperative interactions in the observed actions. Figure 5 illustrates how one such CLS, PHYSICAL-COMPETITION, makes the following hypothesis of a goal and a causal relationship for the interaction of the pitcher and the batter:

(1) The goal of the pitcher AI was to prevent a player on the opposing team BI from hitting the ball; the goal of the batter BI was to hit the ball.

(2) The causal relationship was a 'competitive-physical-enablement."

Triggering the PHYSICAL-COMPETITION schema in Figure 5 results in the creation of a new specific

The system can deal with other aspects of such a spatio-temporal domain, e.g., understanding that the difficulty of executing an action varies with changes in its preconditions.

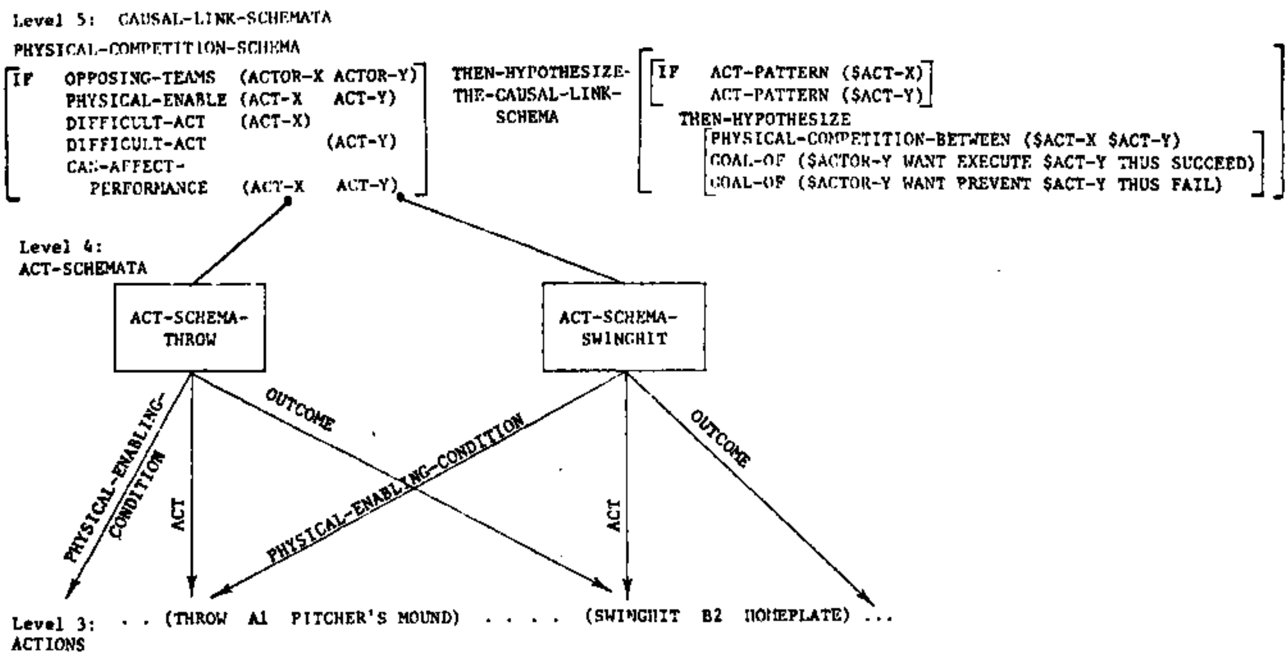


Figure 5 - Moving From Observed Actions to Coals and Causal Relationships by Hypothesizing Specific Causal-Link-Schemata

The Act-Schemata add features to the description of the observations that capture an understanding of non-game activity; e.g., the physical-enabling-condition that Al set up (the ball moving) enabled B2 to execute his act. Then the Causal-Link-Schemata use those features while adding their own competitive game interpretation. The result of triggering a general CLS is the hypothesis of a CLS specific to the observed actions; in this case $ACT-X is bound to the act-pattern (THROW Al ...) and $ACT-Y is bound to the act-pattern (SwiNGHIT B2 ...). The general CLS and the hypothesized CLS have the same production rule structure.

CLS tailored to the particular observations, i.e., the right-hand side of a production rule produces a new production rule. Thus, the hypothesized CLS is structurally identical to the general CLS. Once generalized and verified this acquired CLS can be effectively used in recognizing recurring instances of the episode and can be used in hypothesizing goals in different contexts (different episodes). We can also view the action of a CLS as adding 2 features (goal and causal relationship) to the description of observations. In Figure 6, PHYSICAL-COMPETITION adds a goal feature and a causal-relationship feature to the Throw (act #2) and Swinghit (act #6) actions.

The control structure for the application of the set of cooperative and competitive CLS's is a grammar that characterizes competitive episodes. The grammar is implemented as an augmented-transition network (ATN) parser [Woods, 1970]. Episodes are parsed "left-to-right" with the spatial metaphor referring to the forward movement of time. Each action serves as a state in the network, while an arc connecting two states represents the hypothesis of a causal relationship between the actions. At each arc the eight CLS's currently in the system are tested for applicability. For example, in processing the infield single episode of Figure 6, the ATN creates a state for the Throw action (act #2) and then tests surrounding actions with the CLS's. The activation of the PHYSICAL-COMPETITION schema creates an arc (a hypothesized causal link) between the Throw (act #2) and the Swinghit (act #6). More than one CLS may have its triggering conditions met and thus multiple CLSs can be invoked. This results in multiple arcs representing alternative hypotheses emanating from one state (action).

Analogous to defining a grammatical sentence, we define a grammatical episode to be one in which there is at least one competitive interaction as hypothesized by a CLS (see [Runelhart, 1975]). If at the end of a parse the ATN has not found a competitive interaction, it backs up and looks at the original data at level 1. This is done in order to find an action that may have been filtered out initially, but which now may be a potential competitive act. For example, since in a "called strike" or "ball" episode the batter does not swing the bat, his unchanging action of standing at homeplate is filtered out at level 2. The ATN finds such an action and the CLS's examine its competitive interaction possibilities.

The shift from level 5 to level 6 is one that reduces the data by abstracting from the newly annotated episodes a plan summary. This summary highlights the important goals and relationships of the players occurring in an episode. It consists of all the competitive interactions and the cooperative interactions between distinct players. The dark arrows in Figure 6 indicate the 3 competitive interactions and 1 cooperative

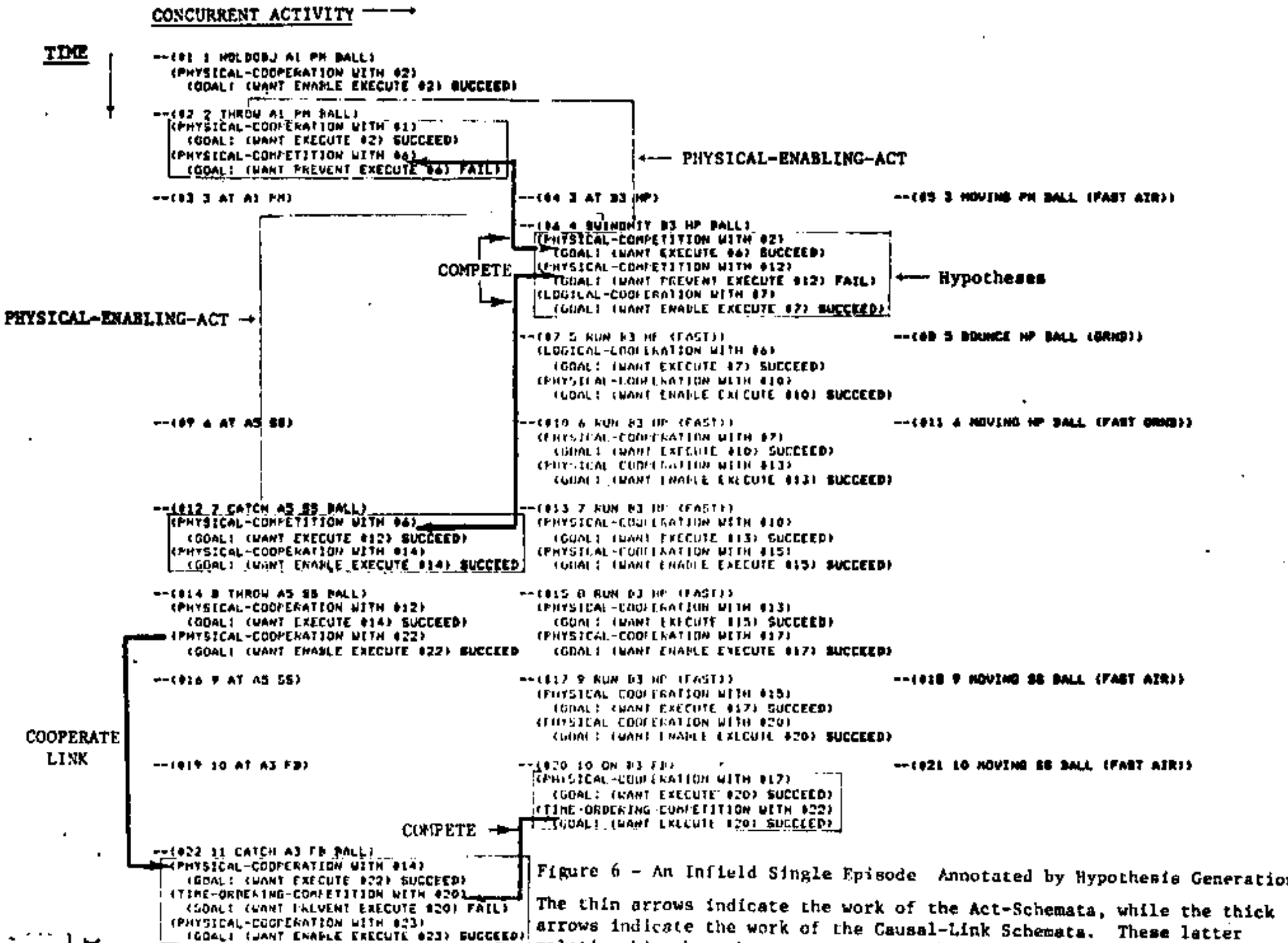


Figure 6 - An Infield Single Episode Annotated by Hypothesis Generation

The thin arrows indicate the work of the Act-Schemata, while the thick arrows indicate the work of the Causal-Link Schemata. These latter relationships have been extracted by the system as the Plan Summary of an episode, e.g.,

cooperative interaction:
(#14 THROW A5 SS BALL) - (#22 CATCH A3 FB BALL)

competitive interactions:
(#2 THROW A1) - (#6 SWINGHIT B3)
(#6 SWINGHIT B3) - (#12 CATCH A5)
(#20 ON B3) - (#22 CATCH A3)

interaction that constitute the plan summary for an infield single.

IV.3 Hypothesis Generalization

The input to Hypothesis Generalization are individual episodes in which each action pattern is described by 7 features: action, actor, location, time, modifiers, goal-of-player, and causal relationship (levels 5 and 6). The task for this module is to generate general classes of similar episodes. For example, infield single episodes at level 6 are grouped together to form a class of infield singles at levels 7 and 8 (Figure 7). The strategy for class formation is to hold a subset of the above features constant. The other features in the pattern description are allowed to vary, thus permitting differentiation within a class. Moreover, a hierarchy of classes is formed by choosing different subsets of features to hold constant. At level 9, flyouts and groundouts are generalized into the class of "outs" (Figure 7).

The system does not initially know what classes should exist in baseball, nor does it have the benefit of a trainer carefully ordering the observations and providing feedback as to the correct classification. Rather, the system is given heuristics which suggest the types of features which should form the basis of classes. At level 7 (Figure 7), classes are formed by holding the goal, causal relationship and location features of a plan summary constant. At level 8 (Figure 7), only the goal and causal relationship are held constant; while at level 9 (Figure 7), the most general classes currently generated by the system are based only on the final competitive goal in the plan summary. Thus, at level 9, flyouts and groundouts are similar and form a class; they have the same final competitive goal of

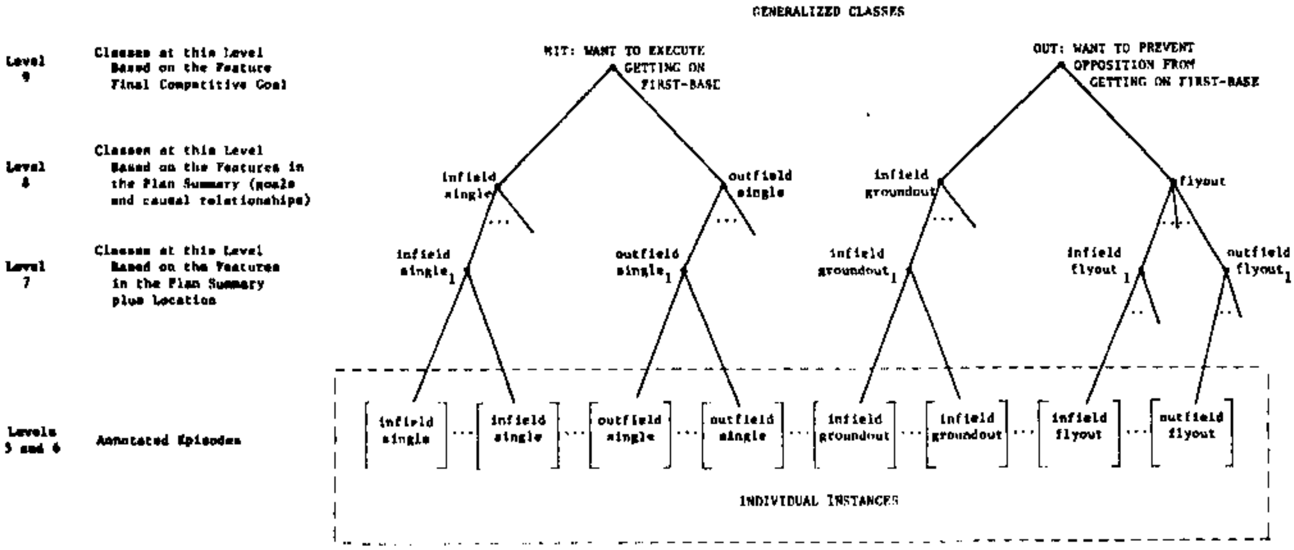


Figure 7 - Acquired Classes of Concepts and Schemata

Different subsets of features of the annotated description of the observations ([action actor location time goal casunl-relaiion]) at level 6 are used *an* the basis for finding similarities. One subnet may uncover *n* similarity between two observations while another will not. For example, based on features of the hypothesised Causal-Link Schemata in the competitive plan summary of level 6, infield groundouts and flyouts are not similar at level 8. Baaed upon only the final competitive goals in the description, infield groundouts and flyouts become similar at level 9.

preventing the opposition from getting on first-base. The choice of features on which to base class formation is dependent on the domain of interpretation; here, only those that suggest meaningful classes relative to the domain of action-oriented games have been used.

Besides facilitating the discovery of similarities in the data and the hypotheses, the multiple levels of generalization also aid the system in avoiding the nasty problem of premature over-generalization. By holding most of the features constant and letting only a few be changed into variables, the space of possible generalizations is drastically reduced. In addition, before the system moves to a higher level of abstraction, i.e., letting more features vary, the system requires that generalized hypotheses at lower levels be verified first (Section IV.4). For example, at level 7 (Figure 7) a class of infield singles will be generated in which the ball hit by the batter goes to the shortstop. Before allowing this class to be merged into the class of all infield singles at level 8, confidence in the hypotheses at level 7 is required. This is a conservative and structured generalization strategy and it may require that a large number of observations to be made. It is an alternative to trying to recover from an over-generalization, a problem which has been little studied.

Those features not used in class formation are allowed to vary and take on values under the direction of the incoming data. For example, at level 7, the "actor" feature is not held constant and thus is replaced by a variable. That variable is allowed to match any actor in the observations. In this manner, the set of matched values allows for variability within a class; the batter in an infield single can be BI, B2, AI, or in general ANY-PLAYER, while the pitcher in that event is required to be some player on the opposing team.

## IV.4 Hypothesis Evaluation

In a learning system there are inherent problems in the production and evaluation of hypotheses. First, hypotheses are just that — unverified conjectures which quite possibly are wrong. Indeed, there are often multiple interpretations for the same events. Second, the " knowledge used in evaluating hypotheses must be general — not specific to the particular game being observed. The generality of knowledge at this level distinguishes perceptual recognition systems (speech understanding) from recognition in a learning system. The former systems usually have detailed knowledge on how to evaluate hypotheses ([Hanson, 1976; Lesser, 1977]). The approach we take is to let hypotheses provide their own evaluation; if hypotheses predict events — and their interpretations — that have not yet occurred and if hypotheses bind together into an internally consistent global view, then confidence in those hypotheses is increased accordingly.

The motivation for this approach stems from the assumption that one has more confidence in knowledge that can be used to accurately predict the future. Both the occurrence and the correct interpretation of a specific predicted event are important evidence. The system makes three types of predictions. One type predicts the complementary success/failure outcome of a competitive interaction. For example, the system hypothesized that the pitcher failed and the batter succeeded when the batter hit the ball; so an obvious prediction is that the system should see the batter fail to hit the ball with the pitcher thus succeeding. Predictions are fed back to the Attention Mechanism for matching against the incoming observations. Events found in this way together with their interpretation are then passed back to Hypothesis Evaluation where the confidence values of the hypotheses are modified appropriately.

In addition to prediction, features such as frequency of occurrence and consistency affect the confidence values on the hypotheses. The system accepts as "truth" those hypotheses with the highest confidence values. Such verified hypotheses (specific causal-link schemata) are then passed back to Hypothesis Generation to be used in further learning. They are also used in Hypothesis Evaluation to eliminate unverified hypotheses that are contradictory. For example, when the system decides that getting on first-base is a desirable goal (for one team), then all the hypotheses which suggest that getting on first-base is undesirable, can be eliminated.

## V. System Implementation and Experimentation

The computer system described in this paper is implemented in LISP and requires approximately 75K on a CDC 6600. An earlier version of the system [Soloway and Riseman, 1977] processed all the observations at one level before proceeding to the next level of processing. The current "system operates in "real-time;" it makes hypotheses, predictions, and generalizations as it is observing events in the game. Since this analysis is sensitive to the order in which events occur, a conservative generalization strategy was employed to prevent the system from prematurely over-generalizing. The current version required the observation of 9 innings of baseball in order to learn the highest level concepts and generalized episodes depicted in Figure 6.

Additional Causal-Link Schemata are being added to the system which would allow it to make hypotheses about the relationship between changes in the scoreboard markers (hits, outs, etc.) and the goals and events in the game. This should enable the system to acquire schemata for episodes such as "strikeout," "walk," and "score," which require this additional knowledge.

A sense for the volume of data in the pattern descriptions is provided by Table 1. The large number of initially observed actions is reduced by the Attention Mechanism's heuristic filtering algorithm; the number of actions per snapshot is reduced from 26 to 2 on the average. Since each action is described in terms of 4 features (action, actor, location, and time), this reduction results in an average of 8 features in the pattern description of a snapshot. Hypothesis generation adds the features of goal and causal relationship to the behavior description. It is these additional 946 features per inning that characterize the observed activity. They serve as the basis for class formation during generalization. Thus, while there is a significant amount of data at the sensory level, the system requires only a relatively small amount of data at the interpretation levels.

| | ACTIONS | | FEATURES | | ADDITIONAL HYPOTHESIZED FEATURES |
|---|---|---|---|---|---|
| | UNFILTERED | FILTERED | UNFILTERED | FILTERED | |
| SNAPSHOTS | 26 | 2 | 104 | 8 | 0 |
| EPISODES | 338 | 26 | 1,352 | 208 | 22 |
| INNING | 14,534 | 1,118 | 58,136 | 4,472 | 946 |

Table 1 - Filtered and Annotated Data at Higher Levels of Description

There are 26 actions in a snapshot if all the actions and all the markers on the scoreboard are considered. On the average there are 13 snapshots/episode and 43 episodes/inning. After filtering out non-changing activity at the level of the Attention Mechanism, the average number of actions/snapshot is reduced from 26 to 2. Four features comprise an action: actor, action, location, time. Hypothesis Generation adds new features to the description of the activity by interpreting that activity as an action-oriented game. Each hypothesis adds a goal feature and a causal relationship feature. Since on the average there are 11 such competitive and cooperative hypotheses per episode, 22 features are added per episode. The generation of classes of episodes and concepts is based on these inferred features. (Read this table: column per row.)

## VI. Summary

The problems of learning in a complex real-world domain require that a significant amount of highly organized knowledge and processing be brought to bear. In particular, a system must deal with large numbers of mostly irrelevant features and must discover meaningful similarities in the new observed situations. To this end, our system employs a multi-level knowledge-directed learning paradigm; it attempts to interpret observations of novel situations in terms of its prior general knowledge. Thus our approach to learning can be viewed as a form of recognition, where the level of initial knowledge is general and where the specific observations mold a particular structure from the general knowledge.

This approach permits the system to filter out nonessential features and to add new descriptive features which have a meaning in the task domain. Based on these semantic features, similarities are found in the observations. Such similarities are represented in the multiple-levels of generalized episodes and concepts. In the action-oriented game of baseball, the system moves via multiple levels of processing through successive descriptions of behavior patterns: from the initial observation of seemingly independent actions such as throw, run, catch; to high-level concepts and integrated activity patterns such as "hit," "out," "single."

A system architecture using multiple levels of knowledge, processing, and pattern description significantly contributed to the successful design, construction and operation of the learning system.

## Acknowledgements

We wish to thank Vic Lesser for his helpful suggestions for organizing the ideas in this paper. We would also like to thank the following persons who read earlier drafts of this paper and provided constructive comments: Kurt Konolige, John Lowrance, Robbie Moll, Hap Peelle, Jon Post, N.S. Sridharan, and Bryant York. Finally, a big thanks goes to Janet Turnbull, who tolerated our inefficiency in the preparation of this manuscript.

## Bibliography

Arbib, M.A. (1975) Two papers on schemas and frames. COINS Technical Report 75C-9, University of Massachusetts, Amherst.

Buchanan, B.G., Feigenbaum, E.A. and J. Lederberg (1971) A heuristic programming study of theory formation in science. International Joint Conference on Artificial Intelligence-2, British Computer Society, 40-50.

Collins, A. (1976) Processes in acquiring knowledge. BBN Report 3231, Cambridge.

Davis, R. and J. King (1976) An overview of production systems. Computer Science Technical Report, Stanford University.

Erman, L.D. and V.R. Lesser (1975) A multi-level organization for problem solving using many diverse, cooperating sources of knowledge. International Joint Conference on Artificial Intelligence-4, U.S.S.R.

Hanson, A.R. and E.M. Riseman (1976) A progress report on VISIONS: representation and control in the construction of visual models. COINS Technical Report 76-9, University of Massachusetts, Amherst.

Hayes-Roth, F. (1977) The role of partial and best matches in knowledge systems. To appear in the Proceedings of the Workshop on Pattern-Directed Inference, Hawaii.

Hayes-Roth, F. and J. McDermott (1976) Knowledge acquisition from structural descriptions. Computer Science Technical Report, Carnegie-Mellon University, Pittsburgh.

Jordan, M. (1968) New Shapes of Reality, Aspects of A.N. Whitehead's Philosophy. George Allen and Unwin Ltd., London, pp. 154-155.

Lakoff, G. (1969) On generative semantics, in Semantics — An Interdisciplinary Reader, D. Steinberg and L. Jacobouits (Eds.), Cambridge University Press, London.

Lenat, D.B. (1976) AM: An artificial intelligence approach to discovery in mathematics as heuristic search. Ph.D. Dissertation, Computer Science Department, Stanford University, A.I. Lab Memo AIM-286.

Lesser, V.R. and L.D. Erman (1977) A retrospective view of the HEARSAY-II architecture. To appear in the Proceedings of the International Joint Conference on Artificial Intelligence-5, Cambridge.

Michalski, R.S. (1974) Learning by inductive inference. NATO Advanced Study Institute on Computer Oriented Learning Processes, France.

Minsky, M. (1974) A framework for the representation of knowledge. M.I.T. A.I. Lab Technical Report TR-306, Cambridge.

Plato (1949) Meno. B. Jowett translation, The Liberal Arts Press.

Rieger, C. (1973) The common-sense algorithm as a basis for computer models of human memory, inference, belief and contextual language comprehension. Theoretical Issues in Natural Language Processing, June, Cambridge.

Riseman, E.M. (1969) Feature detection networks in pattern recognition. Center for Appl. Math., Cornell University, Ithaca, N.Y., Rep. AFSOR 64-1249 TR; also Ph.D. Dissertation, Cornell University, June 1969.

Ross, J.R. (1967) Constraints on variables. Ph.D. Dissertation, M.I.T., Cambridge.

Rumelhart, D.E. (1975) Notes on a schema for stories, in Representation and Understanding, D. Bobrow and A. Collins (Eds.), Academic Press, New York.

Schank, R.C. (1974) Understanding paragraphs. Instituto per gli Studi Semantici e Cognitivi, Technical Report 5, Castognola, Switzerland.

Schmidt, C.F. (1976) Understanding human actions: recognizing the plans and motive of other persons, in Cognition and Social Behavior, J. Carroll and J. Payne (Eds.), Erlbaum Press, in press.

Soloway, E. (1977) Knowledge-directed learning using multiple levels of description. Ph.D. Dissertation, Department of Computer Science, University of Massachusetts, Amherst, in preparation.

Soloway, E.M. and E.M. Riseman (1975) Common-sense theory formation: towards understanding baseball. COINS Technical Report 75C-5, University of Massachusetts, Amherst.

Soloway, E.M. and E.M. Riseman (1976) Mechanizing the common-sense inference of rules which direct behavior. Proceedings of the AISB Conference, Edinburgh.

Soloway, E.M. and E.M. Riseman (1977) Knowledge-directed learning. To appear in the Proceedings of the Workshop on Pattern-Directed Inference, Hawaii.

Sridharan, N.S. and C.F. Schmidt (1977) Knowledge-directed inference in BELIEVER. To appear in the Proceedings of the Workshop on Pattern-"Directed Inference, Hawaii.

Sussman, G.J. (1973) A computational model of skill acquisition. M.I.T. AI Lab Technical Report TR-297, Cambridge.

Vere, S.A. (1977) Inductive learning of relational productions. To appear in the Proceedings of the Workshop on Pattern-Directed Inference, Hawaii.

Waterman, D.A. (1970) Generalization techniques for automating the learning of heuristics. Artificial Intelligence, 1, pp. 121-170.

Winston, P.H. (1970) Learning structural descriptions from examples. M.I.T. Project MAC Technical Report 76, Cambridge.

Woods, W.A. (1970) Transition network grammars for natural language analysis. Communications of the ACM, vol. 13, no. 10.