SDC

TM-4771/000/00

PROBLEM SOLVING AND LEARNING BY MAN-MACHINE TEAMS

Final Technical Summary Report to
The Office of Naval Research

7 July 1971

SERIES	BASE NO. / VOL.	REISSUE
TM	4771 /000/00	

TECHNICAL MEMORANDUM

(TM Series)

The work reported herein was supported in part by the Office of Naval Research contract No. N00014-70-C-0221, Task No. NR 348-009; and in part by the Advanced Research Projects Agency, contract No. DAHC-15-67-C-0149.

PROBLEM SOLVING AND LEARNING BY MAN-MACHINE TEAMS

Final Technical Summary Report to The Office of Naval Research

Aiko M. Hormann, Principal Investigator
Sharon Kaufman-Diamond*
Carlos Martín Cinto**

7 July 1971

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA
90406

* Psychiatrist and acting Associate Professor in Urban Planning, School of Architecture and Urban Planning, University of California at Los Angeles.

** Psychiatrist and visiting Fellow under grant from the Del Amo Foundation Brain Research Institute, University of California at Los Angeles; now at University of Madrid, Department of Psychiatry and Computation Center.

ABSTRACT

This report describes research work in artificial intelligence, human and machine-aided problem-solving and planning activities, interactive languages and visual input/output techniques, and man-machine synergy. The chronological development of concepts, techniques, computer implementation, and experimentation is described in terms of three periods that separate and identify shifted emphases in the work and the different computer facilities used. The emphases during these periods were on (1) developing machine capabilities, (2) understanding man's higher mental functions in complex problem-solving and planning activities, and (3) man-machine synergistic teaming and group planning on real-world problems.

The initial design of an "adaptive" system, Gaku, was a hierarchically structured collection of decision units, each utilizing feedback. Gaku had undergone a training sequence of increasingly complex tasks, coordinating the functions of a problem-oriented mechanism, a planning mechanism, and an induction mechanism. In addition to Gaku's autonomous evolvement through first-hand experience, ways to promote its "secondary learning" (learning from the experience of others) were sought; the two major efforts involved were (1) to gain a deeper understanding of the workings of man's higher mental functions in complex problem-solving situations, aiming toward extending man's limitations and fortifying his superior capabilities; and (2) to develop a man-machine interactive language that would enable the man to express evolving concepts and problem-solving methods dynamically.

A highly complex task environment, Shimoku, was designed and implemented on an interactive computer system having graphic input/output facilities. The task environment contained many conditions and interrelated elements to be manipulated to achieve desired conditions, and cost and payoff functions were associated with the manipulation and its consequences.

Informal and semiformal experiments were conducted with both volunteer and paid subjects. Detailed analyses and interpretation of the results were made both in terms of information-processing needs of the subjects and their methods of meeting the needs and in terms of information-processing psychology. The study showed (a) how subjects' different understanding of the "real nature" of the same problem depended on their search

for problem representations; (b) how the representations selected imposed different information processing demands; and (c) how these representations permitted different procedures for formulating solution steps which, in turn, determined different degrees of success. The study also showed that when man is confronted with an overwhelmingly complex situation he tends to oversimplify the task and focus on only a small aspect of it. Two methods for reducing man's cognitive load were identified: (1) master planning for the overall task by preparing a structural scheme to guide detailed steps, and (2) having the machine's decision-aiding functions assist the man in detailed work so that he can concentrate more on higher-level planning and problem solving and on idea generation.

The design of Gaku has undergone changes to incorporate desirable new features and to remedy limiting factors that were identified. The new design is especially geared to group planning that involves different levels of planning (conceptual, definitional, developmental, and operational) and iterative decision steps of problem solving. Some real-world problems to which man-machine techniques can be fruitfully applied are characterized and the types of decision dynamics influenced by these characteristics are identified.

CONTENTS

Page

APPENDICES

FIGURES

TABLES

1. INTRODUCTION AND OVERVIEW (by Aiko M. Hormann)

This report describes research work in artificial intelligence, human cognitive processes, problem-solving and planning and activities, machine-aided cognition, and man-machine synergy.

The primary objectives of this research are to explore ways in which a man and a machine (i.e., a programmed computer) can augment each other's capabilities and to develop techniques by which those augmented capabilities can be used effectively in a variety of decision-making and problem-solving situations. Teaming man with an "adaptive" machine that can be made to "co-evolve" with him through interaction is expected to be needed especially in those complex problem situations for which complete analyses and detailed decision making in advance is not feasible. Such situations arise mainly from the combinatorial complexity and uncertainty of possible events and their impacts that are too overwhelming to man, and they also arise from the incompleteness and impreciseness of information and the evolving nature of problem situations.

The over-all research can be viewed as an attempt to answer the following four interrelated groups of questions:

- What machine capabilities, including adaptivity, are needed to effectively couple human and machine functions? What can be preprogrammed and what must be left to man-machine interaction?

- What are frequently shared characteristics (both weaknesses and strengths) of human cognitive processes? What cognitive limits can be extended, and what weaknesses can be fortified, by man-machine techniques? How can man's special faculties, such as intuition, imagination, inductive reasoning, and pattern recognition be promoted and put to good use in man-machine partnerships?

- What characteristics of problem situations are especially in need of man-machine synergistic work and from which can substantial payoffs be expected?

- What type of language is needed for man-machine communication toward man-machine synergy--to enable the man to express evolving concepts and problem-solving methods dynamically--i.e., to facilitate interaction even at the problem-conceptualization and definition stage as well as at the exploratory and intuition-guided stage of problem solving?

In an effort to gain sufficient insight into the above questions, hoping to answer them, an iterative design-experiment approach was taken in 1964. The approach was to start with our then-current understanding of human cognitive processes as a partial theory, combined with the then-current state of the art in artificial-intelligence research and to design, implement, and experiment with a system that embodied such understanding; to evaluate the results; and to redesign the system as shortcomings were discerned and as new ways were found to make the system more effective and desirable in terms of the objectives.

This report describes the chronological developments of concepts, techniques, computer implementation, and experimentation as the results of our iterative approach.[1] This section summarizes these developments in terms of three identifiable periods of research progress. Further descriptions of these periods are given in separate sections. Accounts of the work of periods 1 and 3 are condensed since fuller accounts are available in the published literature and in technical reports referenced in the bibliography. The work of Period 2 including a series of experiments, the results of which have not been published before, is described in much greater detail.

Grouping of our work into three periods is made in terms of the shifted emphases that became visible as the work progressed and as new concepts and techniques evolved.[2] The emphases for these periods are on: (1) developing machine capabilities (1964-65); (2) understanding man's higher mental functions in complex problem-solving and planning activities (1966-69); and (3) a man-machine synergistic teaming and group planning and on real-world problems (1970-present).

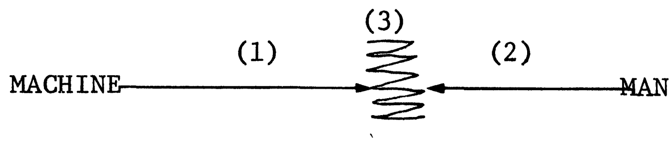
These period identifications stem from our attempts in 1964 to bridge the gap between the traditional uses of computers in problem solving and man's problem solving. As shown schematically below, Period 1 focused on moving the machine into a symbol-manipulating problem solver; Period 2

---

[1]This research project was initiated by SDC in 1961. It continued under joint ONR-SDC sponsorship from December 1964 to September 1969, and under joint ONR-ARPA sponsorship from September 1969 to November 1970.

[2]Shifting of the emphasis, in turn, was influenced by the available computer facilities--i.e., Period 1 by the Philco 2000 (noninteractive), Period 2 by the AN/FSQ-32 time-sharing system with graphic input/output, and Period 3 by the loss of access to the same and the introduction of the IBM 360/67 ADEPT time-sharing system.

focused on understanding man's cognitive processes in complex task environments and on finding possibilities of assisting man in consciously directing his problem-solving activities in a logical, systematic, way (hopefully without hindering, and even promoting, his creative idea generation); and Period 3 focused on developing a means of intermeshing the two.



The zigzag line shown for Period 3 indicates the give-and-take mutual contribution and dependency that is necessary in man-machine interaction and co-evolution. Man's contribution in problem solving may be greater at higher-level functions in guiding machine processes; but the machine can be made to be responsive to man's evolving ideas and hunches, providing quick answers to various probing questions and reporting results that are worthy of man's attention by some specified criteria (as in exception reporting). These, in turn, may spark new ideas in man. The tasks in (1) and (2) are far from being completed and the work in (3) must include continuing research in both (1) and (2) to reduce the "impedance" between man and machine as much as possible.

1.1 SUMMARY OF PERIOD 1: DEVELOPING MACHINE CAPABILITIES

The major emphasis during this period was on developing machine capabilities. The research objective was to construct, by programming on a computer, an "intelligent" and "adaptive" system capable of handling a set of increasingly complex tasks which, when performed by a human being, are usually said to require intelligence.

The initial design of an "adaptive" system, Gaku, was based on a relatively simple, feedback-utilizing iterative process that is often observable in human problem-solving activities. When a man faces a problem with only partial knowledge about how to proceed, his behavior at the outset often seems to be of random trial and error form. As the problem unfolds, however, his behavior becomes more selective, directed, and organized. Inadequate actions are corrected or adjusted by the use of new information gained as a consequence of previous actions. The solution strategy is often discovered in the course of the action itself. This cyclic process was abstracted into three connected components, forming a feedback loop, in order to be embodied as a "mechanism" (a set of computer programs). Three such mechanisms of the same structure but of different functions were implemented one-by-one as each was tested in problem-solving tasks, and as the need was recognized.

The first mechanism of Gaku that was implemented was a problem-oriented mechanism. Its built-in information-processing rules were limited to handling "units" of information defined by the given task environment (in terms of "states" and "operations"). Games and puzzles of differing complexity were used to test the mechanism. Although the self-adjusting rules within the feedback principle worked well for simpler tasks, their step-by-step nature caused the mechanism to attack problems in piecemeal fashion and its limitation became clear as the complexity increased.

A planning mechanism was then added to Gaku to remedy the situation. Its function is to analyze the structure of a given problem and place guideposts on the road to the goal. Concrete objects manipulated by the planning mechanism are "state" descriptions representing guideposts and aggregated action rules. The Tower of Hanoi puzzle was used as a testing vehicle; the planning mechanism demonstrated the power of aggregation and planning (within this circumscribed context) and showed a limited degree of "learning" in handling a sequence of tasks of increasing complexity. However, no form of generalization in the use of past experience was possible without a mechanism for extracting underlying patterns.

An attempt was made to provide the generalizing capability by introducing an induction mechanism--another feedback-loop unit with information-processing rules geared to detect recurrent patterns and group them in different ways to generate possible conjectures toward formulating "unifying principles." This time, a mechanism coordinator was also implemented to take over Gaku's executive function of assigning different problem-solving phases to different mechanisms and coordinating their functions.

A training sequence of graded tasks given to Gaku produced good results in the sense that Gaku exhibited an aspect of learning and produced a form of "generalization." However, no new insight was gained into inductive inferences in a wider sense.

Insights gained during this period are:

- A computer system like Gaku, conceived of as a hierarchically structured collection of decision units, each using the feedback principle, can exhibit autonomous problem-solving behavior;

- The degree of attainment in autonomous evolvement will be slow and limited unless Gaku is also given an opportunity for "secondary learning"--i.e., learning from the experience of others--in addition to primary learning by first-hand experience. A means should be provided for Gaku to interact with a man who can give "lectures" (declarative statements

about conditions, relations, and values) and hints and suggestions of graded sequence (methods and procedures to be tried under specified sets of conditions), and guide Gaku's behavior. The secondary learning capacity would make Gaku more responsive to man's needs and guidance in an evolving, unfolding problem situation for which complete specification of decision steps in advance would not be possible.

- Since realization of secondary learning in Gaku would depend on man-machine communication (in an interactive, time-shared mode), a good interactive language is needed in both problem-defining and problem-solving processes for on-line specification of tentative ideas, including reformulation of initial problem statement, action rules, trial search procedure, etc.

- Concurrently with language development, we need a deeper understanding of human thought processes in complex task situations and to identify weaknesses and strengths under certain conditions (hopefully classifiable). A future man-machine system should aim toward extending man's cognitive limits, fortifying his weaknesses, and promoting and extending those higher mental functions that are already recognized as superior.

For the above reasons, we seized the opportunity to use the AN/FSQ-32 time-sharing system, and set to work on a set of experimental conditions for man-machine problem solving and on an interactive language. Publications covering this work are Hormann, 1964a, 1964b, 1964c, 1965a, 1965b.

1.2 SUMMARY OF PERIOD 2: UNDERSTANDING MAN'S HIGHER MENTAL FUNCTIONS IN COMPLEX PROBLEM-SOLVING SITUATIONS

The major emphasis during this period was on the effort to understand the workings of man's higher mental functions in complex problem situations.

A task environment, called Shimoku, was designed and implemented on the computer. The Shimoku environment has numerous parameters so that we could expose human subjects to many tasks that can be varied in complexity, objective characteristics, and elements of uncertainty. Although Shimoku is an abstract environment, many important problematic features, extracted from real-world situations, are embedded in it. In this summary, characteristics of the Shimoku environment are given and highlights of the experiments are described. Publications covering this work are Hormann, 1966a, 1966b, 1967, 1969a, 1969b.

1.2.1 Characteristics of the Shimoku Environment

Shimoku characteristics chosen (among other possible features) for the experiments are:

- Sequential decision steps are required to achieve the stated objective from the given starting conditions. The environmental conditions or "states" change as these decisions are executed.

- Many alternatives are available at each decision-making step, and the total combinatorial complexity is overwhelming to subjects when many possible courses of action are considered.

- Complex cost and payoff functions (using "point" values) are included so that a wide range of measures of performance is possible, rather than the black-and-white concepts of "success or failure" and "win or lose."

- Payoffs depend on environmental conditions which the subjects can create and change. To do so requires a clever choice of sequence of actions, however, and incurs costs of varying degree.

- The task is in the form of one-person "solitaire" and the computer presents the visual display of the environmental conditions which change as action-decisions are executed. A few computer aids are provided, mostly of a bookkeeping nature.

- The stated objective is to make as many "points" as possible at the end of 30 moves or 60 minutes, whichever comes first. This means that the opportunities for gaining payoffs must be weighed against the cost of taking the opportunities, and complex trade-off implications emerge. Also, the interrelatedness of elements in the environment makes it difficult for the subjects to see cause-and-effect relations between his actions and the environmental changes (therefore, also cost-and-payoff relations).

Added to the Shimoku environment in the computer is an automatic recording of all the actions a subject took during the interactive session with the computer using a display scope and a light pen. Also the computer's internal clock time was recorded each time any action was taken. This is a valuable feature for reconstructing "thinking time," apparent punctuation of move sequences, etc. At the end of each session with a subject, the computer prints out a complete action table, showing a step-by-step record of all the actions the subject took. This record is used later to reproduce the entire session, including the time spent between actions,

environmental changes, cost incurred and payoffs collected, and the cumulative net gain at each step leading to the final score.

This record is called an "objective game record" in contrast to other records we kept for each subject--i.e., questionnaires, interview-elicited accounts of the game, and observation. One experimenter worked exclusively in the "hard" analyses of the objective game records without observing any part of the subject's activities,[1] and another experimenter worked on the "soft" part of the experiment (observing subjects, interviewing, questionnaire analyzing, interviewing again, etc.), and still another experimenter worked exclusively with the Shimoku environment, its interactive facilities, game recording, etc.

1.2.2 Summary of Shimoku Experiments

Both informal and semiformal experiments were conducted. Informal ones were dealt with only a few subjects (coworkers), no systematic procedures or controlled conditions were imposed, and free-wheeling discussions were encouraged on the features of Shimoku and its decision-aiding functions and on closer man-machine participation with dynamic expressibility of new ideas. The semiformal experiments were conducted with a larger number of subjects (34), three Shimoku sessions for each subject (few subjects could not come for all three), and used systematic procedures and reasonably well-controlled conditions (with "objective game records").

The first series of the informal experiments were conducted before the semiformal ones to test the adequacy of cost and payoff functions and their point-value assignments and to elicit subjects' reactions to the Shimoku task with and without decision aids. Suggestions were solicited for other types of decision aids and interactive features.

The second series of the informal sessions were conducted during the last quarter of Period 2 when the design of User Adaptive Language (UAL) had reached a near-completion state so that features of UAL could be used to test the dynamic expressibility of ideas--i.e., on-line specification of evolving concepts, procedures, and rules and conditions. Since UAL had not been implemented, the "hand simulation" technique was used to translate (into UAL expressions) the subjects' English statement about what he would do to specify or change the system's information-processing behavior as new ideas emerged while performing the Shimoku task.

The results of these three groups of experiments are summarized in chronological order.

---

[1] There is one exception to this rule; it will be noted in the main text.

Informal Experiment I: Observations and Conjectures

1. All subjects tended first to make short-term decisions toward immediate payoffs but, later, realized that a local improvement or an immediately observable gain does not necessarily contribute to the total gain. They all commented that a seemingly simple decision that affects one part of the environment can have extensive and important repercussions in the rest of the environment. Therefore, efforts aiming at immediate payoffs and at longer-range potential payoffs can produce conflicting or even nullifying effects. They talked about coordination of parallel opportunity-taking decisions and potential payoffs and trade-offs among them. One subject suggested the business analogy of "investment" when talking about cost incurred in preparation for a bigger return.

   They admitted that their initial method of coping with this complex task was to make drastic simplfications and localize their efforts on a small aspect of the task. This tendency became intensified when in addition to complexity, time constraints were imposed. In discussing possible ways of handling "cognitive overload," some subjects suggested computer decision aids and some recognized the importance of master planning and of making an over-all coordination, at an aggregated level. Within this framework, they felt the business of tending to local details and to immediate information-processing requirements can be managed, while the master plan will maintain comprehensive coordination of the details.

2. After a few bookkeeping functions were provided as decision aids, the subjects acknowledged potential values of such decision aids in reducing the cognitive load. However, they commented that their needs for such assistance arise in so many different ways in different situations that a fixed set of specific aids would not be adequate. They would like to be able to specify new machine functions as they proceed during the problem-solving task.

3. The subjects unanimously agreed that graphic display of their performance measures and of environmental changes caused by their actions was helpful in assessing previous decisions and formulating new ones. However, more sophisticated techniques of summarizing the current "state of the environment" are needed to display information in a variety of formats and at varying levels of aggregation. At one point, the subject might be interested in the overall relational aspects, and at another point, he might be interested in detailed information about one small portion of the environment. Again, the need for dynamic definability of man's ideas and requests became clear.

4. Most subjects ask "what if" questions either overtly or covertly in order to estimate the consequences of the tentative decision steps that have been formulated before they actually execute the decisions. However, the breadth and depth of such "what if" questions vary greatly with individuals and with the task situations.

   The "Look Ahead" feature provided by the computer helped answer some types of "what if" questions. In the Look Ahead mode, a subject can try out a sequence of decision steps, check their consequences, retract some or all of them and try out new ones. It enables the subject to explore the environment and his tentative decisions without commitment. All subjects were very enthusiastic about this feature at first, but they came to realize that its effectiveness was limited because of its serial nature of exploration. Since exhaustive examination of alternatives in depth to more than three steps would be prohibitively time consuming even by the computer, a very selective search and aggregated pattern structuring scheme, not possible by the serial approach, would be needed. Strong correlations between good master planning and selective exploration needed in answering higher-level (aggregated) "what if" questions were considered but no conclusions could be reached.

Semiformal Experiment: Findings and Interpretations

1. Working exclusively with the "objective game records" kept by the computer, analyses of the records and interpretations of the results from the analyses produced a comprehensive grouping of subjects' problem-solving procedures into three main types: Incremental (INC), Master Planned (MP), and Adaptive Master Planned (AMP). INC type procedure constituted more than 85% of the total set of objective game records and the final scores ranged from -64 to 110. All master planned performances scored over 110, with MP peaked at 185, and AMP at 241 points. Thus, task performances describable as generated by different types of procedures resulted in different ranges of score points. The different procedures had detectable consequences in the effectiveness with which elements of the problem situation could be handled to attain the stated objective.

   - The INC type action can be characterized as a short range, piecemeal attack on a problem with a collection of rules of thumb that suggests ways to modify the execution of the basic incremental approach. Attempts to execute incremental procedures, while taking note of many interrelated elements and events, seriously overloaded even highly intelligent subjects and led to drastic simplifications and disorganization in weaker subjects. If the rules of thumb were appropriate and their application well managed, higher-range scores were attained within this INC group.

- In contrast, the MP type prepared a totally preplanned goal conditions of the environment in the form of a structure diagram, and proceeded to construct the desired conditions according to an easily operationalized, nearly "algorithmic" prescription. Little adaptation to the given conditions of the environment was attempted, and imposition of the master plan reduced cognitive load of the subject considerably. MP subjects, while considering their games "optimal," did not score as high as the AMP subject.

- AMP type employed strong features of both INC and MP types to prepare and use (a) rules of thumb to guide on-line decision-action processing adaptively and (b) a preplanned, but flexible, structural scheme. Adaptability and flexibility in both planning and in plan execution seem to set the AMP type apart from the other types. The given conditions in the environment were not ignored, as they were by most MP subjects; instead, many elements of the given environmental conditions were capitalized on toward producing the desired goal conditions.

2. Differences in subjects' problem representations were shown to correspond to differences in problem-solving procedures of INC, MP, and AMP types. Since the stated objective was "score as many points as possible," it only implicitly defined what the desired goal conditions might be. Many other specifications and conditions could be inferred.

   - INC subjects accepted the statement of the task "as is"--i.e., all of the elements in the environment, available resources, permissible actions, payoff and cost functions, etc. were part of the givens of the problem.

   - In contrast, no master range (110+) score was attained without introducing a new subproblem; viz., devise a procedure for finding a better goal statement. Only after a subject's conscious search for an improved description by restructuring and reinterpreting goals and constraints from the givens did a sufficiently articulated new representation of the task emerge.

3. The account of steps taken in attaining new representations borrowed Amarel's (1971) analysis of the program formation task for Derivation (D), Formation (F), and quasi-Formation (quasi-F) types of problems.

- It was found that all subjects who had represented the Shimoku task as a D problem with a weakly specified goal statement played INC games. That is, they played by an incremental strategy that impeded the coordination it tended to seek.

- MP subjects, concentrating on highly structured goal conditions of the environment with little adaptation, represented Shimoku as an F problem--one requiring (a) discovery of a novel representation in a language of structures and relations not given by the experimenter's problem statement; (b) generation of candidate solutions in that language which could express structurally the conditions imposed on the solution as a whole; (c) testing and articulating the class of candidate solutions against all the problem conditions--those stated as properties on the whole and those imposed by the givens--in order to converge toward an acceptable solution; and (d) finding a representation of the procedural scheme for execution of what is now transformed into a new D problem, by taking parts of the givens into parts of the newly structured solution.

- Shimoku could be represented as a quasi-F problem by a subject whose AMP statement insisted on retaining <u>both</u> adaptive and structural schemes, not as a compromise but as a mutually supporting blend. The quasi-F representation had been achieved by repeated cycles of the subject's modeling his own information-processing capabilities in relation to his problem-solving experience in order to achieve successive representations of the task.

4. The task of playing while learning to play was considered in light of the information-processing requirements, not only for executing but also for modeling problem-solving procedures of the three main types. Not only were INC types harder to execute well, but they were harder to model effectively to foster learning. Learning while playing and learning between plays proceeded best in the presence of the conscious modeling of game events as it related to a subject's own information-processing capabilities.

5. Several programlike models used by subjects were considered in our attempts to account for additional aspects of Shimoku events. A subject's model of himself-as-problem-solvers and of his problem-working knowledge and heuristics, contributed in traceable ways to his building of the task model and to his modeling of interactions between himself, with his information-processing capabilities, and the given task.

Informal Experiments II: Observations and Speculations

One subject expressed his frustrated attempts at his Shimoku task by saying, "If only I could tell the machine to do ...!", suggesting that his cognitive load could be significantly lifted and his own mental functions sharpened if he could tell the machine to do many things as ideas came to him. We later requested that he articulate what he wanted the machine to do so that his statements and requests could be translated into expressions in User Adaptive Language (UAL) whose design had reached a near-completion state. This was an attempt to "hand simulate" and test the dynamic expressibility of ideas that come in many forms--concepts, procedures, rules, conditions, relations, etc.

The machine functions he wanted include machine capabilities to (1) search for certain features in the task environment in a sweep; (2) keep track of certain changes in the encironment and let him know when a set of specified conditions is reached; (3) accept a specification of a subgoal configuration and find a "best" sequence of actions to arrive at the subgoal within a specified time period; (4) remind him to take care of a certain aspect of the problem when he is working on some other aspects; (5) take over some of the repetitious condition check and corresponding decision making autonomously; and (6) test out man's tentative ideas in a noncommital basis, creating hypothetical conditions and manipulating them.

These capabilities and decision-aids are stated in general terms. Specific requests for machine assistance, however, are prompted by specific instances in unfolding situations and by man's evolving concepts and skills. Therefore, the subject was asked to work in the Shimoku environment again and express his requests to the machine as the need arose. To his amazement, he found it difficult to express some of these requests in English precisely and completely.[1] Once they are expressed in English

[1] Rather than requiring complete specification at the outset, "partial specification" is allowed in an evolving situation (Hormann, 1970a).

precisely and completely (sometimes many phrases and sentences had to be used to express a very "simple" idea!), UAL expressions could be written, corresponding to the English statements.

It is important to realize that information-processing requirements force man to specify his psychological/intellectual assumptions explicitly and completely; deeper insights are needed into thought-process articulation than into skills in information processing.

1.3 SUMMARY OF PERIOD 3: PROMOTING MAN-MACHINE PARTNERSHIP

The emphases, during this period, were on man-machine synergistic team work and group planning and on real-world problems.

Close examination of some real-world problems was undertaken to identify types of problem situations that cannot be handled adequately by man alone or the machine alone. Major characteristics of these types of problem situations were articulated and the types of decision dynamics influenced by these characteristics were identified. Attempts were made to understand interdependencies of man's capabilities and limitations and the machine's potential capabilities and limitations. Then, man's tendencies in handling complexity and uncertainty were investigated and the concept of "cognitive economy" was defined.

The design of Gaku, which was implemented during Period 1 and has evolved during Period 2, has undergone further changes to incorporate desirable new features and remedy limiting factors that were identified. The new design of Gaku allows group planning and is to provide assistance at many levels of planning (conceptual, definitional, developmental, and operational), and through iterative decision steps of problem solving (goal setting, alternative generation, consequence estimation, and evaluation and alternative selection). The basic framework of Gaku incorporates these levels of planning and stages of decision making.

Although man-machine cooperative planning must be comprehensive, benefits from man-machine techniques are expected to be higher at the strategic level of planning than at the tactical and operational levels. An optimal coupling of complementary man-machine capabilities is being attempted to deal with the conceptual and definitional levels of planning, particularly in situations that are complex, ill defined, and open ended and in situations that involve group planning where different viewpoints and value systems may be present.

Adaptive planning, which is emphasized here, recognizes the necessity of the planning processes to allow for and anticipate changing conditions in the environment and also to allow for incompleteness in man's knowledge and understanding of the problem situation. Techniques used for this

adaptive planning are "partial specification"--leaving some unspecified portions to be filled in later--and "aggregate specification"--specifying decision rules (or, at a higher level, the way rules may be generated or modified) by generalizing conditions and processes.

In the man-machine context, these techniques can be used to bridge the gap between the impreciseness of human thinking and the preciseness and completeness required by the computer. Complete formalization is not necessary at the outset. Man can finesse the computer restrictions by partial specifications and supply details later as they become available and/or clarified. Aggregate specification is a useful technique used by man when he delegates detailed processes to the machine, as these become identified as useful and delineatable. Once aggregated, these processes are handled as a single unit in man's communication to the machine.

The necessity is stressed of allowing a mixture of factual and value-oriented information, qualitative and quantitative measures, and objective and subjective judgments to be expressed and manipulated, especially at higher-level planning stages. A way to handle this is developed by the use of a "fuzzy-set" concept and its associated techniques, combined with the classical utility theory. The handling of implicit trade-off concepts and the handling of planners' diverse views, stemming from different specializations, backgrounds, and value orientations, in team planning, are also investigated (Hormann, 1971a).

It is hoped, that, eventually, the man-machine team approach will lead to the faster generation of higher-quality plans and that it will open up new possibilities of dealing with complex, changing situations that have heretofore been inaccessible to computer assistance.

Publications covering this work are Hormann, 1970a, 1970b, 1971a, 1971b and Hormann _et al_, 1970.

2. PERIOD 1: DEVELOPING MACHINE CAPABILITIES (by Aiko Hormann)

During this period, the research objective was to construct, by programming a computer, an "intelligent," "adaptive" system capable of handling a set of increasingly complex tasks that,when performed by a human, are usually said to require "intelligence." Such an adaptive system, when sufficiently developed, might be used in a man-machine partnership in which the man can demand increased responsibility and participation from the machine in solving more complex and difficult problems than have been relegated to machines in the past.

Consideration of human problem-solving and learning activities permitted analogies to be drawn and suggested the use of certain techniques and processes for the machine. The resulting system of programs, however, was not meant to be a model of human thought processes. Some "nonhuman" (e.g., extensive tracing, searching, and cross-referencing) features of the machine were also explored to evaluate possible design alternatives for optimally mixing human and machine capabilities and to discover fruitful techniques which are not necessarily a deliberate imitation of those used by humans.

2.1 GENERAL DESCRIPTION OF THE SYSTEM

The initial design of an adaptive system, called Gaku,[1] was based on a relatively simple, feedback-using, iterative process that is often observable in human problem solving. When a human faces a problem with only partial knowledge of how to proceed, his behavior at the outset may resemble random trial and error. As the problem unfolds, however, his behavior becomes more selective, directed, and organized; inadequate actions are corrected or adjusted by the use of new information gained as a consequence of previous actions, and the solution strategy is often discovered in the course of this process itself.

Figure 1 depicts such a cyclic process schematically. The cycle passes through an analysis and test phase, a tentative selection or correction phase, and a consequence-generation phase. A feedback loop is formed when the analysis and test phase of the cycle receives the consequence of a proposed action generated in the consequence-generation phase. When the analysis and test phase is reentered, the given task is reformulated or reanalyzed by a comparison between the consequences received and the

---

[1] Gaku, a Japanese word meaning adaptive, is the name given to a system of computer programs that has been evolving for several years. The first generation of Gaku is being discussed in this section.

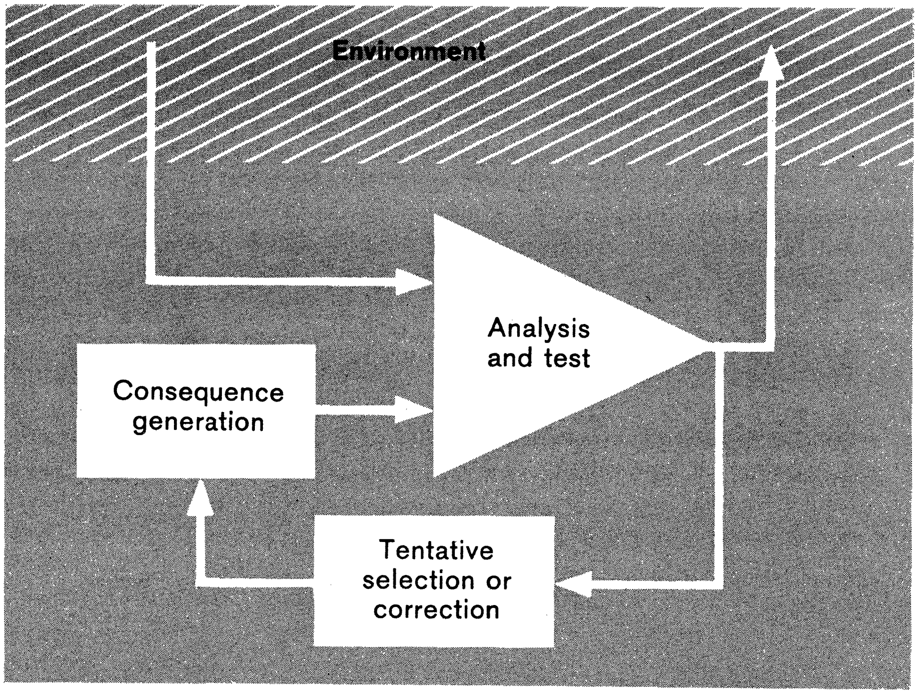


Figure 1. Three-Phase Cycle

description of the given task. Then, in the tentative selection or correction phase, either a new course of action is selected or the previously proposed action is modified. The three phases are passed repeatedly until either a success or a failure is determined in the analysis and test phase. This cyclic process is the basic feature common to all the mechanisms in Gaku.

There are four mechanisms in the system:

(1) A programming mechanism, responsible for manipulation and generation of internal programs;
(2) A problem-oriented mechanism whose actions are determined by a given task environment;
(3) A planning mechanism that takes a larger view of a given task and guides the problem-oriented mechanism by designating a rough sketch of (or guideposts for) a possible course of action; and
(4) An induction mechanism that takes a still larger view by attempting to apply Gaku's past experience to related problems not previously encountered.

These mechanisms are activated and integrated by a mechanism coordinator, as shown in Figure 2.

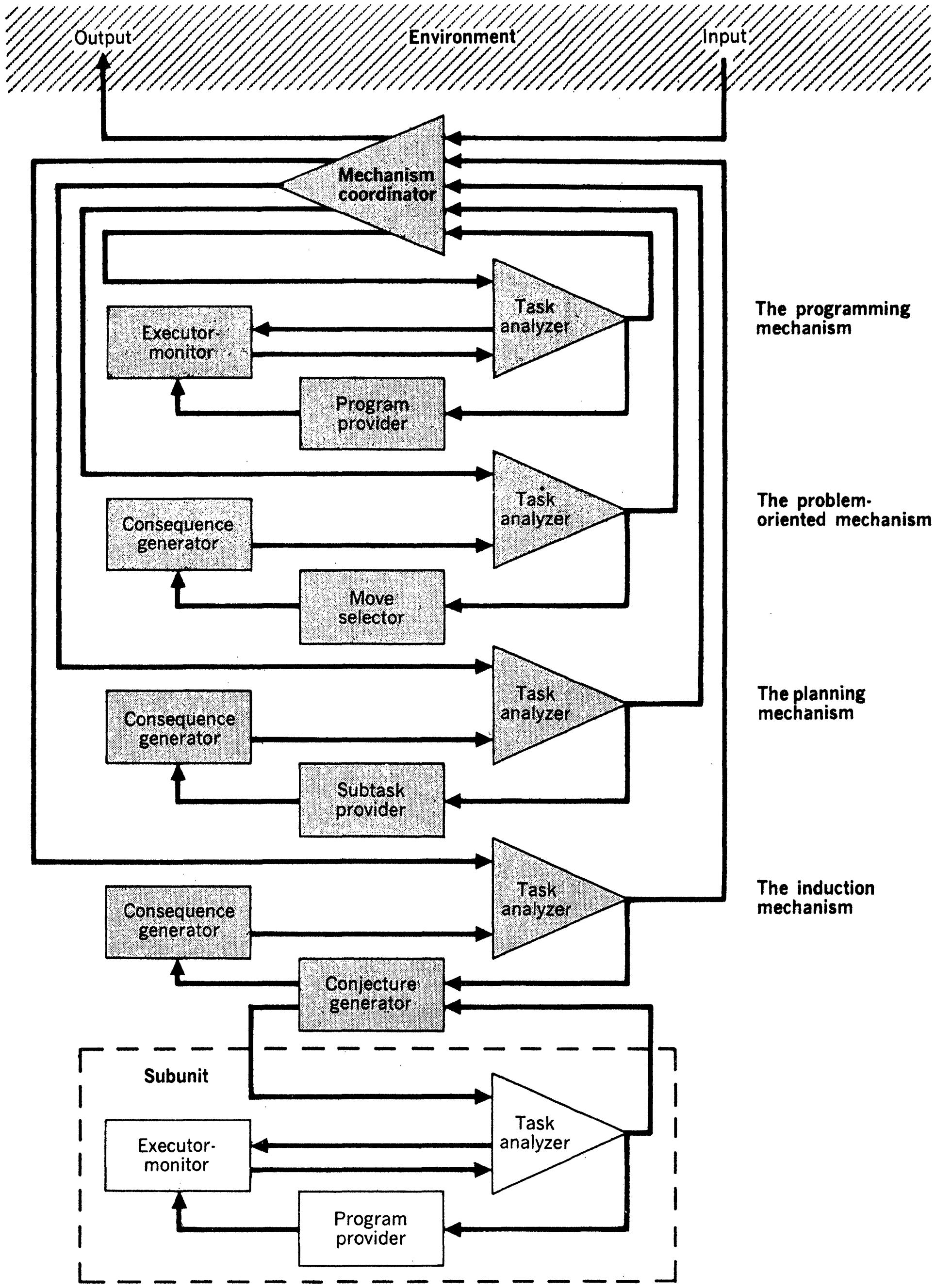


Figure 2. Four Mechanisms Coordinated

2.1.1 Direct and Indirect Specification of System Behavior

The behavior of Gaku is determined by both a direct and an indirect means. The direct means involves detailed, explicit specification of responses or response patterns in the form of built-in programs. The indirect means is supplied by the four mechanisms mentioned above, which, though they are also built-in programs, are capable of collecting, organizing, and transforming information as well as manipulating programs. In turn, the information and programs thus generated or modified influence subsequent actions of the system. These mechanisms represent one way of telling Gaku implicitly how to solve problems. Instead of being told explicitly how to solve a particular problem, each mechanism is given general rules for making decisions within its assigned level of activities, so that a discriminating search through a space of solution attempts can be carried out that has a better probability of early success than random or exhaustive searching.

2.1.2 Other System Features

Other built-in features of the system include rules and criteria for decision making, and rules for changing criteria, that are given to the task analyzer of each mechanism. To execute the decison-making scheme properly, means of internal communication between mechanisms and parts of mechanisms are provided by a set of signal conventions and routines that discriminate between signals and follow a course of action appropriate to the signal and the situation in which it occurred. (Similar features are found in the General Problem Solver [Newell, Shaw, and Simon, 1959, 1960].) There are also tree-growing and tree-locating routines of various kinds. They include a past-experience-record tree, a subtask tree used by the planning mechanism, and a move tree generated by the problem-oriented mechanism.

2.2 MODULAR IMPLEMENTATION OF GAKU

Gaku was first implemented on a Philco 2000 computer, using the IPL-V programming language, and later reprogrammed for the AN/FSQ-32 computer. Since detailed documentation is available (Hormann, 1962, 1964, 1965), only a condensed description is presented here.

2.2.1 Problem-Oriented Mechanism

Initially, Gaku contained only the problem-oriented mechanism with its associated decision rules; simple variations of the Jumping Frog puzzle (Hormann, 1965) were used to test its workability. The self-adjusting rules within the feedback principle worked well for simpler versions, but the limitation imposed by the system's piecemeal manner of attack and adjustment became clear as the complexity increased. The scope of the search in the tree of alternative steps was still less than the amount required for the exhaustive search, but increasing complexity created exponential growth of the tree and quickly made computer processing infeasible. The difficulty was

anticipated in the realization that, for complex problems, step-by-step, piecemeal decision rules and heuristics would fail unless there were also a mechanism for analyzing problem structure and placing guideposts on the road to the goal.[1]

2.2.2 Planning Mechanism: Placing Guideposts on the Road to a Goal

The planning mechanism was then added to Gaku. The task analyzer of the planning mechanism was given a set of characterizing expressions of the given set of tasks (in this case, the Jumping Frog puzzle) such that a particular subset of this set serves to define a particular task category. After surveying the task description, the planning mechanism associates the given task with a set of characterizing expressions and subdivides the task into a hierarchy of subtasks, each presumably easier to perform than the original task. This hierarchy of subtasks constitutes a rough sketch of a possible course of action to guide the problem-oriented mechanism. Concrete objects manipulated by the planning mechanism are state descriptions that represent stepping stones, or intermediate nodes, in the gap between two initially given nodes (the start node and the goal node). Finding a complete solution can be thought of as establishing a valid path (one made up of legal or admissible steps) between the two nodes, while the plan of the solution suggested by the planning mechanism is made up of wider-spaced stepping stones. These stepping stones, or intermediate states, are then handled by the mechanism coordinator, which instructs the problem-oriented mechanism to solve one portion of the problem at a time.

2.2.3 The Tower of Hanoi Puzzle

In order to evaluate Gaku's performance with the planning mechanism added, the Tower of Hanoi puzzle was used as a testing vehicle. Because its solution is know, evaluation of performance is not difficult, but the puzzle task is by no means trivial; human subjects with no prior exposure to the puzzle usually take hours, or even days, to find the shortest sequence of moves for the eight-disk case. The upper half of Figure 3 shows the usual appearance of the puzzle, with three pegs and eight disks. The

---

[1] Minsky (1961) points out that "practically any ability at all to 'plan,' or 'analyze,' a problem will be profitable, if the problem is difficult." To illustrate the point, he says, "Generally speaking, successful division [of a complicated problem into a number of subproblems] will reduce the search time not by a mere fraction, but by a fractional exponent. In a graph with 10 branches descending from each node, a 20-step search might involve 10^{20} trials, which is out of the question, while the insertion of just four lemmas or sequential subgoals might reduce the search to only 5×10^4 trials, which is within reason for machine exploration....Note that even if one encountered, say, 10^6 failures of such procedures before success, one would still have gained a factor of perhaps 10^{10} in over-all trial reduction!"

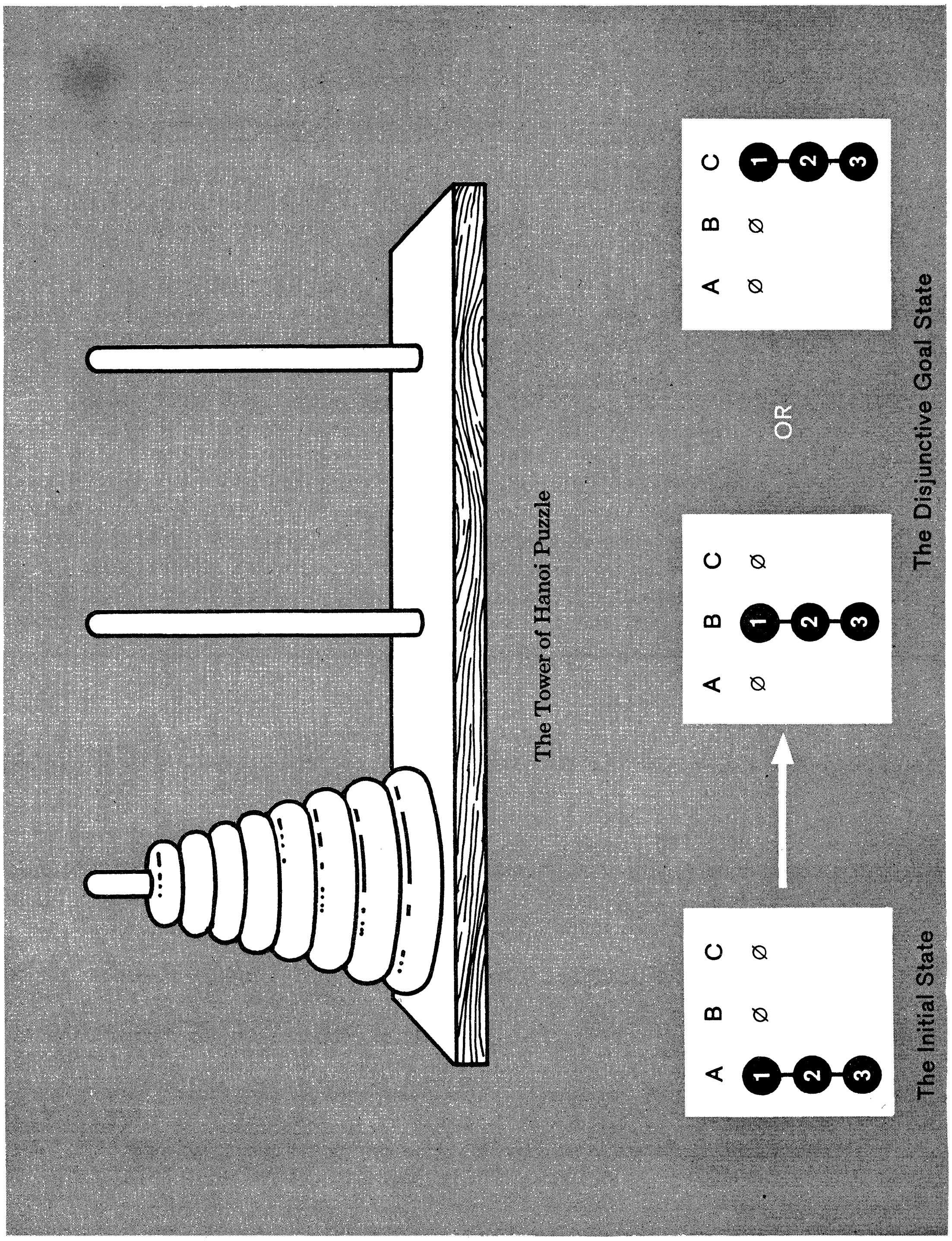


Figure 3. The Three-Disk Case of the Tower of Hanoi Puzzle

problem is to transfer the eight disks one at a time to either of the two empty pegs, never placing a larger disk on top of a smaller one. The lower half of the figure shows (schematically) the way in which the three-disk case of the puzzle was presented to Gaku. The letters "A," "B," and "C" represent the three pegs, and the circled numbers "1," "2," and "3" are the numbered disks, from the smallest to the largest. A slashed zero indicates an empty peg.

In order to present idealized training, the three-disk case was given first (subsequent tasks involved increasing numbers of disks). Lacking previous experience, Gaku exhibited a high proportion of trial-and-error behavior in this first exercise. Nevertheless, the overall pattern of Gaku's behavior was not wholly exhaustive or random; this is because the problem-oriented mechanism activated special routines to analyze and characterize the problem in terms of the relative positions of the disks. After a successful sequence of moves for the three-disk case had been found, Gaku was next given the four-disk case. This time, Gaku activated the planning mechanism in order to find possible structures that could suggest subtasks. The resulting decomposition of the tasks treated the four-disk problem as a combination of the three-disk problem (the task that was completed before) and a one-disk problem. The subproblems thus created were attacked separately. The resulting reduction in the search steps was six (by planning) compared to 65,534 (by exhaustive search) for this four-disk case.

The five-disk case was treated similarly, this time with two possible decompositions: as a combination of the three-disk and two-disk problems or a combination of the four-disk and one-disk problems, always utilizing what have been stored in the memory as past successes.

These exercises could go on to a larger and larger number of disks with similar successes as long as the training sequence did not impose a big leap in complexity. This was not surprising, since human-learning patterns also seem to indicate that graded training sequences usually produce good results. However, human learning and problem-solving behavior includes something more--i.e., some form of induction, or extraction of underlying patterns or "unifying principles," that extends the human's sphere of "coping" far beyond what was actually encountered. How a human discovers such underlying principles is still largely unknown. However, an attempt at mechanical induction was then undertaken.

2.2.4 The Induction Mechanism

A very simple approach of providing Gaku with a means to generalize was tried out, simply by implementing another feedback-loop unit with specific information contents (e.g., conjectured patterns) to be channeled through it. The resulting induction mechanism was geared to detect recurrent patterns and group them in different ways to generate possible conjectures toward formulating "unifying principles."

This time, the mechanism coordinator was also implemented to take over an executive function of assigning different problem-solving phases to different mechanisms and coordinating their functions. The same training sequence, starting with the three-disk case, was given.[1] From the four-disk case on, all the mechanisms were involved. The induction mechanism generated a conjectured pattern of successful moves, which Gaku tried out for the four-disk case as an extrapolation of the three-disk case. Because it was only partially successful, the conjectured pattern was modified by a process of comparing its steps with the steps in the newly found successful solution. This modified pattern was used at the next level, the five-disk case. Again, some discrepancies were found, but fewer than had been found previously. In this manner, Gaku continued to attack progressively more difficult but similar problems, each time using its previous experience. Finally, through the induction mechanism, Gaku found a general solution pattern.[2]

This set of exercises for Gaku was successful in the sense that Gaku exhibited an aspect of learning in a relatively simple class of problems. The mechanical induction, however, although successful in this particular case, cannot be generalized to a wide variety of problems. Besides, the fact that the induction mechanism was implemented after some cases of the puzzle had been tried out created an "impure" experimental condition. No great insights into induction *per se* were gained. A better insight was gained into the workability of the overall structure of Gaku, using feedback loops as hierarchically structured decision units (the mechanisms). Each unit deals with higher-level (or more aggregated) information than its subordinate unit (in the descending order of the induction mechanism, the planning mechanism, the problem-oriented mechanism, and the programming mechanism).

2.2.5 The Programming Mechanism

The programming mechanism, as designed, is to do "internal programming"--i.e., the construction of programs by manipulation of basic operations and prestructured programs. Concrete objects to be manipulated by the program provider are sometimes basic operations, sometimes previously generated subroutines (modified or unmodified), and sometimes a mixture of both, depending

---

[1] All the published accounts of Gaku's behavior in attacking the Tower of Hanoi puzzle describe them *after* all mechanisms had been implemented. One-mechanism-at-a-time implementation was not reported.

[2] The description here is highly condensed. In order to make this report reasonably self-contained, Gaku's performance on the Tower of Hanoi puzzle is included as Appendix A.

on the system's past experience and the particular task requested of the mechanism. A mechanism called the community unit was previously implemented to serve the functions of both the programming mechanism and the problem-oriented mechanism. Rudimentary results on internal programming, using a simple programming task, were reported (Hormann, 1962).

During Period 1, the work on the programming mechanism was dropped, but insights gained during Periods 2 and 3 can be used to resume the work to produce some practical results. Essential ideas are based on man-machine cooperative techniques (in which "secondary learning" plays a part, as discussed below) and flexible man-machine communication languages to guide the machine's internal programming.

2.3 PRIMARY AND SECONDARY LEARNING: NEED FOR MAN-MACHINE INTERACTION

So far, development of Gaku had been mainly concerned with so-called "primary" learning, i.e., learning to adapt to new situations by firsthand experience. The transition into Period 2 began with new emphasis on "secondary learning," i.e., learning from the experience of others (for humans, reading books and listening to teachers, parents, and friends; for Gaku, humans and possibly other machines). Humans learn a great deal about many facts, rules, and implied meanings from other people and from books, without actually experiencing them first-hand. Unless a man is brought up as the fabled wolf-child, he can hardly avoid learning through verbal communication. Indeed, through his verbalization ability, man's learning can extend beyond the limitation of time and space. An average person will make very slow progress in learning, say, chess, solely from firsthand experience. Progress will be faster if he can consult chess books and talk with other players. Gaku should be given similar opportunities for both primary and secondary learning. There could be a mixture of what might be called lectures (secondary learning), and hints of graded sequence (secondary learning evoking primary learning) on each subject area, as well as a demonstration of particular problem-solving techniques or methods and generalization processes.

Gaku's built-in capabilities and secondary-learning capacity would make Gaku more responsive to human needs and guidance in evolving, unfolding problem situations for which complete specifications of machine processes or capabilities in advance would be infeasible. Specifically, functions of the three main mechanisms--the problem-oriented mechanims, the planning mechanism, and the induction mechanism--may be enhanced by human guidance in criteria setting, selective search processes, subgoal generation, conjecture generation, etc. A plan was developed to provide a direct link between the mechanism coordinator of Gaku and a human. To this end, research in man-machine communication, especially on flexible languages and visual display devices, was begun.

2.3.1 Man-Machine Communication Language

We began with a modest idea of providing a set of higher-level programs (macros) with which man can specify rules for subgoal generation, pattern characteristics, etc., for planning and induction mechanisms. After we gained access to the AN/FSQ-32 time-sharing system, a more ambitious plan was developed to design our own interactive language, not only for man-Gaku interactive problem solving but also for problem-defining processes. We were motivated by the following observation.

This traditional method of defining a problem-solving task (e.g., the Tower of Hanoi puzzle) for a computer system was to (1) "program in" the task description directly in the system with the system's language (in our case, IPL-V)and then (2) create a special set of problem-specific terms and action programs (e.g., numbered disks, labeled pegs, and moving of a disk from one peg to another) for any problem-solving interaction between man and the system.[1] Because of these programmed-in sets, each new task had to be programmed anew, and its corresponding set of interactive terms and actions had to be newly specified. These two sets were separately treated because the external representation and manipulation of task elements was usually different from the internal representation and manipulation.

2.3.2 Desired Interaction Features

These restrictions cannot be tolerated if Gaku is to become useful in assisting men in real-world problem situations and to handle (1) a variety of problems, not treating each problem independently but with some "adaptation" to both old and new challenges, each time increasing its repertoire of capabilities; (2) problem-solving interaction with a man who may dynamically create new terms and processes that are not included in the initial set and who may wish to delegate certain details to Gaku by formulating new decision rules (secondary learning on Gaku's side); and (3) ill-defined problems whose definition in the man's mind may undergo changes while he is engaged in problem solving processes, requiring restatement of the problem to Gaku; therefore, the problem-definition stage cannot be separated (by a language or by a programmer) from the problem-solving stage because of the iterative nature of the two stages in most ill-defined or partially understood problems. The same person, using the same language, should be able to iterate over the two stages.

---

[1] Even slight variations of the previous task would require reprogramming unless such variations are explicitly incorporated at the outset into parameter settings.

For these reasons, a part of Period 2 was spent in designing the User Adaptive Language (UAL), which would facilitate man-Gaku interaction for these three purposes. The design was not completed until toward the end of Period 2; therefore, man-machine interaction provided during Period 2 was of a fixed-set type except for some "hand simulation" of dynamic changes in interaction using some features of UAL.[1]

---

[1] A short description of UAL is given in Appendix B.

3. PERIOD 2: EMPHASIS ON UNDERSTANDING MAN'S HIGHER MENTAL FUNCTIONS IN COMPLEX PROBLEM SOLVING

3.1 PRELIMINARY WORK (by Aiko M. Hormann)

A research plan was generated to probe deeply into problem-solving and planning activities of humans and possible decision-aiding functions of computers. The plan included (1) designing a task environment that can be controlled to vary in complexity, objective characteristics, and elements of uncertainty; (2) effecting differing degrees of man-machine participation; and (3) designing a series of experiments with varied groups of subjects under differing conditions. Although it was possible to execute only a small part of the plan[1], the resulting data collected are extremely rich with heuristic "tricks," some discernible cognitive "styles," a variety of plan-formulation and pattern-recognition/utilization procedures, and different degrees of awareness of changing states in the environment and of changing understanding about the problem and about self-as-problem-solver capabilities. These are described in detail in Section 3.2.

3.1.1 Designing a Task Environment

Eight desirable features were stipulated prior to searching for and designing the task environment.

1) The task environment must allow meaningful interpretations, with a reasonable degree of transferability to real-world situations, to be drawn from the problem-solving behavior evoked and studied; therefore, the task environment must contain many important problematic features that are often found in the real world.

2) The environment must be formalizable and abstract in nature, for two reasons. First, in evaluating the performance of human, machine, and man-machine problem solving, detailed records must be kept of decision steps, environment changes, queries, the use of decision aids, etc., all in relation to each version of the environment. To do this, the variables in the problem situations must be known and controllable; it must be possible to isolate them from other variables that are not an intrinsic part of the task. A more realistic, substantive version of the task environment can then be used for the study. Second, an abstract task environment with many important problematic features can provide a valuable resource from which to glean useful techniques, principles, and guidelines for the future construction of man-machine systems and can be expected to have wider applicability than a concrete, narrowly specified task environment.

[1] The original plan was constrained by the reduction in SDC-ONR funds and the loss of SDC's access to the AN/FSQ-32 time-sharing system, on which all the Gaku programs and its task environment had been built.

3) For the purpose of producing a wide variety of experimental conditions, the environment must contain more than one class of problems. In addition, it must be possible to grade the problems in each class from relatively simple to highly complex, so that generalization and the use of past experience (by both man and Gaku) can be observed.

4) For each problem, many steps should be necessary to reach the goal, and each action step causes some changes to the environment. Thus the task can be conceived of as finding an effective course of action--by means of which to change the given environmental condition into a desired condition.

   For convenience, we label such problems "sequence-seeking" problems and contrast them with "enumeration" problems, in which the task is to find out how many possible ways there are to do something, and with "configuration" problems (e.g., magic squares), in which the task is to produce a structural arrangement of parts satisfying a given set of criteria--i.e., the emphasis is on the final configuration, not on how to get there. We believe that many challenging problems are of the sequence-seeking type (e.g., theorem proving, computer programming, and problems of constructing or improving conditions in our environment such as urban planning and transportation system development). We hope to be able to find some underlying principles, operational methods, and decision aids that can be applied to the class of sequence-seeking problems.

5) At each decision-making step, many alternatives must be available so that the total combinatorial complexity of the problem is overwhelming to humans and relatively simple machine processing alone will not make the problem "transparent"; nor will exhaustive search by machine be feasible.

   The rationale behind this kind of complexity is to elicit planning and strategy-formulation behavior and clever uses of machine assistance. We wish to investigate what man tends to do when the complexity increases to a point where previously workable methods are no longer feasible; what separates a good problem solver from a poor one; what type of machine capabilities tend to extend and sharpen human cognition and in what kinds of situations; and what kinds of communication are most effective in coupling both capabilities.

6) The task environment must include cost and payoff functions such that, for each assigned task, some range of achievement is possible. Measurements of performance must avoid the black-and-white concept of "success or failure," "right or wrong," and "win or lose."[1] The environment must also include opportunities for trade-offs and for understanding that local optimizations do not necessarily contribute to the overall optimization.

---

[1] This is the main reason why many existing games (such as chess) and puzzles (such as the Tower of Hanoi) did not suit our purposes.

7) There must be a means of introducing elements of uncertainty into the environment and the experimenter must be able to decide in advance the degree and the nature of uncertainty.

8) There must be a variety of options available in choosing the manner of task performance, such as one-person "solitaire" problem solving, two-person competitive game situation, group problem solving, and man competing against Gaku as well as man assisted by Gaku.

Description of the Task Environment

The task environment that satisfied the requirements above was designed, combining features borrowed from existing games and puzzles. It was given a name Shimoku (a Japanese word meaning "four-in-a-row"). Three-dimensional (3-D) Shimoku can be thought of as a combination of a game of 4 x 4 x 4 tic-tac-toe (for the basic board and for the concept of four-in-a-row), a game of twidle(for action rules), and a pseudo-poker game (for payoff patterns). The 2-D and 4-D Shimoku are also included as part of the training sequence. In the following, the basic version is presented. Other variations are described later in this section.

Figure 4 shows 2-D Shimoku at the top and 3-D version at the bottom with patterns of counters that could give the player some payoff values. The dotted lines indicate examples of "four-in-a-row" positions in which the scoring patterns must fit. Note that 3-D Shimoku is shown in four separate planes; this is the way the environment appears on a display scope. If these planes are visualized as stacked on top of each other to make a three-dimensional cube, it should be easily seen that the dotted lines connect those cells that form a "four-in-a-row" path. The complexity in the 3-D case is easily seen to be much greater than in the 2-D case. The number of discrete entities, i.e., cells, involved in the 3-D case is 64 compared to 16 in the 2-D case, so the volume has increased four times. However, complexity has increased by much more than sheer volume; the number of four-in-a-rows, in the 3-D case is 76 compared to 10 in the 2-D case; thus the complexity, in this sense, has increased 7.6 times. If we extend the same kind of comparison to the 4-D case, the volume increases 16 times, but the complexity increases 52 times (i.e., in the 4-D case, 256 cells are involved but there are 520 possible patterns). Figure 5 shows 4-D Shimoku in 16 planes.

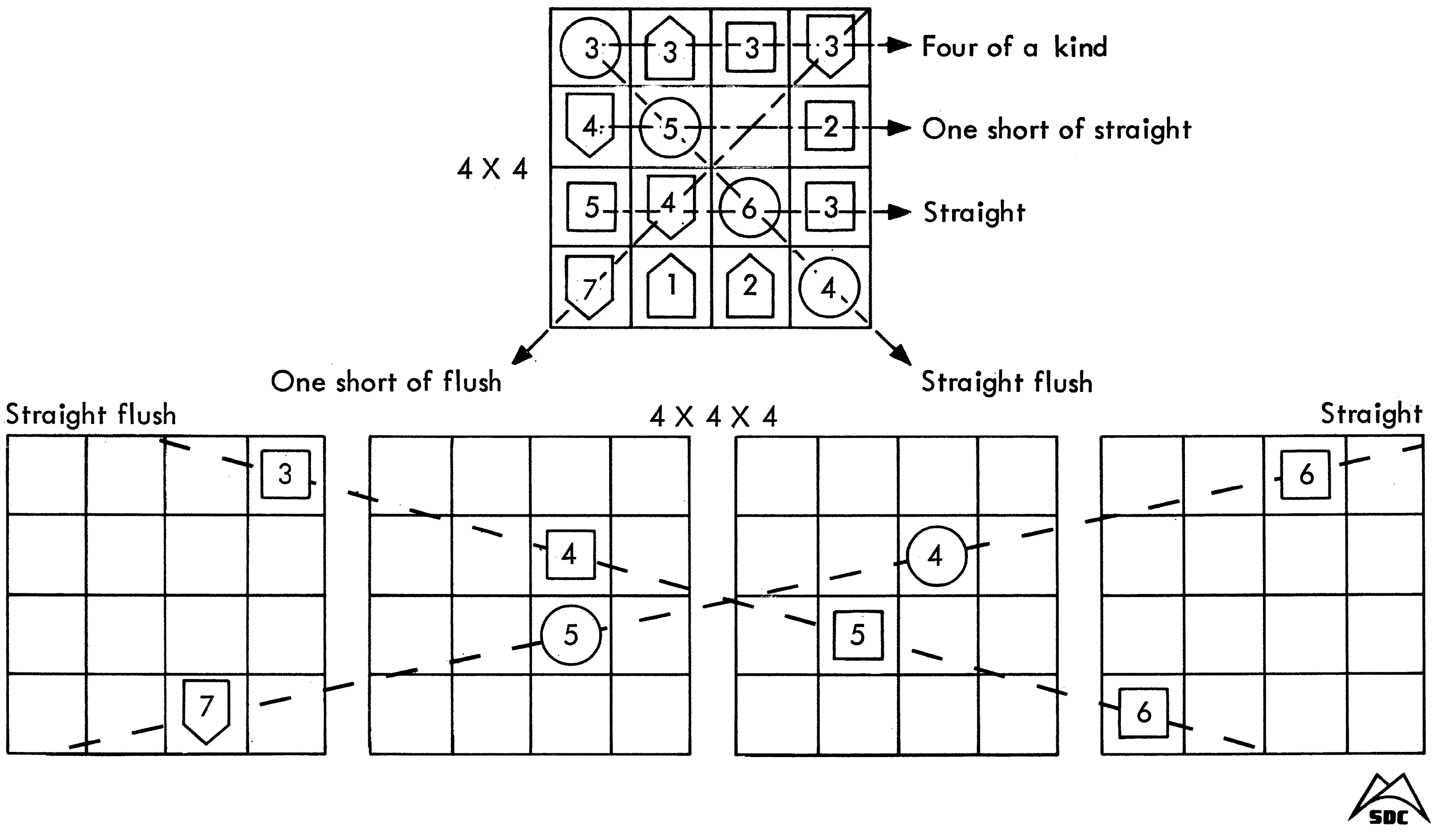


Figure 4. 2-D and 3-D Shimoku

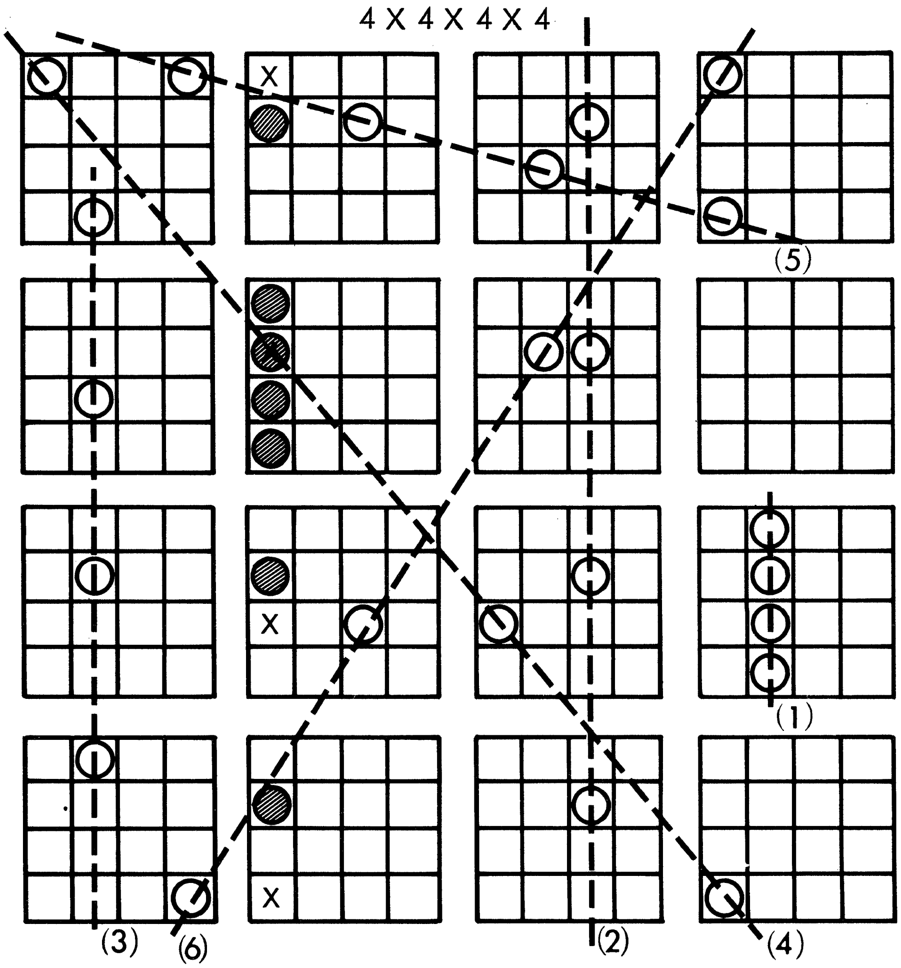


Figure 5. 4-D Shimoku

The top portion of Figure 6 shows the four different counter shapes that correspond to the suits in playing cards. They are called, from left to right, Circle (CI), Square (SQ), Up (UP), and Down (DN). These shapes are easier to distinguish on a display scope than the four symbols (heart, diamond, spade, club) used in cards. For each shape, there are two sets of eight counters (or tokens), numbered from 1 to 8. Each of the tokens in one set has a dot above the number to distinguish it from the corresponding one in the other set. Instead of using colors to distinguish opponents (e.g., black and red in chess), this convention is used for the black and white display scope of the computer. With two sets the same task-environment setting can be used both in the one-person "solitaire" and the two-person competitive game.

Payoff Patterns. The bottom portion of Figure 6 shows examples of the four patterns that count toward the final score and gives their corresponding values. "Straight" patterns need not appear in any order on the board as long as they contain four consecutive numbers. If a scoring pattern is broken by some action after it was made and earned points, the points are subtracted automatically from the total score.

Scoring patterns can be newly defined by an experimenter and corresponding point values assigned; for example, "all even numbers with four different shapes will score 12 points," and "sum of the numbers of four tokens must be equal to k in order to score 10 points," etc. Pseudo-poker patterns are used here in the basic version to facilitate the ease of remembering what patterns to look for.

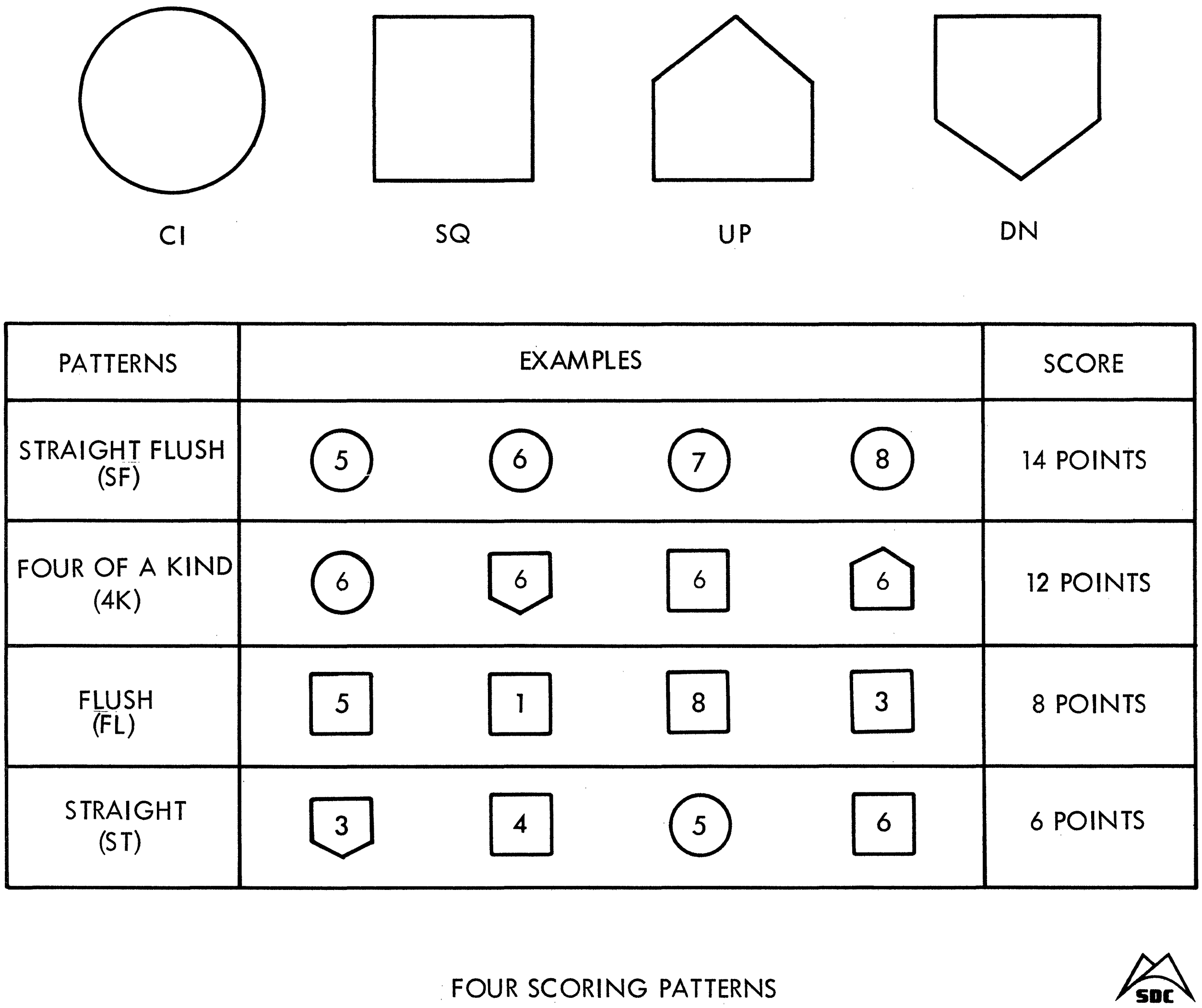


Figure 6. Four Scoring Patterns

Action Rules and Costs. Permissible actions are shown in Figure 7. They are: slide, jump, exchange, and purchase. In a slide move, a token is moved into any adjacent square that is empty; each move costs one point. In a jump move, a token is moved to any empty space; each move costs three points. In an exchange move, two tokens on any two squares on the board are exchanged; each move costs four points. In a purchase move, a new token is brought on to the board from a reservoir of unused tokens; each move costs five points. The experimenter can change the assignment of costs to any values he wishes, but care must be taken to attain reasonable separation between "expensive actions" and "cheap actions" and a good b alance between cost and payoff functions.

Additional complexity can be introduced by differentiating purchasing costs in terms of degrees of specifying purchase orders. For example, a purchase order that specifies both the shape and the number of the tokens will cost more than one that specifies the shape only. This wide range of possible cost and payoff assignments for the experimenter to manipulate was designed to bring out possible different behaviors in relation to cost/ benefit trade-off concepts and to incentives of the players.

Goals and Constraints. The standard objective is to score as many points as possible within 40 moves[1] or one hour, whichever comes first. Note that the goal condition is indirectly specified--i.e., to score many points, many patterns should be constructed; and from the rules of payoffs and costs, the player can infer that high-valued patterns should be preferred, but the "economics" of resource management must be considered and the concept of "return on investment," developed. These are only implicitly given and are inferred by the player; no explicit goal conditions are given as to what pattern configurations to select and where and how to make them.

In contrast, the goal state of the Tower of Hanoi puzzle was explicitly given in terms of its token positions and arrangement. Chess is somewhat in between Shimoku and the Tower of Hanoi. It does not enumerate all possible check-mate configurations explicitly, but it does state relative positions of pieces that constitute "check-mate" as its culmination state--i.e., the game ends with win or lose (or tie) condition. In Shimoku, such

---

[1]The first informal experiment allowed 40 moves with no time limit. Later, it was changed to 40 moves or a one-hour time limit. In the final experiment, we used 30 moves or a one-hour time limit.

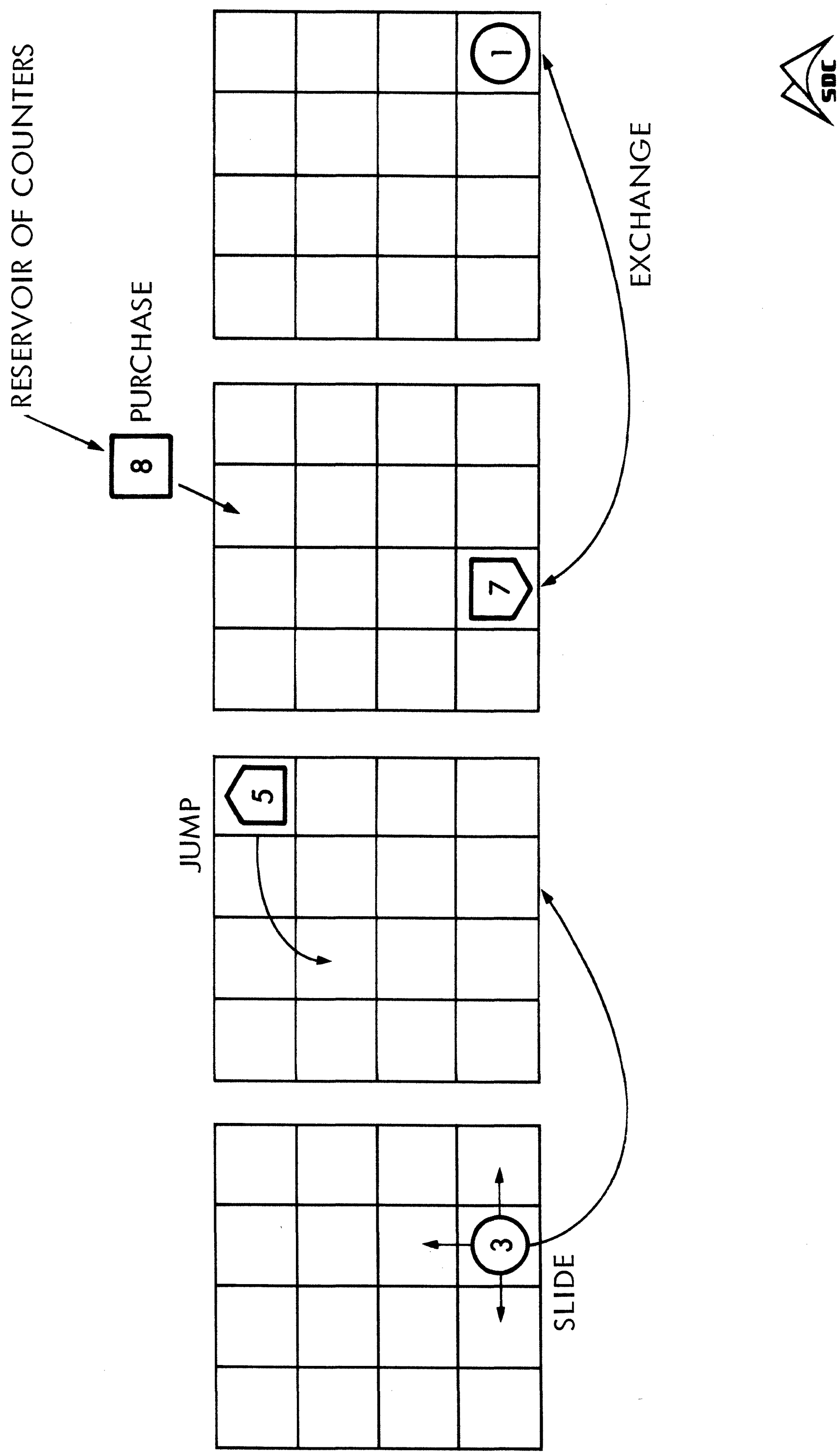


Figure 7. Four Types of Legal Moves

a definite culmination is not sought[1]; instead, range of achievement is measured for a preset time period and number of moves.

The implicit goal statement and gradable performance measure are believed to be important problematic features that are commonly observed in real-life situations. As we shall see later, an individual's understanding of the implicit goal statement and his recasting of it in his own cognitive model often constitute a part of the business of problem solving itself. That is, the problem of how to arrive at a desired condition was partially worked out by the modeling process itself (however inadequate or inefficient it might be). These points are discussed in detail in Section 3.2.3.

If Shimoku is played as a two-person competitive game, each player tries to score as many points as possible and, at the same time, to disrupt his opponent's patterns. Starting with an empty board, each player in turn places one token (from his set of tokens) on the board at a time. When the board is filled, the players take turns making "exchange" moves (among their own tokens) until m prescribed moves have been made. After m exchange moves, each player's score is calculated.

For a one-person Shimoku game, an additional constraint (besides time limit, number of moves, and permissible actions) is usually placed by providing an initial configuration. A sample starting configuration, with tokens scatterered about on the board, is shown in Figure 8. Unlike starting from an empty board, this configuration limits the player's pattern-constructing options. If he is clever and adaptive, he can take as much advantage as possible of tokens that are already on the board, but some tokens and their positions are bound to become obstacles. One subject's interpretation of the Shimoku task is rather interesting: "I am in this environment I inherited from my previous generation and I have certain limited resources; it is up to me to change the current environment to a desired condition, using legally admissible actions and the limited resources available in a most cost effective way."

Introducing Elements of Uncertainty. The experimenter can introduce elements of uncertainty in two ways. One is by random or probabilistic assignments of payoffs, cost, and/or action consequences; the other is by introducing changes at certain intervals in the board configuration, scoring patterns and their values, and action rules and their costs. An example of the first kind is "straight-flush" pattern earns 14 points 70% of the time and the rest of the time earns points randomly chosen

[1]It can be shown that 76 patterns, all of straight flushes, can be constructed on the board bringing the total payoff points to 1064. The final score can be easily calculated if the starting condition is an empty board; 64 purchases (with ideal placement), costing 320 points, will accomplish the task. The final score then would be 744 points. However, the least-cost sequence of moves, starting from a given initial configuration (e.g., see Figure 8), is not known.

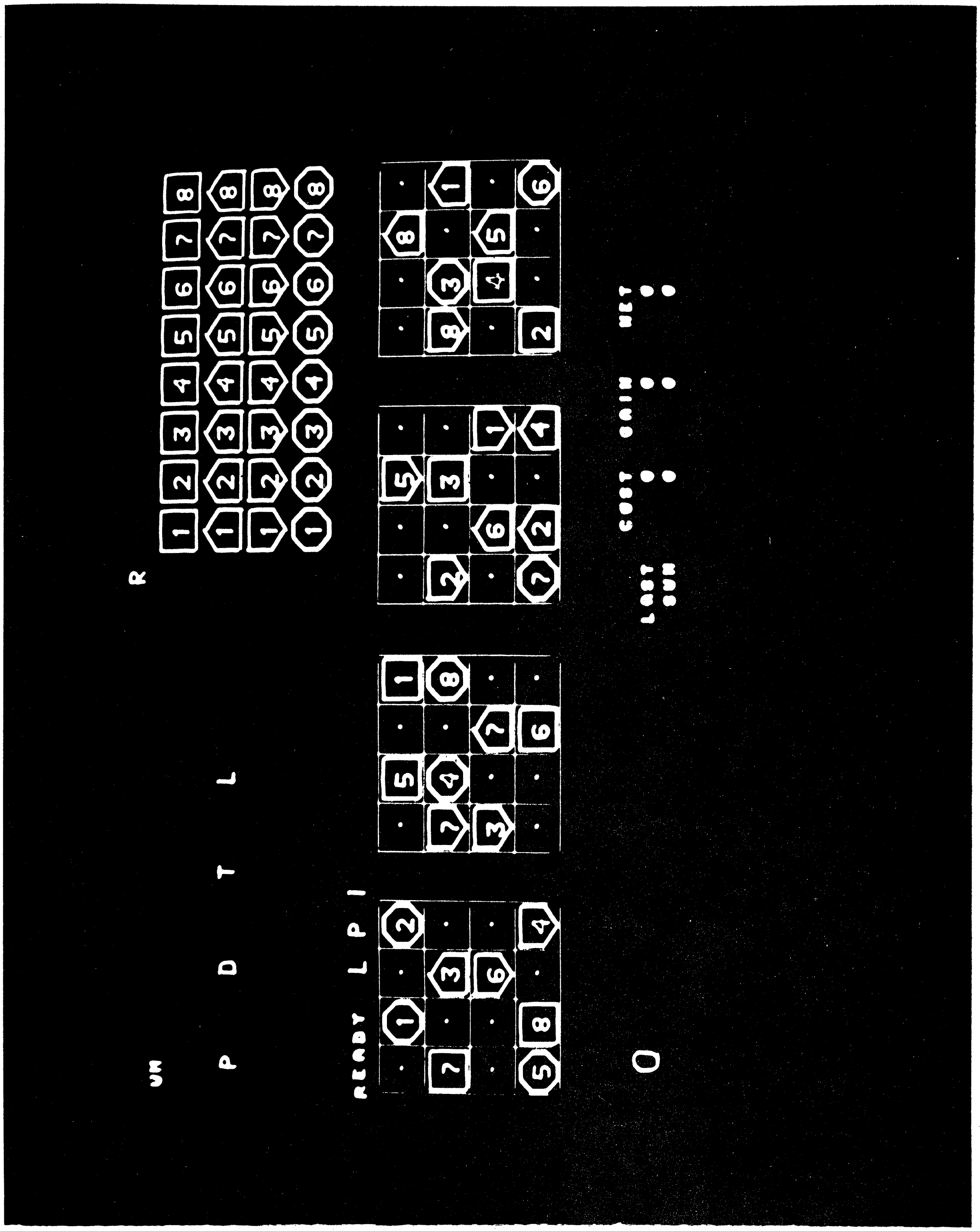


Figure 8. Sample Starting Configuration

among integers between 8 and 13." An example of the second kind is "after the player makes 10 moves, the experimenter declares that from now on, 'four-of-a-kind' is worth 15 points and 'straight' is worthless (0 points)." This may be analagous to a situation in which a company faces changes in consumer demands in favor of one product and the total disinterest in another product.

Variables in the Environment. As can be seen, many variations in the environmental settings are possible. The variables that can be controlled are: the board size, the initial configuration, two-person competitive or one-person solitaire, deterministic or nondeterministic, a wide variety of scoring-pattern definitions and payoff values, four or more legal moves and their costs.

Other concepts introduced later were Kriegspiel-Shimoku and the use of the "initial capital" for the player. Each player in the Kriegspiel-Shimoku has his own display scope showing only his own tokens on the board. He knows what token his opponent has used in each move but not where it is. He can guess where on the basis of two kinds of information. (1) the referee (Gaku can perform this task) informs him that his move cannot be made when he attempts to place his counter in a certain cell, meaning that it is already occupied by his opponent's counter; (2) the referee informs both whenever either player completes one or more patterns (it is possible to complete more than one pattern by placing one new token); the position of the pattern but not its kind then will be revealed.

The use of the "initial capital" concept simply gives the player an initial capital C_0 and allows him to borrow later in the game, using his current payoff value as collateral. Three contractual periods are played with three sets of contractual demands (specification of patterns to be completed), not all of which are completely disclosed at the outset. This variation of Shimoku, cast in a businesslike environment, is described in Hormann (1966).

3.1.2 Implementation of Shimoku and Meeting the Complexity Requirement

The design of Shimoku was implemented on the AN/FSQ-32 computer, in an on-line, time-sharing mode with visual input/output facilities.[1] Informal experiments with a few coworkers were tried out with no decision-aiding functions provided by the computer, using the 3-D, one-person, deterministic Shimoku with the four basic pseudo-poker patterns (later, seven patterns were used).

---

[1]The implementation work was performed by David Crandell, with some assistance from Stuart Shaffer and Terence Ruggles. A few special features added later for the purpose of experimentation were contributed by Antonio Leal.

The 4-D Shimoku was dropped after discovering that the 3-D case is already very complex and that the display scope was used to capacity.

All players remarked how complex and intricate interdependency is among the elements in the environmental setting, even though the rules of the game are relatively simple to learn and remember. Of special interest to them was that even a few steps of seemingly simple decisions tend to set off complex chains of conditions in the environment and that cause-and-effect relations are not clearly discernible (even in the deterministic setting). They all tended to focus on localized, short-term payoff patterns first, but later realized that things are so interrelated that local improvements or immediately observable gains do not necessarily contribute to the total performance measure. They began then to talk about "sacrifice moves," "preparatory investment moves toward a big gain later," and "tradeoff implications," and "strategies." Some decision-aiding functions that can be built into the computer program were suggested. These, together with our own ideas of what might be helpful, were implemented later; they are discussed in the next section.

Let us now turn to a concrete example of complex interplay among Shimoku patterns, action rules, and cost and payoffs. Figure 9 illustrates the need to look ahead and consider tradeoffs before a particular course of action is chosen. Suppose in this simple situation, purchasing of new tokens is prohibited but all other actions (slide, jump, and exchange) are permissible. What move or moves will gain most payoff points? Most players started with a slide move of the token "UP 3" to its left position, forming a "four-of-a-kind" pattern. This gained 12 points but cost one point for the slide and six points for breaking the "straight" pattern when "UP 3" was moved. The net is plus five, still a gain. However, this move eliminated other possibilities: the "straight" is lost permanently since there is no other 3 or 7, and the upper corner pattern of possible "flush" is lost because the token "CI 3" at the bottom is locked in.

Here, a little examination will lead to the realization that two 3's are already in the vertical column, so instead of lining up four 3's on the diagonal, why not line them up on the vertical column? This will not break the "straight" which is already on the board. In addition, the partial "flush" at the upper corner can be completed by moving "CI 3" from the bottom. Thus, the final sequence of moves decided on was "Jump 'CI 3' to the upper corner, jump the remaining 3's on the diagonal to the vertical squares." The cost is nine points for three jumps and the gain is eight for "flush" and 12 for "four-of-a-kind," securing the net gain of 11 points (in contrast to the net gain of five in the other choice).

The above example is a super-simplified condition of the board. Ordinarily, the board is full of partial patterns, not only on each plane but also across the four planes (including diagonals and skew-diagonals not shown in the diagram). Often, moving a single token can make or break as many as seven patterns or potential patterns. Other tokens in these partial patterns, in turn, are influencing other patterns, thus creating highly interdependent conditions in which a single action can create "ripple effects," consequences

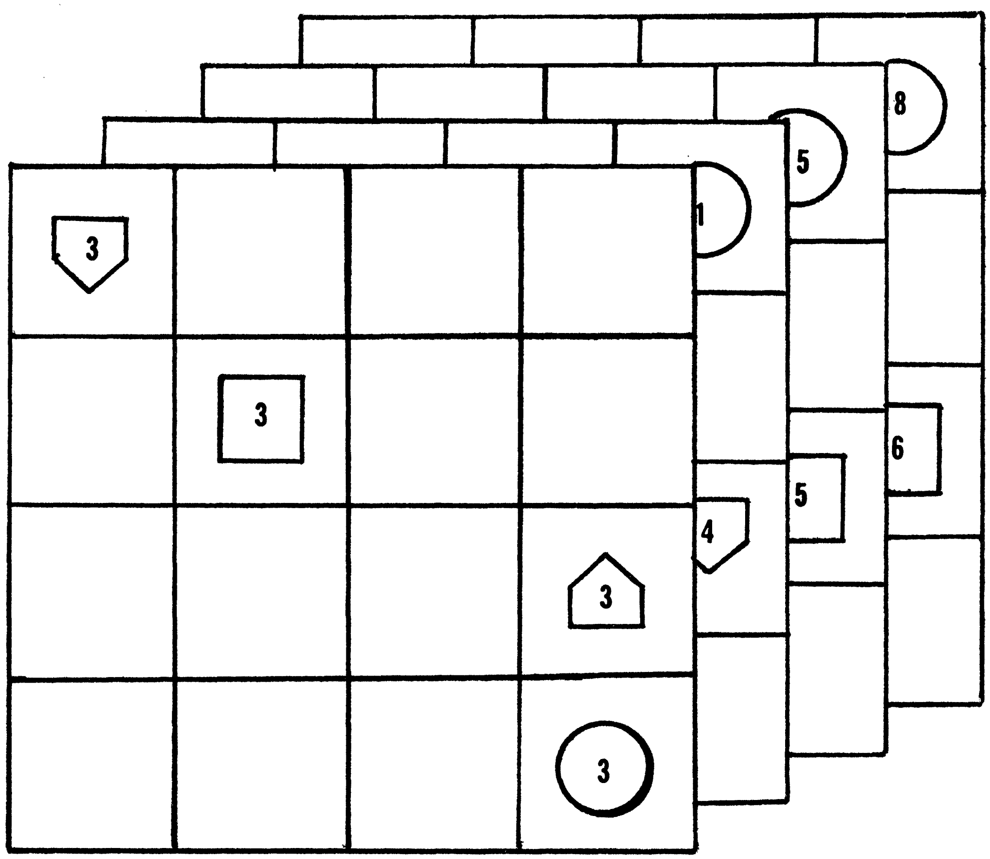


Figure 9. Interrelated Patterns

of which are not clearly discernible. In addition, the cost and payoff functions are such that the player must make judgements of relative merits (rather than yes-no, good-bad judgements) of a large number of alternatives--i.e., tradeoffs and cost/benefit concepts become important.

One way of measuring complexity is to count the number of alternatives that exist in choosing a sequence of moves of, say 40 in length. To do this, we must find the number of alternatives that exist at any one state in the environmental changes. We have the following figures:

4096 possible moves for the empty 4 x 4 x 4 board;

3006 possible moves for the board with 20 tokens;

2544 possible moves for the board with 32 tokens;

2121 possible moves for the board with 50 tokens;

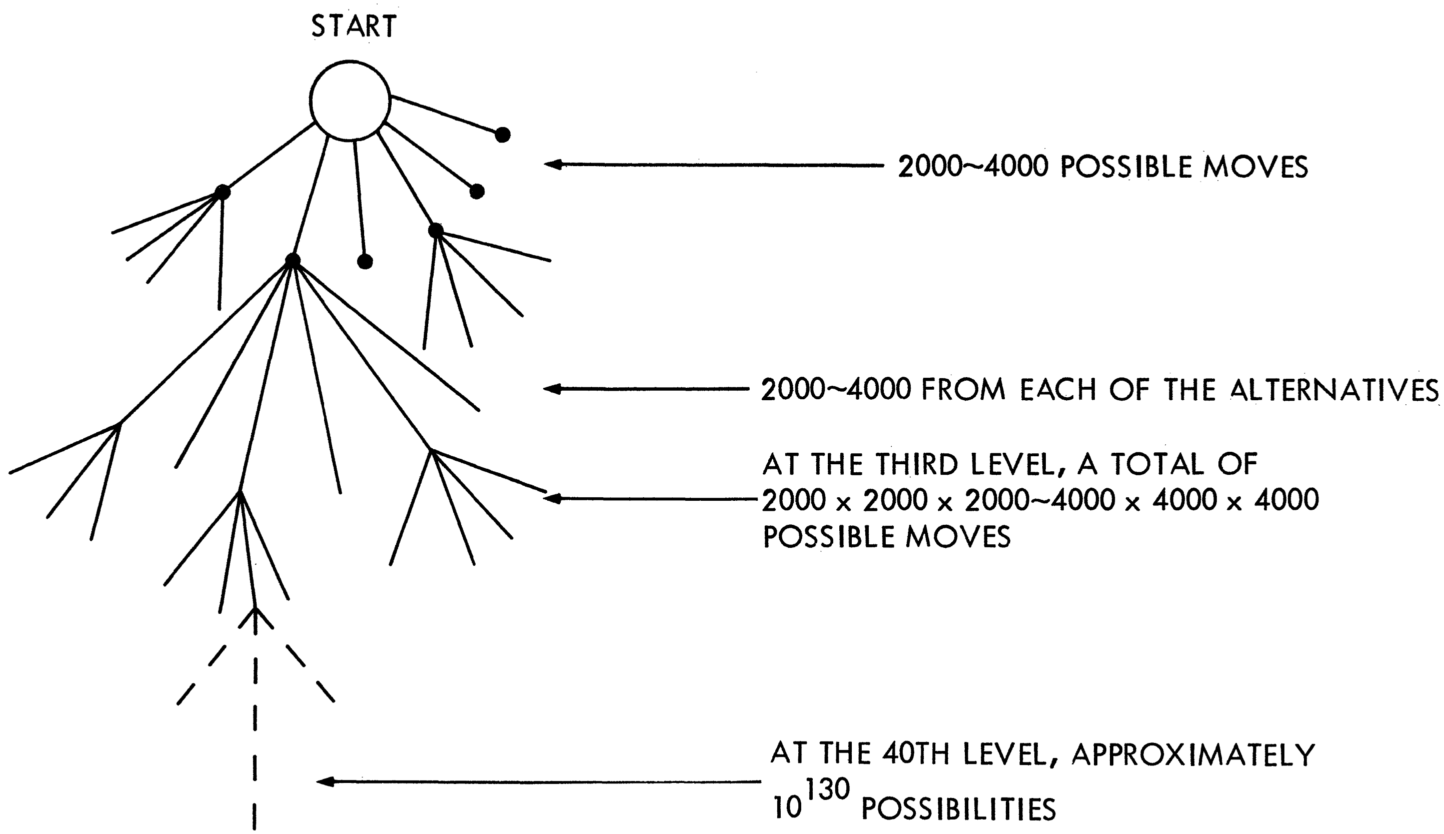
2016 possible moves for the completely filled board (64 tokens).

Let us say that approximately 2000 to 4000 moves are possible at each state. Then a decision tree which grows this many branches at each node will eventually reach, at the 40th step, approximately 10^{130} possibilities (see Figure 10). Clearly, neither brute force nor exhaustive search will be feasible.

3.1.3 Degrees of Man-Machine Participation

Along with a wide variety of problem-solving tasks within the Shimoku environment, our plans included differing degrees of man-machine participation in experimentation. They are (1) man alone (zero degree of machine participation--i.e., the computer is used only for presenting the task environment and for keeping records of man's actions and scores); (2) man with a fixed set of decision-aiding functions provided by the computer; (3) Gaku (described in Section 2) alone, while man gives only a training sequence (by setting the initial parameter values) starting with the 2-D Shimoku; (4) Gaku with man's assistance in higher-level decision guidance (e.g., specifying subgoals to the planning mechanism); (5) man-Gaku competitive game (in the competitive version of Shimoku); (6) man-Gaku cooperation with dynamic definability of decision steps and rules, search strategy, etc. (by the use of flexible man-machine communication language); and (7) two-person competitive game, each having the same version of Gaku to start with but each "co-evolving" with Gaku separately as in (6). These seven variations were planned for both the deterministic Shimoku and nondeterministic Shimoku.

After starting to work on these extensive, long-range programs and after the Shimoku environment had been implemented, we were told that the AN/FSQ-32 computer would not be available much longer (a period of three months was first announced which was later expanded to eight months). We then had to



A DECISION TREE

Figure 10. A Rapidly Growing Tree of Alternatives

decide quickly which one of the seven variations could be used most advantageously with a minimum preparation time. A decision was made to use the second variation for 3-D, deterministic Shimoku, since our major interest during this period was in man's higher mental functions coping with problems of complexity and because some decision aids were already implemented on the computer. The features of this variation are described in detail and features of (6) as planned (and "hand simulated" for some) are discussed to present the scope of the plan and the flavor of man-machine dynamic cooperation.

Man With a Fixed Set of Decision-Aiding Functions from the Computer

Paths through a point is a no-cost option to aid perceptual recognition of "four-in-a-row" legal paths (see Figure 11). Since the display scope shows the two-dimensional representation of a cube in four planes (or grids), the connection among spatially separated cells as "neighbors" is not readily seen. Using a light pen, pointing to a letter "P" on the screen (as one of the light buttons provided) and then pointing to a grid position of interest, the player can see immediate display of all straight-line paths linking four-in-a-row and passing through the selected position. Since these paths show positions of potential scoring patterns and since a given position may have as many as seven paths through it, the player can see interdependency of tokens and their positions. Seeing partial patterns intersecting at one position can help the player plan ahead to prepare for multiple-pattern, token-sharing construction economically.

A scoring graph display (see Figure 12) provides immediate feedback of the player's performance. As the player makes each move, the net gain (any new payoff values minus the cost of the move and any loss of points due to a breaking pattern) is calculated and plotted on the graph cumulatively against the number of moves. This kind of visual feedback is helpful for the player in monitoring his own performance step by step. It is also useful for evaluating the over-all pattern of his performance at the end of the game, self-critiquing and planning an improved strategy for the next try.

For the experimenter, shapes of scoring graphs are indicative of certain types of strategies or of techniques play.[1] For example, a steadily climbing graph followed by a plateau or decline usually indicates that a player is concentrating on scoring patterns on each plane separately (intra-plane strategy), without considering inter-plane patterns. After exhausting (in his judgment) intra-plane possibilities, inter-plane patterns will be tried out but some patterns that were previously completed and scored will be

*These types are discussed in detail in section 3.2.

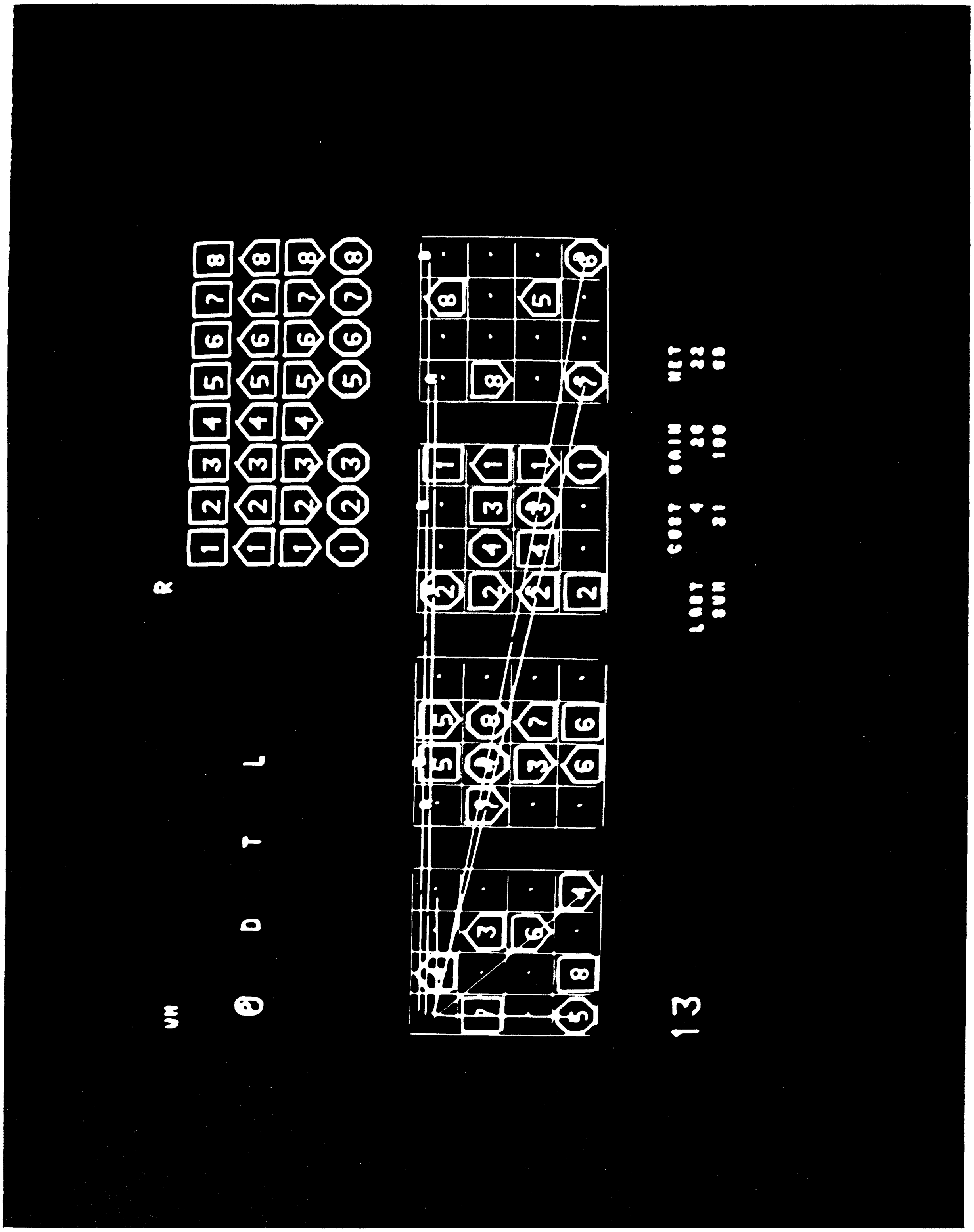


Figure 11. Paths through a Point

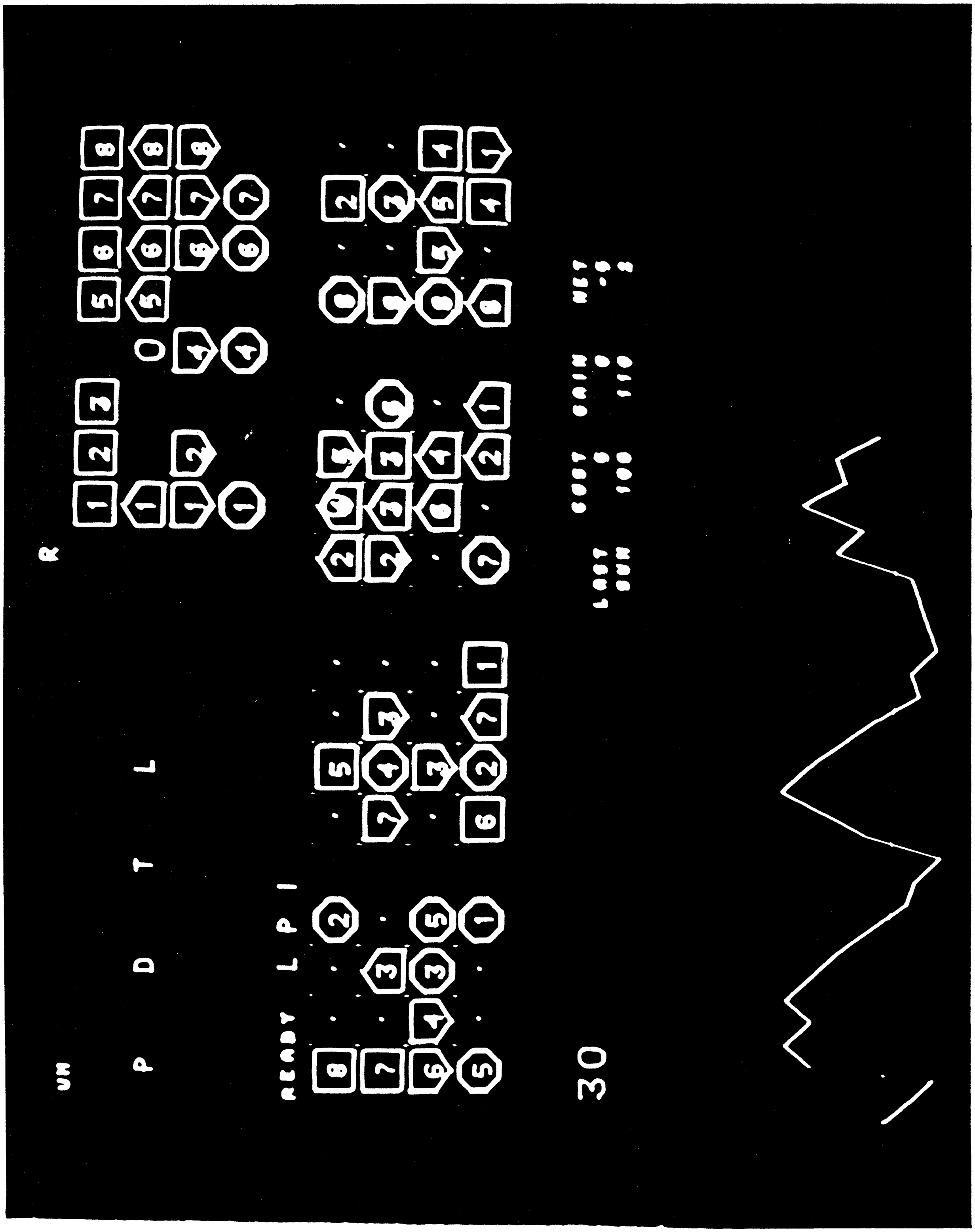


Figure 12. Scoring Graph Display

broken (thus losing points), when some tokens are moved to other locations to make inter-plane patterns. Thus scoring points by new pattern creation is often offset by breaking the old patterns. A segment of a graph that starts with some declining steps followed by two or more sharp gains is usually indicative of a play technique that made preparatory moves for multiple-pattern construction (either intra- or inter-patterns).

In addition to the scoring graph, a table of the cost, the gain (if any), and the net is displayed for each just-completed move (labelled "last" in Figure 12). Below the line for the "last" move is a second line of the table showing the cumulative running sums (labelled "sum") of cost, gain, and net over the course of the individual's game.

The scoring graph is shown for each move made, but it disappears automatically between moves or when "P" is on, erasing in turn the paths displayed. A "Display Graph" light button, labelled "D", permits the player using the light pen to call back the scoring graph to the screen. Because of the scope size limitation, the scale of the graph changes several times as the points above (or below) the line cumulate enough to outrun the available display space. This means that the slope of the graph plotting net points up or down for a given move will be greater, the lower the range of points and that the most dramatic visual plot feedback comes for the person who scores a reasonably high number of points earliest in the game. The plot thus might operate to sensitize an individual most to the impact of his actions when he is just beginning. As the game progresses, if cumulated points reach a high range, the change of the graph per point as shown to the subject becomes visibly smaller.

A "Look-Ahead" option is provided for the player to make any number of trial moves (at the cost of 1/2 point per move), erase some or all of them, revise, and repeat the process until he decides to make permanent-move decisions. This feature permits him to explore many possible alternatives and examine their consequences, showing the complex interplay of tokens and their positions, payoffs, and costs for which a mental tracing to even a few levels deep would be too taxing. Pointing the light pen to the "L" button lets him enter the Look Ahead mode. After uncommitted trials, he may accept all the trial moves or any substring of them which will remain after rejecting one or more of them in the order last to first. He exits the Look Ahead mode by pointing the light pen to the light button "A" (Accept)[1], which will accept none of the trial moves nor any which remain after he has erased any undesired parts of the trial sequence from right to left. "A" accepts and

---

[1] On entering the Look Ahead mode, the position formerly showing "L" switches automatically to show "A". After "Accept" and exit, "L" returns and "A" disappears.

puts into his committed moves those Look Ahead moves he has retained. During action in Look Ahead mode, the graph, table, and move count number have been behaving just as if this were a normal game: the subject can monitor what would be the results of his trial sequence if he were to make it real. The point of entry into this mode is marked by a plus sign (+) on the graph plot having the (x, y) coordinates of the condition at the end of the last move in the normal game mode. During action in Look Ahead, the plus itself moves downward on a line parallel to the y-axis to show an absolute cost of 1/2 point per forward move for the privilege of gathering information in the uncommitted trial mode[1].

A "Remove" move permits the player to remove a token from the board and place it in the reservoir by pointing at "R" displayed along side the reservoir. In the version used in the experiment, there is no cost or payoff attached to the action. This move can yield a severe penalty if the player purchases a token (at the cost of 5 points) and later decides to give it back, because he does not get 5 points back (an immediate "Unmove" will undo the purchase move without any cost, as discussed below). An interesting variation, equivalent to a "partial refund" concept, was considered--i.e., to give back 3 points (not the full 5 points) to the player whenever he moves a token from the board to the reservoir. For simplicity, we did not use this variation; however, most players became aware of the penalty quickly and moved unwanted tokens elsewhere on the board where they could be useful, instead of giving them back to the reservoir.

The "Unmove" (UM) light button permits the player to handle technical errors (e.g., pointing the light pen at a token other than the one he really intended to move). By pointing a light pen to UM on the scope, the player can undo his previous move, either a completed move or part of a move. A completed move consists of "move part 1" and "move part 2" since all the legal moves,

---

[1] During an informal experiment, one full point was charged for each trial move, and practically no one used the option. Even reducing to 1/2 point did not encourage cost-conscious subjects. If it were completely free, some subjects claim, then they would use the option but the time-factor must be weighed against possible benefits to be gained by the use of this feature.

A deeper analysis seems to indicate that the main deficiency of the Look Ahead feature is its single-dimensional nature--i.e., it traces a single sequence of moves at any one time (although the graph also shows the results of the previous string of trial moves along with the current one). Comparing different alternative courses of action in an over-all, "pattern-perceiving" level was needed.

Slide, Jump, Exchange, and Purchase, require specification of two things--what square (or token) and what token (or square)--to complete a move. Although UM was provided for correcting technical errors, players use it when they do not like the results of their actions. However, a player cannot undo his move more than once in succession in normal play mode. In the Look Ahead (L) mode, however, it is possible to erase any number of previous moves one-by-one, retracting moves back to the plus (+) mark on the scoring graph which indicates the point of entry into the L mode (he cannot go beyond the marked point).

All Unmoves are free of point cost. Since the game is under a time limit, however, all cost-free operations--P, UM, D, R, L--have a "cost" in time. The evolution of a player's use of UM for quick exploration of single moves has been interesting to observe.

Other decision aids that were implemented or planned but not used in the experiment include marking and unmarking of the tokens that are being used as a reminder to the player, and machine search that can be requested to locate all sc-1's ("scoring-pattern minus 1" is a partial pattern needing one correct token to complete it).

Marking of tokens is useful for the player to remind himself that these tokens are already being used in making patterns or partial patterns. Since unintentional breaking of patterns is costly and relatively frequent, this bookkeeping device was helpful. Intentional breaking would be made for the purpose of higher gain later; and the player can unmark any of the marked ones. This device was implemented but dropped from the experiment because extra dots used (at the bottom portion of tokens) for marking cluttered the display.

A search procedure was written to locate all the sc-1's on a given board configuration. The procedure was never used because of the awkwardness of reporting the sc-1's found[1]. They had to be either shown on the display scope one-by-one, or the location numbers on the board typed out (requiring the player to switch the output mode to "teletype"). A similar program for finding "scoring moves" (those moves, each of which completes pattern(s) and gains score points) was also written but was not used for the same reason. These aids are discussed again in the paragraphs on on-line specification of decision-aiding function and the delegation of details to Gaku.

---

[1]The procedure was fast enough to be feasible since the maximum number of patterns on 3-D Shimoku is 76. Furthermore, to reduce the number of output, the user can impose additional conditions, such as "sc-1's whose potential payoff values are greater than 8".

Man-Gaku Cooperative Problem Solving with User Adaptive Language

After we began the second informal experiments in the Shimoku environment, it became clear that a fixed, predetermined set of decision-aiding functions could give man only a limited way of extending his capacity. With no provision for defining new terms and procedures as he went along, the player could convey only those queries and actions that could be expressed either by pointing the light pen at prescribed parts of the display scope in a designated manner or by typing in some words and expressions that were prespecified[1].

One subject expressed his frustration by saying, "If only I could tell the machine to do ...!" The machine functions he wanted include machine capabilities to (1) search in a sweep for certain features in the task environment; (2) keep track of certain changes in the environment and let him know when a set of specified conditions is reached; (3) accept a specification of a subgoal configuration and find a "best" sequence of moves to arrive at the subgoal within a specified time period; (4) remind him to take care of a certain aspect of a problem when he is working on some other aspects; (5) take over some of the repetitious condition check and corresponding decision making autonomously; and (6) test out man's tentative ideas on a noncommital basis, creating hypothetical conditions and manipulating them.

These capabilities and decision-aids are stated in general terms. Specific requests for machine assistance, however, are prompted by specific instances in unfolding situations and by man's evolving concepts and skills and, therefore, cannot be anticipated. Most of all, it was desired to delegate detailed work to the machine. But man does not know what to delegate when he first confronts the task; decision rules and conditions to be specified for machine processing gradually emerge only after man becomes familiar with the task situation and explores cause-and-effect implications (such exploration itself can be carried out more extensively by man-machine cooperation). We also found that by adding more and more specific decision aids we would eventually reach a point of diminishing return where the user's capacity to remember and make use of all the possible combinations of available aids would be exceeded. Therefore, it was realized that (except for those aids frequently useful to the majority) a <u>means</u> of expressing these requests and ideas dynamically and easily is more important than having a specific set of decision aids.

As described earlier in Section 2.3, our parallel work (along with Shimoku) during Period 2 was to design a man-machine communication language that will facilitate dynamic expression of ideas to elicit increasingly responsive

[1]For the semi-formal experiments carried out later in this period, interactions permitted were only through the display scope and the light pen.

behavior from the machine as man proceeds in his problem-solving activities. The design of User Adaptive Language (UAL) has undergone several stages of modification and redesigning but a relatively stable and satisfactory stage was reached toward the end of this period. Some "hand simulation" of man-Gaku interaction in the Shimoku environment was tried out since it was still a long way from having the implemented totality of both UAL and the Gaku design.

In Period 3, the UAL design was documented (April 1970) and later implemented features of UAL. Some difficult-to-implement features of UAL were deleted, but features implemented later are available and were documented (Hormann *et al* 1971).

3.1.4 The Original Plans Toward Staged Experiments

The original plans for using the Shimoku environment in a variety of experimental settings had four major stages: hypotheses generation, hypotheses testing, training, and training-testing.

Hypotheses Generation Stage

In this stage, a series of small informal experiments were to be conducted, using only a few subjects from each of several diverse groups (e.g., engineers, artists, clerical workers, children from middle-class neighborhoods, and children from disadvantaged neighborhoods). For the purpose of hypotheses generation, the plan was to examine each individual in the Shimoku environment closely in terms of:

- problem-solving heuristics, both clever and not so effective,
- use of past experience in a sequence of graded complexity, and any generalization effort observed,
- thresholds of cognitive overload, and use of cognitive economy[1],
- cognitive "style" of handling complex tasks,
- interpretation of a given task and possible restructuring of it,
- evolution of plans and subtasks, and master-planning processes,

---

[1] The definition of "cognitive economy" and the related discourse are given in Hormann (1970).

- evolution of self-as-problem-solver models,
- specific use of machine aids in exploring cause-and-effect relations and trade-off implications,
- handling abstract concepts of "cost" and "benefit" (without any tangible substantive representation such as money), etc.

Hypotheses Testing Stage

The plan was to examine conjectures, speculations, and hypotheses generated from the first stage and select those that were interesting or appeared promising and that could be tested in a larger scale, systematic series of experiments.

Some statistical techniques for analysis, classification and evaluation of collected data (e.g., "TRACE" [Shure, *et al*, 1967]) and for examining correlations by man-machine interactive analysis of multivariate data ("IDEA" [Press, *et al*, 1969]) were planned.

Training Stage

The plan in this stage was to create explicit training sequences with machine aids and display techniques in the same Shimoku settings. The previous two stages purposely avoided any contact with the subjects that might influence their problem-handling behavior. (The only contact was in making the objective statement of the task and in interviewing to *elicit* introspective accounts, but no information or hints about problem-solving activities were given). For this training stage, in contrast, we planned to hold critiquing sessions with carefully planned graded sequences of hints, suggestions, and instructions and to discuss with the subjects the possible alternatives that could have been pursued, what characteristics of the Shimoku board to look for, what decision aids might be useful for exploration, possible restructuring of the tasks interpretation to get different viewpoints, possible master planning procedures that might be helpful, etc., while increasing the complexity of the tasks by small degrees.

This stage was to be the pilot study for extracting useful training methods, techniques, and devices that could be adapted to different individual cognitive styles, cognitive capacities and cognitive-economy orientation, and information-processing needs.

Training-Testing Stage

In this stage, the plan was to choose two substantive tasks (e.g., resource allocation and sequential decision making in a business management game and in urban planning) that were of similar nature and complexity; and to observe the subjects' performance in both tasks, one before and one following a Shimoku training period. We were hopeful of discovering and formulating effective training "theories" within this circumscribed context and of constructing training "vehicles" to be used in abstract settings (but with important problematic features) such as Shimoku. We also expected to gain deeper insights into man-machine cooperation that includes flexible role switching in teaching and learning between man and machine for more effective and adaptive divisions of labor.

A Compromise Plan for the Experiment

As soon as we found out about the time and fund constraints, we decided to modify the hypothesis generation stage. Since no other stages are likely to be covered, we wanted to have more than a few subjects to make our hypotheses generation a little more credible or meaningful. However, since we tried to use as many volunteers as possible, rather than paying the subjects, we relinquished the requirements of having controlled groups of subjects representing diverse backgrounds. Detailed descriptions of two groups of subjects we finally managed to recruit and of a modest amount of "control" exerted for the experiment are given in Section 3.2.2.

3.1.5 Compromise Version of the Shimoku Environment Adopted for the Experiment.

After weighing a number of factors, we finally settled on a compromised version of the Shimoku experiments which was not too ambitious but was interesting enough so that there would be a high probability of getting worthwhile results within the constraints of the limited time period and funds. Since there would be no time to experiment with varying degrees of complexity, the "frozen" version we adopted was a deterministic, noncompetitive (or one-person solitaire) 3-D Shimoku with a limited set of built-in decision aids. The same set of four legal moves (Figure 7) and costs was used and seven scoring patterns (instead of four patterns previously described) are used to define payoff conditions and points (Figure 13). The stated objective was to attain "maximum" points at the end of 30 moves or 60 minutes, whichever came first.

The same starting configuration (Figure 8) was used as the opening game "board" for all experimental runs. That is, runs 1, 2, and 3, for all subjects started from the same initial configuration. We planned to generate some of the "topological equivalents" by rotating the same token positions in the cube and slicing the four planes from different sides of the cube. This would retain the "connectivity" among squares in the

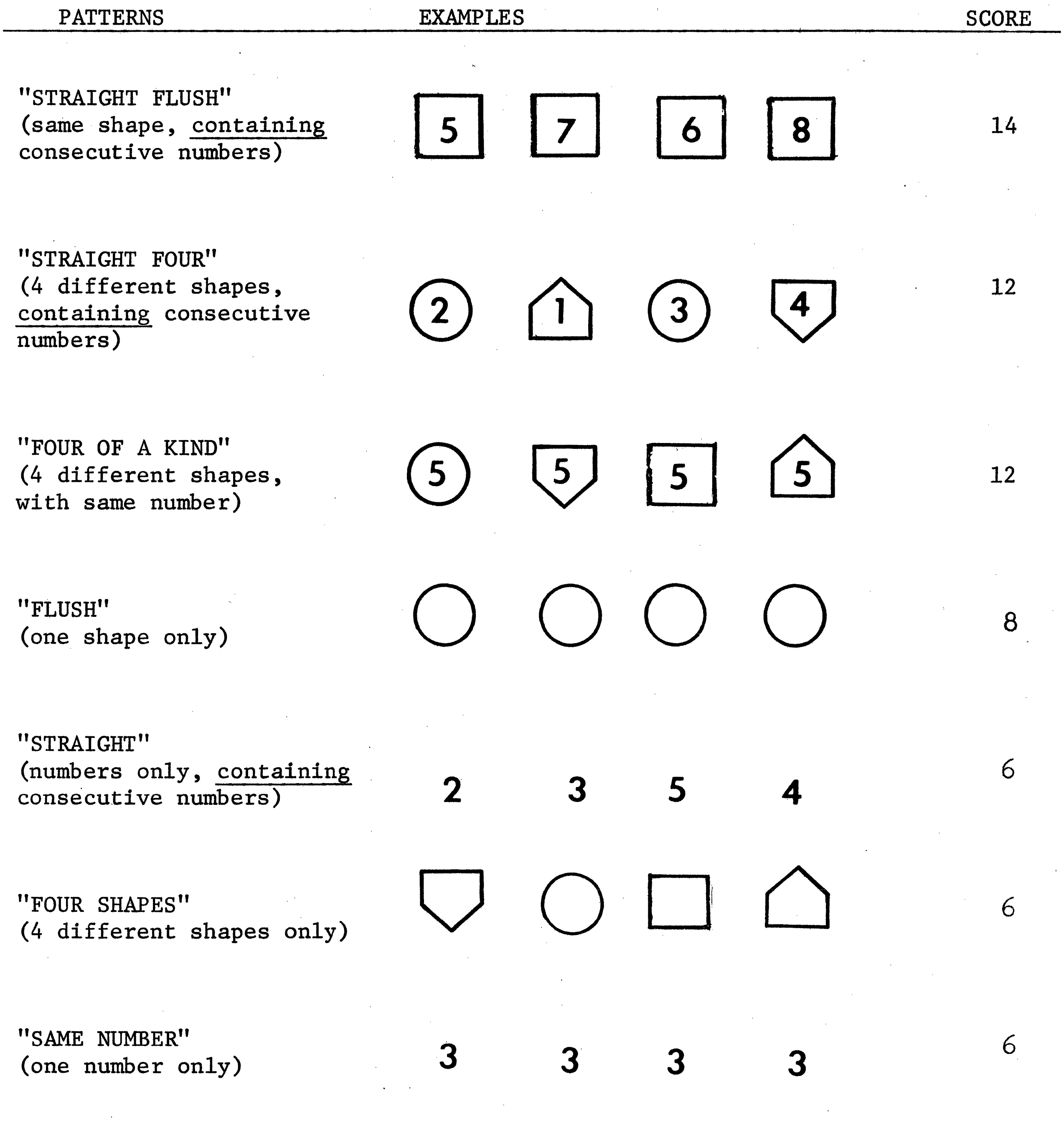


Figure 13. Seven Scoring Patterns in Shimoku

cube and, therefore, would also retain the relative positions among the tokens. Although these would be mathematically equivalent, they might appear quite different perceptually/psychologically. For one thing, the two-dimensional representation of 3-D Shimoku tends to make intra-plane patterns more readily perceivable than inter-plane patterns. It seemed likely to confuse our analysis on a limited sample if we had to wonder about such differences. We, therefore, elected not to use topological equivalents or different configurations, and used the identical opening board. As it turned out, most subjects did not realize that it was the same each time. The configuration had 32 tokens scattered about the board, appearing so miscellaneous, so utterly noncommittal and difficult to characterize in any particular way that many subjects commented on its unsuggestiveness. Only one person suspected that the configuration might be the same, toward the end of the third game; even then he could not remember any of the specific token positions.

Built-in aids provided were "Paths through a point" (P), "Display scoring graph" (D) along with the table of scores, "Look Ahead" (L) option at the cost of 1/2 point per move, "Remove" (R), and "Unmove" (UM), as described earlier.

Additional features had to be included in the computer to facilitate the experiment. It was decided important to have the computer keep a complete record of all actions--not only legal moves with corresponding costs and payoffs (if any), but all aids used and other actions through the light pen designation (P, D, L, R, UM). The record was kept of moves by type (Slide, Jump, Exchange, Purchase) with two actions, token characteristics (shape and number), and loci at move start and end. At the end of the game, the subject exits the light pen mode by stroking light button "T". The mode changes to the Teletype mode that permits the research assistant to start the computer printout of a complete action table. The table includes action number, legal move number, specification of actions as mentioned above, and cost table data comparable to that on the scope for a given move. At the bottom of the printout was given a summary of all the actions by the action categories and the total number of actions executed within each category. Computer printout of an action table is reproduced and commented on in Appendix C.

An additional feature is the "time clock", which records computer internal clock time, starting from the first action and ending at T (Teletype). This feature was implemented and added to the printout of the action table. It is a valuable feature for reconstructing "thinking time", apparent punctuation of move sequences, etc. As we shall discuss later, one subject might pause a long time, scratch his head, then make a few moves rapidly in succession, and pause again; another subject might start with a rapid succession of moves, almost impulsively, then slow down to use some exploratory moves with "paths through a point" and "look ahead" options. Except for the head scratching, these activities are all recorded for later reconstruction and analysis.

Time for the total run starts from the experimenter's signal "go", and is recorded separately (manually), since a subject may wait some time before starting to move. Similarly, end-time is called by the experimenter at the end of the allotted hour. Total time elapsed may exceed an hour in cases where the time-shared system has been "down." The experimenters have tried to record down-time allowances as accurately as possible, but the time measurement could not be precise, since time shared machine response time on some days was slower than other days. Attempts were made to get relatively uniform conditions, using off-peak computer time between 5:00 and 8:00 p.m., but variability was still a problem we could not fully control.

Figures 14 and 15 are the photographs of the two physical models of 3-D Shimoku used in the instruction session. They are shown to a subject side-by-side to illustrate the correspondence between the three-dimensional cube and the two-dimensional representation. We would then change the tokens around within the cubical lucite model and ask the subject to make corresponding changes in the two-dimensional representation. After manipulation of tokens in this manner, the subject was taken to the display scope showing the Shimoku "board" to practice the use of the light pen in moving tokens, requesting "paths through a point," etc. Figure 16 shows a player who is about to use the light-pen on the display scope. More detailed discussions on game instructions are presented in Section 3.2.2.

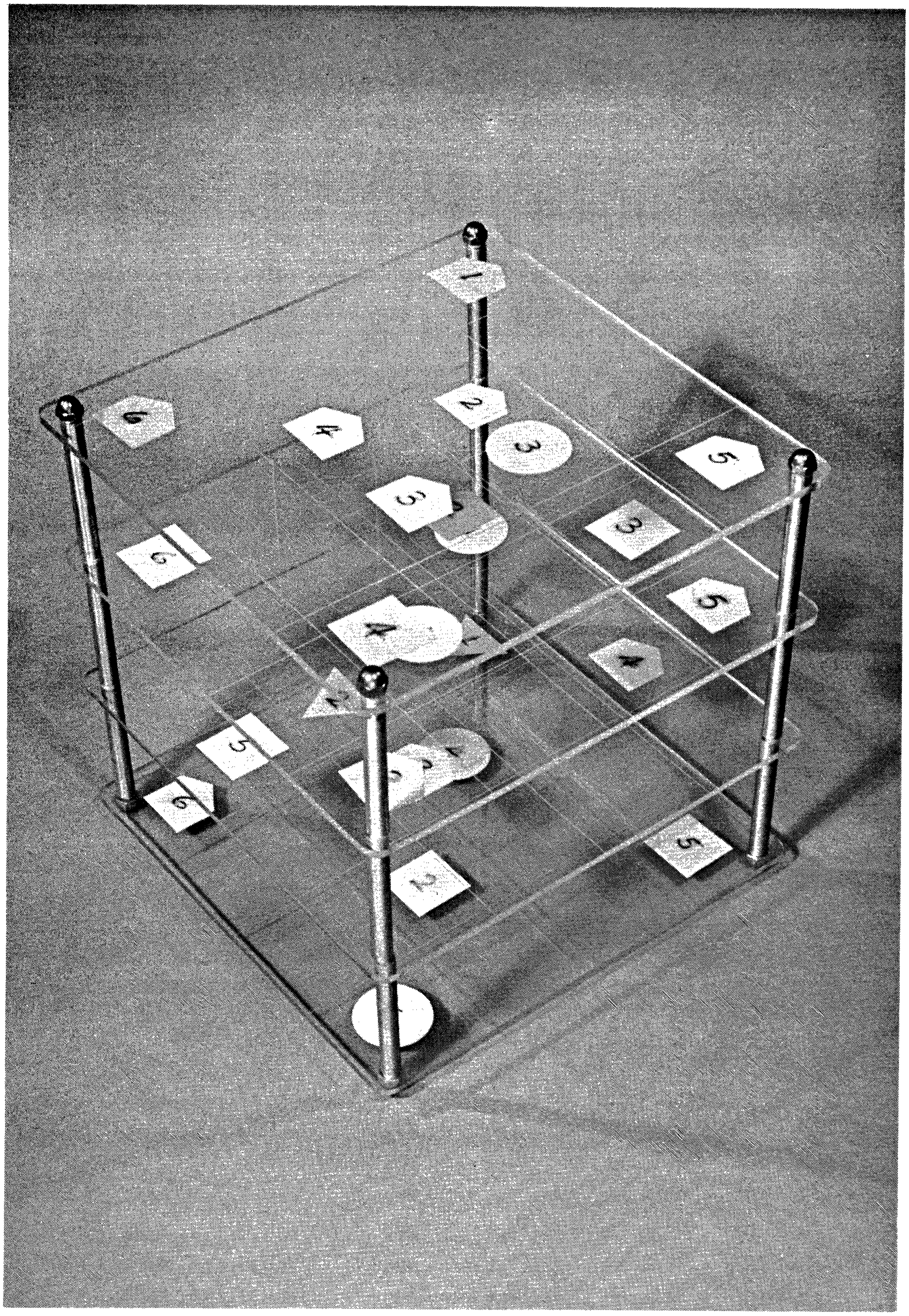


Figure 14. Physical Model of 3-D Shimoku

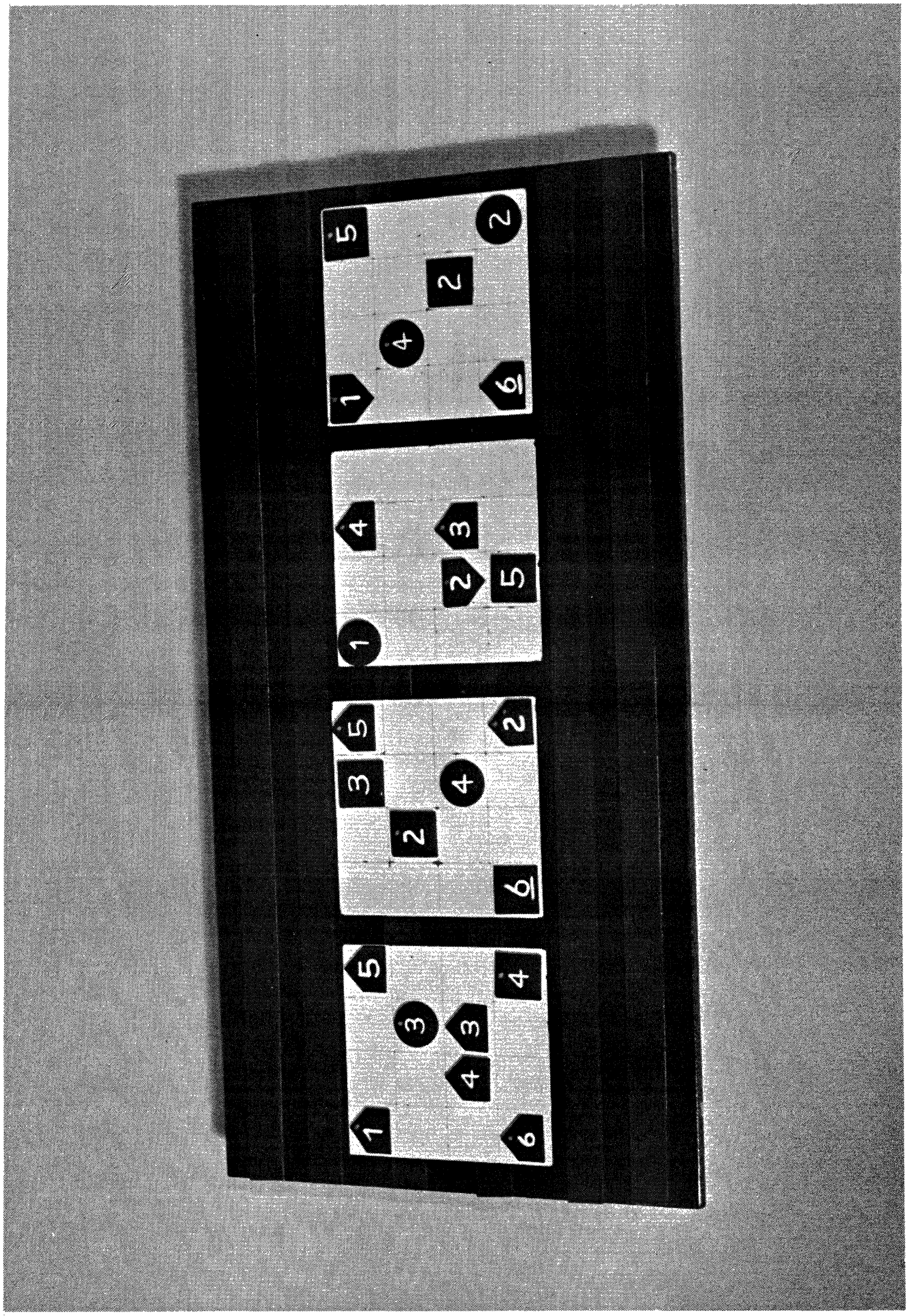


Figure 15. Two-Dimensional Spread of 3-D Shimoku

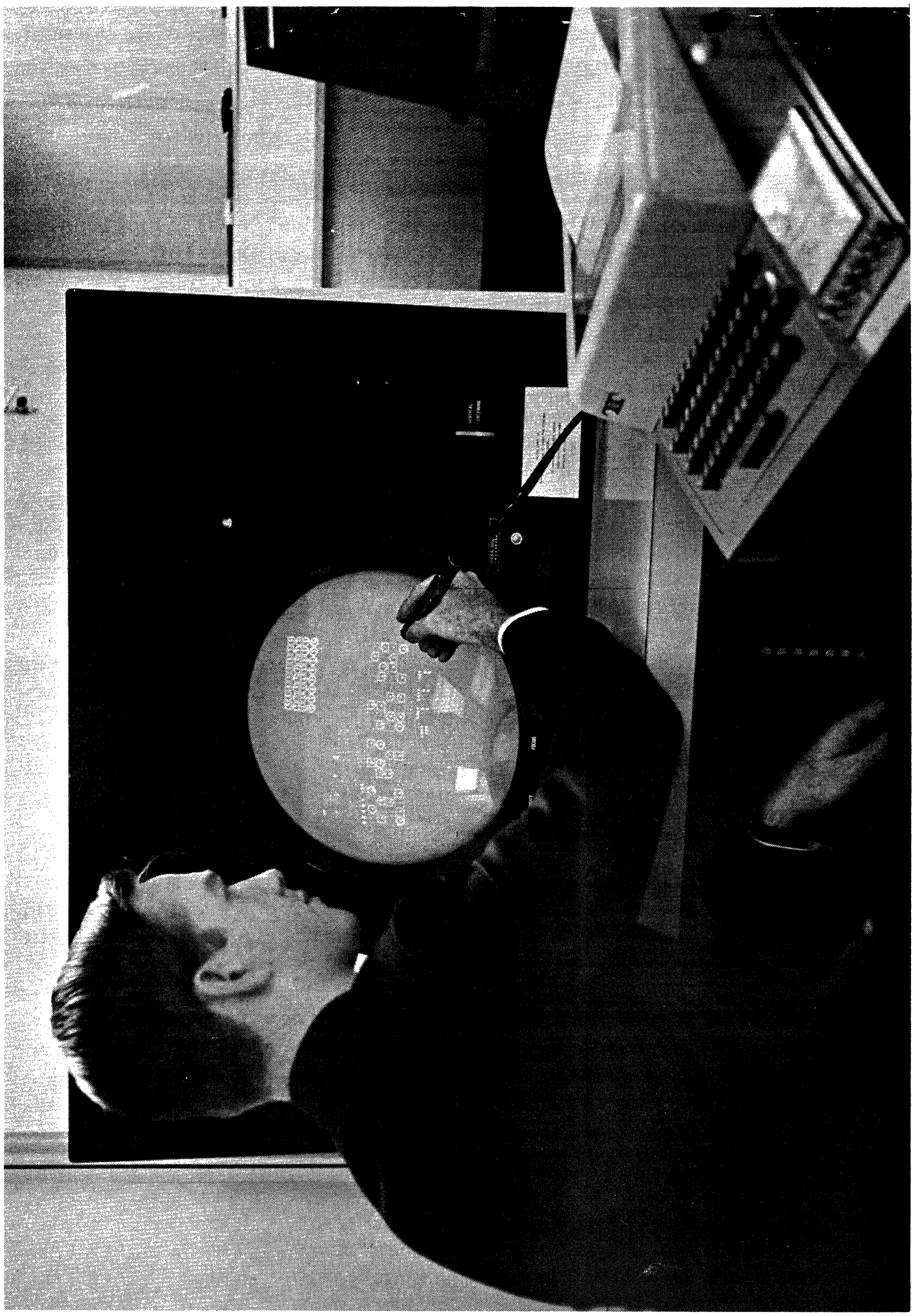


Figure 16. A Player in the Shimoku Environment

3.2 SHIMOKU EXPERIMENTS* (by Sharon Kaufman-Diamond, Aiko M. Hormann, and Carlos Martín Cinto)

3.2.1 Introduction

Purposes of the Study

The study explores a number of aspects of human higher-mental functions in a complex problem-solving task. This is a preliminary, or pilot study. The main intent is to examine the detailed events that occur when a person encounters an unfamiliar task for the performance of which there are many possible options, a task which evolves so that later opportunities are highly dependent on the consequences of earlier actions.

How humans behave in planning and executing many complex tasks and the intelligent functions that are used when people work on such tasks are not well understood. Yet a great deal of effort goes into training people, and, now, into making machines to help people to perform those tasks well. The issues raised in studying work on Shimoku as a model task are some of the same issues raised in discussions of the nature of planning processes. Ordinarily, those issues are raised in terms of entities in the problem description--e.g. military components and their uses or components of an urban building situation. But they are not examined adequately in terms of the ways in which different workers state the same problem, placing very different information processing demands on the planner of problem solver and his auxiliaries. Ordinarily, no examination is made of how the problem-working processes of a planner relate to the planner's statement of "the real nature of the problem" and to the kinds of solutions he finds.

The lack of adequate description of important events in problem-working processes, and of the relation of the events to outcomes, is partly a result of a scientific orientation. Until the recent advent of "information-processing psychology" (Reitman, 1965; Newell, Simon and Shaw, 1958), there was no unifying framework with which to discuss complex, intelligent human functions.

Even now, it is difficult to describe how a whole person, inspired or discouraged, acts in one way or another in his problem-oriented information processing. Information-processing psychology in its early days has not yet dealt much with the way information from many subsystems of the whole person influences how that person will subsequently perform a task--first helping to shape, then reacting to how he senses he is doing.

*This investigation was supported in part by the Institute of Government and Public Affairs and the School of Architecture and Urban Planning, University of California at Los Angeles; and in part by the Del Amo Foundation, in addition to the overall ONR funding.

Our long range purpose is to learn to describe more precisely the events of a human's unfolding problem-working processes in a task that benefits from planning. We want to describe the processes of learning and of changing approaches to a task. We shall try to account for factors impeding learning and change. We will use terms that will suggest links to current work in artificial intelligence as well as in psychology. The description should suggest ways in which machine capabilities could better assist a human's functioning. We want to suggest ways in which broader "ego functions" can be brought into the information-processing framework used for discussing problem solving. We have chosen to try to describe events occurring not only in the good, but in the poor problem workers' processing, in hopes of starting to make a more precise basis for educational efforts to assist both. We hope that the description of the subjects' diverse approaches--and the demands, limitations, and stresses associated with each--will help those who seek, in neighborhood and technical co-planning enterprises, to bring together people who may process the information of a task very differently. In a pilot study one cannot expect to accomplish long range goals. But we hope this is a little step in that direction.

Nature of the Study

In three separate sessions middle-class subjects with post-graduate educations and technical skills and ghetto and barrio youth each played a one-person (solitaire) form of Shimoku with a 30-move or one-hour time limit. They played using an interactive display scope. Each play was followed by a questionnaire and an interview.

The first stage of analysis was undertaken from an "etic" point of view. The term is taken from Pike (1966), who jumps off from the linguistic distinction between phonetic and phonemic descriptions of speech sounds. An etic description is "objective" from the point of view of an analyst who is an outsider and who may not understand the meaning of what he describes to the user. An "emic" description, in contrast, attempts to capture the subjective meaning of an event within the system of the user. In our analysis we first kept to objective descriptions of the step-by-step game records kept by the computer and the slides showing a replica of the game board facing the subject at each move. The primary game classifications and type descriptions were made (almost) entirely on those grounds.[1]

Next, we attempted to infer the nature of some of the mental processes which might have generated the detailed action traces found in the game records.

[1] The one exception is noted in the text. For the first group of subjects it was possible to have the etic analysis carried out by one of us (CMC) who was largely unfamiliar with the subjects and their interview and questionnaire data. For the second group, we used a similar method, but the etic descriptions were done by one of us (SK-D) who had also conducted the interviews.

We compared our inferences from the game records with the subjects' direct statements, in questionnaire and interview material, about their processes. New inferences made from piecing together the interview and questionnaire material and behavior with all the other portions of the analysis were added to the analyses of the game records as we tried to reconstruct something of what the problem-working events had been for each subject from an emic point of view. We sought to know what it was like experimentally for our subjects to work the problems in the ways that they did.

Subjects were recruited from two quite different groups to elicit an array of approaches and levels of ability. We were not so much concerned with detailed comparison of groups to show that they were indeed different. Instead, we sought to characterize kinds of events which occurred in diverse people, who were seen as individuals quite as much as they were seen as members of groups. We emphasized exhaustive, individual case analysis. We sought pattern and structure in unfolding events. We did not merely take statistical averages of selected characteristics of those events--a technique which often destroys the procedural pattern. In our reporting, we move back and forth between individual cases and what appear to be frequent procedural pattern similarities shared by them.[1]

Our subjects were learning to perform an unfamiliar task--yet one which had features in common with important problems in everyday life. They had to develop their own theories for working through a problem with a relatively poorly specified goal statement, no known standards, resource limitations and, initially, a potentially overwhelming number of options. And we had to develop theories of their problem working and their theory building.

The techniques for developing information-processing theories of problem working of individual subjects, or of groups of subjects, in situations as complex and unfamiliar to the subjects as Shimoku, are in the early stages. Studies of chess problem solving by carefully selected subjects who are good and comfortable at thinking aloud (Tikhomirov, 1970a; DeGroot, 1966); of cryptarithmetic puzzle solving (Newell, 1967); of algebra word problem solving (Paige and Simon, 1966); of diagnostic problem solving (Kleinmuntz, 1968); of space allocation in design (Eastman, 1969) and many others have used the think-aloud technique to get at the detail of a subject's thought. They all contrast with ours in three important respects: they were not

---

[1] We often avoid very precise statements when we are generalizing across subjects, because some of the patterns found in individual subjects are not best described in terms of clearly bounded sets or categories, occurring or not occurring in a given subject--rather, the patterns bear family resemblances to each other, in the sense of Wittgenstein (1967).

significantly time limited;[1] if the task was complex, the subjects were generally familiar with similar tasks;[2] and their subjects were preselected for coherence, cooperativeness, and articulateness. The implications of each main difference are worth noting.

By selecting a task where time imposes a reasonably severe economic constraint, and by not preselecting subjects, we lost a chance to use the think-aloud technique. The two subjects who did try to use it were not very intelligible, and could not be prompted without interrupting a timed game. Others declined outright. We had imposed the time conditions (a) because of budgetary limitations on the amount of computer console time we could use, (b) because we were interested in kinds of subjects who are not known for their articulateness and ease in thinking aloud in any case,[3] and (c) because we were very much interested in the aspects of problem working that arise when time constraints and the limitations they impose are "part of the problem." Most studies of problem solving do not stress that part of learning to make a plan for solution which requires that a subject learn to model himself and his resources--in relation to the abstract problem constraints and to time, during the problem working interval under study. Our analysis takes up just that aspect of the problem-working process and its effect on the solutions achieved by our subjects.

By electing to study a task repeated three times by subjects who were initially unfamiliar with it, we sacrificed some of the clarity of the well-practices performance. In return, we were enabled to watch the learning process.

By foregoing preselection only of articulate and competent subjects, we had at times to accept mumbling replies and incoherent records. But we gained access to a wider spectrum of people whose problem-working processes are important to try to understand even if they can be understood only incompletely and inelegantly at first.

---

[1] There often were actual or implicit limits, but they were not set at a level where they operated as severe constraints on the subject.

[2] Kleinmuntz did study diagnosticians who had different levels of training and experience in order to compare their procedures. However, he did not study unfolding learning processes in any given subject over repeated tries on an initially unfamiliar task. Newell's cryptarithmetic task--which may have been itself unfamiliar to the subject--is much less complex than ours, and its defining properties are easier to grasp on brief instructions.

[3] Many designers loathe it, most of the ghetto subjects were not good candidates for it, etc.

We see information-processing psychology as a basic science for education and for planning and design disciplines much as biochemistry is a basic science for medicine. To establish the bridges to those disciplines, however, it should--at least sometimes--undertake to study complex problem working under important time constraints, where learning is occurring, and where subjects come from groups relevant to practical concerns.

A Note on Reading this Study

The main outlines of the Shimoku game should be reasonably familiar to the reader if he is to follow the text. The reader may want to reread the game description in Section 3.1 before reading our instructions to the subject. Then he might turn to the opening board configuration (Figure 8), and try to plan and score a few moves in sequence. Next, a look at the actual game material which appears in the Appendix C might be wise, with the reader trying to trace a few of the subjects' steps both on the records and in the written accounts. The dry, game-type descriptions can be read quickly and returned to later for detail as needed.

3.2.2 Subjects and Methods

This section introduces the experiment. A few details of procedure are postponed for mention where appropriate in the main body of Section 3.2.3.

Recruitment and Description of Subjects

In all recruiting, subjects were told that the purpose of the study was to increase understanding of how people thought about, and acted in, new complex problem situations. To ensure precision in presentation and recording of all actions, and to make the job interesting, the task was presented as a one-person, noncompetitive game played on a computer scope. They were told that a questionnaire and an interview would follow each game, to enable us to learn more about a player's way of thinking during the game, and about how his thinking patterns might relate to other aspects of his life. Results were to remain condifential, and there were no tricks in the procedure. Subjects frequently asked for more detail about our intentions. They were told why the kinds of issues we were examining might be important to education, and so on. However, terms like "strategy" which might suggest a more specific approach to the game were avoided in these explanations. The atmosphere encouraged many subjects to feel that they were sharing in a useful exercise.

Two groups of subjects were selected. Group I was composed of 20 males with technical training typical of corporate research and urban designers and planners. Group II was composed of 15 ghetto and barrio males with no technical training beyond high-school mathematics and shop. None of the subjects had has prior access to the Shimoku game and most were not used to working with our kind of light pen display.

Subjects in Group I were not paid. Subjects in Group II received $2.00 per hour for all instruction, questionnaire-filling, interview, and playing time. The pay was set high enough to compensate poor students who might have lost earnings at about this rate in part-time jobs. But it was not so high as to attract those who had no curiosity or motivation other than money. It was decided not to duplicate money incentives exactly across Groups I and II (by giving all players a fixed amount - or nothing - per hour), since the subjective value to members of the different Groups of any fixed amount would have been very different. Instead, pay was used to generate a basis for somewhat similar subjective incentive conditions in both Groups. We attempted to promote curiosity and motivation to concentrate on the game task itself, under participation conditions that would not cost poor subjects materially. In general, subjects' comments suggest that those goals were acheived reasonably well.

Group I. The members of Group I were occupied in essentially middle-class work and/or study situations--employed by research and computer software companies and/or pursuing university graduate studies. Twelve of the 20 were actively working with computers. Eighteen subjects were between the ages of 23 and 44, one was 17, and one was 58.

Aside from an art institute graduate and the formidable 17 year old who had finished high school, knew five computer languages, and was working as a programmer prior to beginning studies in computer sciences at MIT; all members of Group I had at least one college degree. Eight were completing terminal master's degrees and the other ten had, or were completing work toward their doctorates.

Six of this group were mathematics majors, some had minors in astrophysics, engineering, and operations research. Five were urban design students. The remainder had degrees in business administration (1), political science (1), sociology-social psychology (1), human factors-psychology (1), and journalism (1).

Group I subjects were recruited by invitation. That is, nonintimate colleagues and students who were not in our own classes were randomly offered the opportunity to play Shimoku. Those who agreed were self-selected from a larger group by their interest and the ability to find the time. These subjects knew at the start that they would be asked to play more than one game, but they did not know how many. After three games, enough data were collected to demonstrate that in three games one could develop play reasonably well without becoming tired of the repetitious nature of the play.

Group II. The members of Group II were ten black and five Chicano students between 14 and 19 years of age. They were recruited for a fixed series of three games. Two were recruited from a junior college, 13 were high school students, 11 of them members of a minority program conducted by a local university. Recruitment began at a table in the cafeteria/study lounge at the junior college, after two of us (SK-D and a white female assistant) had been introduced to the Black student leaders by the Assistant Dean. A number of Black students expressed interest and discussed purpose and procedures with us personally. However, only five attended any sessions and three of those played one game and scheduled others repeatedly but never reappeared.

Factors frequently seeming to operate against success with the city college recruits seemed to be racial mistrust, mistrust of experimenters in general, and the risk of looking bad. We shall mention the "incomplete" Black subjects again, but they are not included in Group II.

Two of this group did finish. They constitute the two college freshmen included in Group II. Among the five who started, the two completers appear to have been the strongest, judging from interview and questionnaire data and from first game scores.

Eleven of the high-school students were members of the Upward Bound minority program based at UCLA. The program provides tutoring and campus experience for students in grades 10 through 12 who are judged to have some potential, or spark, yet not to be functioning well enough to get into and to handle college-level work without extra help. The typical member is underachieving, and scores well below the 40th percentile for his grade on the California Test of Basic Skills. Upward Bound members are of particular interest, because educators dealing with them are severely perplexed about how to characterize their difficulties and how to help them.

Upward Bound subjects were recruited at one of their regular Saturday UCLA sessions with a brief announcement about an interesting computer game and voluntary sign up procedure. Recruitment was introduced and supported by the then Director, Mr. Winston Doby, who is himself black and popular with the students. According to Mr. Doby, and to our checks of academic and test data for our subjects in the Upward Bound files, our volunteers are typical of the middle and upper ranges of this (virtually all problem-learner) group.

Given the extraordinary cooperation of the Upward Bound staff, it was possible to engage these subjects well and to finish all games with them. Several have since taken a computer workshop with one of us (SK-D) and have become known to us in greater depth. Occasionally, we shall introduce some of the additional information.

Two Upward Bound subjects were so enthusiastic that each brought a friend to th the sessions. As the two friends were from the same age range, schools, and neighborhoods and were seen roughly as equals by the helpful recruiters, we let them join the game as members of Group II under the same conditions as the Upward Bound subjects.

The socioeconomic status of Group II subjects judged by Upward Bound data and by occupations of parents are from lower-lower to lower-middle class. The eleven Upward Bound students met a poverty requirement to enter the program. Fathers' occupations were mostly truck driver and maintenance worker, extending upward for the non-Upward Bound friends to barber and chiropractor. Mothers' occupations were mostly housewife (all Chicanos, some Blacks) and domestic worker, extending upward to social work assistant and computer programmer, in the cases of Black mothers who had gone back to school in adulthood.

Instruction and Practice

Necessarily, instructions described here will appear more difficult to assimilate than when presented in the context of active manipulations of the game mock-ups and terminals, and with diagrams related to the instructor's explanations and demonstrations. The overall instruction and practice sequence follows.

Two phases of familiarization with the task preceded the start of official play in game 1. The first phase was carried out in a quiet room away from the terminal. The second phase was carried out at the terminal. Phase one usually involved one subject and an instructor. At times it involved two or three subjects and one instructor. Each subject was checked individually for understanding as the instruction proceeded. The practice phase on the terminal always consisted of a single subject with his instructor.

To start, each subject was shown a three-dimensional, lucite model of the 4 x 4 x 4 game board (Figure 14). Here, we give a condensed version of the instructions. They were elaborated much more fully with the subjects, in ways consistent with the game description presented earlier in this report.

> "The purpose of this game is to score as many points as you can, playing for 30 moves or one hour, whichever comes first. You make points by putting together four-in-a-row patterns made up of tokens. You will have a scoring pattern chart (Figure 13) with you when you play, so you need not memorize it." (A diagram showing the game board with the set of tokens are shown in Figure 17.)

> "There are two main kinds of scoring patterns. The low-scoring kind, giving 6 or 8 points, requires only that you pay attention to shape <u>or</u> number. The other, high-scoring kind, giving 12 or 14 points, requires that you have to pay attention to both shapes and numbers." (A subject's attention is thus drawn to differences in information processing requirements for the two types and to differences in rewards associated with them.)

> "The pattern rows must form straight lines. They can go in any direction within the cube." (Possible kinds of four-in-a-row configurations are demonstrated and described, using four blank plastic tokens and the cube model. Each subject practices making straight-line patterns in the model.)

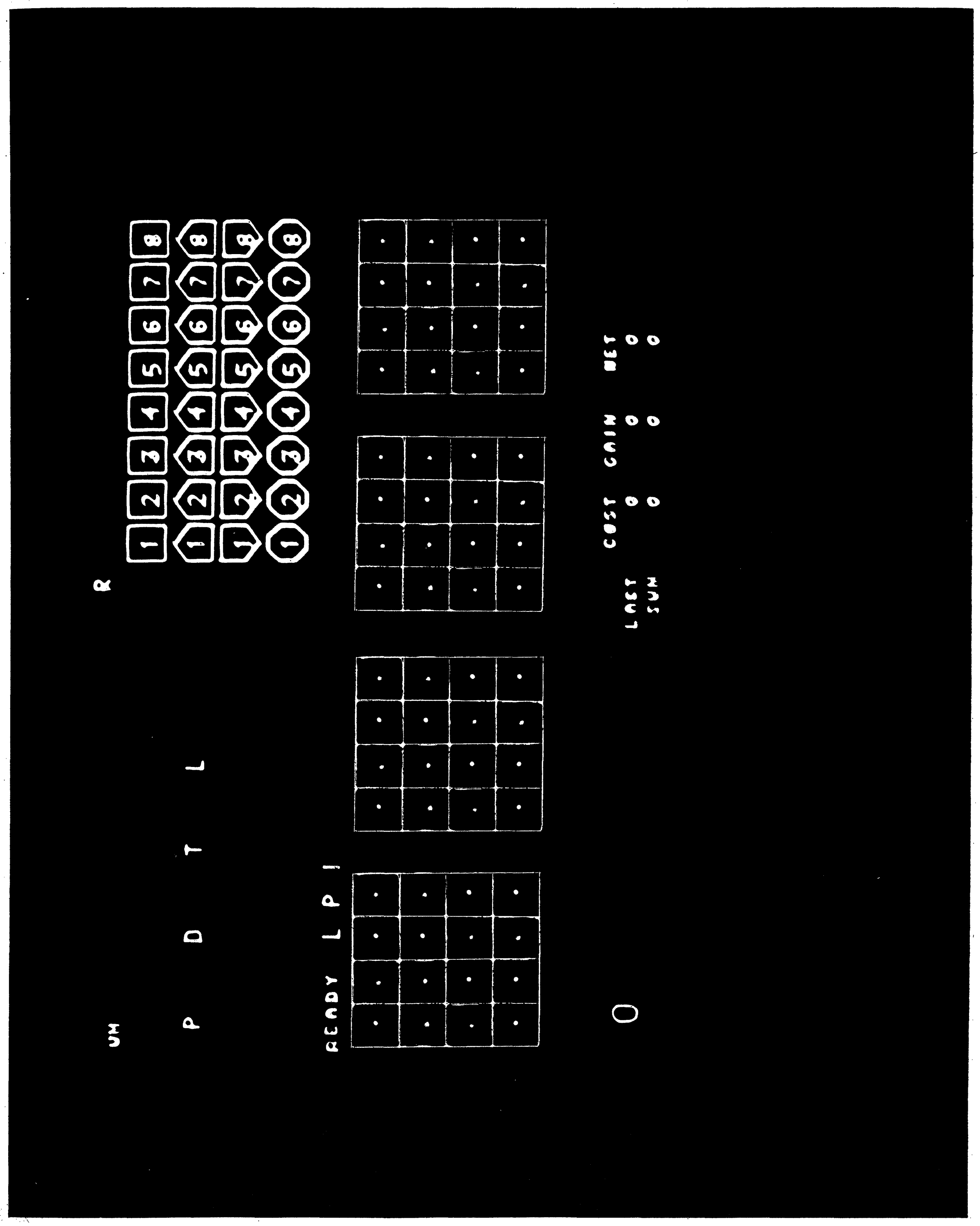


Figure 17. Diagram of the Computer Scope for Shimoku, Used in Instruction

"On the computer we play with a two dimensional scope (diagram of the scope with empty grids, Figure 17, is shown), so we have to show the positions in the cube as if they were on a flat surface. If you were to take each layer (plane) of this cube model and put it on a flat surface, with the removed layers side by side in order, you would get a situation like this,"(four 4 x 4 grids, Figure 15, are shown in a planar array).

"You can see that any four-in-a-row pattern in the cube can be shown on the grids." (Instructor demonstrates. Each subject is led to map each kind of four-in-a-row pattern from the cube into the flat grid array. Each also has active practice in the reverse operation, constructing patterns seen on the grids in the cube.) "So, although you'll probably find it best to think of the game as being played in a cube, you'll be making your patterns with the layers of the cube shown on a flat board."

"How are you going to make patterns? At the start you will have 32 pieces--a complete set of numbers one through eight of each shape--scattered about on the grids. And you'll have another complete set in the reservoir. (Both are demonstrated in diagrams.) "To make points you will have to move the tokens, and making moves will cost points." (The legal moves, including moves in look-ahead mode, their costs, and the computer manipulation options are explained, using Figure 7. The moves and costs chart is available during play. Subject is reassured that he need not memorize those details.)

"Your final score will be the sum of points gained for all scoring patterns complete on the board at the end of the game, minus the sum of points it has cost you to make moves and to use the look-ahead mode. Remember, if you break a scoring pattern, you lose the points gained in making it."

Individual subjects took between 15 and 45 minutes to go through the main instruction phase to a point where each had satisfactorily demonstrated understanding of each feature and rule after it had been explained. Later, we shall treat what went on during instruction in relation to results.

As part of the on-line computer phase of instruction, a subject was introduced to the basic arrangement of the scope with a game array whose grids, being empty of tokens, were clearly seen. After practice with it, he was introduced to a board with the standard practice configuration of tokens scattered on the grids in such a way that almost completed patterns were readily visible. A subject thereby gained experience in making moves to complete patterns, checking the scoring tables and graph, unmoving, using look ahead, and so on. He was prepared for the fact that the actual game would be played with 32 tokens already in place at game's start on the four grids, and a duplicate set of 32 in the reservoir. For a second time, he was remined of the critical

rule about maintaining patterns intact to the end to keep points gained. In this on-line phase he had opportunity to master the light-pen and to review, with demonstrations and active practice, all legal moves, computer options, and scoring patterns.

The subject was allowed to practice until he could perform the most difficult maneuver the instructor might request and felt he understood the manipulations--but not until he felt he had developed an integrated view of how everything would fit together in competent strategic play. The process of developing that view, starting from understanding of the board, tokens, rules, and possible manipulations was what we wanted to study in the recorded games. The traces of that process would be lost if it were to go on in practice. Hence, on-line instruction and practice time was restricted to 15 minutes. Subjects when stopped typically said, "I've got the rules and stuff okay, but I don't yet see how it's all going to hang together when I go to play."

At completion of practice the standard game was brought on and timing began. Subjects played with the scoring pattern chart, legal moves diagram, and a four-grid blank "scratch" paper and pencil at the console. The scratch paper was given with the explanation that it was "just in case you might want to use it for something." Scratch papers were collected and filed as data, as were any notes on other paper which subjects made or brought to any of the sessions.

Throughout the instruction period and during the actual games, subjects were encouraged to ask any questions about legal moves and technical features. Thus, any subject who thought he had forgotten something was free to refresh his memory using his diagrams or questions. However, subjects were given no substantive answers to questions about ranges of points they might expect, characteristics of previous good and bad games, or best ways to proceed. They were asked not to communicate with each other between games. (The few known minor leaks have been taken into account.)

Game Play Conditions

Subjects played individually in the SDC Q-32 terminal rooms. This setting did not afford complete privacy. Despite efforts to shield the subjects from outsiders and from each other, there was occasional distraction. Since subjects ordinarily started at different times and had well separated terminals, it was difficult for subjects playing simultaneously to get any accurate sense of how their own play compared with that of others. Merely having others present led to fantasied comparisons for some. We suspect that graph shape (e.g. a discrimination of smooth versus jagged) may have been vaguely perceptible to subjects occasionally leaving their scopes and passing other scopes. While less than ideal, conditions were not so bad as to call the over-all results substantially into question.

"Think Aloud" Option, Group I

Each subject in Group I was offered an opportunity to "think aloud" into a tape recorder while playing his game. Only two subjects took the option, since most felt that it would hold them back in a time limited game. The option was dropped for Group II. We do not present think-aloud data here.

Three-Game Procedure

Each subject played a full sequence of three game sessions, using the standard opening game. Each game session was followed by a questionnaire and an interview. The interviews were taped and were semistructured, jumping off from the questionnaires. Taped interviews lasted between twenty and ninety minutes, typically running about forty-five minutes, for the first session, and thirty to forty-five minutes thereafter. Intergame intervals ranged from five days to about three weeks, being in the range of five to ten days in most cases. Reviews and practice games were offered each subject before games 2 and 3. They were needed more by Group II members, although there was quite a range of expressed needs in Group I.

The Standard Opening Game

The array of tokens shown on the four grids in Figure 8, taken together with the reservoir of 32 tokens, numbered one through eight for each shape, describes the pieces and positions of the Standard Opening Game. The tokens on the grid are carefully placed in this configuration so that there are no scoring patterns, and no obvious, almost completed patterns. The configuration is sufficiently unstructured to make it hard to recognize or to remember. We have evidence that distributions of tokens on the board which are mathematically equivalent with respect to potential pattern completion are not easily shown to be, and are unlikely to be, psychologically equivalent. Thus, to measure progress in a subject's ability to play the game, it seemed preferable to start him each time with a configuration we knew to be the same, but which he expected would be different, and one which he not only could not have planned for, but probably would not recognize. The alternative would have been to start him each time with a different configuration whose psychological equivalent to the original would be difficult to assess. We shall comment on this later.

Data Collection from Game Play, Questionnaire, and Interview

Game Play. A research assistant sat in the same room as subjects during play, noting anything remarkable about subjects' behavior, computer down time, slow response, and so on. Final game boards were diagrammed, and complete tabulated records of all actions in the games were printed out. Sample diagram and print-out materials are found in Appendix C. Later, the games were replayed so that the boards could be photographed in 35mm slides. This was to permit sequential reconstruction of the visual displays presented to each subject as he worked his way through the game.

Because there was risk early in this work that the computer on which Shimoku was programmed might go out of service very soon, we moved into running subjects before some final details of programming were complete. A result of this is seen in the absence of an internal computer time clock marking the actions in our first few games of Group I subjects. All subsequent games have time, rounded to tenths of a minute, recorded for each action.

Questionnaire. In this project we try to put together behavior in the described task with print-out copy and the results of evocative questionnaires and interviews following each episode of play. The questionnaires, included in Appendix C, are in places vague and/or redundant by design. We hoped to elicit reactions perhaps not usually expressed in words, or formulated at all, by subjects unaccustomed to think about how they think. The questionnaires were to provide clues to the interviewer, who used the answers to guide post-questionnaire sessions.

Interviews. The interviewer (SK-D) is a psychiatrist with experience in interviewing on the subject of problem representation and thought processes. The partially structured, cognitively oriented interview uses ideas in part from the work of Piaget and from psychoanalytic technique.

This method (compared with the "think-aloud" and on-line, mid-task questioning techniques, which are virtually precluded by a time-limited game format) has the disadvantage of catching the subject after completion of the task. Some of what we hear about game-time thought may be wholly constructed post hoc in the interview; much detail necessarily is lost, while some may be accurate in kind but refer to times substantially out of order. However, reference to the print-outs and game diagrams or slides can help us to evaluate interview material with respect to details of play. In contrast to on-line methods, the interview permits probing, floowing up leads, encouraging the subject to dredge up vague ideas at the back of his mind that may have been influencing play--without disturbing the flow of the task. While it does not replace on-line methods, the interview is valuable to us not primarily for retrieval of details but for access to attitudes and motivations, and for gaining a sense of how the problem-working experience is seen by, and influences the self-image of the person. Those data often are not gathered, or are not reported in studies of complex problem solving. In addition, because the first and second interviews are sandwiched between successive plays, there is opportunity to start to look at the influence of reviewing one's experience on the development of play.

During the post-game interview, the interviewer ordinarily had at hand the diagram of the player's final game board, his scratch paper, the print-out and the post-game questionnaire. The interviewer used grossly evident game features from the documents in helping to guide questions in the interview. Since memories would fade and tempers heat, we did not keep subjects waiting while more detailed analysis might be done prior to the interviews. At times

the interviewer made his diagrammed final game board available to a player (usually in Group II) who was having particular struggles trying to say something about his game.

In the first two interviews and during all but the last part of the third, it was necessary to be circumspect, to make sure that our questions did not, in themselves, suggest that there should be different content or approaches than those the subjects were using. At the very end of the last interview, we sometimes allowed more specific comments in order to explore subjects' reactions more fully, as in the case of HS,cited in the text. We attempted to learn from subjects whether they could be aware of any ideas they thought we had given them, or an ways they thought we thought someone should play. In general, subjects were encouraged to think about the game all they wanted, at any time. But they were not permitted during the interview or outside to study or copy the opening token configuration.

Rather than attempt to be formal and rigidly standard, we decided to use the experimenter to as good effect as possible: to create a friendly, relaxed, speculative atmosphere in which anything a subject might venture to say would be accepted as a matter of some interest, and treated benignly and confidentially. Subjects were treated as collaborators in a joint exploration of thought process in the game. The interviews were taped, and subjects gave permission for the taping. Tapes were transcribed. Occasionally, when equipment failed, shorthand notes were taken by the interviewer. At the end of the third interview, subjects were encouraged to ask questions about the research. These questions were answered straightforwardly, with a request not to speak to others who might still be due to play. Cooperation in general was very good.

Conjoint Discussion Session. One conjoint, extra session was held several weeks after the last game play. Present were four subjects from Group I who knew each other as UCLA graduate students. The purpose was to explore whether, following a new task like Shimoku--having comparable, detailed traces of individual problem working activity--group discussion of heuristics, feelings, and the organization of thinking processes would enhance the educational value of the individual events. At the session, subjects were given copies of their final game diagrams and of the print-outs and questionnaires for each of their own games to refresh their memories. The interviewer guided the discovery process when necessary. The session was taped and reviewed and is reported later.

Data Analysis

To limit the tendencies to read into analyses of the hard data any biases we might have from knowing subjects' attitudes and approaches to the game, we divided forces for the preliminary analysis. One of us (SK-D) was familiar with the subjects themselves from instructing and interviewing, another (CMC)

attempted to take the print-out and the game slide sets and final boards for examination without knowing content of instruction, interview, and questionnaire material. Separately, one (SK-D) worked with interview and questionnaire material and behavior notes in light of data in the print-outs and final board diagrams. Because of vagueness in language, it is impossible to analyze the interviews without some access to the hard data. But the reverse is possible.

Because this was a pilot study in which we were trying to explore the merits of different types and intensities of analysis, we chose to make certain measures of the games of each player, more intensive measures of the games in Group I (which contained most of the main types of games encountered), an impressionistic case history of each player, and intensive case analyses of certain interesting or typical players in both groups. Including some interesting games of Group I-type visitors and Group II-type non-completers, we have analyzed about 110 sets of game print-outs and diagrams, questionnaires, and interview transcripts.

3.2.3 Results and Discussion

This section presents our findings. We did not set out to test any hypothesis, but rather to find phenomena, to see what was of interest in them, and to learn to describe and, at least partly, to account for relationships among them.

Overview of Presentation

The report of results starts with the classification of the raw objective data of the game records. The games were grouped by attributes. The collections of attributes characteristic of each group were accounted for by postulating three main types of procedures by which the records could have been generated.

Three main game types are reported. The first, or Incremental (INC) type, accounted for most games in both Groups I and II. The other two were Master Planned (MP) and Adaptive Master Planned (AMP) games. The classification, the type descriptions, and the subtypes and occurrences of all games are described in the next section.

Next a heuristic distinction between Arrived and Learning stage games is added. The Arrived game notion permits us to separate for analysis the requirements for actually executing a game on-line from the requirements for playing while learning to develop the formulations which will guide one's subsequent play.

The rules of the game state the goal as "make as many points as you can." Therefore, every game played constituted a solution to the problem posed by Shimoku. If the view taken of the problem seemed to the subject to warrant an incremental approach, then the consequences of that view endowed the problem with particular problem characteristics. These characteristics, in turn, prompted particular ways of coping.

Later, we explore some of the problem characteristics which followed from three different ways of formulating the Shimoku problem. A fictional creature is introduced. The Human-like Problem Worker (HPW) resembles a human in every way, except that we can be sure he[1] has perfect, active, and accessible memory for all the rules of the game and for all the statements which are given him in any instruction set prescribing play, according to one of the three main game types. In an imagined experiment, one HPW is assigned a set of instructions prescribing the highest level <u>INC</u> game, another gets a set prescribing a classic <u>MP</u> game, and one gets a set prescribing a classic <u>AMP</u> game. In each case the HPW must try to interpret and to execute the instruction set prescribing the game of the assigned type and to do nothing else. He does not have to formulate his approach to the game. He does not have to model what he is doing for purposes of learning, after the game, to improve or to change his play. He has only to execute a game of a given type. The composite prescriptions are those we have assembled from the interview, questionnaire, and game record materials of our subjects who played the better Arrived stage games of each type.

Analysis is then carried out of the requirements for interpreting and executing a game of each type, as those requirements that would appear to a human processor. There is no attempt to present procedural and policy prescriptions in the form of programs which could be carried out by a machine. There is no attempt to prove the completeness of the prescriptive, game-type instruction sets. Instead, there is an attempt to stick as closely as possible to an only slightly edited version of what we could infer by working close to their game records. We believe that the prescriptive formulations of the Arrived stage, main game types are fair approximations of what good subjects told themselves while executing their Arrived stage games. Therefore, the analysis can give a sense of the kinds of differences in information processing demands that the job of interpreting and executing the main types of Arrived games placed on subjects working on-line at the scope.

After presenting the prescriptive instruction sets and analyzing the objective information-processing requirements placed on an on-line, interpreting and executing player of each main game type, we ask, what are the subjective experiences of such players? How do feelings and reactions of subjects' playing

---

[1]HPWs are enough like people to rate personal pronouns.

in each type of high-level Arrived stage fit with the objective requirements placed upon them by their problem formulations? There are marked differences in how it feels to be a player in each situation, and those subjective, experiential differences feed back upon the kind of information processing the subject continues to do as he plays.

The immediate processing required merely to interpret and to execute the prescriptions for those Arrived stage games--the INC and the AMP--which have strong adaptive conditions on play is found to be enough potentially to overload the capacity of an intelligent human player. Having to deal with a potential information-processing overload in forming an approach to problem solution was an important determinant of our subjects' playing efforts and subjective experiences.

Next, we carry along notions of the already formulated Arrived stage games, but we go on to ask what is involved in learning from scratch to play in one way or another. We consider a representation of both the static structure (or program scheme) and the control program (or execution) aspects of each main game type.

We bring in Amarel's notions concerning the contrasting requirements for forming programs to find solutions to problems of derivation (D) and formation (F) types. The three main Shimoku game types are demonstrated to have different representations of the problem goal statement. Attaining those goal representations, starting from the game description given to a subject, is conceptually more difficult in MP and in AMP, than in INC games. The version of the Shimoku goal statement which--like that in a D problem--could be expressed entirely in the language of the "givens" of the problem, was consistent with program formation in terms of an incremental strategy. Generating versions of the Shimoku goal statements that were characteristic of the MP game and the AMP game required the introduction of language not contained in the "givens" of the problem. MP and AMP statements made Shimoku treatable as F and quasi-F problems, respectively. The processes whereby subjects arrived at the novel representations of Shimoku, and the consequences of the new representations in altering the information processing requirements placed on them during play are presented.

The perspective on issues of representation and learning then broadens. We use the notion of models, in Minsky's sense, to look at inferred events in our subjects. Players are seen as using programlike models they already have--of themselves as problem workers, of their heuristics and other problem-working knowledge, and of their broader experiential knowledge--in piecing together an approach to the game.

A subject is seen as needing to use the programlike capabilities in his existing models to take in, and to consolidate into a new model, the separate statements given him of the rules for playing Shimoku. He must make an initial model of the game-as-given, and, subsequently, can elaborate, refine, and perhaps then markedly alter it, passing through a succession of game models. Since what he can build into his successive models of the game will be products of his own thinking and exploring, he can be thought of a gradually building a self-task interaction model which will contain much of the newly generated information about what he has done and can hope to do with the task.

The broader view of subjects' representational activities in terms of modeling is then applied to events surrounding the transition (or failure to make a transition) from INC to MP play. We also consider examples of the uses and misuses of models by subjects in analogic thinking, and their relation to learning to work a new problem. We note the information processing requirements for a subject who undertakes to model his own problem working while performing actions in the game, in order to perform later analyses which can lead to new learning. As would be expected from earlier parts of this account, modeling the relatively hard-to-represent INC games as well as playing them increases an already large load and is difficult to do effectively. Failure to achieve a new goal state representation is shown to keep a player in an incremental situation--one which at once tends to overload his current execution, and to resist effective modeling for new learning. For any subjects who are less than very bright and very competent, the INC situation resembles living on welfare--scrambling to meet current needs, but accumulating little capital which could be used to improve the state of affairs.

The locus, or distribution, of elements of a representation in a system consisting of man and machine working a problem is considered in relation to the man's modeling to learn. A subject who leaves nearly all the work of representing game events to the machine--as some were tempted to do--and who reacts to the board piecemeal, but does not encode events for his own storage during play, is likely to have done so little descriptive modeling that reworking his experience off-line will be difficult.

In the first parts of this account, structure and language are established primarily in discussing regular features of the better games. That framework next forms a background against which to consider some of the less clearly structured events that occurred in weaker games. Later, we present more of the characteristics of weak games--in a way that though not yet very satisfying to us, does capture some of what is essential about the events. We have the sense that the diverse features of poorer games may result from the recombined operations of a limited number of major sorts of troubles, but we cannot demonstrate it adequately here. We note a single interesting question asked subjects that identified a subgroup who were possibly more serious thinkers than the average of their peers, and who were conscious of having difficulties with and seemed potentially receptive to teaching about representation.

Next we demonstrate some of the problems characteristic of Group II performance, as they occurred in the efforts of a very bright and highly motivated player who tried to conceive an over-all structural plan, but whose defective procedures hindered him at every step of the way.

The remaining parts of the Results and Discussion Section introduce more case history material, discuss motivation in the experimental situation, and describe results of the conjoint session.

Main Game Types and Stages

In this section we give descriptions and bases for classification of our subjects' games.

Incremental Games: Type INC

One way to think of Shimoku is that in any given game scoring patterns (sc's) are made by completing linear paths. Four-in-a-row configurations that gain points are made by rearranging and/or adding tokens to cells joined by linear paths in the cube. Thus, incomplete or potential sc's are arrays of empty and/or filled cells. We can call them sc-1 or sc-2, etc., to indicate that one or more tokens are missing (or correct ones are needed).[1] Scoring points requires transforming sc-1's and sc-2's to sc's. All Shimoku games do this. How they do it varies.

If an analyst inspects Shimoku game records, intending to reconstruct key aspects of the processes guiding their production, he is struck by major differences in the local move sequences and in the global patterns of moves. How are incomplete sc's located? How are they generated? How are individual tokens in given sc's made to serve multiple purposes? What kinds of higher-order structures are built that permit more moves to function effectively as multipurpose moves? What are the relations of the foregoing to point costs, to time, to information handling, and over-all resource management based on the requirements of the whole game? Our main typology, based on groupings made from descriptions of game records, captures major systematic differences in the ways those questions can be answered.

Of the three main game types--the Incremental (INC), the Master Planned (MP), and the Adaptive Master Planned (AMP)--the INC includes far the largest number of games played. Figure 18 shows the distribution all game types by game number and group. INC games are the most heterogeneous. They are defined partly in terms of their own characteristics and partly as a residual class left by the definition of the other two main types.

---

[1] We shall see that a player need not always be conscious of an array or an sc-1 or sc-2 in order to make points by completing it. The player's internal state in relation to the patterns is not being treated in this section.

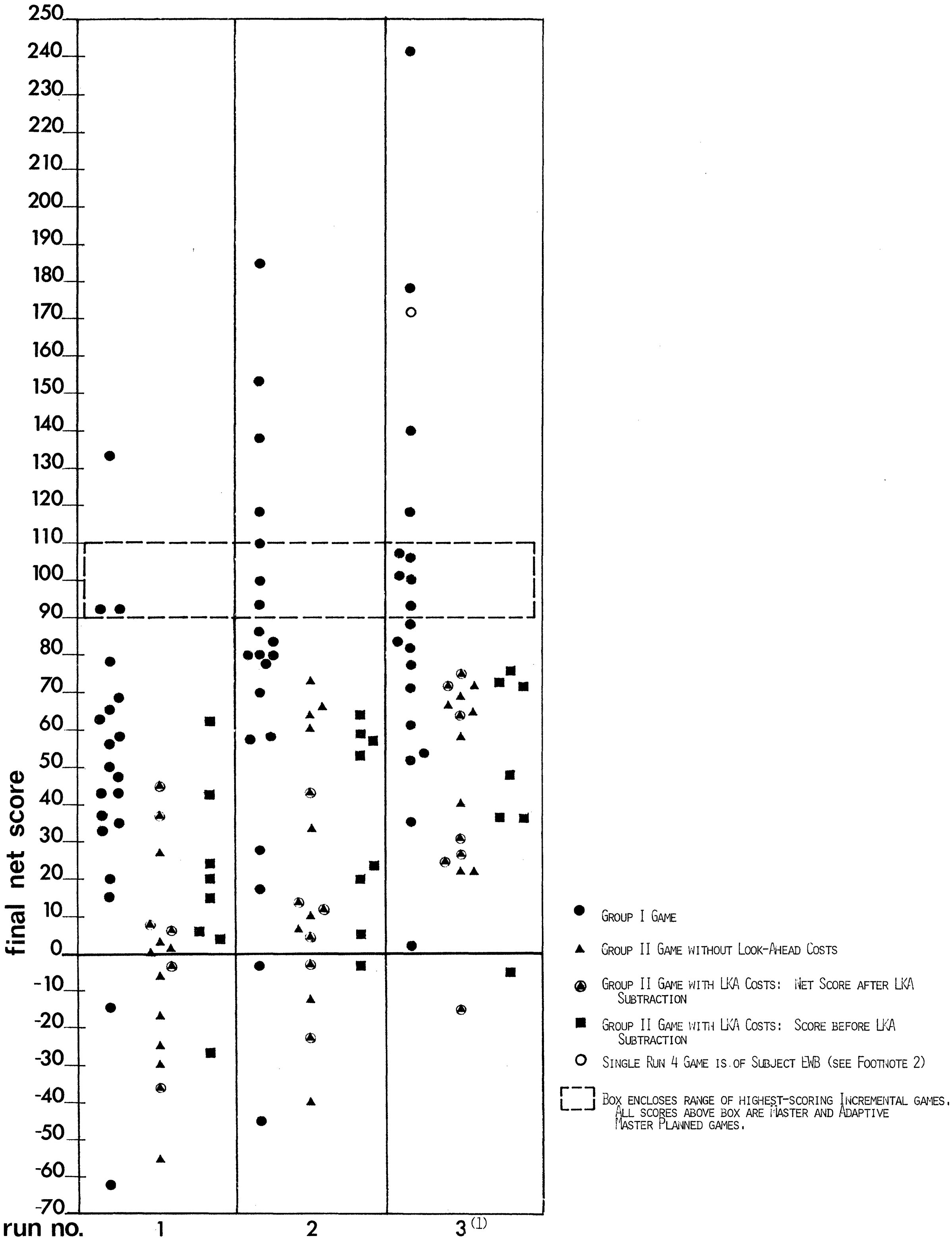


Figure 18. Shimoku Scores, by Game Runs, for Groups I and II

1. Only 19 scores are plotted for Group I on run 3 because time-shared system failure wiped out the game before it could be recorded.

2. Single run 4 game is of Subject EWB, who requested replay of his preplanned but erroneously executed master game 3, an MP-ℓ game which scored 54.

Let us start with the easier part of the definition. We anticipate later sections by stating that MP and AMP games show highly organized final boards. The boards have holistic, readily seen, and simple main designs, made up of interrelated scoring patterns. The printout of an MP or AMP game shows most of the full complement of moves to have been made in a fashion requiring a high degree of coordination. Shape, number, position, and other attributes of tokens arranged in early moves have been so prepared that they coordinate with attributes of tokens arranged in late moves, which may occur 15, 20, or more moves further on. The early moves coordinate well with attributes of the whole structure that may not emerge visibly on the board until well on in the game. Given those characteristics, it is hard to imagine a guidance process capable of generating MP or AMP games regularly[1] without a substantially articulated overall plan. A description which might serve as basis for such a plan can be read from the final diagram of an MP or AMP game.

Now, let us compare and develop the rest of the definition of the INC game type. Inspection of the coordinations of most sc's with each other and with the totality of pieces on an INC game's B_f (final board) shows no coherent, readily perceptible, main design. At times, scoring pattern interactions are quite well coordinated in patches. But other potential sc's are blocked, or incoherent and unproductive of scores at the margins of the patches, and in other regions of B_f.

In the INC game there is no coherent main design involving a majority of the sc's in a large structure. There may be many sc's. They may contain multiple-use tokens. B_f may even contain an entirely filled, single, visible grid, corresponding to a horizontal plane of the cube. But the relations among the moves in sequence and among the tokens in the resulting four-in-a-row patterns do not suggest an articulated whole-game conception that might have been used by a guidance system to generate the moves of the game. The first part of our definition, then: when evidence cannot be found from the game records for a coordinated, holistic use of more than two-thirds of the given 30-move complement, the game is, residually, classed INC.

But what are the positive characteristics of INC games? Both master (MP and AMP) play and INC play necessarily involve locating opportunities for sc's on the board. In the INC game, the current state of the board, as searched by the player, tends to determine which sc's will be completed far more than it does in the other types. In a good INC game, the moves take advantage of <u>current</u> opportunities as they are found. They react to the local possibilities of a situation. Their completion creates new situations, which are in turn reacted to primarily in terms of local relationships. Structures coordinating

---

[1] It is possible to imagine a random guidance system achieving such order, but hard to imagine that order occurring regularly or frequently. Game records show that subjects who achieved MP or AMP games before the last round invariably repeated MP- or AMP-type games.

several sc's appear in many INC games as nuggets of organization floating in the larger cube. They have been built incrementally. At best, each new sc has been coordinated with others by use of tokens which happen to be left (or which have been conserved in a fashion which can be expressed by a general policy),[1] and are available for use in the local structures found on the board at each time (B_t). We do not say that records of INC games give no evidence of attempts to plan ahead. On the contrary, there is frequently evidence--in uses of paths(P), look ahead (LKA), and unmove (UM), and in ways token structures are built--of struggles to handle some chunk of the future. There is a wide range of attempts and successes. But even the successes seem never to have coordinated the requirements of the early and late moves in the manner of MP and AMP games.

Descriptions of the game records support the notion that a hypothetical guidance system generating the INC game could have done so without any overall, articulated, and coordinated pre-existing program. Generally, to produce our subjects' INC games, a guidance system would have to take into account the current state of B_t. It would have a small collection of game-related rules of thumb (but not necessarily a good system for coordinating their application). It would work with no more than a few (usually four or fewer) possible moves beyond the current one.

The problem-working processes of subjects generating INC games produce printout process traces and final score results which are substantially different from those of MP and AMP games. Table I shows the distribution of scores for all players in Groups I and II. With the exceptions classed as lapsed or developing MP games, noted below, scores attained in the MP and AMP games began at 118 points, climbing as high as 245. But no INC game score exceeded 110. High, over-all coordination raises the quality and number of score-producing uses gained by single moves of individual tokens. This is consistent with the Shimoku task environment's structure. Thus, differences in problem-working practices associated with differences in degree of overall coordination of moves have concomitants in score-range differences, separating the main types.

All but twelve of the games played were INC games. Their scores ranged from -63 to 110! Not only scores, but printout process traces and final boards show the differences among them. The differences do not fall so neatly as those distinguishing INC from AMP and MP games. Nor are the distinctions to be made all of the same kind logically. But subtyping them as best we can will help us characterize subjects' problem-working activities.

The first three subtypes all represent relatively coherent incremental games. They are separated by score ranges that coincide with trends in the game records.

---

[1] For an example of such a policy, see 3(u), on page

Table I. Distribution of Shimoku Game Types and Scores

GROUP I										GROUP II									
RUN NO:	1			2			3			RUN NO:	1			2			3		
Subject	S	M	T	S	M	T	S	M	T	Subject	S	M	T	S	M	T	S	M	T
KN	134	-	AMP-d	154	-	AMP-d	241	-	AMP-c										
NQ	36	-*	INC-ℓ,c	185	-	MP-c	179	29*	MP-c										
EWB	43	17	INC-ℓ,ce	58	-*	INC-me	54	29*	MP-ℓ										
							171	-*	MP-c†										
TT	56	-*	INC-me	80	25	INC-m	140	-*	MP-d										
KC	79	19	INC-m	139	-*	MP-c	100	-*	MP-ℓ										
EH	91	-	INC-hpp	119	-	MP-c	118	-*	MP-c										
Below this line all games are Incremental: small letters further identify INC subtypes																			
HS	91	21	-h	110	-	-h	106	-*	-h	VN	37/42	29*	-e	43/53	20	-e	75/76	25	-me
RV	68	-	-me	100	-	-h	107	-	-h	EH	26	-	-e	73	23	-me	72	27	-m
SN	35	-*	-ℓ,c	79	-	-m	101	-	-h	TS	8/20	18	-de	12/59	-*	-ppe	72/73	-*	-mpp
CI	63	-	-m	94	-	-h	77	20	-m	EI	-3/15	13	-de	60	24	-m	64/72	20	-m
NT	65	23	-me	80	26	-m	93	-	-h	SN	2	-	-de	63	19	-me	69	-	-me
IC	15	-	-d	83	-	-m	88	-	-m	LY	1	-*	-d	66	-*	-mpp	67	-*	-mpp
II	58	20	-m	86	-	-m	83	-	-m	VR	-56	-*	-d	33	-*	-ppe	65	25*	-mpp
TN	33	13	-e	59	26	-me	82	-	-m	SO	-6/6	-*	-de	-12/23	-*	-de	58	-*	-m
CC	50	-	-ℓ,c	80	-	-m	††			DY	-17	-*	-d	4/5	-	-de	40	-	-ℓ,c
SE	48	-*	-ℓ,c	71	-	-m	71	-*	-m	TK	4	-*	-de	10	-*	-d	31/37	-*	-ℓ,c
HLF	-14	-	-d	17	-	-ℓ,c	61	-*	-mpp	SS	22	22	-de	30†††	-*	-de	30	-*	-de
SC	20	11	-ℓ,c	28	19	-ℓ,ce	52	13	-ℓ,ce	TD	-25	26	-d	-40	-	-d	25/37	29*	-d
NN	-3	-*	-d	43	-*	-ℓ,c	35	-*	-ℓ,c	KD	-30	-*	-d	7	21*	-d	22	29*	-de
YW	-63	-*	-d	-45	-*	-d	2	-*	-d	DE	45/52	-*	-re	-22/-3	20	-de	-15/-5	-*	-e

* = All moves were completed in < 50 minutes.
M = Accepted Moves. A dash in M column means 30 moves.
S = Final Net Score. In Group II scores above slash are taken after subtraction of LKA costs; score below slash are before it.
T = Game Type.

† An exceptional game of EWB, played just after 3, at EWB's request.

†† Game data completely lost due to computer malfunction.

††† Subject forgot to "accept" last 15 moves in LKA but had meant them to count.

Incremental-High Games. Incremental-high (INC-h) games represent the top of the score range (91-110 points) for the INC type. The bottom of the range is somewhat arbitrary. In practice it appears to have been substantially more difficult to score over 90 than in the range just below it.

There are 9 INC-h games--less than 10% of all INC--played by 6 Group I players. Five of the players are competent and creative Ph.D.-level mathematicians, the other is the most engineering- and operations research-oriented of our designers. One of the mathematicians played two games in this range and one played all three. Our description of the INC-h type will be modeled most closely on their performance.

Examination of the records of INC-h games shows a relatively smooth, net point-accumulation curve, the graph of the game. Typically, its positive slope is greater near game's start and gradually declines toward game's end. (It is easier to find opportunities to make a few high-sc's (H-sc's) at first by making use of the initial board configuration (B_i) and by using tokens more cheaply moved from other grid positions.) Later, if they are on the board, tokens of particular shape and number needed to complete new sc's are likely to be tied up in existing sc's. Their duplicates, if still available in the reservoir set, must be purchased at a cost of 5 points each. Many paths through a given point, or cell, will be blocked in the late game by tokens tied into sc's running in other directions. Their presence may be incompatible with the developing new sc opportunities using the cell of current interest. And many other potential scoring paths passing through the index cell will have been emptied, requiring too many purchases to produce sc's profitably. Even in the very best of our subjects' INC games, opportunities in late situations have been compromised by earlier moves. However, in those games there has been little waste or error, in terms of board conditions at the time the earlier moves were made. Many of the best sc-2 and sc-1 situations of successive B_t's have been used. Opportunities arising from the results of previous moves are ordinarily built upon. There is evidence that clumps of interrelated sc's are developed piecemeal, taking advantage of local opportunities for enrichment of the structure by getting given tokens to work at once in 2, 3, and occasionally even 4 sc's. In INC-h games, where low-scoring sc's (L-sc's) are made, it is generally the result of seizing easy opportunities (by-products of H-sc making moves) at low cost in time and moves. It is rare that a printout shows much time spent just before a move sequence which produces only L-sc's. (Those occasions are found near the ends of games when structures and costs make things extremely tight.)

Especially for an incrementally guided game, the time and move limits throw into relief the trade-offs necessary between gains to be had from devoting time to on-line analysis and planning and gains to be had from using the full complement of 30 moves somehow in the full hour. For all subjects using the full hour, scores give basis for comparison of productivity per unit time.

For those doing the same number of moves--30--in an hour, scores also allow comparison of productivity per move. To get a better picture of games played more slowly, using, say, only 20 moves in an hour, it is useful to take another measure in addition to net score. Figuring productivity per move, where

$$p_m = \frac{\text{net score}}{\text{completed moves}},$$

we shall compare games to see how the analysis time/move complement trade-off operated. Let us take the first game of player I/HS,[1] the mathematician who is the only one to have played all three games INC-h. In game 1, he scored 91 in an hour with only 20 completed moves, for a p_m of 4.55. His second and highest scoring game, the top of the INC-h range, achieved 110 points in 30 moves (using 55 minutes), for a p_m of 3.67. HS's productivity per move has declined with his increase in pace. He shows only a small gain in score despite 50% more moves and increased familiarity. For a good analyst, analysis time and move quality trade off severely, given the way the Shimoku task structure constrains late opportunities in the strictly incremental game.[2]

INC-h printouts show substantial mastery of the mechanics of the game. There are few mechanical errors. The computer appears to be used systematically as a check on whether moving a token on the board breaks an existing sc. Tokens are frequently moved into new positions where they would be well used. But, following a given, promising move, the table and graph may immediately show a marked drop. (Recall that points are subtracted if moving a token breaks an existing sc.) In the INC-h game unmove (UM) is used immediately. A purchase of the duplicate reservoir token, a move of an alternative token, or a period of inaction followed by a spurt of different moves follows. The point-loss-remedied-by-unmove, a sequence originally encountered by subjects during accidental breaks, has gained another adaptive use. This "check" use of the graph and table display and of UM to prevent permanent accidental losses is seen in many INC and all AMP games. It is an adaptive device which saves

[1]The convention used for referring to specific players and specific games includes the group number (I or II), a diagonal (/), the initials of the player, a colon (:), and the game number (1, 2, or 3). (E.g., I/HS:3 refers to the third game of the Group I member identified by the initials HS.) The players' initials are fictitious to protect their privacy.

[2]All other INC-h games used the full 30 moves so we cannot extend this analysis in INC-h. Another example of the pressure of the trade-off on a poorer player, SC, is found in his INC-m and INC-ℓc/e games.

searching visually for all sc's which might be endangered by a move. (Poorer players may experience confusion and desperation on sc-breaking moves, as we shall see later.) The "P" or paths-through-a-point option is used relatively little in INC-h games--usually only in an already late and quite structured board. The computer's look-ahead feature (LKA) is not used in INC-h games.

So far, we have tried to establish some of the objective features of INC-h games--enough to give a sense of what they are like, and to distinguish them from others. We shall wait to present more about them until all the contrasting game types have become familiar to the reader.

Incremental-Moderate Games. The commonest game category, accounting for roughly one third of all games in our study, is the Incremental-moderate (INC-m) game. The INC-m class includes the best games played by the members of Group II, as well as the most typical games of Group I. It is defined by the main INC type criteria, plus a score range of 55-88, and a relatively straight-forward approach to the game.

Before proceeding with description of INC-m games, we pause for a note about categories. Only one INC-h game will be noted in discussion of more than one category (to be described). But from here on things are more complex. A number of games fitting the INC-m and INC-ℓc (Incremental-low, cautious) criteria have special features, and such games are more likely to be classed with more than one subtype, in recognition of those features. For games below 89, score alone is a weaker guide to type. For example, games in the 55-88 score range may be counted as INC-m alone, as INC-mpp (Incremental-moderate, partly planned), INC-e (Incremental-empirical), or as INC-epp (Incremental-empirical, partly planned), depending on their several attributes. Although it is cumbersome, we do this because the more frequently attained and less distinguished scores are more likely to be made in a variety of ways. And the lower in score we go, the more the games are likely to lack clear and consistent enough organization to fit neat categories. Yet, accepting overlapping and somewhat fuzzy categories, some things can be said which help us comprehend the variation in games scoring below 89.

If we compare INC-m games to the stronger INC-h, there are several general trends. Scoring graphs are more varied, reflecting the greater heteorogeneity of ways in which subjects, operating incrementally, guided the pattern-building process. The graphs now are more likely to plateau and even to descend toward game's end. Costs of moves may drive the graph down quite far before ascent is resumed. Such graphs, visible to the players as they work, are more jagged. In these less well-guided games, constraints placed by the cumulative results of earlier moves on opportunities for new moves make late-game, inexpensive sc-development opportunities scarcer. SC's tend to interlock and share tokens less well with each other, and to generate fewer good new potential sc's.

Not only the score, but the productivity per move (p_m) commonly declines. In many games the proportion of sc's which are low-scoring rather than high-scoring increases. There is more likelihood that "thinking time" gaps between moves will be followed by mere L-sc's. This contrasts with their primarily occurring as by-products of H-sc production in INC-h games. The sc's which do occur are more likely opportunistically, but inefficiently, to use up tokens later needed.

In other games scoring in the INC-m range, there remains a concentration on making H-sc's, but existing board configurations are not used as economically as possible in building them incrementally, and high move and purchase costs help to keep the net score down. This sort of thing was more likely to occur in bright Group II players' games.

In many INC-m games pacing was more of a problem. A higher proportion of players failed to use all 30 moves allowed.

<u>Incremental-Low, Cautious Games</u>. Many of the downward trends we have noted in passing from INC-h to INC-m continue, exacerbated in Incremental-low, cautious (INC-ℓc) games. The name derives from the combination of fulfillment of the basic INC type criteria, a low score range (20-54), and some coherence and evidence of conservatism (compared to other games which overlap them in scores). A good fraction of INC-ℓc games occur in the subjects' first rounds. They may be considered to involve the slowness, tentativeness, and restricted activity characteristic of many learners. But in a number of cases the INC-ℓc game is a second- and even a third-game occurrence. A few Group I subjects remained in this category in games 2 and 3, as did roughly a third of Group II subjects.[1]

Our INC-ℓc description is most typical of those instances which are second and third games. In such second and third games, subjects are often repeating previous ways of playing, using modest-cost moves for modest rewards. The games avoid a jagged graph. We are more likely here to see attempts to save points by using two slide moves (costing 1+1) to save one point, compared to a single jump (costing 3). The INC-ℓc game, compared with those previously described, has relatively more of its patterns in the form of low, rather than high, sc's. Interactive, productive uses of tokens in several sc's is less frequent and/or of poorer scoring quality.

We begin to see occasional four-in-a-row patterns that do not fit the requirements of any legal scoring array. These "pseudo-scoring patterns," (ps-sc's) often have everything right for a score except for, say, duplication of shape in two tokens, where a complete set of different shapes is called for to complete an sc. Similarly, we begin to see more than rare instances of low scoring patterns made where it was perfectly possible, and no more costly in points or deleterious to later moves, to make a high scoring pattern instead. It is

[1] Many other Group II subjects had second and third games in INC-lc score range. But their games appear by no means "coherent" and "cautious." They are classed as INC-d, -r, or -e.

difficult to determine how to count all likely instances of apparent scoring-pattern manqués. At the limits, the criteria for judging a four-in-a-row pattern to be unnecessarily a low- or no-score become fuzzy and hard to apply. Therefore, we did not try a rigorous counting procedure. Instead, we noted where opportunities seem most blatantly to have been missed.

The next four subtypes in INC games are named for special characteristics. Their names correspond to ways in which they deviate from the more-or-less straightforward incrementalism of the preceding three.

Incremental-Disorganized Games. As we have descended the score range of INC games, we have found evidence of decreasing quality of resource utilization and of coordination among moves. While this may be thought of as poorer organization, and while we shall find still other categories which show substantial difficulties, the Incremental-disorganized (INC-d) games are distinguished for printouts showing a battle substantially won by the forces of disorganization. To discover the exact boundaries delimiting this category from some of its relatively chaotic neighbors would take better talmudists than we. But the main characteristics warranting the label can be given. The main INC criteria are met. The score range is -63 to a somewhat arbitrary cutoff at +20. INC-d games are not, as some might expect, strictly confined to first games or to subjects in Group II. Two advanced design students scored here in two and in three games, respectively. Two of the numerous Group II INC-d players stayed here throughout. Difficult as they are to try to analyze, we shall try to deal with them. The picture would be misleading were we simply to dismiss these games as failing the instructions, or as not serious.

Often, there are relatively few intact scoring patterns remaining on the field at the end of the INC-d struggle. In some cases many sc's may have been made and broken. Pseudo-sc's are frequent in some INC-d games. Move costs may be extravagant compared to rewards earned. The sc's seldom, if ever, multiple-use tokens. The sc's are relatively likely to remain isolated fragments: few move sequences appear to coordinate the making of one pattern with another. Event sequences can occur which go more or less like this: A potential sc starts to be built; is abandoned; and is picked up later, in the midst of something else, and built at greater cost than necessary. Then, though the new sc made possible another, interlocking, pattern, the opportunity is not taken up, and the region is abandoned once more.

We have said earlier that the INC game is heavily determined by responses to momentary states of the board. Now we see in INC-d games an extraordinary lack of coherence in ways in which subjects react to use current states of the board. A player misses and destroys opportunities that are there, including the very ones he has created. His average increments of useful structure per game operation are very low. And his creations are more likely to be tiny nuclei or largely isolated fragments.

When patterns previously made have just been broken, it is common to see the UM operation, which could have saved them, ignored. A just-made H-sc may even be cannibalized to make another, at times lower, sc.

Some INC-d records show delivery of many error and distress messages from the computer, as things are illegally pointed by the light pen. Exchange moves are frequent, but they occur in situations suggesting that regular move technique has been mixed up since there is no advantage, and often a disadvantage, to the exchange. Yet such moves are often allowed to remain--i.e., no UM was used to retract the move. As if for orientation, there is a high frequency of uses of paths through a point, often in long sequences; and of the display graph command, which brings back the graph when it has disappeared during a path's display.

Incremental-Restricted Games. In a sense, all Shimoku games are restricted: subjects choose, or unwittingly act to avoid, certain kinds of operations. But the games we call INC-r are extremes. Their records have the characteristic that a guidance system generating them would be heavily dominated by a very few rules which, while restrictive, do not appear to an outsider to be particularly well adapted to the requirements of the game. We choose two games in particular to illustrate the idea of severe restriction.

One is the first game of subject II/DE. In this game, none of the tokens originally given on B_i are moved: instead, all 30 legal moves are purchases of fresh tokens from the reservoir! Fifty-nine purchases are tested out, tentatively. Computer Look-ahead and Unmove are used in weeding out those rejected. The net result, after subtracting costs for LKA, is 45 points, with 18 sc's, only 7 of which are H-sc's. Restricted, yet not unrespectable for a first game. Would he improve? No. He fell apart. His games 2 and 3 are INC-d's. Without the restriction he appeared to become overwhelmed and played chaotic games.

The other remarkable game, the first for subject KB,[1] had virtually every sc-building sequence and, sometimes, every move in a sequence follow a succession of applications of the P (paths-through-a-point) probe. The record suggests that P is used systematically as the game's main move generator. Paths are probed through each cell of the board starting at the left, until an sc-1 is identified and completed. In one hour, just 15 moves were completed in 324 actions (usual range is <120), for a net score of 25. Despite promises, KB never returned for games 2 and 3. We shall refer again to KB's markedly restricted game.

---

[1] KB was recruited for Group II. Since he did not complete three games, he is not counted as a group member.

Incremental-Empirical Games. Games having a very high number (>120) of total operations, with many uses of P, and/or LKA, and/or UM, (and a richer variety of uses of empirical techniques than the restricted KB game) are called Incremental-empirical (INC-e). Characteristic of the games is use of actions on the computer scope to test out possibilities concretely and often in great detail before making or accepting moves. All games of one player in Group I and 20 games in Group II show this characteristic. Some highly empirical games fall into the INC-m and INC-ℓc score ranges, but they are not "relatively straightforward" or "cautious" in style and are not concurrently classified INC-m or INC-ℓc. Both INC-r games, many INC-d games, and two INC-pp games are also INC-e. Scores are less than 70. Thus, no INC-h, MP, or AMP games are also INC-e.

Incremental-Partly Planned Games. One INC-h Group I and a couple of INC-m Group II games have tokens placed in relation that suggest a primitive or a partial attempt at a set-up involving most of the 30-move complement or the whole board. Such games are called Incremental-partly planned (INC-pp) games. The set-ups may be good but incomplete (I/EH:1) or quite defective (II/TS:3). They are not worked out in terms of most of the 30-move complement and do not develop move sequences that deal coherently with most of the game constraints and options that we see incorporated into MP and AMP games.

On Naming Games in the INC Series. As noted earlier, where a game has the attributes fitting it to more than one category and those attributes are relevant to the discussion, we combine the names. I/EH:1 is a game which is not purely incremental, and has evidences of partial planning, a token structure of the kind noted in the previous paragraph. Called INC-hpp, it is stronger on the pp dimension than the usual member of the INC-h class.

Master Planned Games: Type MP. We have presented some of the essential characteristics shared by MP and AMP games early in defining the INC games. Briefly, game records of both AMP and MP types show tokens structured so that an easily described main design, involving more than two thirds of the complement of 30 moves, is developed. The main pattern is such that attributes of tokens used in early scoring patterns coordinate well with attributes of tokens used much later. The result is a predominance of high scoring patterns with a high proportion of multiple-use tokens. The final board configurations would be difficult, if not impossible, to generate on a purely incrementally guided basis. The move sequences suggest that developing the main design took precedence over other types of moves. Frequently, potential high-scoring configurations on the board were not completed where their completion would have blocked the larger design.

Given these similarities, what distinguishes the MP from the AMP game? In the MP game in its pure form, we find relatively little attempt to make the most of the positions of tokens as they are given in the initial board configuration in executing the main design. Unlike the AMP game, the MP game appears to

execute a scheme both essentially master planned before the start of moves and little adapted to the peculiarities of local starting conditions.

Master Planned-Classic Games. The Master Planned-classic (MP-c) games are a relatively pure type. The game record shows rapid play--usually less than 30 minutes--in which, typically, nearly all tokens on the board are rearranged. They form a design consisting of two intersecting planes, composed totally of multiply-used tokens, in nothing but high scoring patterns. The record usually includes a sketch for such a final board which the subject brought with him, or drew out on his scratch paper before starting to move. Although the configuration of the opening board is not known beforehand by the player, little time which might be used for study may pass before moves are made which execute a close version of the diagram on the subject's sketch. When a few moves are left over at the end of completing the perfect main design, an additional H-sc is made and the game is over.

Such a game record shows no uses of the exploratory devices P, LKA, and UM (except to undo a technical error, such as pointing to the wrong token). The flow of moves is relatively regular after the pattern has been begun. Once moving has begun, there are no long silent periods followed by strings of related moves, as in INC and AMP games. If there are occasional delays, they are followed by a trial of a possible exchange move, a device for saving a move by trading the positions of two tokens in just a single move. There are no score drops for breaking sc's. The game curve may stay low, droop, and sag for long periods of the game--tokens are being moved in a more or less total board rearrangement, and often do not come together as sc's until a high move cost has been accumulated. Toward the end the curve rises rapidly, as single moves complete more than one sc at a time. Even were one not to be able to see the characteristic design on the final board, the other attributes of pure MP-c game records would speak of a kind of procedure for MP-c's different from those generating INC or AM games.

Master Planned-Lapsed Games. Games are named MP-ℓ, Master Planned-lapsed, when they show part of a main pattern in the fashion of the MP-c game, and occur after the playing of an MP-c game or after the drawing of a diagram for an MP-c game by the same subject who then plays the MP-ℓ game. Two games are MP-ℓ. They are of interest because they show that the MP-c game is not easily retrieved if moves are once accepted which do not follow the over-all design accurately. Their existence tends to confirm the idea we have from examining game records: that it must be difficult for a human player to score in the over-110 range without a well-executed main plan. Here, subject I/HC, whose second game scored 139, fell to 100. Subject I/EWB, who entered carrying a diagram for his third game, came right out at the end of it and announced that he had made a serious error of execution. He claimed to have misread his diagram, made a few changes and, at his request, was allowed immediately to play a fourth game. His MP-ℓ game 3 score had been 54; for his MP-c game 4, it was 171.

Master-Planned-Developing Games. We consider first a single game which is hard to categorize from the objective records alone. It is the third game of I/TT, whose first two games were unquestionably INC-m. His first several moves in game 3 followed an apparently incremental pace. They produced a configuration with multiply-used tokens in high scoring, interactive patterns within a single grid.[1] Thereafter, until his last few moves, all moves went to develop two separate grids, each totally filled with interlocked H-sc's. The last few moves were preceded by path explorations and time lags and were tacked on to the basic, two-grids-filled pattern. The score was 140. This is the only game we call MP in which the two planes are separate grids and do not intersect--hence, somewhat less over-all coordination is required. Should we call it MP at all? It is more successfully coordinated and diagrammatic than our INC games. For the only time in our records-based typology, we add interview data to decide.

I/TT stated that he had been expecting to play game 3 as he had game 2 and described an incremental approach. However, on noticing the larger pattern suggested by the results of this "first few moves," he "blundered into" the idea of an "optimum configuration." The noticing was triggered by his checking the reason for picking up 14 points more than expected on making an sc.

The printout shows his first double score to have occurred on completing move 13. At that point, grid 3 of the board showed incrementally built scoring patterns in place, as in Figure 19.

Note that sc's using the set of numbers 1 through 4 were concentrated on one readily visible plane. And a couple of sc's using the numbers 5 through 8 were already in place on the adjacent grid. At that point he "saw" what he was to call the "optimum configuration."

He described his strategy as one for making two filled grids, in two sets of tokens numbered 1-4 and 5-8, using board tokens only. He said that he had decided to use a diagram if he were ever to play again and felt "idiotic" for not seeing the simple way earlier. Thus, it seems that most of game 3 was played according to an over-all plan that coalesced during play, after move 13, in reaction to perceived larger properties of an incrementally built array of sc's. He said he was unable to execute the plan completely in the current game because of certain moves already made and had to settle for an approximation to it. The game shows an on-line transition to the master plan idea and is better categorized as MP-d than INC.

---

[1] We say "grid" to indicate a horizontal plane of the cube as it appears on the scope. While it is mathematically equivalent to a plane oriented in any other direction in the cube, it is perceptually the most accessible sort of plane because of the display. Subjects are more likely to see patterns emerging in the planes of the grids because grid planes do not require extra perceptual effort (piecing together parts of four separated grids to form a single plane.)

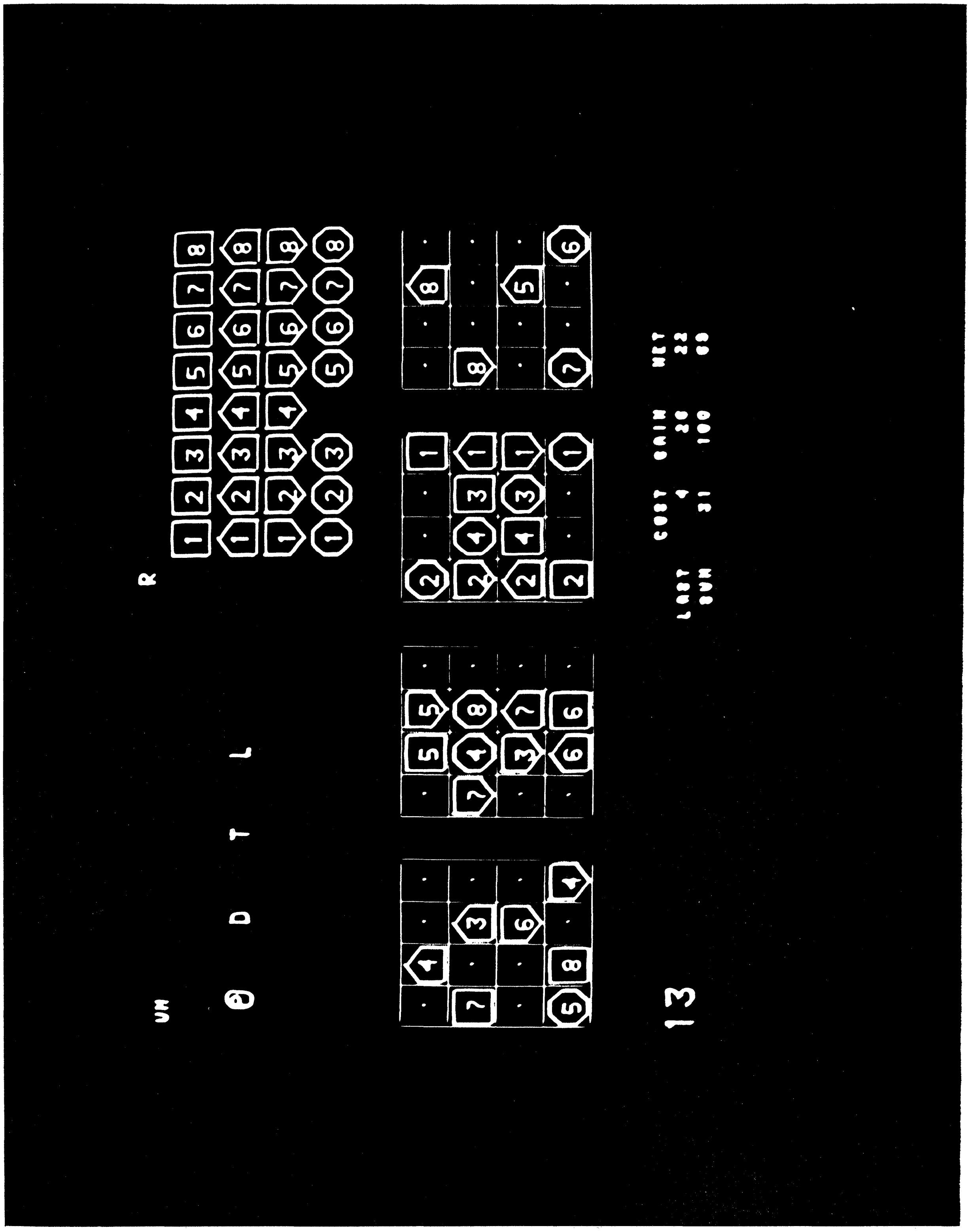


Figure 19. Board Configuration after Move 13, Game I/TT:3

A second case worth mentioning here is the one already classified as INC-hpp. It is a real borderline case, the first game of I/EH who played MP-c second and third games. Like I/TT's game, it has evidences of incremental play followed by the setting up of two main planes; like the other MP and AMP games, the two planes would have intersected. However, the serendipity which allowed I/TT to use most of his earlier moves in the later-formulated design did not operate here, and the design as executed is substantially more incomplete, as evidenced by the much lower score of 91. It shows less good relation of the design to the use of most of the complement of 30 moves.

Adaptive Master-Planned Games: Type AMP. AMP games were played by one subject, I/KN, a graduate architect working in a large-scale, urban-design program. In three games, he scored 131, 154, and 241 points. The scores for games 1 and 3 are peaks for their runs, and the latter tops all games played. The last game is called the classic, and the others are designated precursors. This subject's play is treated later in the main text and program 3 is presented in Appendix C.

Adaptive Master Planned-Classic Game. Like the MP-c, the AMP-c game has, from the start, moves setting up structures for a main pattern realizable in more than two thirds of the 30 moves. But it realizes that main design in a flexible, adaptive fashion. Perfection in the main design appears to have been purposely traded off against the use of already existing board tokens and their positions in sc's contributing to the main design. Again, in contrast to pure MP-c's, board and purchased tokens are used, and sc's interact with the main design. Scratch paper, the existence of the same basic configuration in the second as well as in the third game, and other evidence confirm that the subject has a well developed plan (two planes intersecting as X's, standing vertically in the cube). But, in contrast to the MP-c's, his game was not played so rapidly and regularly, as if read off a diagram. Substantial time lags separate bursts of moves.

Pieces are not simply bulldozed out of the path of the main design's construction. A good deal of time goes into resetting them to fit into potentially productive new places. Even with this handling of detail, completion of the main design occurs at move 23, with more than 20 minutes still to go. At that point, the score of 176 is very close to that which top MP-c games have when they end at 29 or 30 moves.[1] Here, in the remaining 7 moves, new and previously prepared opportunities to make sc's, interacting with the main design and with each other on the side, are brought to fruition. Sixty-five additional points are gained. But, if we simply count points from sc's included in the two design planes of each, the

---

[1]Examples are: I/NQ:2, 185; I/NQ:3, 179, and I/EWB:4, 171. (Given the plans realized in I/NQ's games, there was no way to use the last move profitably.)

actual main pattern as realized in the AMP-c game scores less than the main pattern of the MP-c. In contrast, the MP-c design itself as realized is perfect in consisting of the highest scoring sc's with the highest intra-main design multiple uses. The AMP-c has inelegant compromises in the form of sc's within the main design which do not always fit the design or score quite so well. However, the AMP-c game makes use of 18 pieces in their original board positions, purchases additional tokens with the extra moves, and comes out with a higher sum of sc's and points. Table II compares some critical measures of quality of the two games. Figures 20 and 21 show the final boards of each.

Adaptive Master Planned-Precursor Games. The two AMP-p games are I/KN:1 and I/KN:2. Before moving at all, and without writing on the scratch paper, I/KN sat before the screen for 15.2 minutes at the start of game 1. He made two moves and sat for 12.2 minutes more, starting to make marks on the scratch paper. The time nearly half expired, the printout thereafter shows moves in bursts separated still by as much as 7 minutes. From the start, grid 2 is arranged in a perfect, totally high scoring, maximally interactive design. No move is used which serves only to complete a low scoring pattern. Purchasing begins early, at move 9. Four H-sc's penetrate the cube, interacting with the main single-grid design. Most of the others interact only with additional sc's on their own planar grids. Early moves have placed non-main design tokens where they are used later for high scores.[1] Four tokens on the final board do not participate in sc's. The record shows a great deal of planning time early, and a game with high scoring, relatively well organized uses of nearly all the 30 moves in both early and late stages.

This first game takes little advantage, however, of the possibilities for getting extra service from tokens by using the third dimension, and does not benefit from the higher degrees of coordination possible with a larger main design.

I/KN's second game is markedly different in both respects. The main design has changed to one of X's on each grid, forming two intersecting planes in the cube, involving 32 tokens. Note that I/KN's design intersects the planes not within a cell row, but between rows. This means that the planes do not have to be coordinated so as to share a row, as in the MP-c games, where only 28 tokens then participate in the main design. The only main designs using 32 tokens are the serendipitously planned master of I/TT,[2] and the obviously

[1] We have classed as INC a number of games which fill a single grid. They differ from this game in quality of organization of the grid and/or in quality of coordination of early and late moves of the complement of 30. This difference is reflected in the scores.

[2] The MP-d game described on p. 89.

Table II. Comparison of Master Planned-classic and Adaptive Master Planned-classic Shimoku Games (Subjects I/NQ and I/KN, Third Games)

Subject & Game	Net Score	Total Move Cost	Total Moves	Total Actions	Total Time (Min.)	Total B_f Patterns	Total B_f Tokens	B_f Tokens Not Scoring
I/NQ:MP-c	179	77	29	64	42.8	20	32	0
I/KN:AMP-c	241	121	30	86	60.0	37	51	2

Subject & Game	Productivity in Points/ Move	Patterns/ Move	Net Points per B_f Pattern	B_f Tokens from Reservoir	B_f Tokens Scoring in B_i Cells	Times Paths Requested Through Cells
I/NQ:MP-c	6.17	0.69	8.95	0	3	0
I/KN:AMP-c	8.03	1.23	6.51	19	18	8

* The first 17 minutes were without a move, as NQ was rechecking his Master Plan. The game was played in 25.8 minutes from the time first move began.

** Because of time-shared system down period, actual running time was 64.5 minutes, with 4.5 minutes during which computer was unresponsive. See Appendix C, printout.

B_f = Final board configuration of tokens--the solution configuration.

B_i = Initial board configuration of tokens--the given configuration.

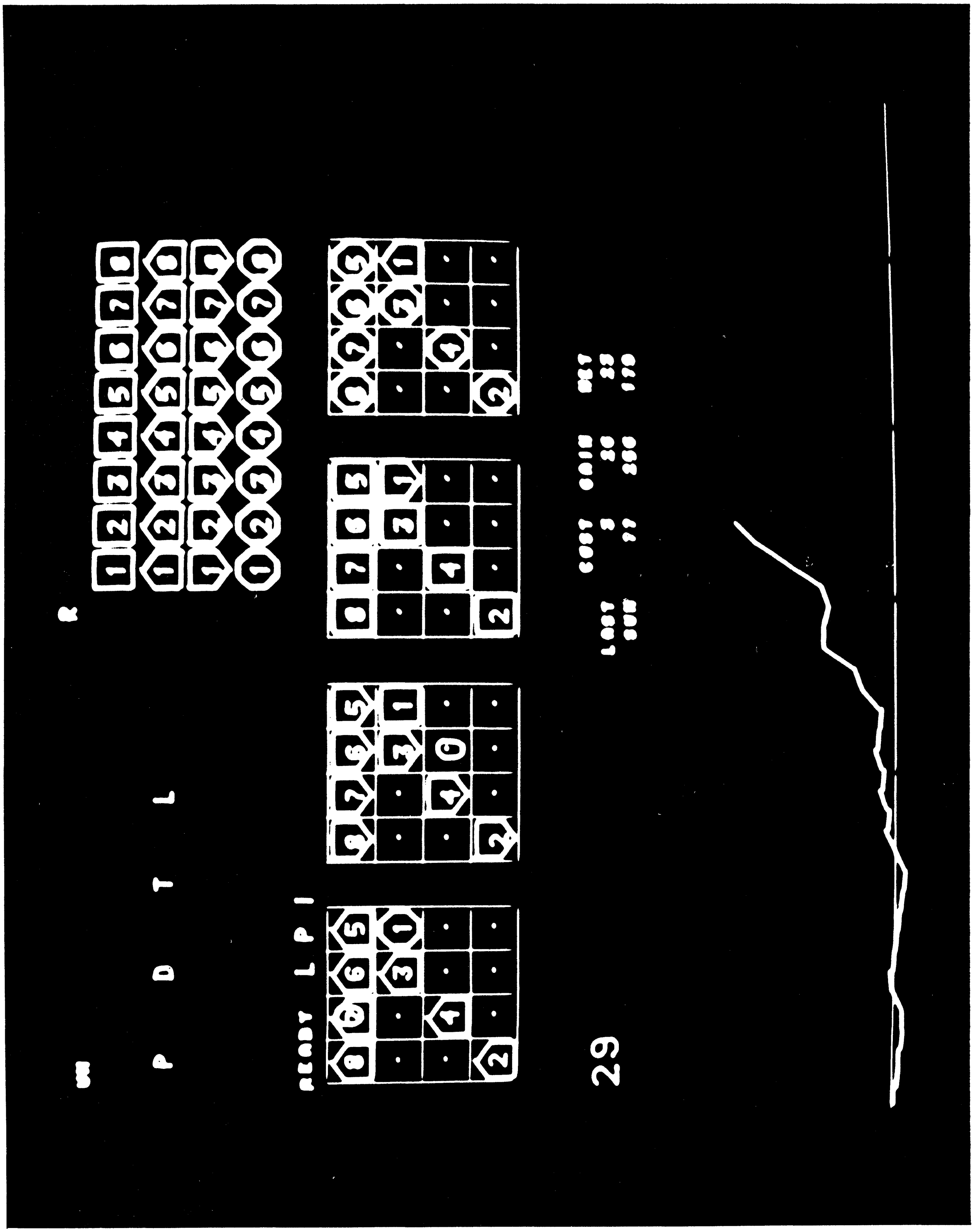


Figure 20. Final Board of Game I/NQ:3, Type MP-c

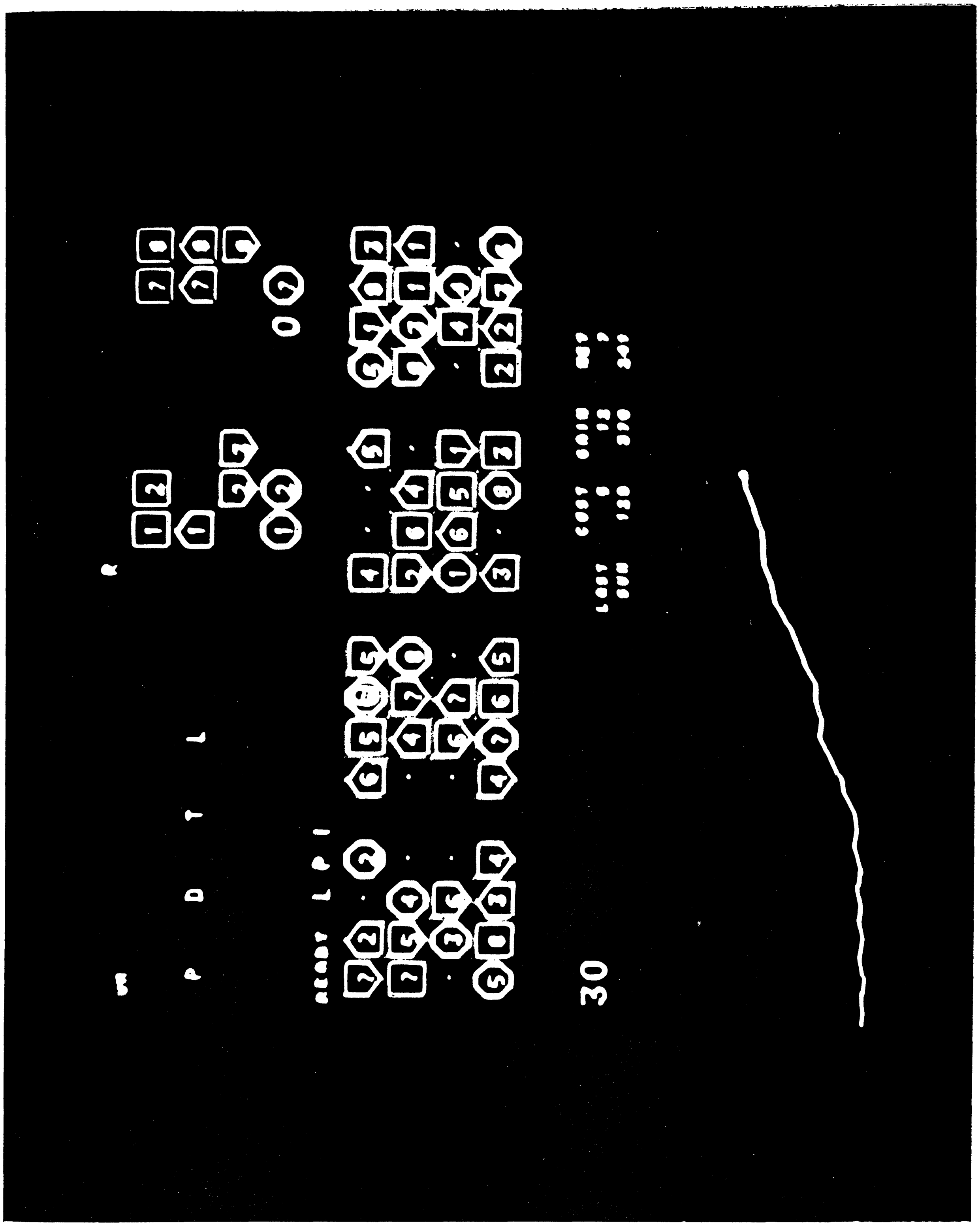


Figure 21. Final Board of Game I/KN:3, Type AMP-c

preplanned adaptive masters of I/KN. Recall that the opening board configuration is unknown before the start. One who has to clear away and build for a 32 piece design in a maximum of 30 moves accepts a situation with more pressure to adapt his design to current token positions than does one whose design requires only 28.

The record of KN 2 shows many of the same features we have noted for KN 3. It differs in still having a long (17 minute) pre-move period. And it differs in the less good adaptation of the main design to local board conditions: a number of token selections for gaining multiple and high scores are less-than-best in terms of the overall configuration of the game. Fewer of the factors influencing the potential worth of a given token in a main design or interactive peripheral position appear to have been taken into account.

Arrived Games: Stage A. We introduce the notion of stages as a further help to analysis. An Arrived (A) game is one in which there is evidence that the style of play is relatively stabilized. Generally, this means the record shows two substantially similar games in a row for the same player. We also use the term where the game is the third and known-to-be-last game of a subject and shows from the start a coherent building upon the previous game.

Heuristically, we use the notion to emphasize that we want to look at games called Arrived in a certain way. We want to ask, what would it take to guide a given game to the recorded outcome we see, if the guidance system knows at the start that it is going to move pretty much as it then proceeds to do; and if, while playing, the system is concentrating only on realizing the single game according to its previous intentions.

Learning Games: Stage L. Learning (L) games are those in which there is not primarily repetition of something a subject has already arrived at. Instead, the subject is unfolding something whose building is essentially unfinished for him. In the L game--from the game records--new things are being tried out, aspects of the game not previously handled are just being taken into account, kinds of actions used in previous games are being abandoned, and so on.

We shall not try to characterize all games as A or L, but we shall choose several which appear to support the distinction and use them in illustrating the information-processing implications of the A and L stages for a problem worker.

Problem Characteristics of Arrived Stage Games of the Main Types

Having introduced the main classes into which subjects' game records may be grouped, how are we to explore the relations which hold between process records and significant events in subjects' games? Let us take a relatively easy part of the work first. We shall try to see what it would take to execute an Arrived stage game of each main type--that is, a game in which the player knew at the start substantially the kind of game he expected to play and the score range he might achieve if he were to execute his scheme effectively.

Some Objective Information Processing Requirements for Hypothetical Human-Like Problem Workers

To throw into relief the Arrived Stage subject's task for this part of the analysis we imagine a hypothetical experiment. Consider three fictional Human-like Problem Workers (HPWs). Our idealized HPW is unlike an ordinary person in this respect: once an HPW has been told to remember instructions, he stores these instructions perfectly in long-term memory and has rapid access to any part of them thereafter. An HPW is like a human in having the same ability to interpret stored instructions and execute them physically, and the same limitations on attention, short-term memory, immediate processing (including visual search), and computational ability that we might find in a very bright mathematician or designer. Each HPW is instructed in Shimoku, is highly motivated, and has accepted the goal to "make as many points as possible." In addition, each has stored and intends to follow as well as he can a set of general instructions to play only a game of a particular type. HPW's have been assigned the best developed subtypes in each type. Thus, naming them by their game-type instructions, we have HPW/INC-h, HPW/MP-c, and HPW/AMP-c. To make sure that each HPW can envision the game-type instructions in the actual context of Shimoku, he plays through a first model game. Each HPW executes on the computer the print-out of a real-life subject whose game exemplifies the assigned type. The HPW sees in what way each move does (or does not) make sense. He gets the feel of how the game-type instructions relate to an actual interpretation in a game. But he does not store in long-term memory any exact moves or token positions. Each HPW how has the Shimoku play rules, the type instructions, and basic familiarity with the mechanics and appearance of a game of his assigned type.

A week later, each highly motivated HPW is reminded that the previously executed game showed it is indeed possible to play in a given point range using the assigned game type of instructions. A range of 100+ is mentioned

for INC-h, 160+ for MP-c, and 200+ for AMP-c[1]. Each accepts an appropriate score range as his goal.

Our questions: (1) What are some important features of the set of general instructions we must give to each HPW to assure that he can play according to the assigned type and that he stands a chance to score in the given range? (2) Once having those instructions, what are some important activities required of each HPW if he is to execute such a game? Recall that the distribution of the set of 32 tokens on the opening board is unknown until game time[2].

First, we shall see what might be the nature of an instruction set supposedly already formulated elsewhere, and now given to the HPW as a set of task, goal, rule of thumb, etc. formulations prescriptive of an assigned game type. The exposition will emphasize highlights, based on a composite of subjects' characteristic games of each type, and is not intended to be exhaustive. Second, we shall see what might be some of the burdens set upon an HPW having to interpret and execute each game-type instruction set, on-line, in the time- and move-limited Shimoku task environment. The HPW's job is not to make high-level descriptions or programs, but merely to interpret received, higher-level instructions at lower levels, and to execute them as effectively as possible during the game play.

---

[1] We know that real subjects playing their Arrived stage third games had the same knowledge before starting: I/HS for INC-h, and I/NQ for MP-c, by virtue of their own previous scores; I/KN for AMP-c by virtue of his own previous scores and a calculation based on correcting of errors he had recognized in game 2.

[2] The opening configuration is assumed equivalent in difficulty to our B_i. Our subjects never knew what opening configuration to expect. Because the B_i was so amorphous, they generally did not know that they were, in fact, reencountering the identical board when they began each game. Subjects who mentioned it in interviews felt that knowing the board distribution to be encountered would be valuable, but felt they could not know it beforehand. They had concluded that there had been no use trying to memorize the difficult mess they had met at the start of any current game. Eventually, several players recognized relations among a few tokens when they saw them the third time around (I/KN on round 2) and thought they might have the same board. But not having encoded their previous game-related knowledge in terms of specific token arrangements, and not having anticipated having this arrangement, they could not use board recognition--beyond faster making of one or two sc's--to alter the over-all carrying out of their games.

Some Objective Requirements for INC-h Games. For the INC-h game, a set of instructions serves as a kind of guiding raw material for forming instructions as they are to be adapted to board conditions at each move.

(1)
1) Find an sc-1
2) If there is no sc-1, find an sc-2
3) Find and move a token, t_i, such that (sc-2)+t_i → (sc-1)
4) Find and move a token, t_j, such that (sc-1)+t_j → sc.
5) Repeat procedure[1]

Additional instructions are given the HPW in the form of statements which modify the handling of (1).

(2) They are in the form of a largely unordered collection of statements.[2] Statements modifying the interpretation and application of instructions 1 and 2 include:

a) Search visually for sc-1 and sc-2.
b) Speed and simplify search by doing the following:
c) Work in regions of the cube whose board representations permit visual path tracing to be relatively quick and automatic, based on ordinary visual habits. Avoid searches which require continual fresh computation of paths.
d) Work to complete the most readily perceived potential sc's.
e) Search preferentially in areas most recently changed by previous moves to find new potential sc's.
f) Increase net rewards of a given search by doing the following:
g) Give priority to search for incomplete sc's that can become H-sc's rather than L-sc's.

---

[1] The stop conditions have already been given as part of the game rules and will be integrated with this little program by the HPW.

[2] The small letters marking the statements permit reference to items later in the text.

h) Evaluate potential sc's by matching their potential patterns against types in the scoring-pattern list and computer scores they would make on completion.

i) Invest little in time or in points to find partial L-sc's. Make them mainly as by-products of primary efforts to make H-sc's. (Also modifies instructions 3 and 4.)

j) Give priority to search for possibly interactive sc opportunities.

k) Find potential sc's which include tokens already participating in existing sc's.

l) Find potential sc's whose tokens will in turn be usable as part of profitable future sc's.

m) Look ahead as far as is practical[1] to anticipate and evaluate such interactions.

Statements modifying instructions 3 and 4 include:

n) Minimize the cost of putting token (t_i or t_j) with appropriate shape and number attributes into a potential sc.

o) Build new sc's with tokens moved from the board, rather than purchased from the reservoir.

p) Where several tokens might suitably complete a given sc, move the token whose continued presence in its home cell blocks developing still another sc.

q) Wherever possible, use exchanges to save points and moves.

r) Conserve the supply of potentially critically needed tokens to maintain flexibility for meeting unpredicated late game needs.

---

[1]Interpretation and execution of this statement is one of the most difficult for HPW to do in a fashion that balances information-processing capabilities, task structure, and time constraints. In Arrived stage games, the subject usually has some idea of what it is "practical" for him to undertake based on his previous games, but the idea still is not clearly developed, and requires continual testing and reinterpretation, as we shall see in the discussion of subjective experiences which follows this section.

s) Avoid using up blocks of tokens of the same number in 4-in-a-row sc's.

t) If you can be certain of making four or more H-sc's in addition to a candidate H-sc, by using 4-in-a-row of the same number, reverse (s) above and use them.

u) Use tokens on extremes of the numerical series (1,2,7,8) in cells completing straights in preference to middle number tokens.[1]

v) If tokens completing straights go into cells which are likely to have high interactive opportunities, reverse (u) and use more easily built-upon middle numbers.

w) In selecting tokens, avoid accidental sc destruction.

x) Before moving a candidate token, test: does it participate in any existing sc (or potentially valuable sc-1)? Use any of the following methods, whichever seems most economical and reliable for a given token's evaluation:

y) Search visually for sc's on all paths through t's home cell.

z) Use computer P option, if necessary for thorough search.

---

[1]Given that one is building sc's on an ad hoc incremental basis, it is more likely that occasions will develop later in which middle numbers will be needed to complete potential sc's. This is readily seen by considering which tokens can participate with each other in straight run sc's:

```
1 2 3 4
  2 3 4 5
    3 4 5 6
      4 5 6 7
        5 6 7 8
```

Thus, if a 4 or an 8 might be equally well used to complete a 5, 6, 7 straight, ceteris paribus, conservation of the 4 is preferable.

OR

aa) Make a "valued locations" diagram: throughout the game mark off the four cells whose contents form a new sc, as soon as each sc is completed. To evaluate the safety of a candidate move of a token from its cell, check the corresponding cell in the scratch diagram: is token in an sc?

OR

bb) Move token and monitor graph and table.

cc) If negative point change is greater than expected, unmove immediately (before making another legal move).

dd) Check to find the broken sc, and re-evaluate intended move of token against alternatives.

Higher-level statements modifying application of the whole set of instructions include:

ee) Pace inputs of time, points, and effort. Higher rewards per move should be earned earlier in the game; relax expectations of high gains toward the end when board may be crowded and opportunities obstructed.

ff) Do not postpone purchases unduly. Opportunities for making extra sc's toward the end increase if purchases have rendered board more densely populated, but not obstructed.

The statements given the HPW are a cleaned up, more articulated, composite[1] collection of those that several INC-h subjects tried to use--judging from

---

[1] We have not arbitrarily combined conflicting or incompatible instructions. Where parts of the instruction set are in open or possible conflict with each other, it is because both sides of the conflict typically were represented within individual subject's data. Each of our INC-h games included something corresponding to, or was consistent with, each of the items in HPW's instruction set.

printouts, scratch sheets, and their statements of principles for use in guiding their games. Thus, at start of HPW's game, we can view him as having the job of an Arrived stage INC-h subject who remembers rules and instructions perfectly. What will HPW's job be like?

Let us imagine HPW at the console, with 30 moves or 60 minutes in which to act. HPW faces first the standard B_i, and then in turn the sequence of B_t's that result from his own previous moves. Perhaps the reader will want to get a sense of the task facing HPW. If so, he should assimilate the rules and the INC-h instruction set. Then, he should try to apply them: first to the opening game, starting from the B_i diagram, Figure 8; then to mid- and late-game situations, using the board configurations of moves 18 and 25 from game I/HS:3, an Arrived stage INC-h game (Figures 22 and 23).

HPW will notice that the instructions vary in vagueness and specificity and in openness or closedness of their constraints. Many of them are in a form familiar to us from our acquaintance with policy statements. The generalizations usually are not in operational form. They require interpretation before meaningful use in a particular concrete situation will be possible. Often, it is clear they can only be taken as recommendations or guidelines--statements toward whose fulfillment HPW should aim. Many of the instructions are in <u>ceteris paribus</u> form. For example, other things being equal, HPW should follow such directives as (o) and (c). But application of (o), saving purchase costs, must confront the desirability of, among other things, (ff), which shows rewards from purchased tokens; and application of (c) and (d), conserving search effort, pull against the thrusts toward thorough search of recommendations (j) through (v).

What is more, the tensions among the statements must be settled for each move (or short move string) in terms of the constraints newly posed by each B_t.

The contents of (2) are given as unordered, "<u>atomic</u>" sentences. But the individual instructions--or instructions which would be derived from them on interpretation--can only be used well in "<u>molecular</u>" combinations with each other. To put it more procedurally, the collection of instructions must be mutually adjusted--in light of the conditions they place upon each other, and the conditions placed upon them by each particular concrete board situation--and must be combined to create executable action sequences. HPW has to piece together a fairly complicated move generation, evaluation, and execution "program," using the INC-h instruction set and B_t at each move. But relatively few parts of that program can then be re-used automatically without further adjustment to later contexts.

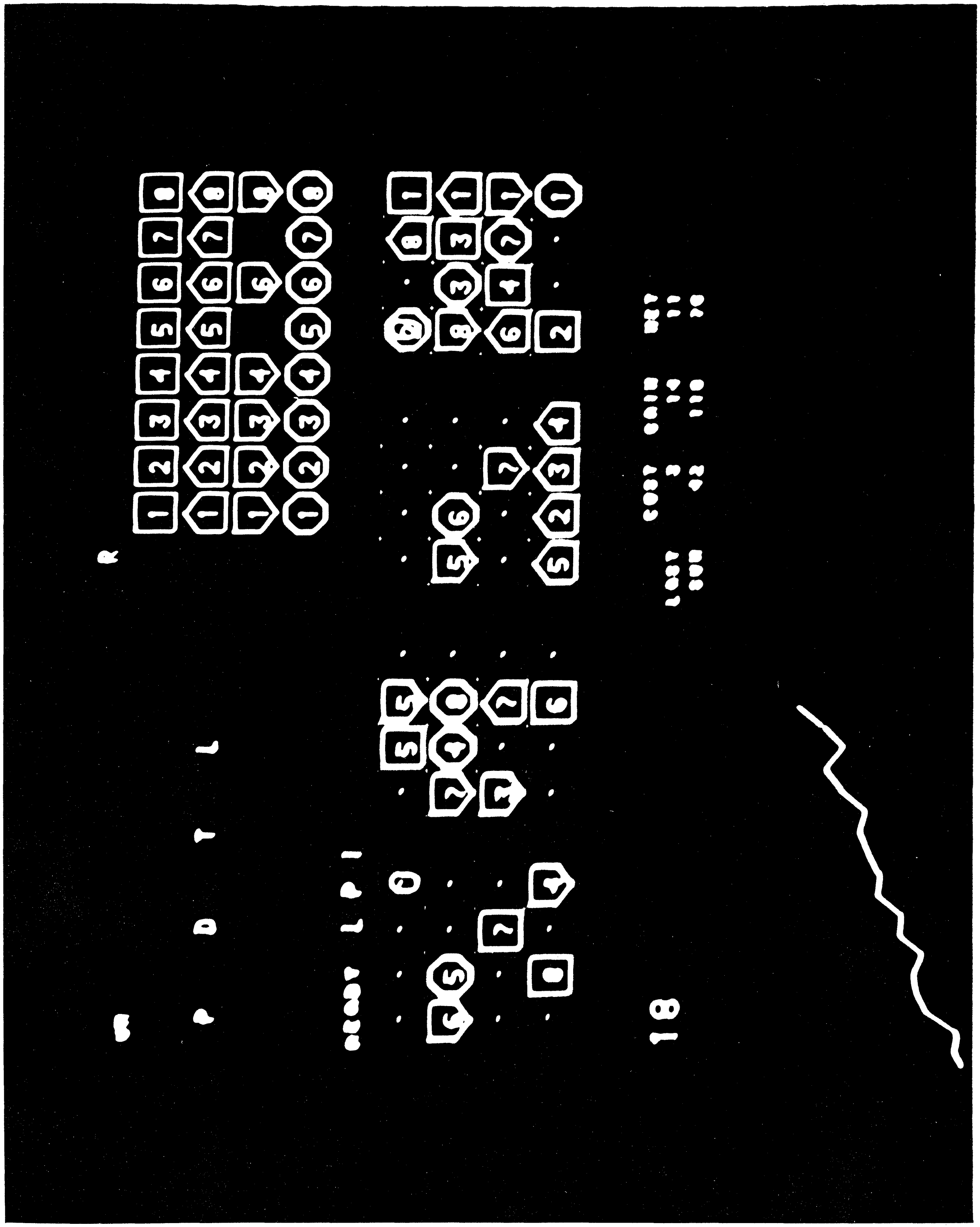


Figure 22. Board at Move 18, Game I/HS:3, Type INC-h

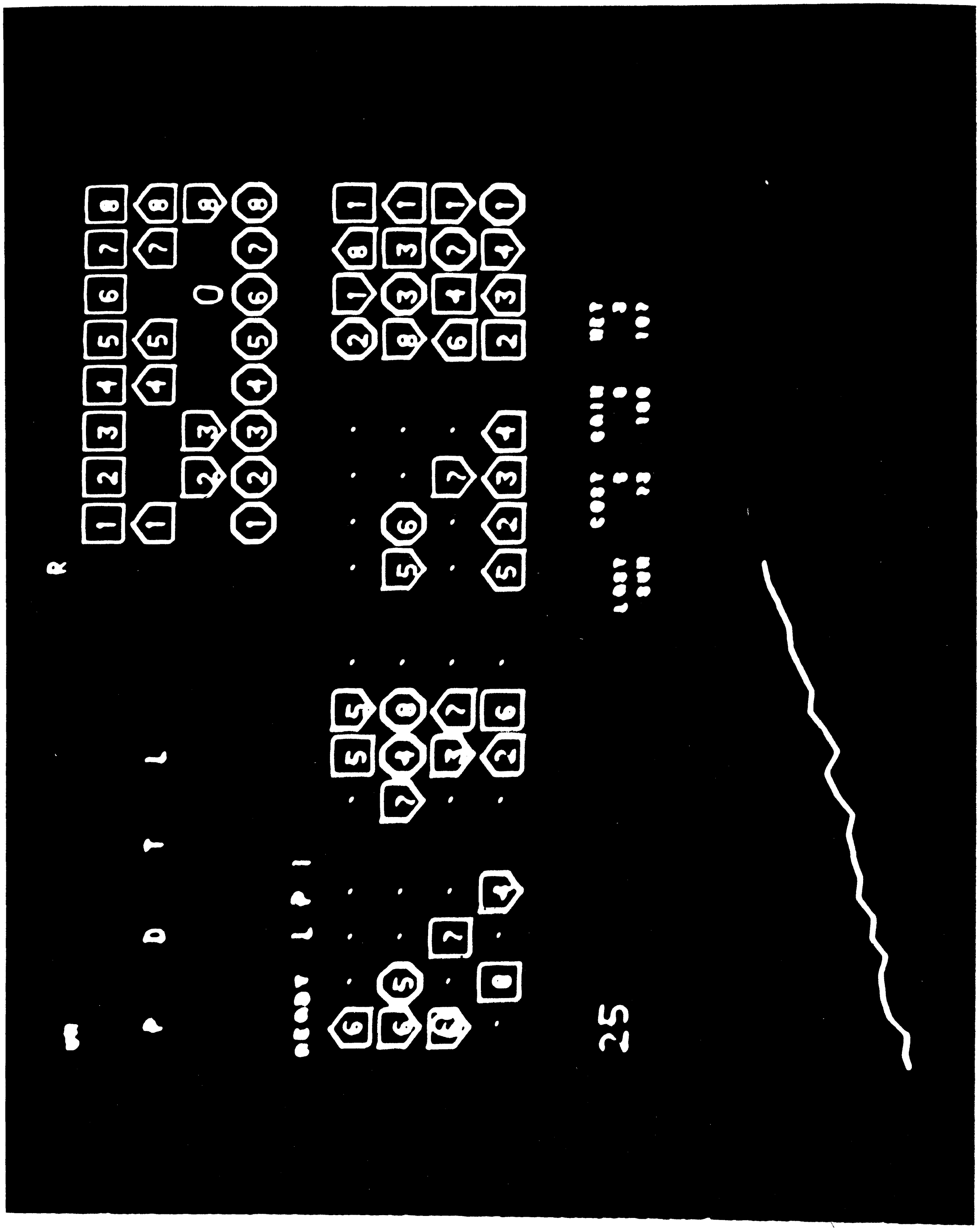


Figure 23. Board at Move 25, Game I/HS:3, Type INC-h

In the INC-h game, the nature of the later board configurations is substantially unpredicted when decisions must be made for an early board. The instructions allow for the incremental game's way of changing from moment to moment the relative merits of different policies. It does so by avoiding tight specification of priorities and order of actions. But this leaves the making of a procedure for actually using the instructions underspecified. The HPW must go through much of the business of converting the instructions into an operational procedure as he confronts each move.

What else does the HPW notice? The incremental recursive program (1) can be taken as the raw material for organizing play. But efforts to apply instructions (2) push toward anticipating needs beyond just the immediate one or two moves of the basic incremental program. For example, instructions (s) and (u) illustrate general anticipatory recommendations. The more specifically one can anticipate later moves, the more finely the uses of (s) and (u) can be modified to guide a higher quality game. If conditions are right, as in (t) and (v), (s) and (u) are best reversed. But to apply (t) and (v), in contrast to (s) and (u), a great deal more specific knowledge of where tokens may be in the future is required.

Assigned his incremental procedure, HPW will approach a task characterized by very many branches at each of very many nodes of a move tree. The tree will be pruned somewhat by the structures of sc's emerging during play and by use of the set of instructions with its rules of thumb. But the possibilities for combining moves into alternative paths through the problem space are still incalculably[1] many until very late in the game. More structure (and limitation of possibilities) can be introduced into an INC game by changing the effective instructions: interpreting some instructions rigidly, omitting others, or introducing new ones. But, so long as the game remains primarily INC, most operations rigidifying or curtailing the effective instructions are likely to compromise the custom tailoring of moves to current boards. The compromise will cost something. And it may not permit enough benefit, in the way of producing a better game, to offset the loss.

Let us review for a moment some of what the HPW, confronting successive states of the game, will have to do. He will have to interpret the basic incremental program and its collection of modifying instructions, readjusting and combining them to make a new executable program for assessing and carrying out each move. His attempts to apply instructions for playing a good incremental game will push agains difficulties inherent in using an incremental framework in a task like Shimoku. The instructions will guide HPW to improve the quality of each move by relating it to the (only vaguely perceptible) characteristics of the whole INC game. HPW will try

[1]For a human-like problem worker with limited time.

to take into account difficult-to-predict future moves whose requirements need to be coordinated with any current candidate move. The task of composing a program which includes a maximum of evaluation and look ahead, given an average of two minutes per move, is a formidable one for an information handler with human-like immediate processing capabilities.

If the reader has by now tried a bit of the activity required to assess incremental moves, he has a notion of the nature of the task. If he has considered each of the instructions and tried to compose them into a context-appropriate program to guide his every move assessment, he has done a good deal of information processing. If he has assessed many candidate tokens for a given move; searched for breakable sc's on paths through the home cell of a candidate move token; and tried to look ahead to anticipate blockage of future sc's, depletions of key tokens, and the interaction along several paths of each token--at home or moved--with others not yet in place, he has done a lot of information processing.

The limits and dimensions of human immediate processing capability are not fully worked out. They may differ depending upon how much is treated in a visual store and how much in a linear, verbal, or symbolic form. But we do know that the capabilities involved place severe limits on what a creature endowed with them can do. The amount of information a human can keep immediately in attention and available for active use in processing a move evaluation is related to the way the information is structured. If George Miller (1956) is right, the range for many symbolic tasks is 7 ± 2 chunks (where a chunk is a group or substructure of informational items that can be treated, for purposes of processing at a given instant, as a unit).

Let us look further at the nature of an INC problem worker's immediate processing needs. In the Shimoku task, visual processing must cope with all or parts of an array of 64 tokens having attributes of shape and number, with their current distribution over 64 cell positions and the reservoir, and with the relations in which they can participate to make four-in-a-row sc's using 104 different paths through the cube. The array is always available on the display scope in its current state, and can be referred to. It functions as an external memory of the state of the game.

But the display cannot be made at once to show, for comparison, the consequences of taking different alternative paths several nodes down into the move tree. Such simultaneous presentations of alternative arrays[1] would help an incrementalist to use the display to look ahead. In contrast, the actually available look-ahead computer option, LKA, permits a player only to develop a single move sequence at a time, paying 1/2 point per move for the privilege of keeping the sequence tentative. HPW, in imitation of our INC-h subjects, decides not to use LKA.

[1]Or, perhaps, quickly and easily alternated.

Working in the head, to permit comparison of alternative move strings, the different arrays would have to be projected, each envisioned with proposed transformations of the scope board. Then with the projections held in attention, strings of inferences would have to be made for HPW to compare the merits of the alternatives. If the transforming moves are not easily grouped, and their number exceeds about two in each compared array, the job may be impossible (or so subject to error and so time consuming as to be impractical) for one with human immediate processor capabilities. Thus, HPW tends to abandon attempts at really careful, more or less simultaneous comparison of two or more projected move sequences[1]. Instead, HPW is likely to favor projecting a single short sequence at a time to see whether it appears to satisfy standards developed from the rules of thumb. If one does not satisfy, another short sequence will be projected.

Again, the degree to which the HPW can see ahead on any single, short proposed move path will depend on how much processing goes on in selecting and evaluating each move of the path. The more the evaluating and the poorer the chunking, the more the load on the immediate processor and the shorter the move series that can be accumulated and retained while evaluation of the next potential move goes on. For coherent, usable evaluation, an upper bound of 4 ± 1 examined future moves is expected for the HPW.

HPW INC-h will require great concentration, alertness, ability to keep track of his place in a projected move series and to chunk and deal methodically with information both in the program forming and search parts of the move-making task. Chunking is easier wherever there is hierarchical structure in the analyzing program or in the problem analyzed. But it is hard to build quickly a hierarchical structure which will guide really good INC-h play. With the many tradeoffs, ground keeps shifting. It is hard to develop fixed specialist subroutines that can be run off quickly. Slowly, new procedures reinterpreting general statements must be continually composed. But little mileage can be got from each. The amount of processing necessary at the level of each move is such as to press even a very competent time-limited player.

---

[1] Given the game limits, it would be prohibitively cumbersome and time consuming for HPW to use scratch paper to draw even parts of the board configurations to be developed by alternative more sequences.

Some Objective Requirements of A/MP-c Games. Let us contrast the foregoing with the task of our second fictitious Human Problem Worker, the one assigned an Arrived Stage Master Planned-classic (A/MP-c) instruction set. Here a hierarchical structure can guide play, the grounds do not shift continually, specialist move-making subroutines can be run off quickly, and new procedures need not be continually composed. The player does relatively little on-line processing, and time is in good supply.

What is the instruction set we give to HPW:MP-c? The key portion deals with a diagram. Either we give an MP-c B_f diagram directly to the HPW, or we give instructions on how to generate such a diagram[1].

Thus, HPW receives either[2]

(3a) Given any B_i, transform it without purchasing[3] and in not more than 30 moves into the B_f diagrammed in Figure , or its equivalent.

OR

(3b) 1) Draw a B_f diagram such that two 4x4 planes are completely filled by H-sc's made from tokens in the set found on any B_i. The two planes are to intersect to share an sc.

Half the sc's are to be four-of-a-kind/4-shapes sc's, the remainder are to be straight flushes. All tokens in the main design must participate in at least 3 sc's. The four tokens outside the main design are to make 1, 4k/4c sc.

---

[1] HPW, like our subjects, knows from the game description that all B_i's that may be received will have a complete 32-token set distributed somehow on the four grids.

[2] We use both (3a) and (3b) because we have seen both basic versions in subjects' games. Good examples are I/NQ:2 for (3b), (an Arrived game in the sense that I/NQ had figured out the basic method and constraints for developing the diagram before coming to play); and I/NQ:3 and I/EWB:4 for (3a). (I/EWB:4 was the only fourth game permitted on that subject's request after his abortive execution of I/EWB:3, mentioned earlier.)

[3] The pure type MP-c game does not purchase. This program is a composite of the pure type. A couple of games classed as MP-c do purchase a little, and their programs are slightly different thereafter.

Given (3b, 1), actually drawing the diagram is easy and not sufficiently problematical for HPW to warrant more attention here.

The supplementary instructions in the set are contained in the following program.

(4)

a) Compare B_i with your diagrammed B_f and orient the process of construction which will realize the main B_f design within the cube so as to retain as many tokens as you can in their original cells.

b) Give priority to conserving moves over conserving move costs: do not slide twice to effect a jump.

c) If applying (1) does not permit retaining at least 2 tokens in original positions, give priority in executing all subsequent instructions to making moves necessary to complete the main design[1].

d) If a non-main-design token blocks a main-design cell, move it as soon as its B_f position is empty.

e) Search the board for each pair of tokens which are currently occupying each other's desired B_f positions, and exchange them.

f) Take each token remaining in a non-B_j position into its empty B_f position[2].

---

[1]There will be a move shortage, failing savings on exchanges, since 32 tokens must be in precisely determined places to complete B_f. Only 28 must be in place to complete the main design. The highest returns per move come from completing the main design, because of the high degree of high-quality sc interaction.

[2]Execution of the instruction performs the further sequencing necessary: tokens will be moved first which can be put into their emply B_f positions.

Clearly, the arrived stage MP-c instruction set given to the HPW is very different from that given to HPW:INC-h. The instructions are ordered. They are in or, relatively, close to operational form. We do not see needs for INC-h-like balancing and trading off of numerous, complexly interacting prescriptions. The priorities are clear, as are the rules for dealing with them. The requirement of (4a) is perhaps trickiest to execute, since it involves trying to fit the B_f diagram on B_i in different orientations in three dimensions. However, it is not a demanding prescription--there is leeway in its application for most B_i's, and for the actual B_i we used. Either (a) several different orientations of equivalent B_f's would be easily adequate to conserve the minimum two moves, or (b) possible exchanges would be sufficient to conserve them, or both. Essentially no envisioning of projected moves independent of a written external memory is required. Barring truly gross sloppiness causing errors in executing the instructions, there is no danger here, as there is in INC or AMP games, that sc's will be broken. There is no need for on-line evaluation of proposed moves in relation to current or future sc interactions, token depletions, and so on.

Thus, there is no push to overload the HPW's immediate processor in attempting to follow the A/MP-c instructions. There is little likelihood of wasting large amounts of time and effort by overloading, and then losing track of one's place, or becoming confused and forgetting parts of an unwritten move series evaluation or plan. There is little need for information processing calisthenics to stretch immediate processing capabilities with super-skillful chunking.

Instead of all that, the main requirement for the HPW:A/MP-c is fastidious attention to the precise execution of moves, to generate the diagram without exeeeding the full complement of 30. Potentially disastrous, inaccurately placed moves which are not caught immediately and unmoved may compromise the game badly (as we have seen with games I/KD:3 and I/EWB:3, the lapsed MP games).

Having seen some of the information processing requirement of play for HPW's starting with A/INC-h and A/MP-c type instructions, let us turn to the remaining main type. This one calls on all the skills needed for both types examined so far.

<u>Some Objective Requirements of A/AMP-c Games</u>. The classic, Arrived stage Adaptive Master Planned (A/AMP-c), game goes farther than the A/MP-c type in attempting to conserve moves by retaining B_i tokens in their home cells while making them participate in sc's. The modification of the A/AMP-c main design to adapt it to a given B_i is quite extensive, compared to A/MP-c. At the same time, it is still much more ordered and hierarchical than the A/INC-h game, being controlled according to a main design. The incremental sort of processing necessary to effect the adaptation is always guided by and referred to a basically preset structural scheme.

The written scheme permits the player to achieve many of the desirable gains of an effective look ahead. The HPW's immediate processor can employ his on-line capabilities in conjunction with the over-all, externally stored master plan. To the extent that it is operating incrementally, HPW's immediate processor is partly unburdened by the externally stored plan. On-line, therefore, it can take into account more of the unpredicted attributes of the board, moment by moment. It can assess more effectively the requirements for coordination of local conditions and the needs of future moves.

What sort of instruction set is given to HPW:A/AMP-c to prescribe the game type play?[1]

It starts with

(5) Given any B_i, in 30 moves[2], transform it into an approximation of the B_f. B_f consists of a main design of H-sc's, straights with 4 shapes, and straight flushes consisting of 32 tokens arranged in two crossed planes running vertically through the cube. The ideal main-design drawing is presented to HPW. B_f also will contain an unspecified number of sc's, most or all to be H-sc's, in addition to, and interacting as much as is feasible with, the main design.

The collection of rules of thumb which guide the interpretation and execution of (5) is

(6) a) Develop a systematic use of the scratch paper so that the main design tokens to be retained in place at each stage of the game, and the value of retaining them are represented, along with indications of certain key, proposed moves which might be forgotten. One main representation should use different colors and over-writing for efficient encoding, and must be kept substantially up to date.

---

[1] Again, as always, HPW knows from the game description that all B_i's that may be encountered will have a complete, 32-token set distributed in some fashion over the four grids.

[2] The adaptation and presence of potential sc's off the main design are intended to permit profitable use of all 30 moves, regardless of the ease of completing the main design.

b) Purchase tokens since they facilitate making a higher total of highly interactive H-sc's.

c) Search B_i visually for all high sc-1's and sc-2's and note their locations on scratch paper. Compare results of search with the requirements of the B_f ideal diagram and, both before starting and during play, adjust token numbers and shapes so that (d) through (i) are heeded.

d) Use exchanges to conserve moves in carrying out (e) through (i) whenever feasible.

e) The principal elements of the main design (all sc's as straights, flushes, and 4 shapes; all tokens participating in at least 3 H-sc's; the plans arranged, as in (5)) are to be substantially preserved when adjusting to B_i conditions.

f) Adjust the main design, and the off-main-design-but-design-interactive sc's, to take advantage of the following property of the 4 X's design: the design assures that preplanned token placements occur in all cells of the cube which are penetrated by the maximum of 7 four-in-a-row paths[1]. Give priority to placing mid-series tokens (3,4,5,6) in 7-path cells to increase chances for their simultaneous participation in the design and in extra, non-main design H-sc's.

g) If they are parts of potential H-sc's and do not block the design, tokens are to be marked for retention in B_i cells.

h) If they are usable in an adjusted main design, tokens can be marked for retention in B_i cells.

i) Need to employ both the tokens marked off in (d) and (f) and the tokens that must be removed from blocking the main design, by placing them in interactive H-sc's, should act as a further constraint determining adjustment of the design.

---

[1]The main design itself cannot assure that more than 3 or 4 H-sc's will pass through the key 7-path cells. The 7-path cells are those in the 4 corner positions of the outer 2 planes, and the 4 central positions of the inner 2 planes.

(7) In executing the game, give priority to moves essential to completing the main design.

(8) In addition, explicitly or implicitly, the AMP-c instruction set is to be taken to include almost the entire collection of statements modifying the application of the recursive incremental program of the HPW:INC-h game: Item (2,a) of the INC-h instructions, suggesting limitation of visual search to areas defined by easily perceived paths, is handled in a different way here. Having the cube basically prestructured by the 4 X's of the main design, with its known critically important paths, permits more rational limiting and ordering of search. This is because the properties of many token positions in relation to the main part of the desired B_f are known. INC-h instructions (2,b through s, u through dd, and ff) also can be thought of as absorbed into the AMP-c set. (We shall consider that the senses of the INC-h rules of thumb, with the changes noted, are presented to HPW:AMP-c, but shall not repeat them here.)

We see the substantial inclusion of the sense of (2)--the INC-h rules of thumb for adapting moves to local board conditions--in the instruction set of the AMP-c game. In contrast, the incremental recursive program (1), dominant in guiding INC-h play, can be recognized in only a weak and much-modified, plan-constrained form in AMP-c--for some of the tacking on of non-design late-game sc's.

The idea of a substantially prestructured goal state in AMP-c, and some of the principles for achieving it, resemble MP-c's instruction set, (3a,b) and (4). A/AMP-c is a game which integrates many of the prominent features of both A/INC-h and A/MP-c games, while avoiding some of the costs of inching incrementalism, on the one hand, and brittle imposition, on the other.

What is demanded of the information processing capabilities of the HPW playing A/AMP-c? Certainly, the fastidiousness and precision in execution of the MP-c player. But, more than that, we are back to extensive needs for on-line immediate processing. Executable programs again must be made for single moves which are composed in part from instructions interpreted

and balanced in relation to the board. The rules of thumb are, if anything, more demanding than in INC-h in their imposition of interaction-promoting and move-conserving constraints which must be related to building a main structure. They urge continual relating of on-line first-formulated or reformulated local moves to the on-line readjusted whole. They are imposed requirements for disciplined and thorough look ahead.

If HPW takes all the instructions and tries to cope with them, once again the immediate processor will be overloaded in assessing many of the possible moves. As in the INC-h game, compromises and limits will have to be discovered during play and in relation to the information processing requirements of individual concrete situations.

However, in AMP-c, the presence of a main structure helps to establish more coherent and stable bases for interpretation, reference, precedence, and coordination. Permanent features of the planned board can be used to advantage. The structure that accumulates during play clarifies the board's key relations. In contrast, in many INC-h games, the structure developing during play obscures the nature of many relations[1].

With the AMP-c game, the main design is rationally related to the properties of the cube. The cells which offer seven path opportunities for making sc's are identified and given priority use.

The instruction set for the A/AMP-c game shows recognition of the need to cope with the load on the immediate processor, in its prescription for a quickly built and reliable external memory. The INC-h HPW had instructions for a limited form, an optional "valued location" diagram. The MP-c's external, constructed memory was the rigid design guide. Once settled, it never needed updating and had simply to be followed without change. It cut the on-line processing to a minimum, but at a cost in adaptiveness. The HPW:AMP-c starts with an initial ideal diagram, as in instruction (5). HPW uses it to make a new working diagram adapted to B_1 in an early set of adjustments, and thereafter keeps updating and reevaluating positions to get the most benefit for guiding adaptive play.

Previously, we have seen that complete redrawing of different stages and different alternatives is cumbersome and too time-consuming to be good for INC-h play look-ahead. But here, the giving of a main design idea, which substantially orders the board and play, is followed by an instruction to make a single main working drawing. Because the drawing uses color codes, different conditions can be noted and clarity can be maintained without

---

[1] The main design in MP-c facilitates visual perception of relations. But there this fact matters much less to the quality of the game--relatively little on-line search and projection and evaluation of alternatives occur to be facilitated!

extensive rewriting. This external memory device can function well to improve over-all comprehension of relations on the board. It can support HPW's immediate processor and free more of it for look-ahead evaluation. In contrast to INC-h games, time and effort are better spent on an elaborate sketch in the AMP-c game.

We have been drawing out information processing characteristics of the Arrived stages of main game types. Our purpose has been to try to make sense of the experience patterns of our subjects.

The use of composite instructions to fictitious HPW's has helped us capture some of the requirements for interpreting and executing the three main approaches to the game. HPW's assigned instructions were composites of statements we obtained from subjects or from analysis of their records for each main type. Our subjects said they used the statements they gave us and diagrams they shared with us as guides. But, in the case of general statements with conflicting implications and in the cases of overload and imperfect function, they did not, of course, carry out all the instructions. We have treated the HPW's handling of the instructions in the same fashion. We pointed to ways in which attempts to follow the instructions might founder or succeed--given the Shimoku task structure, the structure added to the execution problem by the game type's approach, and the limitations on human immediate (and perhaps intermediate term) information processing. We have not tried to be exhaustive in the descriptions of game type instruction sets[1]. Nor have we attempted to write all the instruction sets as executable programs. The ways in which rule-of-thumb sections of the instruction sets were used by our subjects varied sufficiently with local conditions and individual subjects, that they are best appreciated by study of case history printouts. Even so, of course, some of the fine detail of mutual regulations among rules of thumb and context cannot be retrieved from subjects.

---

[1] We do not consider them completed theories of problem solving activity in the game.

We have tried to show that key instructions with which a player starts each type of game differ substantially in just those properties that make them usable in particular ways to guide action. Only the MP-c instructions are in the form of a clear-cut, ordered, and relatively easily operationalized program. The INC-h and the AMP-c sets are far from that. And their nature is such that they cannot easily be transformed into as succinct and readily operationalized programs as the MP-c's, if they are to guide the sorts of INC-h and AMP-c games we actually found. That the generating instructions prescribe different actions would have been expected from finding the different games. But we see that the ways in which they prescribe or suggest actions also vary among the three main types. Do the specific instructions, the ways in which they function together, and the objective information-processing requirements they entail for the users have concomitants in the subjective experience patterns of our subjects?

Subjective Experiences of Players Executing Arrived Games

Recall that the notion of Arrived stage games helps us to separate out those parts of the data that are related to executing a game, once one knows pretty well how one views the task and expects to undertake it. For purposes of this section, real Arrived stage subjects are seen to be in situations comparable to those of the hypothetical HPW's interpreting and executing already formulated game type instructions.

Let us take a moment for another bit of orientation to what follows. We use an analysis from the point of view of information processing. Historically and conceptually, it owes much to the advent of the computer.

However, we do not analyze the information-processing aspects of performance in our task merely as though they were carried out by the commonest stereotype of a computer. We do not look upon the subjects' Shimoku games as if they were generated by a device that registers no feeling about what it is undergoing and does not use a developing, over-all sense of its cognitive-affective experience in any undertaking to influence what it will do next. On the contrary, it is crucial to the subjects' problem working that they do those very things.

The phrase "cognitive-affective experience" needs to be emphasized: our language and culture so separate treatments of thought and emotion that a reminder is sometimes necessary. Normal experience in the world finds perception, thought, and feeling thoroughly enmeshed with each other, moment by moment. From now on, no matter what sort of expression we use to speak of a subject's experience, we mean that the experience itself is the inextricable intertwining of perception, thought, and feeling. And we mean that experiencing events--in all aspects of perceiving-thinking-feeling--necessarily entails information processing and can be thought of in information processing terms. We ask the reader to make the corrected interpretation whenever we fall back, as we must if we are to use currently available language, on words that suggest only a single aspect of the process of experiencing.

We return now to the experiences óf our subjects. They played their Arrived stage games substantially as if interpreting and executing the instruction sets we have just seen.

Are there patterns of experience associated with the information processing requirements placed on a human executing the main Arrived games? And, of so, what are they like?

To take the first question: we feel secure in speaking of patterns for the A/INC-h and A/MP-c games, for there is great similarity in the reactions of at least four different individuals in each. In the cast of the A/AMP-c game, however, there is just the one outstanding subject.

Subjective Experience Patterns in A/INC-h Play. Aspects of the Arrived stage playing experience discussed here are characteristic not only of INC-h subjects, but of players with high INC-m scores (in the 80s) in their Arrived stage games.

Subjects felt the game necessitated strong concentration. But even with strong concentration, they were severely taxed by the job of taking into account all the policies they had developed and all the relevant assessments of candidate moves in terms of current and projected incremental structure. It was impossible to do it all well under the circumstances.

Yet, on trying to assess their limitations and to avoid fruitless strain, players often had trouble calibrating the information-processing requirements of a given move assessment against their capabilities and available time. They still tended, in Arrived games, to overextend themselves in search one place and to do less analysis than they profitable could have done in another. They tried to stretch in considering alternative move sequences but very quickly ran into points of diminishing return. Although it occurred less often and less severely in Arrived stage games than in earlier games for a given subject, loss of earlier parts of an analysis while pursuing later parts was still, at times, discouraging and disruptive. It was especially so for those who took pride in their abilities to deal with complexity. Often, they belabored themselves as "going crazy, thinking too much", or as "sloppy, sinking into bad habits". Both executing lower-level instructions to search and test on the board and interpreting and coordinating policies with each other and the board involved on-the-spot information processing. And both interrelated levels of activity were hard to stabilize in relation to their abilities and time.

There tended to be swings and overcorrections at the policy level. Temporarily fixing one policy and ignoring its relation to others was a frequent, but not always very satisfactory, attempt to simplify things and to reduce the move-by-move information-processing load. For example, sections of games, or whole games might be dominated by blanket experimental interpretations of the policies about making early purchases of large investments or deferring them. For

some players this was not so much an intentional exercise in investment policy as an attempt to stabilize the operation of a rule and, thereby, to avoid having to test for its applicability continually. For a discussion of "cognitive economy" as what man tends to do when complexity exceeds his ability to cope with it, see Hormann, 1971a.

Most players felt uncomfortable about, and tried to damp, their unintentional or ill-considered swings. Some did it by developing calibration rules based on their prior games of the same type: it was common to hear of doing enough processing to assess whether a move or a very short (2 or 3) move string would "average me 3 to 4 points per move net"; to hear of considering a proposal passing the test to be sufficiently scrutinized to be executed; or to hear of similar pacing rules for the use of time. INC-h subjects spoke of the great need for efforts to be methodical and for getting control of a process that could easily get out of hand.

One of the troubles--given the time pressure, the need for intense local evaluation, and the absence of a coherent large structure--was a tendency to work excessively with some details and to fail to "come up" to get an overview which would take stock of any emerging structure and of available resources. Actually, the INC-h subjects appear to have handled this better than many poorer INC-h subjects. Still, they were likely to say, "I meant to stop every so often" to assess the situation on the large scale, but "I forgot", or "I felt to pressured" to do so.

After alert and vigorous starts in their early, and in their Arrived games, subjects usually experienced a decline in vitality. Late in the games the A/INC-h players felt pushed to relax their (never strict) H-sc priority search rule (2,g). Arrived stage players generally felt that it was the nature of the game task to plateau and become relatively stale, flat, and unprofitable in the last moves. There was little tendency to see the trend as also a result of their approach to the task.

The motivation to struggle to maximize returns on each incremental move would begin to wane on their seeing limitations on improvement in their Arrived games. Most said they sensed having reached a point where gains to be expected from continued polishing--of search thoroughness, of attempts to look ahead, and to check candidate moves against a list of policies, etc.--would bring little net change in the rewards of their performance. Although their abilities to do one sort of operation or another might still improve, the constraints on immediate processing made doing each bit of assessment a trade-off for another which might be equally important.

Arrived INC-h players said they had on the whole enjoyed the experience of playing three times but would not want to go on: they understood the structure of the game as it could be played and expected to learn nothing new. We shall see a similar response among the A/MP-c players.

Subjective Experience Patterns in A/MP-c Play. As would be expected from the great cut in on-line information processing required to apply A/MP-c instructions as compared to those of A/INC-h, the subjective experiences of pressure, need for intense concentration and energetic application to the task, and the stresses and disappointments of overload found in INC-h are not found here. The poignant sense is absent that tradeoffs in uses of one's own limited capabilities for immediate processing must severely limit aspiration to improve. Since the tendencies for sc's to block each other, and for the game structure developed in play to plateau in profitability, are absent, the sense of struggle for diminishing returns is absent as well.

Once they had the diagrams worked out, players experienced execution as a mechanical task. Subjects spoke of having "formalized" or "algorithmized" the game--with a sense of mastery followed immediately by let-down. They felt it was disappointing to see that the "real nature of the game" didn't meet their expectations.

The simple processes of ordered token finding and position changing to match the diagram used only a narrow band of their cognitive capacities. There were no well remembered sequences of difficult sc coordinations, no critical moves that deserved review. Similarly, the A/MP-c players were less involved personally. During the INC-h game's execution more of the whole person was involved. Wherever complex efforts to follow under-specified instructions and judgements must be made under conditions of high motivation and time and resource pressure, much more of the self is called into action. One pushes to outreach one's self for the brilliant and elegant. Or, one warns one's self to assure expectable, respectable progress. And so on. In contract to INC-h, the experience of executing an A/MP-c game involves none of the rise and fall of aspiration, none of the complex balancing and judging that rouses large chunks of a personality. A/MP-c subjects did describe both themselves during execution and their methods as "aggressive" and "insensitive."

All MP-c players had played at least first games in INC. Two (I/NQ and I/EWB) with the purest A/MP-c games had scored below 50 in round 1 INC play, which was undistinguished. They felt INC play required a certain brilliance to perform really well, while this did not.

They felt certain that their (essentially equivalent) diagrams were "optimal configurations" for the game--given a human player, and the time pressures, move limits, and an unknown opening configuration. Improvement was likely to be trivial--one might at best save a couple of points more in orienting the diagram to the B_i or in finding exchanges.

Subjective Experiences in A/AMP-c Play. Subject KN's experience in game 3 appears to have been very closely shaped by the information-processing requirements of the task as his plan structured it.

Intense concentration, attempts to study and get a sense of active areas of the board that might interact with the main diagram, the sense of stretch and load on immediate processing capabilities during move assessment and look ahead were characteristic of his play. The careful use of his more elaborately coded game diagram enabled him to free himself to dive into detail for quite thorough and intensive evaluation of each board situation and still not lose his place. He felt that his handling of policies and their adjustment to context and his search were of quite high quality. Although still taxing, they did not often result in a sense of frustration and waste. He experienced the game as exhilarating and responsive to his approach, and he had no leveling off of feeling or motivation. He regarded it as wholly involving and not merely a mechanical execution. Despite his having a master plan, the continued exercise required to adapt it as well as he could kept it pleasing and interesting. Despite his requiring extensive incrementalist sorts of processing at the local move level, the presence of a main plan gave him enough structure to permit both balancing policies and quite thorough searching, with a sense of being effectual.

In the interview, he remembered structures and tokens; he spoke accurately of the errors he had made and ways he could improve the game a little. He thought that, with those improvements, he would have taken into account as many of the exploitable game conditions as possible--but emphasized that he was by no means sure.

Relations between Objective Requirements and Subjective Experience in Arrived Games

We emphasize the importance for the human problem worker of the relations between the objective information-processing requirements of a task and the subjective experience in performing it. The amounts and kinds of information processing that occur in work on tasks, pushing toward overload and requiring a great deal of interpretation and judgment, seem very sensitive to the over-all sense a subject has of how he is doing--how competently he is performing, how much he is enjoying it, and how much he hopes that inputs of great effort will be productive and worthwhile. In the absence of an essentially algorithmic approach as in MP-c, the INC-h and AMP-c subjects use the feedback of information drawn from their entire subjective experience to influence the interpretation and application of strategy.[1]

---

[1] Execution of an MP-c procedure has fewer ways to be influenced by the player's sense of the requirements, value, and promise of his effort. But, after satisfaction at the achievement of the master idea, I/HC said his sense of letdown and dislike of mere diagram-following led to lapses in his execution of game 3, his second MP game.

We have noted both coping and the strains on coping in Arrived stage games. We have seen that it is easy for the MP-c player to carry out his highly specified, largely predetermined intentions. It is more taxing, but still satisfying and highly profitable, for the AMP-c player to interpret his rules of thumb and to develop them into parts of an on-the-spot program for executing an adapted version of his sketch. His elaborate working drawing, and many other features of his play, show a fairly well worked out knowledge of his own capacities in relation to his formulation of the task and an ability to enhance their effectiveness by assisting them with invented devices. He has worked to relieve his immediate processor of whatever can be handled elsewhere, thereby permitting his on-the-spot, locally adaptive decision-making to take a lot into account without swamping him. The Arrived stage INC-h people did not fare quite so well, but they still achieved a partial peace between their conceptions of the game's requirements and of their resources. They felt they knew pretty much what to expect over-all from their application of their procedures to the task as they saw it.

Arrived Games and the Learning Process

The notions of the Arrived stage game and the hypothetical HPW, who had to interpret and execute instructions to play such a game have let us separate two aspects of our real subjects' activities. They helped us distinguish learning how to play--ascertaining the nature of the problem and developing a procedure for dealing with it--from executing play once one has the formulation.[1]

So far, we have chosen mature games for analysis. They have been treated as if little or no formulating of their nature was going on during play. When and how did people develop what ideas they had of the nature of the task? How did the problem-working process develop? How was that process related to its products--the performances and experiences of the problem workers? In the sections that follow we try to shed a little light on these questions.

The Developing Problem Working Process

So far, we have not treated the processes of becoming players of the types of games we have found. To go on, we shall find it useful to consider issues of problem representation and modeling.

---

[1] It is true, especially for AMP-c and INC-h Arrived games, that the real subjects were still learning in a sense--seeing how to adjust the requirements of execution to make the game most effective. But indeed, they had started their Arrived games with formulations of the sort seen in the three instruction sets. The distinction is a relative one.

Representation and the problem working problem

A key statement to introduce this section comes from a mathematician subject, I/HS: "What I am up against here is not really an abstract problem, it's a practical problem."[1] For him, an abstract problem was one whose well-defined abstract features were the sole matter of importance. He saw it as something for which one had, in principle, unlimited time. And it was one in which the problem working process was a relatively unimportant incidental, being of no interest in itself. Whether one solved the problem or failed to, it was not generally worth considering any properties of the human processor--the problem worker--as part of the problem[2]. He spoke of a practical problem as one in which conditions usually would not permit exploration and appreciation of all the potentially important abstract properties it might have. A practical problem was one in which allocation of one's information-processing resources in relation to all or part of the abstract structure of the task and the limitations on time was "the real problem."

We have set up the Shimoku experiment to try to elicit, in a controlled and relatively simple situation, many of the kinds of behaviors that are important in real-life work on complex, unfolding problem situations. There are many different facets of work related to competent problem-oriented behavior. People having to deal with planning and design problems--in the broadest senses of the terms--frequently must work with more complexity, and less ability to anticipate future requirements which should be coordinated with present actions, than they can handle. The constraints on them come from information processing costs and time pressures, and from difficulties in the structures of unfamiliar problems, inter alia. Their "real problems," then, are not just some day, at some cost, to find solutions to what abstract properties the target, problematical situations may possess. Their "real

[1]Of course, this view runs into difficulty in the area of problems which are suspected to be, but cannot be proved to be unsolvable. The question is raised whether unsolvability is part of the abstract characteristics of the problem, or merely an attribute projected onto the "objective" problem by an insufficiently adept solver. But, in this tradition, relatively little explicit attention is given to characterizing the processes of attempted solution.

[2]Operations research has made its business the transforming of "practical" problems involving embedded abstract content, in I/HS's sense, and resource allocation and time use into more fully abstract problems. The distinction between abstract and practical problems is not one which can be tightly maintained. As we shall see, the abilities of subjects to get some clear notions of their processing is a job that requires some formulation. Such a model may, in turn, have abstract properties.

problems," like those of Shimoku subjects, are to find out how they can cope with as much of the target situation problem as they can, under the practical pressures of action and decision.

In considering the issues of problem representation for players of Shimoku, therefore, we want to consider not simply whatever static representation of the abstract features of the game they may have used. We shall want to consider also the ways in which subjects' actions--and their plans and representations for actions--were related to their representations of the static structure of the task. The way has been prepared by our demonstration of (a) parts of static task representations given in the INC-h, MP-c, and AMP-c instruction sets, (b) by the representations in programs and rules of thumb of actions to be used to deal with the static representations, and (c) the actual execution requirements posed for an on-line, limited human information processor trying to interpret and execute them. But we have not dealt explicitly with issues critical to the development of the different game representations.

Representation of Static Structure and of Dynamic Execution Aspects of the Problem. The ways of speaking of "abstract," "practical," and "real" problems suggested in the foregoing will not carry us far. Let us stop for a moment to develop a more differentiated view.

If someone says to you, "Here is problem X. What is the real nature of this problem? what might you answer? You are likely immediately to take the problem as cast in terms of one, or possibly several, domains of your knowledge, suggested by the representation in the statement. Your answer may state, or tacitly assume the questioner knows, that X is a problem in group theory or in membrane biophysics or in diagnosis of jalopy illnesses. You may go on to elaborate and interpret the problem statement in terms of the entities and relations in that domain.

If he says, "Ah, but what is the real nature of the problem, apart from the specialist jargon and world view of the biophysicist or the mechanic?" What then? Perhaps you would search for an alternative problem representation in a domain of knowledge which may be equally special, but more familiar to your questioner and try to elaborate it there.

Or, you might try to find an equivalent statement, however cumbersome, in everyday language and related to the domain of everyday knowledge. Seeing the problem cast in a more familiar domain, or seeing the entire correspondence between the two versions, may give your questioner an enhanced sense of knowing something about the problem's real nature. But it is hard to say that the one or the other statement is more real or wherein lies the "true nature" of the problem, apart from the minds of individuals and of their fellows in the larger community of problem workers.

If you transform the problem to an alternative statement related to the same or another domain, you may call it the "same" problem if you can somehow demonstrate the equivalence, with respect to a standard, of the new version of the problem to the old. Certain relations within the old must be preserved in the new. But the second statement may require or permit a wholly different program of approach to elaboration and computation. Thinking about "the" problem in one representation and its related domain may not be the same as thinking about "it" in the other. A restatement which is "equivalent" for some purposes may not be for others. One may sense having illuminated the "real nature" of some aspects of the problem (often its static, abstract structure) by a mapping into a more easily apprehended domain. But it is likely that "the real nature" of other aspects (often, its computational requirements for a problem worker) have been changed.

In the case of Shimoku, all subjects who went from Incremental to Master-Planned play transformed their statements of the problem. By working to specify the abstract static B_f diagram, the MP players felt they had found the "real nature" of the problem. The execution task, being well within their information processing capabilities, was markedly changed by the representation and became no longer genuinely problematical for them. In contrast, for good INC players a more detailed, differentiated conceptualization of the static, final abstract structure was taken as not feasible or desirable in the circumstances, and therefore was not seen as problematical. For them, the most problematical part was to meet the demands for dynamic processing in the on-line interpretation and execution of their basically correct self-instructions for the incremental game.

Especially since we shall be dealing with the inner views of various players who see the game and what is genuinely problematic about playing it quite differently, it will be important to be specific.

We shall use the abbreviation, Pr S, for "problem statement".

Pr S	the static abstract aspect of the Shimoku problem as stated in the Experimenter's instructions and the rules of the game and constrained by the objective characteristics of the board and tokens.[1]
Pr INC MP AMP	the static abstract aspect of the Shimoku problem as stated and elaborated characteristically by players of the game types.

[1] The Pr S given to the subject says nothing about desirability of modes of play or about how the goal statement, "make as many points as you can in 30 moves or 60 minutes, whichever comes first," is to be interpreted.

For example, if we speak of Pr INC-h, we shall mean the typical or composite INC-h subject's statement of the static abstract aspect of the Shimoku problem at the start of an INC-h game: his conception of the rules, the properties of the token sets and boards, and of the goal--as well as the rest of the knowledge he has elaborated about how those things are related to each other, and to his other, closely game-related knowledge. (By the last, we mean such things as knowledge of the numbers 1 through 8, abilities to recognize and distinguish shapes, to visualize a cube and its mapping into two dimensions, etc.). We shall say that "XXXX" is true of Pr INC-h if, at start of an INC-h game, it was true of the problem as stated in the INC-h player's representation.

We shall want to know whether the player was capable of dealing with a given representation--whether he was capable of relating it to the necessary domain of his knowledge, and how well he could judge whether he was capable of performing the computations and actions necessary to deal with the problem and achieve his version of the goal in the time available. How does he work to elaborate the static problem statement in terms of the on-line interpretation and execution constraints? If we ask this, we are asking how accepting or building the static problem statement is influenced by the subject's estimation of the execution problem. When we want to refer primarily to the problem of planning and achieving the dynamic on-line interpretation and execution, we shall put an E (for Execution) before the static problem notation. Thus, E Pr INC-h refers to the player's representation in an INC-h game of the problem of realizing his static conception dynamically. Subjects in some game types went much further than others in articulating instructions to themselves for devising representations of static relations in the solution structure and for specifying procedures for execution of solutions.

Representation and the Main Game Types. We have presented enough to suggest that subjects were using different problem representations to guide them in playing INC, MP, and AMP games. But we have said little about how INC, MP, and AMP representations are related to the original problem statement. In this section we shall compare the tasks of getting INC, MP, and AMP problem representations, starting with what is explicitly given to the subject. Notions of Amarel (1971) on the issue of representation in forming programs to solve derivation and formation types of problems will be used.

The treatment will try to develop deeper descriptions of some of the Shimoku events which would ordinarily be labelled with the standard problem solving literature notions of "set," "overcoming set," "insight," and the like.

To consider the issues of problem representation for the Shimoku players, we must look again at a key part of any representation: the goal statement. The goal statement given in Pr S says, "make as many points as you can in 30

moves or 60 minutes, whichever comes first."[1] Given the exact meaning of the Pr S goal statement, we see that any score attained by someone who made as many points as he could qualified technically as a "solution" to the problem posed by the game. Our subjects' scores are a collection of solutions. It is convenient to think of scores as solutions to the statically defined aspect of the problem stated in Pr S. The dynamic processes whose (partial) traces appear in their printouts and sketches are the closely related, concrete solutions--realizations of the on-line program formation and control program execution aspects of the problem representation Pr, E Pr. We can also think of the scores as performance measures of the execution aspect of the problem-working process.

We think of problem working as a goal-oriented activity. In general, the more we know of conditions which the realized goal must meet, the better we can guide the problem-working process.

The Shimoku subject is introduced to the game as a study in problem working. He enters an entirely unfamiliar task environment. His chances for preplay analysis of the situation are only those he can snatch during the insturction period. He finds himself on-line in game 1. Toward what is he to aim?

To most subjects the goal as stated seemed very amorphous. It was hard to get bearings on what score to work toward, what to expect of their procedures in playing, and what to use in guiding the formation of those procedure. Over the three-game sequence, subjects had to develop for themselves further conditions with which to specify new versions of the goal statement.[2]

First, let us look at how the goal statement is specified and elaborated by good INC players. The language of the Pr S goal statement expresses the goal in terms of an unknown number of points. "Make...points..." is given semantic content directly in the rules of Shimoku: "You make points by making four-in-a-row scoring patterns." The Pr S statement can be restated, without introducing new meaning, as "make as many points as you can by making four-in-

---

[1] The Pr S goal statement is not to "make as many points as is theoretically possible in 30 moves, etc.," a statement which recasts the problem and the status of scores as solutions. That is another problem. And the Pr S goal statement is not "make a total configuration of the final board which will be realizable in 30 moves, etc., to produce as many points as you can."

[2] They could do it on-line, using their concrete experience of ways in which the possibilities of play--their own play only--related to goal states attained. And they could, if they wished, analyze further without the board, off-line: between games, during the questionnaire and interview periods, and at home (without any Shimoku materials, of course).

four-in-a-row scoring patterns." The Pr S statement is substantially accepted "as is" by INC players. It is, for present purposes, about the same as Pr INC.[1]

All of the entities and relations necessary to make scoring patterns are explicitly "givens"[2] in Pr S, which now is also Pr INC. No entirely new major concepts need to be introduced to go from the concepts of the "givens" to the concepts used to represent the concrete situation required to satisfy the goal statement at game's end.

To put it another way: In the version of the problem presented to and accepted by the INC subject, the language of the "givens" and of the goal statement is the same.[3] And the language of INC players' descriptions of the game situation at the end of his 30 incrementally generated moves is the same. The syntax and semantics of the goal statement and end-game description are included in the syntax and semantics of the givens of the initial instruction period game description.

---

[1] The reasonably good INC player may set himself a point range in the goal statement. But ordinarily this occurs only after experience with game 1. He may use it to specify an expected-points-per-move subgoal condition to be applied to each candidate move. He may use the "60 minutes" condition in his version of the goal statement simply as a stop rule, or may use it to place a pacing condition, say, two minutes per move, which will be used to monitor grossly the amount of information processing he does. Further elaboration will start to take in many of the conditions of play that we have seen in (2), page 99. But the concepts in the goal statement of the incremental problem representation are not substantially different from those directly given to the subject in Pr S.

[2] We use the term givens in the same sense as in mathematics, where a situation is laid out: "Given: xxx, yyy, zzz," and a task is presented, "Prove kkkk," or "Find kkkk." The givens are xxx, yyy, zzz. The goal statement (which also is given to the subject, but is not part of the "givens" in this use), is "Prove kkk" or "Find kkk."

[3] The language of the givens is a subset of English precisely defined in terms of the game (and a subset of geometry). The language included, e.g., "scoring pattern," "token," etc., but nothing like "main design," or "main token structure."

The INC player's program formation task--his need to devise a procedure which will start from the givens and arrive at a game result consistent with his INC goal statement--is handled without any novel, game-related representation.[1]

It may be no wonder, then, that the great majority of our subjects' games were INC. To play incrementally, one could take the Pr S directly and try to run with it. But one could not do that for AMP or MP play. Among our subjects, no one who achieved master-range scores (over 110) did so without recasting the representation of the Pr S goal state.

Let us consider the situation of a subject who is dissatisfied with the Pr S goal statement but has yet no idea what an adequate alternative will be. He wants a better-specified goal statement in order to guide himself in generating a procedure to play. He has a new subproblem--devise a procedure for finding a better goal state description. He is in an awkward spot. In what terms should the goal state be specified? "Many points" and "completed scoring patterns" give too little hint.

Polya (1954, 1957, 1965), Amarel (1971), and others have made a distinction between two main types of problems that is relevant here. Using Amarel's terms, the distinction is made between derivation (D) and formation (F) problems.

First, a note to avoid confusion. All Shimoku players have one formation problem--the problem of program formation. As we have mentioned, it is the problem of devising a scheme that can be used as the basis for an executable series of actions in a game, and making the executable program from it. We have called making the executable program the E aspect of problem representation. Devising the program scheme may mean devising a static representation of relations among the structures required for solution of the main Shimoku problem. But some Shimoku players--the MP players--treat Shimoku itself as a formation problem. They then have the problem of program formation for a formation problem. INC players treat Shimoku as a derivation problem and others, have a problem of program formation for a derivation problem. To keep things clear, we shall speak of the problem of program formation when we mean the first formation problem and the F problem when we mean the second.

---

[1] To say that the program formation task is carried out for the INC player using only the language of the givens of the static problem statement Pr S, does not say that the program will be completely specifiable before play brings substrate and context information to each move. Nor does it say that program formation using preliminary formulations and immediate game information will be easy to complete, or that a formed program will be easy to interpret effectively during play. It says only that the game-related terms in which the program formation will be conceived and represented are easily available to the player. Those terms need not first be sought before INC program formation can begin.

Let us become acquainted with the polar D and F types, keeping in mind that they are ends of a spectrum. Our subjects' own INC and MP versions of the Shimoku problem will be seen to be closer to the D and the F ends of the spectrum, respectively. Here is the distinction as seen by Amarel (1971, p.414).

> In problems of derivation type...we are given specific problem conditions in the form of parts of a solution description, and we are asked to complete the description by using given rules for solution construction...Usually, the specific problem conditions specify boundary parts of the solution, and the problem-solving process consists of finding a connecting structure between the given boundaries. It is characteristic of a derivation problem that the specific problem conditions are given in the language of solution structures. A typical derivation problem is the problem of finding a proof to a theorem in a formal system. (Emphasis added.)

In the very simple theorem-proving problem one is provided with boundary parts--an initial situation (the givens) and a final situation (the theorem to be proved). The theorem is accepted as the specification of the state at which reasoning is to terminate. All of the rules for solution construciton--the legal transformations--are provided in the formal system. Their stepwise application completes the solution description by finding a structure of links between the initial situation and the specified end situation. Any new entities which appear en route are generated directly by the application of one or more of the already provided operations to preceding states.

Recall that we are trying to deal not with some hypothetical, absolute "real nature" of the Shimoku task but with the way the task appeared to, and was represented by, different subjects. We have noted that the INC players presented with Pr S accepted something like "make as many points as you can by making as many scoring patterns as you can," as essentially the full specification of a boundary part--the terminal state of a process, to be attained by operations provided in the system, to transform, stepwise, the initial into the final situation. The additional 30 moves, 60 minutes conditions of the Pr S statement were represented by INC subjects essentially as gross "stop rules" which could be used more or less crudely to pace the making of transformations on the initial and succeeding states. They did not serve as problem conditions to be used in elaborating a structural scheme of the terminal state.

To prepare the contrast with MP versions of the Shimoku problem, we borrow from Amarel.[1]

> In formation problems...(the) problem conditions are in the form of properties that the solution as a whole must satisfy...
>
> ...the problem conditions are not given in the language of solution sturctures.
>
> ...no choice of solution element...can be determined directly by the given problem conditions.
>
> The solution process cannot proceed by (direct, stepwise transformation-applying) reasoning from the problem conditions to specific parts of the solution--as is possible in derivation problems.
>
> The general approach here is to generate candidate solutions in the given language of solutions,
>
> and to test them against the problem conditions.
>
> An important problem..is to find ways of bridging the gap between the language of solution structures and the language of conditions.

Let us return to a Shimoku subject who is about to become an MP type player.[2]

> He is dissatisfied with the adequacy of the Pr S goal statement as a guide to program formation--to helping him shape his course of action.
>
> He sees that gaining "as many points as you can" can be better achieved if one builds sc's to interlock and share tokens.
>
> He consciously seeks an answer to the "interlock-maximizing" problem.[3]

---

[1]To condense and to emphasize parts of Amarel's statements, we have at times changed the order and have obviously separated the format. Material in parentheses is added.

[2]What follows is based on interview material from all MP players. Where they diverged on the ways conditions were reintroduced to narrow the candidate solution space, we have followed subjects I/NQ and I/EWB most closely.

[3]We use words like "maximizing" here as a loose shorthand of the sort that subjects told us they used in thinking to themselves. It should be taken to mean "increasing-toward-a-maximum." There is no evidence that a thorough search-and-test for an abstract maximum cccurred. In general, pure MP-c subjects thought it likely that they had achieved not only a maximum of sc interlock for the number of tokens used but for a 30-move game as well.

He introduces a new condition which changes the Pr S goal statement (but keeps it still in the language of the givens): "to make as many points as you can, make as many interlocking sc's as you can."

He must represent the new condition in relation to the rest of his altered goal statement.

An attempt is made to represent the new condition verbally or to incorporate it into an imaged representation of the board as containing interlocking paths. This first effort shows nuggets of organization, as if accumulated by incremental building of individual sc's.

His poorly articulated sense of what a helpful representation of the interlock condition might be is not satisfied by the representation he can achieve, as long as a very strong incremental, adaptive condition is imposed at the same time. The representation achieved seems not to be easily usable to guide his actions to realize the interlock condition.

He puts aside momentarily much of what he thinks he knows about the game, and concentrates on representation of the new condition.

What might a situation look like if it had the "most" interlocks? What would be the properties of representation of candidate solutions which might fulfill the condition "make as many sc's interlock as you can?"

Attempts to answer such questions move the subject's problem into a new space, determined heavily by the currently most problematic condition.

Given the ease of seeing three possible interlocks for every token on a plane, the interlock situation sought is taken to require that each token in B_f ideally be part of at least three sc's.

To guarantee that, a high degree of coordination is needed in placing tokens in relation to each other over the whole board situation.

The subject has progressed to a position to generate a sequence of representations of candidate solutions which are all in a theoretical space of holistic, precoordinated game boards.[1]

Candidate solutions can be progressively articulated, and the space can be narrowed by reintroducing the temporarily neglected game conditions against which they must be tested.

---

[1]The total "problem space" for Shimoku is that of all possible states achievable by all possible move sequences on all possible B_i's. The space with which the subject deals now is greatly limited in size, and the solutions have been limited to candidates having easily representable, highly ordered, descriptions.

For example, the "most" interlocking sc's in the theoretical space would require 64 tokens in the 64 cells. But the number to be placed must deal with the condition of 30 moves.

The 30-move condition, formerly the basis of a gross stop and/or pacing rule, is reintroduced as a strong determiner of the size and constituents of the holistic structure and the relations within it.

The way it operates as a determining condition for narrowing the space of candidate holistic solutions depends on the subject's estimation of the amount of highly coordinated order he can build in 30 moves, starting with an unknown distribution of the 32 tokens in B_i.

All 32 tokens in B_i may be unfavorably placed. The MP player seeks the safety and assurance of achieving very high coordination. He decides that the size of the holistic structure should assure placement of the largest number of tokens into a highly interlocked structure of sc's within 30 moves.

The obvious elaboration of "as many sc's as possible" into "as many H-sc's as possible" is introduced to help determine the positions of tokens in the holistic solution.[1]

The board situation which permits the subject easy representation of interactions is the plane. The easiest plane to see on the scope is a grid plane. Twenty-eight tokens can be arranged into all H-sc's on one grid and another which intersects it so one sc is shared by two planes.

Without going further, we can see the fashion in which the candidate solutions can be articulated and tested against reintroduced problem conditions.

Now let us look back on the process. Nothing in the Pr S givens seemed automatically to entail the notion of a structure of the whole. There was a jump from the givens language of tokens and sc's to a theoretical space in which an interpreted problem condition--taken as a condition strongly

---

[1] I.e., obvious to virtually all subjects. (Much may be "obvious" in the calm of the reader's chair which was not obvious under conditions of action to even very bright subjects.)

determining the whole goal situation--could be represented.[1] This required representation of a structure attainable in 30 moves.[2] The new "language"--not contained in the system of the Pr S--was that of the holistic "large board structure." Once the candidate solutions could be represented in the required (verbal and/or visual language), the procedure converged toward a solution structure. The parts of that structure, arranged according to the structure's requirements, were, conveniently, the parts which were givens in Pr S.

The subject has arrived at a point where the problem is bounded by more satisfactory parts of a solution description. Once a solution representation, meeting all the conditions of the expanded goal statement and the givens, has been achieved, the rest of the F problem becomes a D problem.[3] The rest of the program formation process needed for working the transformed problem can be guided directly. The operations will take the givens of the game and map them into a solution description whose newly structured parts are in the same language as the givens.

---

[1]It is this jump that was associated with subjects' labels of "sudden insight." It is understandable that the introduction of a novel representation is felt as an important discontinuity. Their accounts, however, seemed influenced by popular talk about sudden insights to seem more as if the new representation of the problem had sprung <u>ex nihilo</u> than was the case. If asked to try to relive the transition moments, starting from some prior stable place, and given other suggestions to mull over in an unstructured fashion, most subjects who made the transition to MP were able to retrieve so many closely related steps as to make the transition seem unmystical and inevitable. They found the exercise somewhat surprising. (It is possible that some of the steps are only post hoc constructions in some cases. In others, there was enough material associating thinking of a step to a particular historical event to merit our giving some credence.)

[2]Compare the INC subject: while knowing that the executed situation would, in fact, result from 30 moves, he did not furmulate beforehand a goal statement which represented in any detail the results of 30 moves. The end situation of the executed game would consist of many sc's. But their relation to each other was taken to be not substantially specifiable during elaboration of the meaning of the goal statement. The INC subject's formulation assumed that the need he saw to adapt individual sc-making to the original B_i distribution (and to the successively emerging opportunities of B_t's at each point in the game) precluded realistic attempts to represent the result of 30 moves. His language of representation, in the Pr INC goal statement and its elaboration, is a language of individual sc's, of tokens, points, etc. The language does not include the words or images for a large (30-token-or-so) structure which is first to be represented and then achieved.

[3]The solution may be a specification of a class of equivalent solutions.

The program scheme, the static representation of relations among the entities required for solution of the main Shimoku problem, is the large structure diagram or main design. It is easy to use it to develop an alternative problem-working representation, E Pr MP, which is a control procedure for execution of the scheme on a given B_i.[1] It is so well within the immediate information-processing capabilities of the subject (barring plain sloppiness) as to be unproblematical for him. Because the abstract program scheme formed happened to have such an easy E representation and posed so little difficulty in actual performance of E, the pure MP subject could afford to put little effort into relating the static scheme to E and to action. But is was his discontent at the Pr INC representation's having failed to guide him to build an easily handled execution program that prompted him to seek the MP program scheme at all.[2]

It was not only a desire to be sure to achieve total sc interlock in the main design, but a desire to assure against mishap in their own on-line, adaptive local move tailoring that, they told us, influenced the pure MP-c subjects to narrow the space of candidate solution structures to just those having a totally prespecifiable main design of 28 tokens. Thus, representation of the abstract, static problem was indeed driven and shaped by MP subjects' knowledge of their own information-processing characteristics. The process of formulation of the abstract nature of the problem (its "real" nature, to them) and its program scheme occurred in relation to their model of their own properties as problem solvers.

Let us turn now to issues of representation in Learning and Arrived AMP games. Because the one AMP subject showed continuing development in representation from games 1 through 3, we shall use material from his interviews and game records to trace his progress across all games.

> I/KN never plays an INC game. Before and during the nearly moveless half hour that starts his first game, the approach to representation characteristic of AMP play is taking shape.
>
> He has elaborated the Pr S goal statement during the instruction period, and has introduced as a strong condition a version of the "maximize sc-interlock" condition before reaching the scope.

---

[1]In the fashion of the program given IHPW:MP-c.

[2]This requires qualification: a combination of discontent at INC-type processing usually combined with curiosity to find a "trick" or "puzzle" solution, as we shall see later.

Immediately, on seeing the motley B_i distribution, and remembering that eventual high interlock will pay off, he recalls the Future Three game of Olaf Helmer.[1] In both the analogous situation and this one, he sees a necessity to project ahead from current conditions to an integrated notion of a state of affairs a specific number of moves (or a specific number of event-filled years) in the future.

He sees that scoring pattern interlock--as a condition of the whole result--can best be met by an attempt to "forecast" as well as to plan an end-game state, using a theoretical space of large, high interlock structures.

The high costs of arranging high interlock make H-sc's necessary as products. "High risk, high gain" is the nature of the game.

The 30-move limit is introduced to function as a condition on the set of structures which can represent parts of a goal statement (but in a fashion slightly different from its functioning in MP games).

For use with "30 moves," KN retains as a strong condition "maximize use of tokens in home cells as parts of sc's to make more total sc's."

Thus, in AMP, the 30 move and high interlock conditions are to operate in conjunction with a strong adaptive condition in determining the large structures to be considered as candidate solution representations.[2]

For Pr KN:1,[3] candidate solutions can be represented as within a space of large structures with three-way interlocked H-sc tokens, but their nature is not more closely specified. And they are not tested before the game against all the problem conditions.

---

[1] Institute for Studies of the Future, Middletown, Conn.

[2] At times during I/KN's analysis, the adaptive condition will be partly set aside but will always be reintroduced more strongly than in MP. Comparing goal representations, the MP, dominated by the first two conditions, easily employed a structured scheme; the INC, dominated by the last condition, could not be represented by a simple static program scheme.

[3] Because there is just one subject in AMP, Pr KN:n is the same as Pr AMP:n for n = 1,2,3.

The representation of candidate solutions is of a mixed sort consisting of static, program scheme structure diagrams and modifying verbal conditions which cannot easily be translated before the game into specifications on the whole board.[1]

Game I/KN:1 is taken as an experiment "in my logics", i.e., in KN's relating the requirements of large structure and interlock conditions with those of on-line local adaptation. KN undertakes to remember features of the B_f in terms of their relative satisfaction of his strong conditions, in order to analyze his strategy after the game.

He views the game 1 B_f as itself a holistic representation. It represents the results of his (only partly specifiable) ways of balancing his strong goal statement conditions to produce their determining effects on the outcome.

In game 1, the high-interlock, large structure was interpreted in a single 16-cell grid plane.

The non-main design sc's were carefully built, often from main design-blocking tokens, and of reasonable quality, but could hardly interlock with the main design by making sc's in space: only three tokens had been left through which paths could pass on grid three.[2]

On leaving the scope, KN said, "This isn't really three dimensional: you can play the game...in two. It's a two-dimensional game." He had neglected to take much account of the conditions determined by the cube when forming his problem representation.[3]

---

[1]The verbal conditions will operate in Pr AMP and E Pr AMP. They include conditions to be placed on further developing abstract structure representation, and on the control and execution representation during the game.

[2]Because the grids of the scope display are readily seen as planes, and their within-plane four-in-a-row paths are readily seen together we distinguish "planar" games as those whose sc's are predominantly made within the grids. Four-in-a-row paths which belong to other, non-grid, planes of the cube are not presented with the cells together visually, of course, and must be computed by the player's taking one cell from each grid along the path. We speak of all paths not within displayed single-grid planes as "spatial" paths.

[3]Relative neglect of the three-dimensional aspect of Shimoku was perhaps the commonest simplification--especially in first games. Under-representation of the conditions imposed by the cube is consistent with both I/KN:1's confining of the main design to a grid and with his relatively few sc's in space interlocking with it.

Discontented with the weaknesses of his game 1 representation for guiding him on-line to use the full complement of 30 moves and to interlock non-design sc's with the main design, KN goes to a theoretical space of large structure representations which better fulfill the 30-move and interlock conditions.

While writing questionnaire 1 (immediately post-game) he arrives at a whole board structure, the 4 X's composed of H-sc straights mentioned earlier. (The adaptive condition has influenced his expectation of being able to place more than 30 tokens in the main structure.) It links up all grids of the cube. And it affords good visibility for on-line construction of adaptive local sc's to interlock with the main design in the grid planes.

In forming Pr AM:2 the cube now has been introduced as a strong condition on the candidate solution's main structure, and the structure has been expanded to 32 tokens. The adaptive condition is strongly reapplied to the new version of the program scheme as a verbal prescription to modify its realization and supplementation during execution.

The Pr AMP representation for KN:2 still contains both a static, structural scheme and rules of thumb for adaptation which cannot be integrated with the diagrammatic part of the representation in any easy way.

The next stage of evolution of his representation occurs in the interview following game 2. He comments on his poor use of the main design's central tokens of the middle boards in extra, design-interlocking sc's. ("In tic-tac-toe, diagonals were important and corner cells were controlling...") His preplanning of corner cell tokens gives the outermost board good tack-on interlock opportunities but it did not help much on the inner boards. ("To figure out cell position advantages for a 4x4x4 cube seems too much...") At this point, he sees the correspondence between the path "control" of outer board corner cells and inner board center cells in a 4x4x4 cube.

Thereafter, his representation of candidate solutions marks off the 7-path cells for high-priority planning and placement of key middle number tokens. This completes the abstract large structure part of the solution representation to be used in game 3.

Strong conditions, stated as rules of thumb for modifying the static abstract structure in relation to the unpredictable B_i, remain part of the problem representation.

The application of the rules on-line to complete program formation has, in games 1 and 2, led to immediate-processing overloads. Therefore, for game 3, the verbal conditions limiting search are strengthened further, and search is related more closely to a memory-assisting device.

> Culminating the evolution of ad hoc uses of scratch external memory, an elaborated description specifying a coded working diagram is introduced into the E part of the problem representation. It is both a mnemonic for the static Pr AMP scheme, and a guide for adapting, updating, sc-protecting, etc. Thus, part of the E aspect of the AMP 3 problem representation is an instruction to build during the game a representation to enhance KN's ability to complete program formation and execution on-line.

In the series of Learning and Arrived AMP games, we can trace especially well KN's growing knowledge of his own information processing capabilities and his needs for representation. Their growth occurs in relation to his successive restatements of the problem presented in the Pr S.

KN felt able to retain the potentially profitable--but potentially disastrous--strong adaptive condition despite the recognized chance to get known payoffs by total preplanning. This was partly becuase of a sense of his own relative success at handling demands of on-line processing.

Yet, even he was at times taxed counter-productively. His being taxed prompted finding the 4 X's design--enlarging the number of tokens which could be handled in the main structure. The sense that he could, under the right conditions, profit from the strong adaptive condition influenced not only his specification of the size and sc contents of the main design but also his discovery of seven-path cells. The discovery changed his representation of the abstract properties of the cube.

Thus, learning to cope with the information-processing loads his problem statements placed upon him repeatedly generated subproblems in representation for him. Some of them were solved by further articulation of the properties of a desirable structural scheme, in a theoretical space of candidate solution representations. Some were solved by the invented, external memory representation of the ongoing game's status in relation to his plans.

In Amarel's terms, it is probably correct to call KN's handling of the Shimoku taks an effort to treat it as a quasi-F problem.

> Many important real-life problems can be considered ...quasi-(F) problems.
>
> The position on the scale (between D and F) depends on the amount of theoretical knowledge which is available (to the problem solver) about the structure-performance relationships in the space of solutions, and... on the quality of the...problem-worker's methods for using...(his) knowledge in the solution-finding process.

> In quasi-(F) problems, theoretical knowledge is not complete...there are no perfect ways for using it for guiding problem-solving actions.
>
> To solve problems in these areas we must combine modes of reasoning that take us from problem conditions to parts of solutions, and...from candidate solutions to the problem conditions.
>
> Many examples of such problems exist in engineering and architectural design, in economic planning, in the interpretation of scientific data, and in complex diagnostic tasks.[1]

The Shimoku problem situation can be represented as a D problem with an extremely weakly specified goal statement, determining an incremental strategy that impedes the coordination it tends to seek. It can be represented as an F problem, by a subject emphasizing the strong interlock condition and de-emphasizing adaptation--a problem requiring (a) discovery of a novel representation in language not given with the original Pr S; (b) generating, in that language, candidate solutions which can structurally express the problem conditions placed on the solution as a whole; (c) testing and articulating the class of candidate solutions against all the problem conditions in order to converge toward an acceptable solution; and (d) finding a representation of the program scheme in a control program which can execute it, in the form of a D problem transforming parts of the givens into parts of the newly structured solution.

It can be represented as a quasi-F problem by a subject whose statement insists on retaining both the strong adaptive and strong interlock-maximizing conditions and yet, temporarily sets each aside during formulation in order to apprehend and to represent the theoretical solution requirements of the other.

It is interesting that the different approaches to Shimoku and the kinds of representation-making and representation-using efforts they entail are similar to those occurring in urban planning (where most problem-working situations are probably best treated as quasi-F). There the burdens and costs of information processing are often more than humans know how to handle,

---

[1] Amarel (1971, p.415). We have changed the format, added emphasis, and inserted the contents in parentheses. "Quasi-F" is our version of Amarel's "quasi-formation," preserving the distinction we established earlier. Amarel's text tends to treat "existing methods" and available knowledge" as though it is assumed they are or could be the same as the "problem-worker's methods" and "knowledge available to the problem worker." For subjects such as ours, with situational limits on time and on sources of knowledge, qualification is necessary. In considering real-world practice, it may be necessary much of the time.

and the compromises in problem representation--in the formulation of problem statements which recognize some strong conditions for a solution and ignore others as too unwieldy--often have been disastrous.

It has been perfectly possible, in many urban planning circumstances, to appear justified in choosing a problem representation which is primarily incremental or in choosing one that is rigidly prespecified and to say that either corresponds to the needs posed by the true nature of the problem. The former foregoes the attempt to conceive a theoretical solution space dominated by one or several strong conditions even temporarily and may miss an opportunity to articulate some of the structural requirements of a good solution. The latter assumes that attempts to meet the detailed conditions necessary for adaptation will preclude the coordination essential to an adequate solution.

Frequently, neither planner is particularly aware that he is making a decision which deeply involves his model of his own information-processing capabilities (and those of others who will shepherd action)--in relation to the problem conditions of the task.

In Shimoku play, the work to form a program to cope with, and retain as strong determiners, the conflicting conditions of adapting of found structure and high coordination of new structure was best carried out by subject I/KN, who was especially well aware both of the representation issue and the models of himself and the task that he used.

Issues of representation and modeling are intertwined. We shall develop them now in relation to a more inclusive description of the modeling events in Shimoku subjects' problem-working processes.

Aspects of Representation and Modeling. A feature of the Shimoku task given our subjects was that learning to work the problem and working it for the record occurred at the same time.[1] At least during the first two games, and for many subjects during the third game as well, the task had those two interlocking aspects.

We have distinguished Arrived and Learning games to separate conceptually some of the information-processing aspects of playing according to an already learned procedure from playing while learning it. As it can be applied to the actual experimental situation, the distinction is relative. It prompts us to look further now at the task of learning while playing (and learning in the inter-game intervals).

---

[1]This is like real-life problem working situations and less like most traditional laboratory psychology experiments where even much simpler tasks tend to be drilled and fully absorbed before the recorded experiment.

The issues of representation which we have laid out in the preceding section are critical to the learning process. We have delineated some of the characteristics of INC, MP, and AMP problem representations. And we have shown some of the steps necessary to arrive at them starting from the Pr S presented to each subject.

We have shown, in a sort of composite case history form, some of the steps taking a subject with Pr S≡Pr INC, through elaboration of E Pr INC during and after play. Certain subjects' dissatisfactions with their abilities to work with the INC formulation prompted generation of the very different Pr MP, with its unproblematical characteristics for representation of, and "algorithmic" performance of, execution. We have shown in KN's case the evolution of the successive Pr AMP:1, 2, 3 abstract program schemes and rule collections. That evolution was shaped each step of the way by KN's ability to relate his subjective problem-working experience during play to (a) his pre-game representations of the task (b) the concrete action sequence B_f, and (c) the score which actually resulted from play. KN learned progressively that he could use more and more of the attributes of the game situation in his favor in this fashion.

The discussion of task representation so far has prepared the way for broadening our notion of representation in learning games. Now, whenever we need to refer more broadly and roughly than in the last section to subjects' representational activities we shall speak of them as modeling activities. The task representations already discussed are (partial) models of the Shimoku game presented to, or made by, the subject. But the subject needs a number of other models in order to make those representations and to function with them in relation to the task. Already existing models and model systems in a subject are used in building new ones. He needs a model of himself as a problem solver, a model of his problem-working knowledge, and a model of himself-in-interaction with the Shimoku task.

We use the term "model" in the sense of Minsky (1968, pp.426-7).

> If a creature can answer a question about a hypothetical experiment without actually performing it...his answer to the question must be an encoded description of the behavior (inside the creature) of some submachine or "model" responding to an encoded description of the world situation described by the question.
>
> We use the term "model" in the following sense" To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A.
>
> The model relation is inherently ternary. Any attempt to suppress the role of the intentions of the investigator B leads to circular definitions or to ambiguities about "essential features"...

> When a man M answers questions about the world, then (taking on ourselves the role of B) we attribute this ability to some internal mechanism W* inside of M. It would be most convenient if we could discern physically within M two separate regions, W* and M-W*, such that W* "really contains the knowledge" and M-W* contains only general-purpose machinery for coding questions, decoding answere, and general administrative work. However, one cannot really expect to find, in an intelligent machine, a clear separation between coding and knowledge structures, either anatomically or functionally, because (for example) some "knowledge" is likely to be used in the encoding and interpreting processes. What is important for our purposes is the intuitive notion of a model, not the technical ability to delineate a model's boundaries. Indeed, part of our argument hinges on the inherent difficulty of discerning such boundaries.[1]

We shall use the "model" concept, then, as a way of speaking of aspects of a man's representations of the world and its parts and processes, including himself and his parts and processes.

It is important to note that the relations among the various models we shall discuss are not closely specifiable in most cases, and their boundaries may be fluid. If we say a person "has" a model of a thing at one time, we do not mean necessarily that he will be using exactly that model of the same thing at another time. The working model of the thing which he uses at any moment may be a fraction, or perhaps a new version including parts of other models--worked up for the present moment's purpose. To say that a person "has" a model says nothing about the quality of the model--its fidelity in representing the significant aspects of what it models. To say that someone "has" a model says nothing about the degree of articulation or coherence of the representation which constitutes the model. Some models are wretched, others superb--but only and always with respect to particular needs and capabilities for making use of them. A model, which may be quite well developed in some of its aspects and serve one purpose of the man well, may be incoherent in other aspects and serve other purposes poorly.

An important aspect of our discussion will hinge on a subject's awareness that his models <u>are</u> models: (a) that they are partial representations of some piece

---

[1] Minsky's discussion, to which the reader is referred for more of its development, encounters quickly the problems of (a) specifying just what knowledge within the man is to be assigned to what level (embedded, meta-, etc.) model; and (b) representing the relations among models, since it becomes clear that a simple idea of one model within another breaks down. "...(the) notion "contained in" is not sufficiently sophisticated to describe the kinds of relations between parts of programlike processes and...the intuitive notion of "model" used herein is...too unsophisticated to support developing the theory in technical detail. It is clear that in this area one cannot describe intermodel relationships in terms of models as simple physical substructures." (p.427).

of internal or external reality; (b) that they have been made often for purposes different from those he has at the moment; and (c) that their nature will reflect the differences.

The further a subject is from realizing that the representations, with which he thinks, are artificial tools, the less likely he is to be able to criticize the appropriateness of the representations he sets out to use in any new problematical situation and to adjust them to suit his current needs. The naive realist, the subject who identifies his model of a situation wholly with some idea of absolute existent reality, is less likely to see his representation as corrigible.

1) Consolidating a model of the game-as-given. Of course, in Shimoku, there are certain elements which any game model, or representation adequate to deal with the task must possess. The subjects' representations are likely to be inadequate unless they somehow include the entities and relations given in the instruction Pr S with most of the features of the game board. (We have seen, however, that one can score in the middle and even high, ranges (110+) taking relatively little account of space paths and 7-path properties of the cube).

   We begin discussion of important "models" that the subject must have with the game statement model.[1] In our previous discussion of the game statement given in instruction as Pr S, we have spoken as though the Pr S, given the subject, could be considered substantially absorbed and then used as raw material for making Pr INC, or MP, or AMP. The absorption of Pr S was taken as occurring during the instruction period and was tested and consolidated during the first game--at times with reference to the "crib sheets", given the subjects to keep during play. The instructions and rules could be appreciated and made more fully meaningful in action.

   However, we must now point to the apparent marked differences in ability to make an adequate first model of the game as given in the initial statement. It was necessary to assimilate the parts of the game statement given sequentially during instruction. They had to be kept in mind so that they could be used (a) to draw out implications of the instructions and rules for understanding the nature of the game and (b) to guide minimally coherent action.

---

[1]Frequently, to avoid monotony and to specify further we shall refer to given models by slightly different names. Thus, e.g., the game-as-given model is the first model a subject assembles of the game as it has been given to him in the instruction period. It is one of a succession of game "statements", or models, which he will use over the course of the experiment.

Strong subjects were not only able to grasp the instructions at a level that met our criteria in as little as one third the time of others but were able to start asking questions which showed that they were reasoning during the instruction period about implications of the instructions for guiding good play.[1] Other players who were high INC-m, and INC-h spoke directly of consolidating their notions of the game at the start of and early in play as a first task. The need to consolidate a model of the game-as-given was a first task in one form or another for most Group I players and for the stronger starters (LY, EH, UN, EI) in Group II. But a conscious modeling effort involved in testing the behavior of attributes of the game, and relating them both to other attributes and to developing notions of how to play, was much less extensive and/or much less successful--in at least half of the Group II players and in two who were at the bottom of Group I.

Those weaker players often tried to play with weak and haphazard models which omitted some of the rules. The contents of their models omitted or included mistaken versions of computer options, the sc-breaking rule, the use of unmove, and so on. Most significantly, they did not take perception of anomalies as an indication to search regularly or systematically to improve the models.[2] Recall that (a) they were given sheets with reminders of the sc-preservation rules, move costs, scoring patterns, and legal moves, to use whenever needed; (b) they were told that if they thought they forgot any rule or how to work any manipulation on the scope they might ask the assistant who was in the room. An effort had been made to make asking easy and unembarassing: we said casually that there was "a lot to remember," and that we expected people would want to check and to ask.

The failure to establish a reasonably coherent model of the game as given extended even to game two for about half of the weaker subjects mentoned.[3] A few of the weaker Group II subjects (and a number of strong subjects as well) took advantage of offers of review and warm-up practice before games 2 and 3. Others said they didn't think they needed them--but in the

---

[1]This was true of I/EH, who played his first game INC-h pp;I/KN, whose first game was AMP-d; I/HS, who played all INC-h; and I/KC, who played a strong INC-m and went to MP-c on his second game.

[2]They did generally check scoring-pattern types, however. The sc-preservation rule was writ large on the same sheet.

[3]And for a two or three, even into the third game. We recognize that many subjects just decided against using LKA, the look-ahead computer option, and did not store its uses in their game models. We count a game model as essentially adequate without LKA, provided the subject indeed does not try to make use of it.

post-game 2 interview said they still did not have in mind all the rules and options! Rather than dismiss them from the sample and ignore them, we should like to consider what appeared to be going on. We speak of trends. There was a spectrum with no precise boundary.

First, it appeared that subjects who made seriously defective initial models of the game situation did not have strategies for assimilating the various single instruction statements to get a good sense of the whole and of how things worked together. In interviews, they recognized and remembered having heard the parts that were not assimilated. Usually, they could recall having actively used the rule or statement in trials on the models of the cube and grids or on the computer screen during familiarization. Yet, it appeared that they had stayed "centered"[1] on a given rule or option in trial exercise during instruction and had not stored it in any organized way with the other features of the game. Moreover, weak subjects, recognizing during play that something must be wrong, tended less than stronger players to question themselves about what it might be, to search for sources of error, to review rules from the "crib sheets," or to ask for clarification. Their diagnostic procedures--parts of their problem-working-knowledge models--were deficient, and helped maintain the deficiencies of their original game statement models.

At the bottom of the spectrum in Groups I and II were players who for substantial periods of time seemed to accept whatever happened on that highly responsive scope, upsetting and disorganizing as it might be to them, without making coordinated attempts to develop a more coherent model of the game situation. Others tried to adapt by constricting their games so as not to encounter unexpected events (such as breaking of sc's). But, not knowing clearly the causes of the events, they made blanket efforts--e.g., merely playing by checking everything several times according to the rules they did remember, working only with paths they could see well, investing little, and so on. Still others tried to play fast and "blindly": evaluating little, and blindly unmoving (without thorough search for causes) anything that made graphs plummet. Of course, their attempts to play with defective models of the game-as-given were not only costly but left marks on any further learning. Typically, the very weak game modeling subjects had had, according to their accounts, relatively poor, rural or ghetto early schooling. They tended to keep as fragments statements describing parts of problem situations and not stand back and check on the quality of their initial representations.

---

[1] This adapts Piaget's use of the term. In perception tasks, young children overestimate the size, importance, etc., of objects at the center of attention and fail to relate them coherently and in proportion to other objects and larger structures in the field. (cf. Jean Piaget, The Mechanisms of Perception, translated by G. N. Seagrim. New York: Basic Books, 1969.)

2) Relating game models to other models. We learned in interviews that what we had been seeing with those who had trouble getting basic game conditions into a working model was quite typical of them in other new problem-solving situations. Their models of themselves as problem workers included expectations that they were likely to leave things out and to be unaware of doing so, to have to struggle in disorganized fashion, unable to find the sources of errors they might detect, and so on. Their coping with the difficulties initially encountered in building a game model was restricted by more pervasive models of themselves as more or less doomed to incoherence as problem solvers and capable of fighting only a painful rear-guard action to fend off the worst defeats. Those self-as-problem-worker models had been establihsed, in turn, by years of difficulty in working problems without a sese of how to go about building problem situation models or models of their own problem-working knowledge and resources.

To convey a sense of the mutual influences among the model of a given problem, the model of a subject's problem-working knowledge, and the model of a subject himself as a problem worker, an example will be helpful. The two Group I designers who were very weak and became incoherent and stressed in game modeling were able to function creatively in design with visual statements that "just came on inspiration"--statements they could make and store in their own drawings. But they were poor in assimilating quickly numerous complex conditions (which would have to determine characteristics of their designs).

One designer, subject I/YU, had coped with technical subjects in his architecture training by intense rote drill, applying fixed methods to problems. In new situations, he became anxious and fell back to a relative inability to structure a problem such as ours--given in verbal conditions as well as diagrams--or to develop a plan for carrying out a solution.

With tears of anger and frustration, this subject spoke of an early (upper- or upper-middle-class) education in an Arabic country, where great indulgence at home in early childhood was followed abruptly by a miserable forced attendance at a classical boys' school. There, rote memorization of texts under an authoritarian discipline which included humiliation and physical punishment had been the basis for years of unhappiness associated with learning. He had come to identify bold structuring of a situation with aggression and the hated instructor. He had been taught not to question or to plan for himself. Confronted with the need to organize his understanding in Shimoku to make a coherent model of a situation having a vague goal statement, incalculably many options, and known time pressures, he got into trouble immediately and was unable to do research on the nature of his troubles. In game 1, after losing points for poorly understood reasons on a number of moves, he did not go on to

diagnose his errors or request clarification. He decided (as he put it, defensively) to abandon any attempt to play for points and, instead, to "play like a little child," moving things just to see the scope respond. He scored -63. In interviews he said that in this game experience, he was starting to appreciate the needs for structuring and planning more vividly and concretely than he ever had before.[1] He was going to build a game like Shimoku for his small children, so that they could learn now to do what caused him such trouble at age 30.

Despite reinstructions, practice, and a warm relationship and sense of ease with us, in later attempts to play the game straight (not "like a child"), he never scored above +2 (game 3). It is interesting that this subject, despite a higher class background and opportunities that later included a European phase of education, played even in games 2 and 3 much like, but generally worse than, our young ghetto subjects. He played quickly (e.g., game 3 took less than 24 minutes) and "could not resist" moving long enough to plan. Post hoc, he could be more articulate about some of his difficulties than many of the weaker Group II subjects, but, like them, he did not have differentiated ideas about what "planning," which he saw as so important, would entail. Without a familiar context which would suggest rote-learned procedure components, he was at a loss to order a course of action effectively. His models of problem-working knowledge, including heuristics, planning, and evaluation techniques, were largely fragmentary or empty--often containing mere catch words with no procedures associated with them. This vignette of I/YU's experiences has enabled us to suggest informally where notions of models--of the initial problem situation, of the self-as-problem worker, and of one's repertoire of problem-working techniques and approaches--fit in discussing a subject's function. Again, the models are inferred, programlike structures within the subject, whose functions are thought of as accounting for parts of the subject's behavior in working on and discussing Shimoku.[2]

---

[1] Despite a number of university courses he had taken including planning aspects of large design projects.

[2] Note that the models themselves, and their use, involve the encoding and experiencing of feeling, in close relation to the knowledge of how to proceed in a novel and reasonably complex information-processing task. Our statements about models and the processes involved in their use can be taken as much a part of "ego psychology" as of the usually more narrowly construed "information-processing psychology." Ordinarily, psychoanalytic ego psychological treatments stay very general, speaking of "highly cathected" mental processes such as "creative thought," and their relation to a person's sense of competence and self-esteem. Or they refer to the thinking of "primitive" characters who do not qualify, distinguish, plan, etc., and who have simplistic models of the world of human (behavior-problem) situations in which they live. But the treatments do not usually try to specify further what are some of the information-processing skills that differentiate the "primitive" from the "complex, insightful" person.

The questions I/YU could ask and have answered using his game statement model--which omitted critical features of the game--would not have been adequate substitutes in thought for, or predictors of, experimental actions on the computer. Eventually, in the first game he stopped trying to query his model very much at all and engaged in random activity to watch the tokens twinkle about the board.

The questions that I/YU could ask this model of himself as problem worker, concerning his likelihood of finding a satisfactory way to proceed in the game gave answers to stunt aspirations and create anxiety. The model contained statements of his avoidance of "aggressive structuring", of passivity, helplessness, and little or no past success in the face of such a problem.

The questions he knew to ask of his model of general problem working knowledge were in themselves impoverished questions dependent for their generation on that impoverished model. For example, there was virtually no recruiting of model parts which might suggest seeking uses of analogy with other games, different strategies of representation, or uses of strong problem conditions to elaborate one's understanding of opportunities.

3) The self-task interaction model. Our treatment of models important to the Shimoku problem-working process must be expanded to include a model suggested earlier in our discussion of representation. An essential part of the process of elaborating and changing the subject's initial game statement representation occurs as a subject explores what he can do with the game. The successive representations made by a subject who progresses well in his understanding of the game are products of an interaction between the outside-world embodiment of the game in the computer and the player with his action effectors and his models--himself as problem worker, the initial game statement, and his problem-working tools. Gradually, a new, more or less continually changing, model is made. It contains records of interaction among the models within the actor, his actions, and the computer embodiment of the game. It is a growing model of self-task interaction. It has both historical and predictive guidance aspects. The later, more developed representations of the game "task itself" are models which overlap both the initial game statement model and the growing self-task interaction model of what can be known about and done with the task, by virture of what information processing the subject can do. The shape and quality of the self-task interaction model will be determined by materials and actions gleaned from the other, older models, and reworked in the context of the present experience of events in the game. The quality of the self-task model will first reflect and then determine, at the next stage, the subject's capacity to alter his conception of the task's requirements to suit his capabilities for analysis in the time given, to alter his notion of what is desirable and should be sought in the task, given new ways of enhancing his capabilities, and so on. The case of subject I/KN, presented earlier to show the co-evolution of representation

and execution capabilities in terms of the needs for program formation in Shimoku taken as a quasi-F problem, also fits our present purpose. It illustrates the notion of a developing self-task interaction model and its use in learning to improve his game.

It should be clear that the self-task model, its growth, contents, and functions will be very susceptible to inputs from each of the other main models we have discussed and to actual feedback from experiences with the machine to itself and to each of the other models. Let us try to apply the notions of the program-like, dynamic models within our subjects to some of the problems with which we are familiar from earlier sections.

4) The operation of models in the transition from INC to MP. The reader may have continued to wonder about the subjects who remained in INC-type play and about what happened to their occasional urges to go beyond their current fashion of play. And the reader may have asked further whether we might have any more detail about the INC→AMP transition, recasting the problem in a theoretical space of large structures. We shall not be able to satisfy ourselves or the reader completely, but we can add more to the analysis which may be helpful.

Let us return to the situation of an INC-h player. Take the mathematician I/HS, about to do his second or third game.

His self-task model says it is time-consuming and somewhat difficult to do extensive on-line processing.[1]

His updated game model says that the late game plateaus and that the game requires incremental play to adapt moves to immediate board conditions.

He considers the benefits of satisfying a high sc-interlock condition and whether there is any way to change the game model to meet it.

His self-as-problem-solver model says that he is relatively good, compared to other skilled people, at searching methodically, weighing and balancing, and so on.

---

[1]According to rules very much like those we presented earlier for HPW:INC-h, q.v.

If the game must be handled this way--incrementally--he can compete.[1]

Since he has already modeled the task largely as a "practical," not an "abstract" problem
because of the time constraints on solution, his model has biased him against inquiry to find whatever abstract parts of a solution description might have been attainable within the time available.

His notion of an "abstract" treatment is of so beautiful and elegant a product of demanding, pure, "timeless" thought as to bias him subtly against attempts to find working, convenient abstractions to recast the "practical" task.

The foregoing aspects of his usual problem-working-knowledge model influence the way he builds his current game model.

Contents of his self-as-problem-worker model enter.

Earlier in his career, I/HS had worked in pure mathematics and had won substantial recognition. In recent years, however, his love, pure mathematics, had been something done only on his own in spare time.

His job--which he regarded as a compromise, or comedown--required practical tasks in developing computer system software, and had a lot of day-to-day, local incremental patching to it.

He was good at it without devoting what he regarded as his highest faculties to it.

His typing this problem as a "practical" problem carried into the growing, Shimoku self-task model statements limiting his investment in and motivation for "practical" tasks. The imported statements were characteristic parts of his self-as-problem-worker model.

---

[1] We shall refer more than once to competing even though the Shimoku task was played as a one-person puzzle and scores were confidential. Each subject was aware that he was not the sole player. The absence of a known point score range or a general knowledge, such as a chess subject might have, of what constitutes good play prompted many subjects to imagine standards. They tried to set expectations of themselves in accord with past experience in other tasks, where they had ranked themselves competitively.

That he did not think much about the task after plays 1 and 2 is consistent with his relatively low motivation to reexamine--a motivation established by his models of the task and himself.

Let us return again to I/HS's problem-working knowledge model. It contained a strong form of an inspiration theory--"insight" was hard or impossible to prompt, and came on an all-or-none basis. There was little one could do if it failed.

This model applied to his pure mathematical work, as well as to any other kind of problem. There was a faintly desperate quality when he spoke of it.

The only heurisitic device of which he was readily aware was to leave something stumping him for a while and return, in hopes of avoiding fruitless persistence on a wrong track.

He virtually never discussed with anyone--put into words and/or symbols--how he was thinking about a problem.

In his job, he often relied on others to check his fundamental assumptions or to provide the context for his work.

He was relatively unaccustomed to describe, or to take a stance meta to the kind of processing his mind was doing on a problem, with intent to find possibly markedly different strategies. He was accustomed to use a variety of tools to polish a situation, construed in the terms given him, into less costly, smoother-working versions. But he spoke of questioning the basics of a situation as something he had learned not to do in his commercial position.

The model of problem working approaches currently active and available for application to Shimoku--a "practical" problem, which reminded him of his work--was likely to have hindered him from going to a preplanned large structure situation.

We have already suggested, in an earlier section, the kinds of events that occurred in I/HS as he considered the limitations of the INC-h approach that he used in every game.

He perceived the potential benefits of an approach which would achieve high interlock.[1]

[1]We are working here from his attempts to relive the events in the interview. Apparently, he had cycled through sequences close to this one more than once, over the three-game series, with the same result.

He made an "intuitive" assessment of the costs of building interlock situations.

Such building meant to him moving everything in a fashion that would ignore adaptation. Being "brute force," it would be too costly.

This evaluation occurred very quickly and was not more detailed than what is written here before rejection of the idea occurred.[1]

A number of other INC-m and INC-h subjects seem to have gone through similar "intuitive"--here, incomplete, vague, often at the edges of awareness and without examined premises or tests of feasibility--rejections of the idea of a nonincremental structure. Their assessments, too, were often biased gently by material from all of their own models--just enough to prevent development of the dim, nascent, fragments of an idea of a large structure. Data from three Group I subjects who did not go on the MP, and data from two who did, contain quite clear evidence of arriving at a state of "intuitive" rejection of the nascent idea similar to that noted above for HS.

One of the components often urged (cf. Polya, 1957, 1965 and Gordon, 1961) as part of a creative problem solver's model of problem-working approaches and heuristics is the rule of thumb: avoid very early practical evaluation of a budding and still ill-formed idea. In Shimoku play the pressures of time, as sensed by most of our subjects, pushed them to try to evaluate a new idea quickly. Here, to avoid premature evaluation, a subject's problem-working approach model required strategies for protecting against rushed evaluation.

The only subjects who avoided premature evaluation and consequent abondonment of a large structure idea on "intuitive" economic grounds were subjects who intentionally shielded their evaluations of the idea from severe time pressure. We have seen that I/KN, who never played an INC game and who achieved a large structure

---

[1] I/HS responded to the questionnaire after game 3 that he would be interested to know what others had done. But said he had no idea of what might be an alternative approach as successful as his under the time circumstances. When told of the large structure solutions, he became deeply serious. He reflected that he had been getting by piecing together solutions to day-to-day problems over and over again from a kit of tricks he knew well. He had been avoiding thinking deeply about the nature of problems on which he was working. This, in turn, was making him less able. He stayed for some time piecing together the relations between his problem-working style, his old abilities, and his current sense of the relatively unrewarding characteristics of his intellectual life. He felt the experience to be valuable and was glad, though shaken to have had it.

realization in the very first game, used almost the entire first half of that game for move-less meditation. And thereafter, he used questionnaire, interview, and post-interview periods for detailed, off-line analysis which improved the large structure schemes he was to use in games 2 and 3.

In the case of I/KN, we know from interview material that characteristically he tries to model his own problem-working processes. His model of problem working approaches contains many rules of thumb to check against premature evaluations, over-responses to time pressures, over-credulity for first statements of a problem, and so on. He is aware of building a model of what goes on in interactions between himself and the task. In a practical problem whose solution requires fine tuning between his abilities and the problem statement, the self-task interaction model comes to contain the revised working problem statement within it. The same was not so clearly true of the subjects who made transitions from playing first in INC play to go on working with preplanned, large structures. All of the INC subjects who went on to achieve preplanned high interlock large structures knew they would want to devote time to building and testing their schemes off line.

> All players who went on to use preplanned large structures had had in their developing models of the Shimoku problem something like this tentative notion: "creating a preplanned high interlock large structure is likely to prove too expensive."[1]
>
> But the notion of the structure was at that point portected by heuristics from the problem-working knowledge against dismissal on economic grounds from the growing problem model.
>
> The subjects' models also contained a countervailing notion: a very high reward from a large, high interlock structure might overbalance high costs.
>
> That notion could not be tested adequately until after further, heuristic pursuit of the large structure idea: that is, until candidate solutions from a theoretical space of large structures had been articulated. Evaluation at that point was in their favor.

---

[1] Since subject I/TT's MP-d game 3 arrived at a large structure partly serendipitously, rather than by pregame planning, his course does not quite fit this discussion.

All the subjects who went from INC to nonserendipitous MP games reduced pressures to evaluate quickly the large structures they were to use off-line. The structures were often modified at game time, but the work of evaluating whether an idea of a large structure was economically feasible had been done between games, on a precise, non-intuitive basis. All the foregoing behaviors were guided by parts of the problem-working-knowledge models of the large structure subjects.

Let us look at how the self-as-problem worker models influenced the INC→MP transition in those subjects.

We have mentioned that those subjects all seemed to have a stronger than average sense of frustration with the INC play.[1]

Their discontent was with (a) the enormous amount of detailed processing and patient repetition of application of their collections of rules and heuristics to each board situation, and/or (b) the seemingly pedestrian results and within-game score (and for EWB, across-game score) that were part of their INC play.

Subject I/EWB modeled himself as worse than average for handling all the evaluation necessary for INC play, but felt he must master every element of Shimoku incrementally and slowly first, before considering anything else. Having satisfied himself that his usual, cautious, methodical--"simplify, then add one element at a time"--approach to new problems had brought him a basic grasp of Shimoku, he felt both secure enough and annoyed anough after game 2 to be daring.

Subject I/HC had enormous dislike of repeating anything in problem solving, and of any meticulous polishing in order to execute any once blocked-out idea or method.[2]

Subject I/EH modeled himself as characteristically likely to become

[1]Compared to other subjects in Group I having similar background, ages, education, and point score ranges for their INC games.

[2]It was I/HC who, once having made the transition from INC play in round 1 to an MP-c game in round 2, lapsed to a much poorer execution on round 3. In the retrospect of interview 3, he felt the whole performance--impatience, bold restructuring, and shipshod work on a replay he felt must be done the same way--to be utterly characteristic of himself as a problem worker.

erratic in repeated tasks requiring careful searches and balancing of judgments: at times he did well and at times felt like throwing everything and "speeding fast over a cliff."

Subject I/NQ resembled I/HC in his dislike of repetition and weighing and balancing details. And like EWB, he modeled himself as probably not competitive if he had to play Shimoku in INC style.

Thus, the self-as-problem-worker model of each of the subjects who made the INC→MP transition on a preplanned basis pushed in the direction of a solution which would cut detailed on-line processing and the need for continual making of subtle judgments.

The problem-working-knowledge model of each contained material about "puzzle" problems which could be solved by key rearrangements.

Particularly I/NQ and I/EWB--the two who played the purest MP-c games with the least adaptation and no purchasing--had like, worked for fun with, and in NQ's case, invented rearrangement puzzles which were representable by recasting statis, mathematical, spatial structural schemes.

I/EWB gave more detail of the transition of his problem statement, Pr INC to a Pr MP in these terms: for the first two games (INC-1c and INC-m) Shimoku had been approached as a game of skill, where skill in foreseeing, in searching and weighing, etc., might be expected to improve with practice as does putting together one's coordination in playing basketball.

When the gains in skill seemed to be plateauing, along with his scores, after games 1 and 2, he considered an alternative.

It might be best not to see Shimoku as a game of skill but as a puzzle, something which must have a best way or an optimum.

The change in problem model recruited new parts of his problem-working-knowledge model. He started to search for best possible arrangements and drew on heuristics related to the puzzle-working domain of his knowledge.

5) Problem statement and approach modeling based on broader knowledge. We should like to make just a few comments here on the uses of subjects' problem-working-knowledge models in relation to the subject's broader store of knowledge based on his entire experience. All interviews followed up the questionnaire item which asked a subject

about other experiences that he thought might have influenced his play in Shimoku.

Whenever a subject responded that experience X influenced him, we tried, using neutral probes, to find out in what way he thought it had done so.[1]

Many of the better-scoring Group I subjects had to think twice before being able to articulate in what way previous experience had influenced them, but then were eventually able to do so relatively clearly. (Here again, there was a spectrum.) They were able to state in what way knowledge in their stored experience models had been extracted for modeling the new task. Relatively often, they could state explicitly in what ways the situations of the new model pair were analogous (and in what ways not). In contrast, except for the strongest couple of subjects in Group II, those Group II members who did report influence of analogies based on their stored experiential knowledge often could not specify in what ways the analogy was justified, and/or in what ways useful, or limited in use.

Those weaker subjects were much more likely to have been misled by their own analogies in playing Shimoku. It appears things typically proceeded like this:

> If any correspondence was seen between the stored model and Shimoku, attributes of the stored-model situation which were not relevant to it were carried into the Shimoku game model, essentially without scrutiny.
>
> The inappropriately imported attributes then subtly influenced the subject's further thinking about Shimoku.[2]

---

[1] E.g., by developing skills in search, by serving as a strategic analogy, as an analogy of structural relations, etc.

[2] This is a familiar finding that recurs in many guises. Importing implicit, excess syntactic or semantic content into a new problem by analogy is involved in susceptibility to demagogic rhetoric, in processes of invention and design (e.g. features of the horse carriage were carried unnecessarily into its analog and replacement, the horseless carriage), and in neurotic expectations (that a current situation will be as disastrous as one in the past that is in some respect analogous, but with respect to fewer attributes than are necessary to warrant the expectation). Despite the familiarity and generality of this negative side of analogic thinking as a source of error, practices based on recognition of its implications for teaching effective problem working have not become widespread.

Analogies are often made--on the basis of attributes shared between the model and the new situation--in order to guide the search for still more attributes of the new which may be shared with the stored model. If the right attributes are found, they may render the new situation susceptible to solution techniques known to be useful for the stored model.

Here, a common error of many Group II subjects involved assuming that the techniques known for the old situation were applicable without adequately testing the new situation for the presence of all the attributes warranting applicability.

They acted as though excess attributes had been imported into the new situation's model.

A preventive heuristic needed for inclusion in a subject's problem working knowledge model would be, "for purpose A, X is like Y with respect to characteristic Q. What extra attributes of X might I tend to attribute, or act as though I were attributing, gratuitously to Y? What would be the consequences?"

In the sorts of difficulties we just mentioned, the subject's problem-working knowledge model does not contain active procedures for assuring effective uses of analogy. The difficulties are related to the adequacy of description of relevant features of each situation, as well as to strategies for comparing them.

We have already seen two examples of analogic reasoning in building a subject's game model. Let us reexamine one of them.[1] Use of a model from the subject's experiential knowledge was successively refined.

Recall KN's somewhat vague notion that, as in tic tac toe, corner grid positions in Shimoku must have the greatest path "control."

In game 2 he had not got the expected extra opportunities to add on to corner positions on the middle boards.

At the same time, he had noticed missed, or poorly handled opportunities to make extra sc's through the center cells of inner boards.

---

[1]The other was I/KN's immediate use of similarities between Helmer's Future Three and Shimoku in an appropriate and clearly specified fashion to suggest the "forecasting" aspect of his game model.

The two experiences suggested that corner control, taken as a generalization on all of the Shimoku grids from the tic tac toe model, had to be modified: "control," (or interlock opportunity) depended on the way attributes of the cube, not just those of the plane, determined additional path-making opportunities through key cells.

The idea of key positions penetrated by more paths than their neighbors in a square playing board could be taken over from the tic tac toe situation.

But the idea that the key cube would have the "same" (corner) positions had to be qualified by taking into account attributes of the one situation which prevented its being simply analogous to the other.

This refinement process did not happen all at once. It was triggered by the subject's constant monitoring of the data coming in from play, and his use of vaguely perceived discrepancies between data and expectations to trigger further articulation of the similarities and distinctions of the modeled situations.

That sort of refinement of analogy in modeling may sound simple enough. But it is something some weaker Group I and mid-range and weaker Group II subjects neglected to do in a plethora of instances, which occurred at all levels in their play. Instead, their use of stored models as analogies to Shimoku tended to be expensive. A gross but not atypical example, with subject II/DY.

The presence of poker language for the scoring patterns ("straight flush," etc.) was taken up by his model of available heuristics, which was loaded with ways to try to find patterns in arbitrary, or chance situations.

He spent a great deal of time trying to treat Shimoku as a game with chance aspects and a secret, hidden pattern of events (as if in superstitious gambling).

This occurred despite the fact that there is nothing else he or we could see in the game or instructions to suggest the applicability of gambling models. After spending much of his time during games 1 and 2 at trying to apply it, he finally concluded that the model triggered by his identifying the

poker language really was not applicable.[1]

Here is another example.

A number of subjects in Group II spoke in interviews of trying to apply the notion from tic tac toe that the center cell was important.

That would have been worthwhile as a hunch. But, typically, the weaker subjects never tested the applicability of their expectations for a center cell of a 3x3 tic tac toe grid to a cell in a 4x4 grid.

They did not clearly state to themselves that a 4x4 grid has no single center cell at the cross of the diagonals.

Instead, the typical response was simply to retain a (correct) sense that "the center is probably important (based on tic tac toe)," but to say "something's fishy and I can't handle it, I'd better drop it and try something else."

Such a typical event would

- tend to fragment play (since many such efforts were parts of attempts to work out sc's);
- fail to build the Shimoku game model by failing to clarify and make usable anything from the analogy;
- decrease the subject's attempts to develop and to call further upon his models of problem-working approaches and experiential knowledge; and
- in subjects, whose self-esteem components of their models of themselves as problem solvers were already low, lower them further.

---

[1]As one might expect, subject II/DY was also one of those who had the most difficulty in organizing the instruction statements into a coherent initial game model. We speculate that this difficulty in assimilating game elements into a coherent functional model may make his experience of events in life appear more as if they were driven by chance. And, closing the circle, his bias to use his heuristics to investigate a situation as a chancy gamble may preclude his assimilating what deterministic structure is present.

Each tiny instance of this kind of miss had costs in terms of confidence in, and exercise and development of each of a subject's models: of problem statement, of problem-working approaches and heuristics, of self as problem worker, and of self-task interaction.

Let us turn the discussion of modeling to one more facet, which will enable us to appreciate more of the learning subject's task. We need to look more closely at some neglected parts of the business of modeling while playing.

6) Modeling while playing and its relation to learning in INC games. We have already treated some aspects of modeling while playing under other headings and names. For example, we have seen the need for a subject hearing the sequential instructions and seeing the teaching materials to compose for himself a coherent model of the game as given. The consolidation of the model goes on while playing. The adequacy of that first model would influence any subsequent learning possible. We have seen the subject's need successively to model new versions of the game and to model his interaction with it--in order to develop new representations necessary to change play from INC to MP, or to start AMP play and keep it evolving. At least, the data gathering for any modeling which is associated with major changes in game representation must go on while playing. We have shown some of the information-processing requirements for executing the main game types, once learned. Execution time applications of adaptive policy rules require at least short-range modeling while playing, although not necessarily the amounts and kinds of description and storage of important features that would be wanted for a post-game, off-line analysis. But we have by-passed some important aspects of the modeling that is necessary even to learn to play and to improve an INC game. And INC games were all but twelve of our sample.

Recall that the hypothetical HPW who had simply to execute an Arrived INC-h game was kept very busy with enough calls for on-line information processing to stretch him to and beyond his capacity. What would he have had to do in addition, not merely to execute the given game, but to gather data on the whole performance, while playing, and to use it for intercurrent and post-game analysis? And what of the game performance modeled in play where a subject changes approach in midstream?

Clearly, any player trying to model what he is doing while doing it cannot afford to spend all his playing time just figuring out and executing moves. Probably we must interpret the loose "while" in the last sentence substantially to mean "in periods alternating with" rather than only "during exactly the same time as," for a human

processor in Shimoku. In any case, more of the game time and effort potentially available for concentrated processing of particular moves when playing a pure Arrived game must be allocated to modeling in a Learning game.

In this sense, again, the Shimoku environment resembles many real-life tasks in which the learner is in an environment he does not fully understand, and where (a) he is unable to predict fully the short and long range responses to his actions; (b) he must be intensely absorbed in performing those actions; and (c) he must at the "same" time construct some picture of the relations of the principles or rules guiding his actions to many features of a complex, cumulating situation.

What must go on in the modeling time? First, if the subject is to be able to develop knowledge based on moves, he must have some memorable description of the moves and/or move sequences he is making. Second, he must have a memorable description of the rules or principles, or essential features of the programs he has used in guiding the moves. Third, he needs a description of the ways the rules were interpreted, mutually adjusted, and carried out in the moves whose effects will be used, in turn, to judge the worth of the individual rules and of the procedures for their application. He will need apt ways of describing end-game board effects resulting from play.

To be able to assess an idea for any kind of change in midgame, he will have to use the descriptions right then to assess the effectiveness of events in the recent past, and will need to have generated descriptions of alternatives. If at any moment he should want to change a single policy rule on the basis of on-line analysis, he may need to probe the effects of the change on the meaning of other parts of his model of the game before devoting moves to it. If he wants to make a major over-all strategy change in midgame (as in the case of I/TT:3, the game which became MP-d part way through), he will have to generate a description of a game-level alternative strategy and test preliminarily whether it can be effectively applied in a situation already partly developed under a different strategy.

In any case, we see that to try to learn from the game in any systematic way requires an on-line effort to describe and to store important aspects of the events of the game. The more sophisticated the analysis is to be on which learning and improvement can be based, the greater are the demands on description. To give aliment to the modeling and self-correcting effort, a subject will have to be able continually to play, to store some result, then to move one and two levels meta to the actions of play and to store some result.

Let us take actual search actions and move actions as being on more or less the same level. Take the description and prescription of the organization of search as on another level, meta to the first and saying something about it. Then a comment which criticizes the organization of the search and proposes another is still meta to that. It is not so important to worry about assignments to levels as it is to recognize that a subject needs to be at home in moving quickly from one level to another and in alloting time to his processing in each. Often, he needs to be able to keep track of the status of programs in the meta levels which have been left temporarily, to search and execute. For example, a subject who, say, changes a top-level strategy may find himself still using some habitual lower-level rules which are inconsistent with it. If he does not note this, his later evaluation of the new strategy may be falsely based, and his learning from experience will suffer.

We have sketched quite a heavy load to be placed on the game-executing player who would learn at the same time. Consider now implications of the different main types of approach to the game for the modeling that should go on to permit learning to occur during and after play.

First, let us look at a learning subject who is trying to develop his model of the game and of what can happen as he interacts with it. He is playing a basically incremental game (though one not yet so carefully elaborated as the Arrived stage INC-h game whose instruction set we have seen on page 99). Recall that even in its most developed and polished form, INC-h is relatively difficult to represent compactly for guiding a player. Similarly, it is difficult to model an even less well developed INC game in a way that can be extracted on line or later to make new generalizations. What is a subject to describe to himself, and to note for later analysis? Token positions will be difficult to remember, having been generated by many moves, building adaptively atop an unpatterned opening distribution. Because for good play, nascent policies often must be balanced against each other in the context of local conditions, it will often be hard to see just what the effect on the whole game of a particular policy or rule one is trying out might be. The procedures whereby the <u>ad hoc</u> mutual adjustments of rules to local needs occur may be complex, partly out of awareness, and hard to describe. The game will be hard to reconstruct from memory, particularly if one wants to search out effects of move order, which is important in highly adaptive games.

If he has postponed much of his attempt at modeling for later learning to the postgame period, in favor of concentrating during game time on assessing and executing, move by move, a subject will have forgotten most of the board and will have--by nature of the INC game--a few

rules with which to generate the essential structures attained in his play. For the ambitious INC player of whatever level of ability, the push toward the large amount of on-line immediate processing necessary to effect good moves in an INC type game will be in constant conflict with much of his need to take time and effort back to model what he is doing in order to permit learning.

Sophisticated modeling attempts were largely abandoned by most INC subjects. Crude and often inadequate modeling while learning in the weaker (INC) subjects diminished their opportunities to learn from play.

Modeling while playing requires getting critical data about game performance into the subject. Let us see how the locus of representation of game data during play interacts with a subject's problem approach heuristics when he goes to model the game events.

7) <u>The locus of representation in a man-machine system and modeling for learning.</u> One way to look at the modeling problem for a Shimoku player is to consider the locus of the parts of a problem model or representation that he is using, and the control he has over access to those parts, for use in learning to correct his procedures on and off line. Let us think for a moment of the man and machine functioning together as a single information-processing system, with the problem representation somehow distributed over the entire conjoint system. So long as he is at the console, the player can act as an executive function, whose final implementation, bookkeeping, storage, and display functions are available to him through nearly instantaneous communication with another part of "his" system. The internal models which the executive absolutely must have within it can be relatively sparse and incomplete, for many purposes, so long as it can refer continually to the auxiliary parts of its system.

Certainly the executive can apply the most basic incremental program ([1], page 30) without there being any holistic representation of the game or board situation within the executive. For a simple example, it can apply that program using a crude move generator--systematic use of the computer's "paths through a point" option to find incomplete sc's--to guide search, as in the restricted INC game of subject I/KB.[1]

---

[1] In the section on INC-r games we described KB's game records as consistent with such a description. In his interview he himself explained that he had used P, probing cell by cell starting at the left of the board, as almost his sole method of finding moves.

The modeling effort needed to guide current or future game play for such a game can be minimal. The states need not be stored and indeed would be hard for a person to store. Play can occur essentially as if guided by a first-order Markov process which, moment by moment, assesses only current states for their incomplete sc's and applies the basic rules for making patterns which are in the subject's model of the game-as-given.

Consider a game actually played only by responding at each moment to key features of sequential states whose sc-1's and sc-2's are "algorithmically" sought and closed according to a few pattern-forming and legal move rules. What are the consequences of playing such a game for an "executive" subject's internal representational needs? He needs to describe and to store almost nothing of the patterns evolving and of their relations to policy rules, regions of the board, and playing order descriptions, and to particular new strategies guiding their formation. One who plays in this fashion, using the scope to represent nearly all the game data, and with little modeling or storage effort seeming necessary or desired, is likely to be unable to use his game experience for analysis (after play), to learn to improve a subsequent game. While he will be able to ponder his rules and their narrow range of application, he will have little else to examine. The scope display will no longer be before him. Yet he will not have encoded enough of the vents formerly represented for him by the display to have a rich descriptive history of his game. A game whose play can only be represented by a simple search guidance rule and a simple incremental program in one's memory is likely to remain fixed between games in its current rule encodings. A subject needs to have some descriptions of objects and relations in his updated game model (and self-task model) which could serve him as raw material on which to operate between games in considering alternatives.

We have taken an extreme example to illustrate what we believe to be a general principle. Any subject who leaves virtually all of the conjoint man-machine system's play-time representation and storage functions to the machine during play, allowing the machine wholly to relieve him, runs the risk of failing to describe and to store what he needs to permit intelligent change. He will be lost without the machine when later, away from it, he goes to analyze and to learn. In contrast, consider a player who does not keep virtually all of his game representation stored in the machine, merely referring to it, moment by moment. He will have--during the game--to do a lot more tagging, chunking, sorting, describing and storing of events to establish a modeled base within himself for later inference. In the rest of this section we consider learning which depends upon the man's having developed a substantial degree of the man-machine systems' distributed problem representation in his head and diagrams.

Take an INC player who is playing under a new and still untested, high purchase rule.

> Many things are going on in his game at once which make relationships unclear.
>
> Toward the end he finds he is picking up more interlocking sc's than he had realized were possible.
>
> If he encodes and stores these data in his model of the current game, he may be able later to suggest a correlation among the high purchase policy, dense concentration of tokens, and enriched pattern-making opportunities.
>
> The new parts of his updated game model can be remembered, extracted, tested and eventually generalized to improve his subsequent play.

The preceding vignette, shows a typical example of descriptive modeling and later inference which occurred in a number of better INC players who had a complex set of rules, and game patterns which were hard to represent compactly. It is evident that a player using an AMP-type pregame, flexible diagrammatic plan with several articulated policies, and following it during the game with the formulating, and notating, and recording efforts involved in maintaining an up-to-date color-coded working diagram, will have an easier time storing adequately for future learning and analysis. His game's structure, worked-out notation system, etc.--all of which, as we have seen, make it easier to guide execution during play--make it more efficient and easier to store the events for the longer term and to have them available for pondering. Thus, modeling while playing, and learning through the on-line and retrospective uses of the models made both easier with the more coherently interrelated and easily describably AMP game, compared to the INC.[1] The very experience of doing the information processing necessary to establish in the man the memorable imaged structures and their articulated relations to policies for adaptation in the AM game probably should be thought of as having changed the cognitive equipment with which the person thereafter approached the game. We are reminded of Bower's comment (1970, p. 504) on the induction of effective memory by use of linguistic tags related to vivid, interacting images: "The important ingredient appears to be the cognitive constructive activity itself..."

---

[1] We do not mention the MP game here, because most people who arrived at the classic MP design assumed no further learning was necessary or possible within the circumstances surrounding the game. Their games, were, of course, the easiest of all to model!

Summary of Conceptual Scheme

We have components of a scheme for speaking about a number of key aspects of developing Shimoku play. Now is a good time to pull the elements of the scheme together.

Processes in the subject are thought of as occurring in a system of models which can be related to actions that perform experiments on the world. The subject has a world model which "contains"[1] models of himself. His models of himself "contain," inter alia, more or less accessible models of (a) his general experiential knowledge (with many submodels); (b) himself as a problem worker (his style, success history, usual information-processing capabilities); (c) his problem-working knowledge (heuristics, capabilities to model a new problem, and how to approach it); (d) the succession of models of the game problem just given; and (e) the growing model(s) of self-task interaction (what the problem is as the subject can learn to know it and cope with it).

Potential inputs to the subject's representation-making and -assimilating capabilities were first given in the experiment during the instruction period. What happened to the statements offered for input during that period, and immediately after it during the first game, determined the nature of the subject's own model of what constituted the game-as-given. The instructions were given sequentially. Once inputs were registered in the subject, active structuring of them was essential to the subject's modeling. He had to model both the abstract relations among entities represented in the problem situation as given and their relations to the procedural options available to him.

Initial formation of a model of the game-as-given was carried out, subject to contents of the problem-working-knowledge model the subject already had. That model provided the subject's repertoire of ways of organizing inputs into a coherent whole, testing their consistency, testing his understanding of their mutual relations at some basic level, and querying the outside system (instructors, assistant, and computer, via exploration of effects) in order to correct and to consolitdate the model.

Contents of the self-as-problem-solver model could operate at this stage in terms of motivation to find and/or to impose structure, or to avoid structure; in terms of expectations to be overwhelmed, or to master; and so on.[2]

---

[1]The word is to be taken with the caveats mentioned earlier.

[2]Similarly, contents of the self-as-problem worker model can enter into problem processing in this fashion at each of the subsequent stages. We shall not always state this kind of influence at every point of the scheme, but will expect its operation to be taken as understood, even where not explicitly mentioned.

Our detailed treatment of problem representation spoke as though early making of a consolidated version of the instruction statements--an adequate game-as-given model--to be well within subjects' capabilities.

Thus, Pr S as the abstract, static statement of the game given in the instructions was treated as if taken in by the subject. Pr INC, the static statement in the subject's game-as-given model was a direct takeover from and organization of Pr S.

A subject whose own initial game-as-given model became a version of Pr INC then was seen to develop successive refinements on and elaborations of his initial E Pr INC (the part of his model that referred especially to execution aspects of the problem). His representations came, more or less, to codify the properties of the game-as-he-could-know-it.

All subjects but I/KN went through early stages of attempts to articulate and to develop successive versions of Pr INC and of E Pr INC. I/KN transformed Pr S immediately while assimilating the instruction statement. He then worked with successive versions of Pr AMP and E Pr AMP statements of the static, abstract structure and of the execution and control aspects of his Adaptive Master Planned games.

To extend this scheme to weak players, it was necessary to look more carefully at the comfortable assumption that Pr S was directly assimilated into Pr INC, and to specify that the assimilation of Pr S was, in addition problematical. For most Group II subjects, and the weakest two subjects in Group I, this certainly was the case.[1]

In all cases, the successive versions of each subject's models of the game were models of the game as he could come to know it--as a product of the interactions between himself, his information processing capabilities on-line, and his models of heuristics and relevant knowledge. The time limitation on the game, the complexity of it, the unfolding character, the cost structure,

---

[1]Stronger Group I subjects also had occasional difficulties, which they were readily able to overcome when they asked about the facts after game 1 if not before. A most notable example, suggesting the strong interaction of personal characteristics with the early game modeling task in a very competent subject, was that of subject I/IC. He accepted the point drops from accidentally broken sc's throughout game 1. He never unmoved the sc-breaking moves, even if he noted the drops before a second move (which would have precluded unmoving) had been made. That he did not recall either the breakage rule written on his "crib sheet," or that the UM button before him could be used to undo--despite his ease at assimilating everything else--was consistent with his world model: "you don't get easy second chances;" and with his self-as-problem-solver model: "once I start to mess up, it's straight downhill from there."

and the largely uninterpreted, unstandardized goal statement, all made Shimoku more like many real-life problem situations than like a problem given in the usual, find-an-abstract-solution sense.

There were basically two ways that a subject who became acquainted with at least some of the heavy demands of high-quality processing for an incremental game could go. The first has been studied in the form handled by eventual MP subjects in Group I. In it, the impetus from discontent with the results of applying their own limited information-processing capabilities to the demands for playing a high quality, incrementally formulated game pushed them to search for an alternative approach. For success in finding that approach, they had to recognize the power of the high interlock, H-sc condition, and make a decision consciously to seek a candidate solution which would fulfill it. The search for the alternative had to be accompanied by intentional maneuvers to resist time pressure, and to resist intuitive dismissal on economic grounds of a costly large preplanned structure idea. This shielding was most commonly accomplished by reserving to off-line periods the articulation and precise evaluation of candidate solutions.

In Group I, no one who sought methodically, an alternative representation, off-line failed to find it (within about a half hour, by reports). No one played a large structure, high-interlock game who thought of the idea in a vague way but failed to articulate and then test it against costs. That more of the subjects who were discontent with INC processing made no transition to any kind of master-planned games may be regarded as caused by the power of a "set" to represent and to execute the game according to notions established by the instruction "you can make points by making scoring patterns." But that sort of explanation can be deepened by considering the discontinuity in representation that was required to make the transition to "overcome set."

In considering the D, F, and quasi-F problems and their correspondence to views of Shimoku characteristic of INC, MP, and AMP games, we look at the relation of the main working problem representations used by subjects to the language of the problem's givens. There was need to go outside that language and to project a theoretical space dominated by the strong interlock condition, to engage in MP or AMP play. Program-scheme representations, characteristic of D, F, and quasi-F views of Shimoku had correspondingly different properties for transformation into control program statements and for execution-time requirements. In turn, those requirements placed different demands on immediate information-processing capabilities in the player operating on-line. And they offered subjects markedly different opportunities to model effectively and to learn to improve games treated in one or another of the main game-type representations.

We have seen that it required no novel concept--and in that sense was easier--to stay in a D problem representational framework than to leave it. But it was far harder to execute a skillful, high-scoring game using it. And it was far harder to model that game in order to analyze it and improve one's performance. The leap to an MP game showed a marked change in play based on

a single, major step of re-representing the task and the player's relation to it. MP large structure users felt they had exhausted the possibilities as they defined them and had no more to learn.

In contrast to both INC and MP players, we saw continued growth of the one subject who started with a partly articulated idea of a large structure. He relentlessly reworked his large structure ideas in relation to his information-processing capabilities and to their needs for assistance as he opted to retain and to cope with demands of a strong adaptive condition. His learning shows most clearly the effect of cycles of conscious effort to represent the problem. His reworked representation made modeling and off-line learning more effective. In building the representations he examined the effects of the two (partly opposing) strong conditions first somewhat separately and, then, as they determined each other. The elements of his approach which permitted him partial relief from excessive detail and frustrating waste in on-line, incremental processing were the same elements which permitted more coherent modeling of the situation for planning purposes and for later purposes of analysis. The person who appears to have learned the most about the game, in terms of taking into account the attributes of the problem and using them profitably in play, was also the person who had the most articulated mental model of the game as he, interacting with it, could come to know it.

Our scheme for analysis enables us to speak in a common language of information processing to describe and, partly, to account for demonstrated differences in objective performance on the Shimoku task and to describe and to account for some of the marked difference in subjective experiences in playing it. We have shown that the solutions to the abstract aspect of the problem and the experiential aspects of play are closely intertwined.

Understanding Group II and Lowest Group I Performances

We have set out to see if we could effectively study not just competent problem workers but some who might be less than competent or articulate. Until now, we have said relatively little about members of Group II and their performances. Good problem workers tend to converge toward one or a few approaches which are heavily determined by adaptation to task and human information-processing requirements. But poorer problem workers manage to have a greater diversity of ways of going askew. Like others, we have been hard put to find compact ways of formulating just what it is that goes on with them.

A profitable way to start to look at the performance of the poorer workers in Groups I and II is to think of an analogy. Any complex metabolic pathway in biochemistry will serve us. Take a hypothetical pathway consisting of a dynamic and, at times, looped chain of 10 to 20 or so reaction processes, interlinked in a number of ways. Each requires particular local conditions--

of available substrate concentrations, environmental pH, critical catalysts, and cofactors--at each step. We shall suggest events in a fragment of the hypothetical pathway, leaving the reader to extrapolate.

> Assume that the product of step P is required in a certain concentration for step Q' and the product of Q is in turn a substrate for R' and a by-product of step R acts as a cofactor for step C in another part of the pathway. Then anything which curtails production of the compound resulting from step P may affect a whole chain of steps immediately following P, as well as another part of the pathway.
>
> Often, in seeing the game records of a poorer subject, it is as if we could identify a shortage of substrate in P, shortage of an enzyme cofactor in Q, excess of an inhibitor on the enzyme catalyzing R--and thus a complex of sources for difficulties encountered in step C. That might be the pattern for one subject. His friend might have a slightly different collection of analogs to blocked cofactors, enzyme defects, substrate shortages, etc. impinging on his processing in the Shimoku problem. Another fellow might have still another different set of microdefects cumulating to a slightly, or vastly, different collection of troubles.

The different subjects' trouble collections might show some family resemblances, in the sense of Wittgenstein (1967). Subject XX might have difficulties p, q, r; subject YY, p', p'', r; subject ZZ, q', r'', c; and so on--where each small letter and its primed versions represents a defect, omission, or block as it is detectable on observation of steps P, Q, R, etc. The results of such collections of difficulties operating at different parts of an interdependent and cumulative process with internal loops can be quite different in all-over configuration. But there might still be a limited number of them which, recombined in different orders and intensities, might account for much of the variation in what we see.

To account for the subjects' performances and experiential patterns in Shimoku, one would have to know the typical blocks, inhibitions, etc., for each and how they were applied to, and affected, the subjects' individual attempts to form and to execute programs for playing. In studying in detail the 42 games in the final Group II sample, and the lower 6 games in Group I (as well as a number of games from the non-returned Group II dropouts), we got a sense of a collection of fragmented, defective procedures which could be seen as containing large numbers of critical nodes. At each node, representation and execution could be biased or erroneous.

Many of the kinds of troubles we saw resonated with the general statements, and even clichés, of the literature about poor problem workers and, especially, about ghetto students as problem workers. We think that eventually it will be possible with studies of the sort carried out here to tease apart and to describe more precisely the information-processing characteristics of

students that are described as "impulsive," "fragmented," "distractible," and so on. The detail of operations from the printouts and the overviews and occasional details from interviews make a tempting start. But in face of much diversity in a small sample, and other limitations, what we have here is only a start. At this point we still do better at understanding the individual cases than at making trenchant generalizations across cases. Let us look more closely at some of the frequently shared characteristics of the weaker games, keeping in mind what we have said in the biochemical analogy about influences of combinations of factors and of factors operating at different stages of procedures.

First, let us recall our discussion of difficulty in modeling the game-as-given. We have pointed to what appeared to be lacks in the subjects' models of problem-working approaches--lacks of heuristics for developing a composite and integrated model of a problem from a set of instructions given sequentially from the practice exercises on the cube, practice board, and scope. We have noted that rules might be omitted, but their omission often would not be noted. For examples, the subjects were not alerted by anomalies in the incomplete or erroneous set of statments they had taken together; or, if trouble was noted, it might not become the stimulus for a corrective inquiry; or, statements which had registered--had entered the subject's working collection of statements descriptive of the problem situation--might not be taken together and examined for their relational properties. Some subjects functioned as if they had few, or no, good procedures for keeping in mind a body of new statements, relating them, and modeling them coherently. They tended to work with fragments. Naturally, they experienced distress as some of the negative consequences of this accumulated. Moves might be planned and execution begun only to be found illegal, unrealizable, or unprofitable. Failure to organize scope maneuvers according to instructions led to accidental disruptions of moves.

What we have seen as a difficulty in pulling together fragmentary statements into a checked-out internally consistent problem model at the start can be seen as part of a larger, persistent pattern in many subjects. For example, their own internal instructions to themselves about how to evaluate and execute moves seemed to be used in a fragmented fashion. Take a typical, composite attempt to plan a move sequence.

> An idea occurs to "do X."
>
> Three or four of the main elements required for doing X--search for tokens, identification of cells, plan for building a given sc--are worked out. The subject goes ahead.
>
> However, he has omitted consideration of two or three additional elements which are essential to "doing X."

He puts in time planning X, starts performing the actions required for it (running up move costs), and only then discovers that something is wrong.

He then simply abandons the attempt to "do X" in frustration at the still undiagnosed difficulty, and starts afresh somewhere else--more uneasy and mistrustful of his own playing but with no more knowledge to guide him.

Or, he stops, to try to diagnose the source of difficulty. He finds one, where there are two. He tries the plan again after changing the diagnosed single fault, fails again, and goes off to something else.

Or, he makes a new theory of good play concentrated so much on avoiding the diagnosed fault that other important features, formerly grasped, are now freshly dropped out.

The new theory doesn't work well, because of the new omissions. The subject decides his theory making is defective. He should stick only to the safest, restricted approaches he thinks he knows well--stopping attempts to learn.

If the subject does not already have an area of safety--a way of playing that at least functions in a restricted way, he becomes increasingly desperate, struggling "just to stay above the (zero points) line if I possibly can," moving empirically, unmoving, moving, unmoving.

Examples consistent with these can easily be identified in Group II and lowest Group I players' records. Our hypotheses about the mental events occurring find general support, and occasionally specific and precise confirmation in the interview material.

Everything we discussed so far in this section can be seen in terms of the difficulties keeping in mind simultaneously the critical features of a modeled situation or plan or the critical reminders to check certain features before proceeding. The difficulties can be seen as (a) problems in making representations of situations and of courses of actions, and (b) as problems in using representations effectively to guide the actions as they occur.

We have noted earlier the serious problems presented to any player who needs to encode efficiently and to use efficiently the rules, policies, and plans for playing <u>INC</u> type Shimoku. Subjects, who did not have good strategies for checking to see that they had a representation which included the significant aspects of a situation or for checking while executing actions to see that they were keeping their places in the unfolding programs they had set for themselves, got into a great deal of trouble. Generally, such weaker subjects got out of it, if they did, by holding down aspirations to levels which could

be met by relatively simplified and rigid approaches to play. If they did not get out of the trouble, they might continue to the end to execute little chains of operations based on defective partial models, relatively unrelated to, or even thwarting, other chains of operations within the same game.

Another related defect in Group II play was the difficulty in relating current moves to larger views of the game's emerging structure. It was more severe and exaggerated here than in Group I. Many Group II subjects told us of being able to concentrate only on a current move and often of forgetting how they had meant to relate it to another. Those who spoke of trying to plan for several moves so overloaded their heuristically poor immediate-processing capabilities that they "forgot" or "got mixed up." Either they fell back to very modest approaches, or they decided to play large parts of their games in the computer's look-ahead mode (LKA).[1] The effect was marked enough that we note the scores on our main graph both before and after LKA cost subtraction.

The category INC-e, which represents the highly empirical games, captures one way in which subjects chose to cope with the modeling, preevaluating, and ordering and place-keeping difficulties in INC games. The extensive uses of LKA that occured in INC-e games were, typically, accompanied by many unmoves outside of LKA, as well as by extensive unmoving within it. Often, there were many probes for scoring opportunities with "paths through a point; and, interestingly, many applications of the "display graph" option.

To understand the possible use of the "display graph," recall that when the "paths" option is on, the scope cannot show the cumulative graph of the subject's net points at each move. Subjects who had trouble maintaining a sense of where they were and what was going on in the game often said they grasped at the graph to help give them reminders of where they were and how they were doing.[2] The problem of getting a stabilized sense of where one is with the problem and a sense of what one could expect of one's actions is highlighted by frequent and often strong reactions to the graph--reactions of dependence on it or of anger that it was too close a testimony of one's troubles, "It makes me look stupid."

---

[1] They forgot that they had, or in some cases decided it was worthwhile, to pay 1/2 point per forward move in LKA, whether the trial move was retained or not. They were very certain that they were "likely to mess up" and needed protection. About one third of Group II games show LKA costs cutting deeply into their net scores.

[2] The graph would have come on, anyhow, as soon as the paths were erased and a move undertaken.

Typically, about two thirds of the Group II subjects appeared to have construed the need to adapt moves to multiple aspects of each local board condition much less strongly than did players in Group I. Refined and close adaptation--which required the abilities to represent opportunities for scores and to develop them under the guidance of a collection of policy rules--was an excessive burden and therefore unrealistic for some to attempt. Several subjects told us this explicitly--they felt that a stronger adaptive condition on their playing "solutions" would have been a good thing, could they have handled it.

Others seemed simply to operate without criteria for good adaptive play. Scoring patterns were made at much greater costs than necessary, or than apparently advisable in light of what then developed in the game. A number of subjects who played such games told us they had in mind only a few criteria for guiding moves, omitting many of the desiderata for good adaptation. They forgot at times--or during whole games--how expensive it was in points lost and in moves to purchase. They moved tokens out of the way to prepare a path and then moved them back, on changing their minds. They removed tokens from the playing board to the reservoir to clear particular cells, rather than take the removed token into a new, potentially rewarding situation elsewhere on the board. Removes, which are nearly always bad moves, occurred 43 times in Group II, but occurred only once in Group I. A majority of Group II games appear at one time or another to throw away cost criteria, partly by failing to get more than the simplest, minimal use from a move. The weaker the game, the more heavily neglected were the opportunities to cut costs by getting more mileage from a move.

We have mentioned earlier, in describing game types that multiple uses of tokens in sc's also decline in the game types taken up by many members of Group II, and the poorer playing members of Group I. The work, that we have demonstrated to be required for getting high multiple uses of tokens in sc's in an INC game, is just the sort of work to require a strategic chunking and ordering of information in the immediate processor--work of representing and of executing in orderly fashion. Instead, weaker subjects' play showed expensive building of individual sc's which interacted little or relatively unproductively.

The sc's might be built according to some simplifying rules. The rules serve to cut out much of the analysis of local opportunities, interlock possibilities, and number series. They allow the subject to consider only patterns of a very limited kind and, thereby, to unburden himself of a great deal of processing.

Some subjects told us that they would decide for a while to concentrate on one attribute of tokens--shape, for example--in making sc's, ignoring anything that could be gained by considering the additional attribute of number. A

subject following such a rule would have been likely to produce instances that appeared to the game record analyst that an unnecessary forfeiture of high scoring opportunity had occurred, leaving only a low sc.

Some of the H-sc manqués and the pseudo-sc's we found are attributable in different subjects' accounts to (a) failure to carry a precise enough description[1] of the token needed in a slot, (b) failure to apply it in selecting a token, (c) failure to check the accuracy of the placement once the token had landed in its new cell.

Some Group II subjects appeared to feel that place keeping was such a problem that checking moves out for accuracy then would risk getting lost. Others were not in the habit of checking anything in their problem working in school if they could possibly help it, and they saw no reason to change their procedure here. What came, came.

Only 20% of 40 Group II games had their peak score as their final score.[2] In Group I approximately 90% of the games had peak score as final score.[3] This means, of course, that most Group II players in most games did not retain a score once they had it, and they tended to go on moving even when they could not adequately evaluate subsequent moves. Experientially, they had a sense of destroying and losing which led some of them to feel trapped by the dangers of handling what they could not fully master. Yet they went on moving and losing.

Several factors contributed to the tendency to continue to play despite losses. First is the game's attractiveness because of its responsiveness, acting for its own sake is more lively and enjoyable than analyzing. Most subjects at all levels did enjoy just getting a response out of the machine

---

[1]Relative insensitivity to the problem of adequate description came in the frequent tendency of beginners to use a search and placement description for a "pointed" token, rather than a "point up" or "point down."  and  were, of course, two different token shape sets.

[2]We speak of 40 instead of 42 games because data are incomplete on 2 games for this part of the analysis. In II/DY:3, the computer failed at printout time, though after the final board had been copied. In II/NN:1, the subject executed an astonishing 480 actions, when the uppermost limit for storage of game operations had been set at what seemed a safe 400 (all but INC-e games had $<$ 120 actions, and no one had reached 300 before). Therefore, only the last 80 of his operations were printed out; the first 400 were bounced out of storage when the game actions reached 401.

[3]The 10% which did not, included games of the two weakest players mentioned earlier.

and commented on it repeatedly. However, differences in preference for actions versus analysis cannot wholly account for the phenomena.

The weakest subjects in Group I and many subjects in Group II expressed a feeling of an almost irresistible push to move, in spite of wishes not to. The push was to do something even if they did not know what. The players who scored well in Group I also frequently reported this sense of pressure, but they resisted it more effectively. Not only did they use slogans of the sort employed by weaker subjects ("take your time," etc.), the stronger subjects deliberately called on and rehearsed heuristics from their problem-working-knowledge model, reexamined their current game models, and usually soon found a procedure which would be useful in the game. Subjects frequently spoke of experiencing anxiety when they did not know, and had not yet found out, how to act.

Subject II/VR, who said he never thought about problems as problems, but usually just "saw" answers that were "ok" and went on, had sketched at the start of game 3 several token numbers without shapes. He executed moves for H-sc's very rapidly, using only 52 actions, 25 moves, and 12 minutes to earn 65 points. That was enough for him, being a big improvement on -56 and 33 in games 1 and 2, and he stopped. We have only a poor understanding of what was going on with him, since he could not be induced to write or say much. All of the other subjects who played rapidly, however, did speak clearly of being "nervous" and wanting both to see what would happen quickly and to prevent getting too hung up.

Thus, it probably is fair to say, as would most of the psychological literature, that the apparent "impulsiveness" of our subjects was realted to anxiety and was one way subjects chose to diminish stress. What is often slighted in accounts of impulsiveness in students is consideration of the close relation of the anxiety and "impulsiveness" to actual and estimated information-handling capacities.[1]

On interviewing the very rapid players and those players who substantially repeated a previous, somewhat limited game, we heard repeatedly of their having "no idea" of how their games could be improved. Or they told us of simply repeating over and over to themselves slogans like "take your time," "plan ahead," while being unable to know what actually to do to translate slogans into behavior appropriate to the problem-working situation.

We have remarked a relative poverty of readily available heuristics in some very bright Group I subjects, who were little accustomed to thinking

---

[1]Except where the subject is known to have a sensory or neurological defect such as dyslexia.

explicitly about how they developed problem-working strategies. The lack of consciously available heuristics appeared more debilitating in Group II. Group I subjects, who often did less well than they might have, were nevertheless equipped with an experiential store of problems solved; had special integrated problem-working techniques, as taught in higher-level academic disciplines; and had standard built-in procedures for ordering an approach--all of which could do some of the job for them almost automatically.[1] Group II subjects had fewer old reliable routines, which could be mobilized almost without conscious effort, and were thus even more at the mercy of their poor abilities to build new effective procedures consciously on the spot.

One of the most important things that might come from detailed studies of the information-processing activities of any subjects would be an appreciation of their points of leverage and potential growth. Group II subjects were by no means homogeneous with respect to the contents of their self-as-problem-sovler models and problem-working-knowledge models, the quality of their abilities to make game-as-given, and later game and self-task interaction models. We considered it promising that more than hald of Group II subjects were ready to take on the paid, experimental, "fun" Shimoku task as a

---

[1] A striking example of this is afforded by subject I/RV, who held a computer system design job while accumulating a straight A average in a Ph.D. program requiring mostly mathematics and physics. He said repeatedly that he was quite unaware of his own problem-working processes and, when stumped, left a problem, brooded over it in no particular way, and returned to it eventually for a solution--especially if he knew a solution existed. For the Shimoku problem, his INC-m and INC-h games were respectable "solutions," and he did not seek another approach but assured himself that he had one appropriate to the circumstances. In interview 3, he noted that although it usually didn't matter, his lack of awareness of ways of going about a problem had got him into trouble in his Ph.D. oral exam. He was asked to solve a problem which he did not immediately "see." The faculty encouraged him to show ways he might approach it to find an answer. What he did to demonstrate approaches was disorganized and inadequate--but not just because of nervousness. It was, he said, typical of his inability consciously to mobilize an orderly process of solution when his (apparently very talented) unconscious processes were not automatically producing for him. He had been criticized and admonished to improve by his committee but had thought little about it again until now. The conversation stimulated him to reexamine his approach to Shimoku. (Also, we think, he may have glimpsed at a distance the highly ordered nature of the board on the scope of MP-c game I/EWB:4 when leaving the computer room after his own game 3. This would have suggested that another kind of solution did exist, though he could not have seen any details.) He telephoned in a large structure, high sc-interlock MP-c type solution as the "optimum" on the day after game 3, having spent an "insomniac" night.

potential serious learning experience.

Eight of the 14 subjects in Group II spontaneously and enthusiastically tried to link experiences in the game to learning how to play and how to control urges to act before thinking. The remaining six took the present Shimoku situation more nearly only as a pleasant diversion with a fascinating gadget.[1]

Eight who "turned on" seriously, stayed strongly involved in the task as a learning experience, despite the troubles they had in learning based on defective representation and execution skills, and despite the lack of any help. Five of these who struck us on other grounds as among the most curious and thoughtful, had an unusual answer to an item on the first questionnaire.

Question 11 asked, "Is it usually easy or hard for you to tell someone else how you're thinking about a problem?" The other nine members of Group II had answered it was "easy." When queried, they turned out to have relatively less of a sense of what it felt like to tackle a difficult intellectual problem.[2] It appeared that they did not engage often in what they themselves would consider to be serious problem-oriented thinking. The kind of thinking they did do was, so far as they could tell, easy to communicate with the tools they had.

The five Group II subjects who answered that telling someone else how they were thinking about a problem was "hard" were very different. Each was aware of trying to do "hard and deep"[3] thinking about mathematics,

---

[1] Several of them, however, have come up since to ask if we had any more "neat games to figure out." Because of the experimental situation, we could not explore much of a potential teacher's uses of the task to stimulate thought and to suggest new directions for thinking. Hence, the constructive reactions we found in the unguided situation may represent the minimum of what could be expected if there were unobtrusive and skillful prompting in a Shimoku-like, purposefully educational set-up.

[2] If school problems in math were hard, they tried a few times and gave up, generally not liking them and thinking them artificial and worthless. Life problems--the first association they had to the workd "problem"--were often so upsetting and murky that they did not regard them as subject to coherent, problem-oriented thinking.

[3] Quoting subject II/SO.

social problems, or gambling. Each was used to trying to look more deeply than his friends into a problem and to seeing many situations around them as intellectually geniunely problematical. Each said he tried very hard and, frequently, got very confused and frustrated in trying to think for himself but liked to keep trying anyhow. All five felt seriously hampered in efforts to conceptualize clearly and to put into words the kinds of more serious and problematical things they were thinking about. Four felt none of their friends or family could understand them when they tried to talk about new or different ways of thinking about a problem. Three had at times been mocked by their schoolmates, called "dumb" or "far out" when they tried to explain something nonstandard. Four felt their teachers found it too difficult and time-consuming to try to find out what they meant; one was greatly helped by a math teacher. Three frequently asked themselves whether a math problem could be done a different way from the standard one demonstrated in class. Four thought math was really interesting--unlike most of the rest of Group II! All five took learning very seriously. Yet four felt very isolated in the enterprise, were subject to great tensions and frequent depressions, drove themselves to try to learn, considered dropping out altogether, and then drove themselves again, at great subjective cost.[1] Compared with the average of their fellow Group II members they seemed to us more intense, alert, and vulnerable.

The responses to question 11 suggest that among our ghetto subjects there were some quite troubled, isolated students with a great awareness of their own diffidulties in expressing thought in adequate representations, and in using these representations to communicate. The subjects answering "hard," appeared readily capable of a great detail of excitement in using their minds to do hard work, despite all that thwared them. They were more likely to experience elation at new learning and discovery and to experience marked disappointment at being unable to understand or to get help with their investigations of a problem.

One of the attractions of mathematics for them was its providing representations which permitted new kinds of thought. Three of the students had been curious about computers before, and after their games made suggestions to us about how computers should be used to help people "learn to think straight."

Two additional subjects who answered "easy" to question 11 also were aware of occasionally doing serious problem-oriented thinking and qualified their questionnaire answer in discussion: talking to someone about how they

---

[1] Four of the five had spotty academic records at best, frequent headaches, disciplinary problems, and difficult-to-terrible home situations--but these features, while reported with more intensity, do not clearly differentiate them from others in Group II.

thought of some serious problems was often hard for them. They, like the "hard" group, were relatively more likely than the remainder of Group II to speak of doing intellectual work, though not necessarily school work.

Although this is a pilot study and the sample is small, we were struck by the way question 11 pulled out of the sample boys who had a great deal in common. We wondered whether they were potentially the original thinkers in their relatively inarticulate peer groups. A reasonable hypothesis would hold that while ordinary, superficial speech would do to communicate thoughts for most who were not thinking too hard or too differently from their peers, the limitations of language (in the broadest sense of internally and externally communicable encodings of meaning) hit the deeper thinkers harder.

What surprised and intrigued us was the idea that the question might be used as a screening probe.[1] It might help in identifying those ghetto students who felt a need for, and might especially benefit from opportunities, to express themselves precisely about thinking and problem working processes. Standard means--their California Tests of Basic Skills and their grades and school reports, as we have checked them from the Upward Bound records--would not permit us to differentiate them from their fellows. Their ways of playing Shimoku were diverse and not necessarily coherent or well organized. But the game records, coupled with interview material, suggested ferment--ability to question, a tendency to try to reconsider and reshape play--which was, again, greater than the average of their fellows. The results of their efforts to think hard might be almost as likely to be disastrous as productive; to be stressful and disappointing as to lead toward growth. They were questioning and experimenting and not shrinking back to some of the extremely conservative and unambitious approaches of other Group II members. But their questions and experiments still suffered from many of the kinds of defects we have mentioned as characteristic for weak subjects in Group II and the bottom of Group I. They were more likely than their fellows to acknowledge the defects and to push to try to understand them. But they were not necessarily more successful in doing so. The picture is one of subjects making often intense efforts, for unreliable rewards, and being aware of suffering from their combination of curiousity, high

---

[1] Answering "hard" to question 11 does not necessarily mean that a subject has trouble thinking within his own conceptual system. It might mean only that he has trouble in communicating--in translating out of the private system. But Group II subjects who reported the one reported the other, as if for them the two were often directly related. They did feel, in addition, that at times they were able to have insights that were good, if uncommunicable. But they were so aware of their tendencies to err, that they mistrusted their own insights unless they could get validation. Moreover, they often felt at the edge of something, but could not get effective help in clarifying and going further in their thought because of difficulty in describing what was going on inside them.

aspiration, and lack of tools.

Learning to Plan and Planning to Learn

In Group I those subjects who thought of a preplanned large structure solution and took the time to articulate and to evaluate the idea invariably played successful 110+ games which substantially realized their plans. Was there anyone in Group II who thought of an over-all large structure? Was there anyone who took time to try to articulate a design? If so, why were all Group II games classed in INC?

There was at least one person who considered over-all designs and took time to try to articulate plans. But no coherent preplanned large structure, involving high sc-interlock and most of the 30-move complement, ever was worked out and effected. Let us look at this young man's effort. To us it seemed to tell a great deal about what happens to a bright person whose procedures are just defective enough to hinder him at nearly every turn.

Subject II/TS was one of the subjects who answered "hard" to question 11. He spoke with fervor of his attempts to understand things and to get new ideas but of having a mind which "has no pattern" into which "everything (was) jammed every which way"--hard for him or for anybody else to understand. He was used to teachers avoiding his attempts to do the textbook geometry proofs differently, and he had experienced rare but heady triumphs that broke his isolation when he could both show what he meant and have it turn out right. He doubted himself much of the time and at times agreed with others that this ideas might be just plain "out of it." In his first interview, we speculated aloud that he might be an original thinker and asked him to tell all his ideas about playing Shimoku. We would try--but could not guarantee--to understand them.[1]

Let us describe highlights of events in his play.

> In II/TS:1 Incremental-disorganized-empirical (IN-de) game II, he netted 8 points, having had 20 before LKA cost subtraction, adn a peak of 37 earlier in the game.
>
> He had made 255 actions but only 18 accepted moves in 60 minutes.

---

[1]We came to feel both for him and for his teachers: despite our total concentration and later reanalysis of tapes and records, his excitement, imprecisions of language, and tendency to jump from a fragment of one thing to a fragment of the next, on and on, made him at times beyond us to follow.

On starting out, he tried immediately to make elegant, triply or quadruply interlocking sc's, and immediately he ran into trouble. He forgot partial plans, lost his place, broke up previously completed sc's and disrupted partly formed new ones.

To try to cope with his troubles, he used P (paths through a point) 41 times to try to check against sc breaking and to find new sc making opportunities. He thought of P as "plan ahead" option, a way the computer could tell him what sc's to make.

To try to maintain a sense of where he was, he repeatedly displayed the graph (23 times).

He felt he was "messing up" so much that he used LKA for 171 actions, so he could reverse as many moves as possible.

He tried scratch paper to work out a few sc's and their interconnections, but soon mixed up those related to one interlock constellation with those related to another.

He watched the scores go up and down and felt the machine made him "look stupid." He got a headache. He felt making points was a "hardship."

TS had, among other things, an immediate processor overload syndrome.

In interview 1, he said he could see that he had to plan because his head couldn't handle things without.

Asked how he was thinking about the game and what it suggested to him, he mentioned:

an analogy with chess: some tokens in some positions might have more value than others, as a king has more than a queen (he raised this several times, also in interview 2, without developing or clarifying it).

an analogy between grouping animals by similarities in biology and grouping tokens by shapes and numbers in Shimoku.

that learning to view the board in three dimensions would help him think better about the univers.

considering making patterns with all four tokens kept to the same number, to simplify and reduce what had to be kept in mind

considering making patterns only by attending to shape, and

ignoring number entirely, again to simplify

considering putting key numbers in corners and working in three dimensions from multiply shared corners tokens

using P and LKA more to check himself

considering making as many different approaches as he could think of, taking plenty of time at home

considering using more high interlock sc's to save moves better
considering getting a way to see his whole problem before him at once

considering "doing something so (he) could play the whole game and never get lost," and "trying to plan the whole game"

Like INC→MP transition subjects and like the AMP player, TS (a) recognized the difficulties of overload in an incremental game, (b) recognized the value of sc interlocks, (c) protected himself from time pressure by trying to plan off-line, and (d) had a notion that an over-all plan or design might be useful. He was by no means a victim of unexamined "set," and had a long tradition of trying to do a problem first in at least one way opposite that overtly or covertly suggested to him by any instructor. He took the game more seriously than almost any of our subjects.

At home between games 1 and 2, he practiced playing chess and tic tac toe, just to warm up his mind on games. He drew two sets of diagrams, which he brought with him to the game and turned in to us as part of the record afterward.

The first drawing showed four versions of an abstracted series of lines drawn on 4x4 grid boards without tokens, shapes, or numbers. They represented ideas of making sc's by going in a plane vertically through the cube in one case; by radiating from a corner of a cube in another; by making a vertical column of the four central cells of each plane; and by making filled X'd end planes. Each version he saw as an alternative game plan, or at times, as an element of a game plan which would somehow incorporate the best features of all the separate plans.

His second drawing contained four grids across the top with patterns made by filled and outlined tokens without numbers. The filling in and a method of circling the tokens were to represent the fixed versus the optional nature of a shape requirement in a given position. There was an arbitrary tabulation of scoring patterns and their values which he had realized (rightly) was probably incorrect that he had used in

working out his diagrams for practice. In addition, he had drawn a large, three-dimensional representation of the cube and all its cells, with shapes-only tokens used to make four X's, one on each plane[1] plus some additional tokens in same-shape patterns. A calculation of points to be earned according to his scheme completed the picture.

Taking the diagram in the scoring system he provided, its rewards were incorrectly calculated. His attempt to estimate not only figured wrong points made, but: failed to account for the number of tokens on the drawn board (42) compared to the opening game board (32), or to account for the number of tokens on the drawn board (42) compared to the opening game board (32), or to account adequately for move costs. Some supposed shapes-only sc's would have been inconsistent with his own scoring table.

He had not worked out a strategy which could start from an opening board and arrive at a structure like the main drawing.

He had intentionally omitted numbers from the plan in order to reduce the burden of on-line processing. This was costly simplification, since move costs were fixed and no scores over 6 could come from shapes-only sc's.

He considered that he had accomplished a reasonably satisfactory "plan" to play according to a large structure idea.

He had generated a candidate solution which we can think of as being in a theoretical space. But he did not treat it as if in a space of alternative large structure solutions which had then to be narrowed and refined.[2]

He did not systematically reintroduce the temporarily set aside problem conditions in order further to articulate the candidate solution. He did not refine the candidate solution into one which would meet information processing requirements for himself or the problem.

Most readers of our account of the INC→MP transition are likely to have such built-in routines themselves that "getting the idea" of a large structure, high-sc interlock solution may seem "all it takes"--the rest of the elaboration simply "follows." But the rest did not, apparently, simply "follow" in this subject. His heuristics for generating new flashes of

---

[1]As in KN's basic idea.

[2]Wholly unlike Group I subjects who got this far.

partly formed ideas functioned much better than did his rules for guiding reintroduction of conditions, balancing of their mutual implications, and testing all aspects of a candidate solution against them and the "givens."[1]

> On returning to play game 2, TS was disappointed. He found that he had omitted so much from his thinking about the large structure plan or diagram that he felt he had to abandon it entirely.
>
> After getting off to a rough start when confronting the inadequacy of his main plan, he switched to a plan of working in a single plane going vertically into the cube. He created an Incremental partly planned empirical (IN-ppe) game with a fairly good degree of coordination of four main paths on all four boards.
>
> The plan he used was indeed only a single notion of how to orient a plane. It was coupled with a rule to concentrate on shapes only in making sc's. The plan and rule were sparse directives. They could give him scant guidance for most of the actual move finding and assessing he had to do. Therefore, he soon again felt "messed up."
>
> He switched into LKA to do 128 actions, with 94 forward moves that took a whopping 47 points off his final score, leaving a mere 12 net.[2]
>
> He experimented in LKA with 30 exchange moves, recognizing that if he mastered them he would conserve moves by getting two tokens into desired new locations (each other's) at once.
>
> He made 30 sc's instead of game 1's 18 sc's. But still he kept 2/3 of them as only low scoring sc's, mostly made on a shapes-only basis.
>
> He completed his 234 actions (for 30 accepted moves) in 49 minutes, raring to go, nervous, and afraid too much thinking would "get me lost."

As in his first game and despite all his efforts with "plans," he was

---

[1] We see that to teach TS, it will not be helpful to concentrate on stimulating him to think "divergently," and to expect well-oiled machinery to process the creative flash into something he can use.

[2] He said he had "forgotten" LKA costs. However, he said he was glad he used LKA despite the costs, because he felt such a great need to be able to take back his actions. But he would not do it again.

depending heavily on seeing empirically constructed, and subject-to-repeal, concrete realizations before him on the scope. It was his main way of assessing what he had done and what he wanted to keep.

A measure of TS's make it-and-then-evaluate-it method is seen in his having actually constructed 382 points worth of scoring patterns during the game, counting everything made in regular game and in LKA, and then kept, broken, or purposely erased![1]

He did not make a new, workable diagram plan for the game. Nor did he make a "valued locations"[2] diagram to reduce processing load and to protect against sc-breaking. He did try to cut the load in massive and costly fashion by largely just ignoring numbers.

Game 2 had evidence of some interlock and of structuring using the four boards in an on-line, partly planned fashion.

In contrast to game 1, and as part of his continued experimentation, he had hardly used the "paths through a point" or "display graph" options, and had tried to carry out their functions in his head.

Thus, in game 2 there was a prodigious amount of action, a continued testing of new ideas, and evidence of great uncertainty about, and an empirical attitude toward, the results of his own information processing. At the same time, an improved organization of the board resulted.

In the interview after game 2, he spoke of his game 2 plan as inadequate. But said he had a new plan. He would fill a grid richly, with sc's going every which way. And too, he would try to spread out his sc's over the other grids as well. He might try to pay attention to numbers, and avoid LKA "by planning better in my head." The new plan was not further specifie in the interview.

At home between games 2 and 3, he used playing cards in numbers 1-8, placing random arrays of them down, as if on an opening board, and trying to make sc's from there.

Although he was more conscious of the importance of 30 moves in the interview, he did not try to apply the 30-moves condition seriously to

---

[1] This was the highest Total-Points-Made score of any game but I/KN:3. Especially for a subject whose sc's were often low-earning 6's, it shows a great deal of construction and demolition.

[2] Marking cells whose contents participate in sc's. Compare the INC-h game instructions set in section

Although acutely aware of wanting to make his on-line processing more effective by using plans to help him remember where he was, he did not work out a diagram with tokens and numbers that could actually help him to keep track of his place in the game.

He retained some notion of an adaptive condition which would influence all token placements. Yet he did not articulate for himself the partly conflicting relations between the adaptive and the preplanned high inter-lock ideas. Nor did he work out techniques for trying to cope with both.

TS's game 3 was Incremental-moderate-partly-planned (INC-mpp). It netted 72 points (73 before an irresistible, momentary entry into LKA just at the end).

He completed 30 moves in a mere 21 minutes, and kept the highest score he made. TS again played fast in order "not to get lost" and "not to mess up."[1]

He used only 1 unmove, and 71 total actions.[2]

The final game board, Figure 24, shows grid 1 completely filled with 3's, 4's, 5's and y's arranged in rows of four-of-a-kind numbers and in columns of straights.

At first glance, it looks like the perfectly arranged, "maximized" grids of our MP-c players. But, in contrast to an MP-c subject's getting 10 H-sc's out of such a configuration, TS got only 4, with the remaining 6 being only L-sc's.

An MP-c "perfect" grid would be well coordinated with another in space, sharing one set of four tokens with it and contributing to eight more interlocking H-sc's. But the plane which intersects TS's grid 1 makes only a small fraction of the points possible, and coordinates only minimally, in part accidentally, with the filled grid.

TS said, further, that he only made the straights by chance! He was surprised to notice how well the numbers went together--though he had had an idea to do something like that in game 2, and had abandoned it.

---

[1] But slower than in 1 and 2, in terms of actions per minute.

[2] The minimum for a 30 move game would be 60, so game 3 was sparing and to the point.

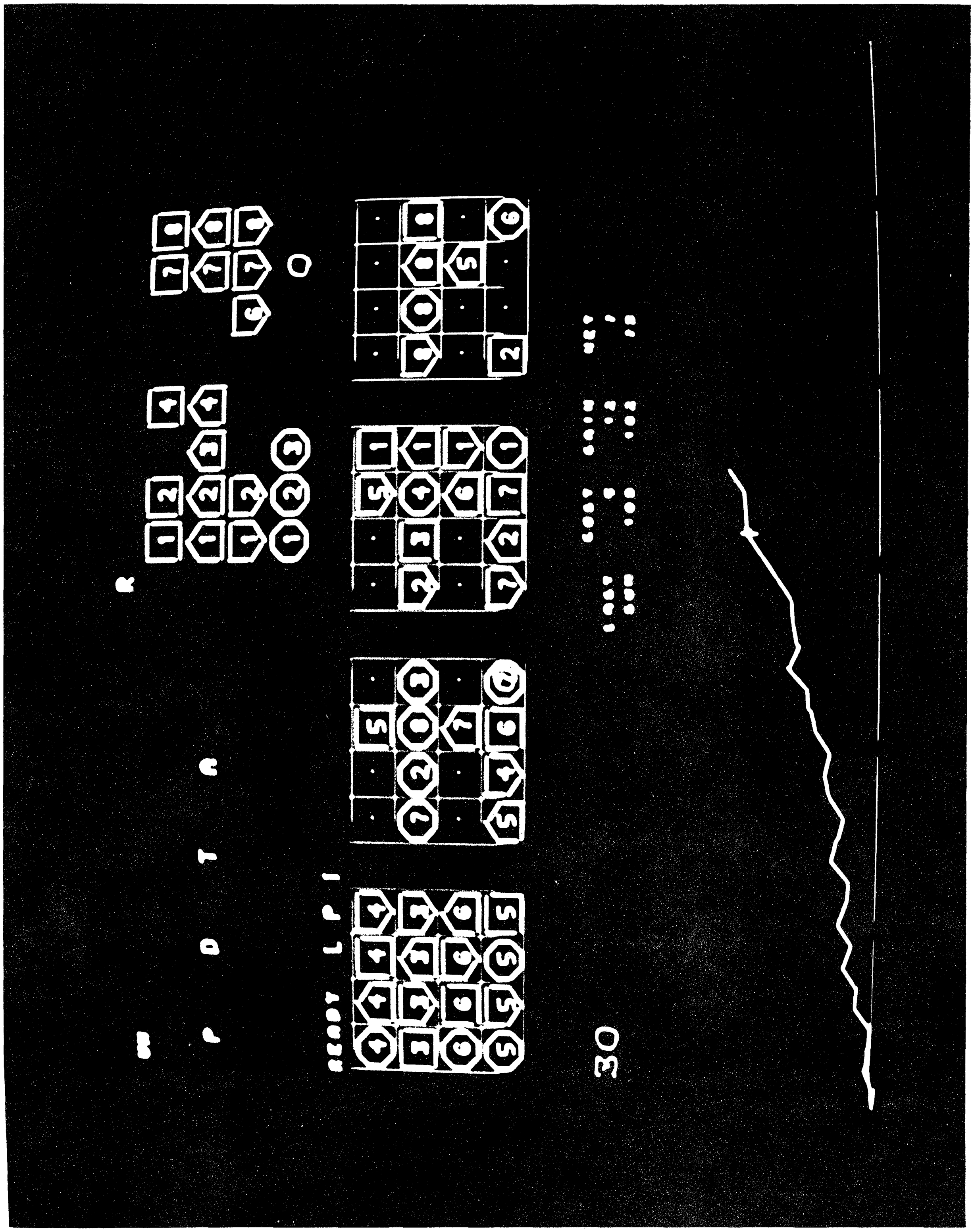


Figure 24. Final Board of Game II/TS:3

Even so, numbers are better arranged in grid 1 than shapes, for the patterns which would score for four different shapes often fail. They contain a duplicate of one shape.

In interview 3, TS requested a chance to review records of all of his games together, to learn from the progression, and to identify his mistakes. He would have liked to play more--to see if there was some way to put all his plans together in one game.

He also wanted to know in detail how others went about working, to learn from their approaches.

He said it was a new idea to be able to make a lot of different plans and keep trying how well each one did. To quote him, "This was a new approach for me because...it's more real...more realistic than just thinking of it. If I...was to play with the computer like I think just in everyday life, I probably would have just about 40 points. I wouldn't gain nothing, because I would try to go through it (once) and that's all. If I could play again I could see if my ideas would work. If it wouldn't work, I would see why...and maybe try it again to see if if would work.

The highest score obtained in Group II was 75. Thus, no one in the ghetto-barrio group did much better than TS's 72. No one in Group II showed as much evidence of efforts, however flawed, to keep his game evolving, and to work toward a new representation.

Yet at the end TS still was not progressing to think of how to handle the adaptive and interlock conditions and the 30-moves complement in any systematic way.

His best attempt to elaborate a holistic structure had occurred with working out diagram 2 after game 1. Once the plan based on it failed, for lack of having been tested adequately against problem condtions, he never returned to so ambitious a representational effort.

In playing Shimoku, TS became more aware than before of the need to describe problem properties and to seek a representation which could help relieve his immediate information-processing overload. He was rich with the germs of ideas, but not nearly so well equipped with ways of unfolding their implications and composing them into the whole that he sought. His procedures for monitoring his own execution of even very short and uncomplicated subplans--e.g. to complete sc's of given shapes--could not be relied upon. Procedures for comparing the merits of differentially costly ways to cut his on-line information-processing load were not in evidence. Procedures for following through his insights into need for external memory support were flawed, and

then most of the insights were dropped. As noted above, once the idea for an over-all, integrated structure arose, procedures for developing and refining a main design representation were not adequate. Systematic introduction of problem condtions for test against plans or tentative solutions of any kind was lacking.

TS was still working after game 3. He spoke about a need to assimilate the accumulating pieces of his knowledge about the game and about his capabilities to play it. He looked ripe for more experiences of this kind. We thought high potential had been made visible by the Shimoku situation.

We hypothesized that he would have an intellectual growth spurt given opportunities to work independently, but with help available, on computer-based tasks that had opportunities to construct, run, debug, and re-run candidate problem solutions. The hypothesis was shared with the Upward Bound staff, who tried to find opportunities for him. He worked with one of us (SK-D) the folowing summer, taking a self-instructional computer-based programming course with other Upward Bound students. He was the fastest and best student. He became our assistant, and a consultant for other students. He grew interested in using the program records to diagnose his fellow students' difficulties.[1] He learned more computer languages and paced a junior college class in computing while still finishing high school. His leadership in Upward Bound and in his community grew.

One case proves nothing, and we had not set ourselves up in the Shimoku project as systematic diagnosticians of talent in disadvantaged students. However, the hunches we had about the four Upward Bounders who happened to rejoin us in the later programming project were confirmed. The Shimoku situation, because of its responsiveness, its concreteness-cum-abstraction, its opportunities to go back and reexamine, and its relative independence from teacher-student hostilities, may be good as a method of finding and describing problem-working gifts, flaws, and flawed gifts in the educationally disadvantaged, which are so hard to describe usefully by ordinary tests.

---

[1] The the end of that project, he said in an interview: "I'm really learning a lot about my thinking from working with the computer--now I can think clearer, less random, like less just stuffed into my head any old way. And a lot about how other people think--from looking at those printouts. I can tell a lot about how a kid is doing--like where my friend Jose needs some help. I want to use this if I become a math teacher so I can know how my students are thinking. I never thought so much about how other people think differently! I think that will help me, too, in trying to work with my people, being a leader in the Movement..." (The "Movement" refers to the Mexican-American struggle for jobs, rights, and dignity). He did appear to be thinking differently then, compared with the time we first met him.

We could not simply have looked at scores, productivity per move, or any of the simple, nonstructural, statistical ways of approaching over-all quality in order to pick TS (or the several others somewhat like him) out of group II. At this stage of our knowledge, it was only by trying to reconstruct the processes he was going through in some detail that we could get an idea of the potential. The many measures on his games could be used in attempts at process reconstruction. But there was no single, uniform pattern of static measures to identify him, or the other serious and flexible, but flawed players. Possibly a more thorough study, using interactive computer-based sequential pattern-finding analysis of subjects' process records would help to establish better methods for talent finding in disadvantaged subjects.

Now let us put TS, and other top-scoring Group II members into the larger perspective of the whole Shimoku subject population. A check of the game score plot, Figure , shows some overlap of scores with Group I. Scores of 70 to 80 could be achieved fairly readily. The difficulties of straight incremental processing tended to level off the scores, perhaps acting as a bottleneck. Our findings on the leveling off of INC players suggest that more attention to representation may be useful in educating anyone to use the full complement of his intellectual capabilities. If one representation makes heaviest demands on the most strictly limited parts of a human's information processing equipment, to learn to re-represent the problem in order to bypass bottlenecks is to permit functions idled by the jam to come back into play.

Comments on the Case Material

Material from individual case histories has been introduced in a number of places in the main text of this Report. At the end of the Introduction within this section we recommended that the reader take a quick look at selections from the raw case history material itself, before starting to read the accounts of subjects and methods, and results. Now that the conceptual framework we used to analyze the results has been presented, we refer the reader once more to Appendix C. A better sense of the relation of the raw material to the analysis that was built to deal with it can be obtained by a rereading at this point.

Motivation and the Shimoku Task

That all our subjects were volunteers who were at least moderately receptive and curious assured that initial involvement in the task was good. As the study progressed, the degrees and kinds of involvement in the moment-by-moment work on the problem came to parallel closely the kinds of subjective reward a player felt.

The subjective sense of how he was doing, and of how he might do subsequently in working the problem, was closely related to the success of a player's information processing as he monitored and understood it. The self-as-problem solver model contained feelings about what might happen, and how he might react in the situation.

Gradually, the self-task interaction model developed. In it a subject's motivations, hopes, expectations, and sense of competence were recorded and updated, as parts of the history and projected future of his interactions with this particular problem. The problem representation, as it evolved from the initial, game-as-given model of the subject, came more and more to be a product of the results of motivation to search, and of the nature and effectiveness of the search for a solution, each influencing the other at every step of a spiral process.

Motivation and Cultural Values

If performance in the task was a function of motivation, were there culturally based motivational differences we must take into account in understanding our Group II subjects? Did the experiment strike them as just one more white middle class gimmick with a potential "put-down" built into it for them? If so, that would have thrown doubt on the interpretation of their records. Or, again the Shimoku structure rewarded cost-consciousness, planning, point-by-point evaluations of many attributes before making a decision, and so on. Is it necessarily a matter of heuristics and problem-working capability, or is it a matter of different cultural values that would cause subjects not to "do well" under the circumstances?

Those questions are hard to assess fully. Here is our over-all sense of the answers to those questions. In the Appendix, the interested reader will find details based on individual cases which illustrate, or which differ from the trends reported in this section.

No one who stayed and played the series appeared to have had much sense of hostility toward, or fear of being demeaned by, the experimenters. Several who stayed were militant and angry about much that they saw wrong in white-dominated society and admitted they were ready to read insult if they were to find grounds for it. But they did not find the grounds. Group II subjects appeared to be people who had sustained hurts but were nevertheless mostly trying to be constructive when they could. Several spoke of themselves as on the fence between still trying, and giving up: "getting too depressed," or "hating like the Panthers." How they might do in the next couple of years in learning skills to cope with problems seemed to several of them one of the key questions that would help determine their direction in the society.

The computer was seen as primarily--but definitely not necessarily--a white man's tool, and, as a fascinating thing that could be used for their own purposes.[1] Subjects who said they had played badly and were made to feel "stupid" by the computer were able to see that the machine was not maliciously stacked against them.[2]

The game's "values" were taken quite straightforwardly: subjects were nearly all familiar with a variety of games that required strategy and had points, and they did not take them or Shimoku as representing alien values.

The last of our opening questions is the subtlest to answer. We can find no hints that a substantial element in Group II subjects' reactions to Shimoku was a simple, pre-emptively value-based rejection of the task, with a refusal to engage their problem working efforts fully as a result of the rejection. On cultural, educational and just plain age grounds, Group II subjects were likely to have had less experience trying to analyze how to get the best value for an action, in a complex new situation with many criteria, and to have had less experience in planning ahead in such situations. Beyond that, on a more subtle and still serious level, there is no way to factor out the life-long effects of the values of their cultures on the formation of their heuristics. There is no way neatly to separate cultural values and the development and use of certain kinds of thinking, since values prompt finding ways of guiding one's activities in the world, and support their uses. Heuristics and advanced problem-working capabilities are, largely, learned cultural products developed

---

[1] II/TS has since set out to get the skills to "build a Chicano computer."

[2] Although sometimes only after explanation of the technical errors by which they had wrought unexpected havoc (e.g., waving the light pen about the scope with the beam on).

under the promoting of values. In a study such as this, we can say simply what we find and that we are looking at skills which anyone trying to cope with, and to determine his own, and his people's own course in post-industrial American society is likely to need.

Motivation and Acceptance of the Given Goal

Having made generalizations about straightforwardness of approach, we must add that not every subject in every game was trying primarily to make the most points immediately. Playing incrementally, recognizing the double load of performing INC type play, and trying to model and to learn from it while and after playing, several subjects in each group took their early games as times for exploring the properties of the problem, in order to score better later. Several subjects in Group I, and two in Group II played partly to test their minds in reconstructing paths in space from the flat scope array, and to pursue other aesthetic dimensions of playing, more than they worked for points in their early games. By the third game, however, everyone had decided to see how he could do playing seriously for points--though often still for aesthetic pleasure, new learning, and so on.

In interviews, we always sought to know the subjects' versions of their goals, and we did not assume that the assigned goal was operating. Thus, we were able to avoid bending our notions to cover games or parts of games whose actions were in fact directed primarily by a motivation different from that imposed by the Shimoku problem statement. The analyses that appear in this Report are based on those instances where actions were guided by a goal to make as many points as a subject felt he could, unless otherwise noted.

Motivation and the Intrinsic Reward Value of the Shimoku Procedure

In discussing the motivation guiding our subjects during play, we have alluded to their wishes to explore, to enjoy aesthetic pleasure, to learn more problem properties, and new ways of tackling them, and so on. Group II subjects spoke happily of "blowing my mind" at the scope, and thinking about things in a "groovy" new way in the interviews. Even those who actually became quite restricted and conservative in their play still wanted more opportunities, and said that if they had more chances (and no time limits!), they would learn to play much better. But wait--haven't we said earlier that they often had felt distressed at seeing themselves break sc's, at getting scores they felt must be low, at being overloaded and losing their places? And haven't we emphasized that this was discouraging, saying that it often recruited memories of previous failure in problem solving. Why then should they want more?

It is probably true that facing any task which is intellectually challenging risks bringing the problem worker some painful moments. Those moments are especially likely when the problem worker has not only deficits in his problem-working knowledge but can recruit and bring to bear on each new problem the history of frustrations and defeats that resides in his model of himself as a problem solver. Therefore, it is particularly important to find kinds of problems and ways of presenting and discussing problems, that are intrinsically rewarding, and stimulating enough to make a subject able to stick with the enterprise through the painful moments. The kind of motivation which the Shimoku experiment was able to sustain, albeit for short periods, is a hopeful pointer toward possible future uses of our techniques (in forms better adapted than the experiment) for minority--and other--education. Indeed, the task was so intrinsically rewarding for almost all of Group II that most subjects offered to come back without pay any time and even to work fun puzzle kinds of problems off the computer.

The Interview and Motivation. It should come as no surprise that an interview about how one has been thinking about a problem and relating that thinking to other problem working in one's life can be very interesting, and that the experience can help to motivate people to continue and to intensify their efforts to understand their work. Yet, from what teachers themselves have told us, it is rare that they are trained to discuss students' procedures in thinking effectively.[1]

Thinking in man is continually put to new tasks, where it must peice together new ways from old--and possibly yet-to-be-invented--procedural components. If man is to learn continually, and not merely to run off old unexamined routines, he will have to be able to pay attention to the building and fitting of new thought procedures to new situations. The better equipped he is to pay attention to the building--to describe, to diagnose, to monitor, to test, to model--the better he will cope. Can we expect him to learn to build new thought procedures as well as he might if we ignore discussion of how to do it? A "finding" of this study is simply that discussing the building of new thought procedures was novel for almost everyone in both groups. For some, it seemed awkward at first, since they had little vocabulary for it and were not accustomed to the point of view. Some had defensive responses since, ordinarily, when people speak of someone's thinking it is more to praise or blame than to describe it. Neutrally valued description, however, is important

[1] Some have even been trained not to do it on the ground that students will become paralyzed by the examining process, like a man suddenly asked for a description of an overlearned and ordinarily unconscious motor response--e.g. tying his shoe. The great fallacy, of course, is that it is only that thinking which does as well by its every task which probably does not stand to benefit from examination.

to "debugging" a procedure in man as well as in a machine.[1] Most subjects caught on quickly and eventually found they were interested in, and looking forward to doing more description of their thinking about the problem. Those Group I members who thought of themselves as sophisticated professional problem solvers frequently felt a jolt on recognizing how little prespective they had on their problem-working processes. Many soon thought of instances when the lack of a descriptive approach had been costly to them. Most Group I and many Group II subjects felt that opportunity for neutral discussion was a worthwhile part of the total experience[2] of relating patterns in the approaches to the game to other aspects of their problem working in daily life. The opportunity to have an independent task followed by a value-neutral descriptive examination was seen by those subjects as worth considerably more than the task alone.

The Conjoint Session

Section 2.84 introduced the conjoint session. In the session we experimented briefly with group discussion of individuals' approaches and reactions to the Shimoku procedure. The setting was informal. All members of the group knew each other as classmates before the session. The group included both the strongest and the weakest of all our players, I/KN who played AMP games, and I/UY who played INC-d, and two who had played INC-m and INC-h games. Each subject had before him complete copies of his objective game records--print-outs, final board diagrams, and scratch drawings, if any. No subject was asked to reveal his scores or to speak about errors. No subject saw anyone else's records, unless a player himself showed one of his diagrams to assist a point he was making. In short, the individual's privacy was protected, at the same time as he was encouraged to explore with others how each had gone about working the same problem.

---

[1] We first heard "debugging" applied to human thought procedures from Seymour Papert of MIT and Wallace Feurzeig of Bolt, Beranek and Newman, Cambridge, Massachusetts, in connection with their LOGO Project. Many of our ideas about the importance of teaching description- and procedure-oriented problem working are similar to theirs--although ours arose originally from trying to teach behavioral problem solving to character disordered psychiatric patients and theirs arose from teaching mathematics. See Feurzeig, et al., 1969.

[2] Many felt it would have been more valuable had the task been still more complex and their opportunities more developed to use the computer flexibly and personally to help solve it.

It was necessary at times for the discussion leader (SK-D) to prompt the proceedings along, for the subjects were still somewhat unaccustomed to discussion of their heuristic procedures. However, the subjects themselves handled most of it, and proceeded circumspectly--avoiding trying to seem bright at each other's expense--to learn how each had formed a series of solutions to the problem.[1] The subjects worked, largely on their own, for more than three hours trying to consolidate and to deepen their knowledge. They reviewed initial reactions to the overwhelming number of possibilities, hypotheses about valuable generalizations and rules, problems of testing them, ways in which ideas of nonincremental solutions had arisen and had been either articulated or dismissed without evaluation, and so on. The review of their detailed records allowed them to relive parts of the experience, while putting the whole sequence of learning to play in a single perspective. A halt had to be called near midnight.

Subjects were told that this session was just to see whether group discussion was appropriate for following up work on a task like Shimoku, and their criticisms and suggestions were solicited. They were emphatic that the session had been valuable to them and said that it had helped them to described and to capture elusive and interesting processes. They added that having a long talk with a faculty member about how their individual heads and feelings worked with any problem would have been great. But having played Shimoku had enabled them to engage in such a discussion as individuals in a group. It had given them a common ground which could serve as a basis for very precise reference and communication, and had thereby enhanced their ability to learn not only from an interviewer or instructor but from each other.

3.2.4 Critique, and Suggestions for Future Research

In this section we shall confine ourselves to comments dealing with the limits of the current study and with suggestions for future research.

Features Limiting the Present Study

Because the experimental work with subjects was a pilot study and funds were limited, many technical features were less satisfying than we should have liked. This account works around them. But we should have liked to have gone further and, in some cases, to have made analyses which would have required more precise control of conditions than was possible. First, the terminals,

[1] Although these subjects were used to learning how each other had conceived design solutions to an assigned problem, the problem statements had usually been so broad and so free that no direct investigation had occurred of how others' minds dealt, condition by condition, with a problem, having complete information, and all structure and no art in its definition.

though shielded, were not private, and the subjects were occasionally interrupted. Funds were not sufficient to permit dropping from the sample subjects whose records had small interruptions from men or from machine troubles, although a number of games and subjects were dropped when greater difficulties supervened. High costs kept the sample sizes smaller than we should have liked. The pilot nature of the work and the fact that the computer on which it became inaccessible prevented developing computer-based analyses of the printouts which would have allowed less tedious and more refined examination of the records than could be done by hand practically.

Features at Once Limiting and Beneficial

We have mentioned that work with subjects who were not preselected for articulateness and competence interfered with our understanding at times. However, there was a worthwhile gain in access to sorts of subjects different from those typically being used in information-processing studies of complex problem working. To appreciate the importance of parts of a problem-working process that might be taken for granted when working only with the skilled, we believe it is worthwhile to observe as well the processes of those who are not. Having a spectrum of the sort provided by our subjects was valuable--each group's behavior threw the other's into relief and at times, by overlapping, showed what was a common bottleneck in the task.

Similarly, foregoing the think-aloud technique afforded the opportunity to study other ways in which modeling the self-task in interaction could be critical to the heuristic preocesses for finding alternative problem representations and new procedures for solution. Questions of how well a problem worker knows how to use his own resources are important and little treated elsewhere. The time limit made those questions more poignant. Frequently when think-aloud protocols are used, other interviewing is omitted. While the post-game interview is time consuming and hard to code, it does broaden the view of what was going on in the subject. It allows the experimenter to start to relate the information-processing events of the problem-working process to their context in the person. Hence, it provides a beginning for relating ego and information-processing psychology. It adds to what a think-aloud protocol might provide for understanding subjective, experiental aspects of information-processing events.

Suggestions for New Research: Implementations and Conditions

Increasingly, new projects have the stated purpose of enhancing human functions in complex problem-solving tasks by allowing them to interact with sophisticated computer assists. Increasingly, it is recognized that more knowledge is needed of the ways human minds function in complex tasks. That knowledge is wanted to improve the education, and the working competence, of humans at many levels in a post-industrial society. And it is needed to help build better computers.[1]

[1] "There is a failure of (man's) self-observation: most of our inability to write intelligent computer programs is our inability to understand ourselves." Professor John McCarthy of Stanford, in a talk on June 4, 1971 at UCLA.

As things are now, new sophisticated computer assists for men rarely are examined systematically for the ways they affect the thinking of humans working in concert with the new tools. And although new interactive systems exist, their potentials to help generate basic scientific data on the precise details of human problem-working function are under-used.

Ideally, new research using man-machine systems to study human complex problem working and to assess machine assists to human function should:

1) give working subjects reliable, priority access to machines under controlled laboratory conditions suitable for psychological experiments. Most computer rooms do not fit this condition, and most systems not dedicated to careful psychological experimentation have enough delays to disrupt a subject's thinking and add noise to the data.

2) compare both time-limited and non-time-limited studies of work on the "same" problems--studies with, and without think-aloud--with physiological measures to find objective concomitants of hypothesized mental events and, where the problem is suitable, with eye movements.

3) provide for computer-based analysis of protocols, not merely for "horizontal" measures, such as productivity ratios and score counts. They should provide for the investigator's working interactively with subjects' procedural data to find sequential event patterns related to the inferred "programs" of subjects.

4) allow the subject to define new functions, and to devise easily some new ways in which the computer can be made a better tool for his own purposes.

5) provide the investigator and the subject with ways of going back over the recorded events, making them the topics of systematic interview discussion.

6) use two-person and other team arrangements, where both (a) the problem-oriented information processing, and (b) the communicative description and negotiation aspects of a task involving judgment and strategy can be studied.

7) allow experimental arrangements to be systematically altered to isolate (a) effects of problem conditions on subjects' processes, and (b) effects of the structure of interactions between subject and machine, and between subject and subject, on a worker's ways of dealing with the problem conditions.

Many investigators[1] have recognized the importance of the foregoing in doing systematic work on complex problem solving. Elements of the list can be found in a variety of projects. We should like to be able to do concerted work by piecing results from systematically related studies, using a variety of techniques no one of which is adequate alone. In the Shimoku project, the severe limitation on ways a subject could get the computer to assist him in handling the task made the task conditions, throughout play, for each subject more nearly uniform and dependent on their individual, relatively unassisted mental functions. But it prevented our seeing how subjects could learn to build a problem-working approach that put together human and machine information-handling strengths complementarily in tackling the problem. The ability to build helps for himself is one of man's most interesting characteristics. Given the full use of advanced Gaku concepts and of UAL's extensible language capabilities, we should have liked to study ways that humans diagnosed their needs and generated designable computer functions which could have remedied the felt lacks.

One questionnaire item elicited many suggestions from subjects on computer helps. They would have liked a function to mark sc-participant tokens, one to scan paths, one to analyze structure-enrichment possibilities, one to permit rapid examination of alternative structures on-line, etc. Their suggestions were just the first few that came to mind from struggling to work with their own limited immediate processors and with the limited scope options provided. Much more invention--possibly worth using in later system development and in teaching problem solving--can be expected to become evident in a more flexible task environment.

We should like a task environment which, like the Shimoku version used, would be able to engage people easily and at a number of levels. But unlike our current problem version, it should have more potential for developmental variation of the problem-working process and for growth in depth. In such a situation the same people should be studied over time as their thinking and their skills grow in relation to the problem, or problem area.

Suggestions for New Research: Key Issues and Events

Problems having potential for strong adaptive and strong high coordination conditions in their solutions need to be better understood. Many of those problems may allow more leeway than a problem solver realizes in representation of the task. They may be treatable as D, F, and quasi-F problems--depending upon the problem worker's views of "the real nature of the problem" and upon

[1]To name just a few whose laboratories are currently dealing with one or several items on the list: Shure (1969), Tikhomirov (1970a, 1970b, 1971); W. Grey Walter (1965); Reitman (1969), and Newell (including current studies mentioned in talks at University of California, San Diego in November 1970, and material to be covered in the forthcoming book, with Herbert Simon, see reference under Newell (1967)).

the heuristic procedures he uses to remodel his problem-working processes and his goal statements successively, through cycles of search for improved representations. We can understand both the tasks and their handling better than we do at present, by examining the nature of a worker's representation of a task, the factors that go into his building it, the demands that it places on efforts to execute a solution using it, and the ease with which the critical events can be modeled for learning and potential improvement.

In the Shimoku situation, the transition to a representation employing one of a set of candidate solutions in a theoretical space could be traced quite closely. The transition entailed getting a novel representational element, the high interlock large structure. But the search for it was stimulated, and shaped by subject's criteria for an information handling device. It was shaped by criteria for a representation which would help them to treat the entire "practical" problem with available time, move, and immediate-processor resources. The conditions on the practical solution were not fully articulated in the information given to them in the original statement of the problem. The problem conditions, that acted effectively as conditions on the solution, came from the subject's modeling the set of interactions currently, and potentially, occurring in the entire problem-working system consisting of the man, the machine-task embodiment, the original task description, and time. The "inspired leap" to a new representation had to be guided by modeled properties of this problem-working system, taken as a whole, as well as by properties of its parts. The microevents of the leap, were sculpted, traceably and sensibly, by a subject's applying criteria derived from the properties of the problem-working system.

By examining closely the event sequences whereby some workers come to change their representations (and others start--or "have the idea" in some weak sense--but fail to proceed to change theirs), we can tease out procedural elements of the critical representational transitions on which much of creativity in problem solving appears to rest. The closer we can get to the microevents of the transitions, the closer we shall come to being able to teach procedures by which more humans can guide themselves more of the time to what we now call "inspired leaps".[1]

In his discussion of space "the transition from the initial representation of the formation problem to its improved representation", Amarel (1971, p. 464) notes that the first step, is finding "a theoretical model of program space." He points out the significance for advances in artificial intelligence of studying transitions to new models.

Processes of model finding are exclusively in the domain of humans at present. In order better to understand these processes and to advance their possible

---

[1]This view agrees with Polya's view that for the person who does not know how to handle it at the start, solving any problem involves creativity and can be made a pleasurable challenge (Polya, 1957).

mechanization, it would be most helpful to study them in a man-machine interactive environment. This would force us to clarify the nature of models and of their relationship with the phenomena that they are intended to model; it would also induce the development of appropriate descriptive tools for computer manipulation of models, and the identification of key processes that occur in model-finding activities (p. 465).

We should like to focus on the microevents of representational transitions in next stages of our work on complex problem solving.

Another, related focus for further study is analogy formation and use. We have mentioned the faulty procedures for analogic thinking that wasted potentially good ideas of our weaker subjects or burdened them with bad ones. The use of analogies as a basis for predictive tests of high-level academic ability is well known. We are thinking now of a variety of ways to use computers interactively to elicit, to describe and model, and to teach effective procedures for analogic thinking--and for weeding out covert erroneous imports into the models of new situations by analogy. A system which would assist a subject to learn procedures for effective, and to curb routines for defective, analogic thinking might reach a critical deficiency in students like subjects II/TS and II/DY, whose weaknesses have a large procedural-organizational component.

In the next section, on practical applications, a few projects are suggested, which should, at least in early stages, have a research component as well.

3.2.5 Possible Practical Applications

The practical applications suggested here, and the ones already mentioned--implicitly or explicitly--must be considered in the light of the preliminary nature of the study and the possibility that the confirmations of results of our small sample and more sophisticated sequel steps may be necessary.

Suggestions for Education of Planners and Executive Decision Makers

Instead of being run as an experiment, a modified form of Shimoku, or a similar but more sophisticated task situation (possibly including elements of uncertainty), could profitably be run for teaching and for diagnosing the problem-working skills of planners and executive decision makers.

At its present level or, even more, in an advanced version that would permit a subject more inventive ways to use the computer to assist him, a task like this could be used as a sensitizing device, alerting the student to ways he characteristically estimates, sets goals, assesses the economics of time and information-processing resources, uses the contents of his self-as-problem-worker and his problem-working-knowledge models, formulates new representations, regulates his execution of an error-prone and complex solution procedure, and reworks initial formulations in order to learn. Merely playing, of course,

would not be enough. There would have to be opportunities for analysis and critiquing sessions.

A teacher's role--in stimulating a student's awareness of his own information-processing practices as they interact with his developing notions of the task--could be more active than that of our interviewer in the experimental situation. We would suggest that teacher and student work directly from the game records, encouraging habits of description of the student's own planning procedures and the corresponding action sequences. The student's descriptions should characterize and comment in such a way as to make clear aspects which are erroneous, useful, changeable, and so on.

Putting the student into a teacher's role with other students and arranging for reciprocal role switches in a supportive and noncompetitive setting might be a very useful way of giving him a comparative perspective on mental functions. The switch could illuminate the differences in his own and other people's ways of thinking. That sense can be very useful to the student who, in his role as executive or planner, must be able to communicate in problem-oriented fashion with other people in a team or in a community who may process the givens very differently.

It is only a small step to realize the potential usefulness of this kind of technique--studying a detailed trace of a student's problem-working effort with him--as a device for teacher training. It is often very difficult for teachers to re-create from an erroneous result, the procedures that took the student astray. In consequence, teachers themselves are often more answer- than procedure-oriented in their corrections, to the detriment of the teaching-learning collaboration. The approach suggested is likely both to teach the teacher how to work more effectively and to give him the wherewithal to do it.

Teaming working pairs of students who would have to articulate and agree upon their procedures would engage them in a double-sided, problem-oriented and inter-personal negotiation task suitable for later analysis.

The use of conjoint sessions following individual play and analysis appeared to be potentially useful in furthering all the kinds of learning. The sessions, like our single example, could be run with subjects, having their own results in front of them, being encouraged to question and to contribute to the discussions of how each saw, and went about forming an approach to, a new problem. Our group procedure followed a procedure less open than the reciprocal role-switch teaching suggested above. It showed that a great deal of information could be shared without embarrassing a very weak subject despite the presence of the strongest player.

In discussion with subject I/KC, the idea emerged to use individual Shimoku, followed by interviews and a conjoint session, as a management device for helping to establish executive task- or problem-oriented teams. Such a procedure would help to stimulate a meta-level, descriptive approach in which

each member of the new team would become aware of his own characteristic heuristic procedures and their similarities to, and differences from, those of his new team-mates. The involvement most players feel, and the recruiting of large portions of the self as problem solver models, would bring much meaning to the task. For this to be useful, the time constraints, underspecification of the goal state, existence of a plethora of "solutions" as the task is defined, and the complexity, and unfolding character of the problem all are important.

Suggestions for Education of the Disadvantaged

We have pointed out earlier some reasons that problem workers whose skills are weak for representing problems, and for controlling the execution of solution procedures might benefit from exercises based on the principles discussed in this report. Many of our Group II subjects, and the weakest subjects in Group I, commented that the Shimoku situation made the usually vague and abstract notions of planning more vivid and concrete to them than ever before. Many of them had no use for "set" problems--as they called arbitrary and static problems presented in books. And many of them had such complex and hostile relationships with teachers that they "turned off" at ordinary discussions of their errors. The present format, which promoted involvement, action, reflection, and reinvolvement appeared promising. However, much more specific guidance of student discussion, relating problem working techniques to printouts, and suggesting alternatives, probably would have been necessary to keep the more easily discouraged and self-defeatingly disorganized subjects in a real learning situation.[1]

Possible Applications in Community Planning Work

A number of efforts are under way, at the UCLA School of Architecture and Urban Planning and elsewhere, to bring technical "experts" directly into give and take with local community people in disadvantaged neighborhoods. They seek to co-plan for local amenities, health and education facilities, etc. Local issues are often heavily charged with feeling, and local people are often unaccustomed to think in terms of cumulative effects of events represented in short and long range plans. Other ideas in the development stage include a mobile capability which could take a computer terminal into the neighborhood, for use in demonstrating and engaging participation in co-planning, using model situations. Such a terminal would connect to an urban laboratory equipped to

---

[1]Since the Shimoku project, we have worked with four of the same Group II subjects in a computer-based computer programming course and have found that two of the four, especially, needed continual extra work on procedure description, heuristics, and overcoming negative contents of their models of themselves as problem solvers. Many of their peers in the computer course had similar needs.

handle models and gaming.[1] Again, a neutral and enjoyable task that provides concrete referents for discussion of ideas fundamental to planning could be an interesting start to co-planning activities. The techniques and the lessons of the Shimoku experiment could be applied in the model task, and in subsequent procedure- and representation-sensitive planning exercises. Giving more neighborhood-relevant semantic content to the elements of a modified task might make bridging the gap to the real-life issues easier--as our subject I/KN, who has experience in neighborhood co-planning, suggested after the experiment.

Possible Practical Applications of the Descriptive Framework

So far, we have noted ways that a Shimoku-like teaching or sensitizing procedure might be useful. We have always assumed or specified the presence at Shimoku interviews and discussions of someone who could facilitate the self-questioning and the descriptive, modeling processes which appear important to assure the usefulness of the technique for many people. We have noted that teachers might be trained using this set-up and procedure.

But the sensible skeptic is likely to ask whether it does not take a special frame of reference and special knowledge to be effective as a facilitator of self-corrective, descriptive modeling processes. He might say that most teachers, even with exposure to the set-up themselves, are likely to need a new outlook on human information processing and on procedure description to be able to function well.

Most good below-college teachers with whom one of us (SK-D) has discussed thinking in problem solving have a deep feeling of need for a coherent descriptive framework for considering possible events in the minds of their students. A number of them who have tried using computers to teach mathematics have had flashes of a sense that they could carry on detailed dialogues with individual students, referring to the programs they have written to solve problems. But the teachers have said that they greatly under-use this approach because they do not themselves know a useful way to think about crucial issues in problem-oriented thinking.[2]

We believe that the kind of representation and procedure-oriented descriptive framework developed in this report, coupled with guidelines on cognitively oriented discussion technique, could be extracted and adapted to help introduce teachers to ways of speaking about the procedural and representational aspects of problem solving. The adapted version would, preferably, supplement concrete Shimoku-like task experiences of their own.

---

[1]The notion of this sort of Urban Laboratory was originally proposed by Professor Peter Kamnitzer at the UCLA School of Architecture and Urban planning.

[2]A helpful source of information of this sort has been Dr. Robert Haven, Director of Project LOCAL (Laboratory Program for Computer Assisted Learning), a sponsored and jointly funded by five communities in suburban Boston.

3.2.6 Summary of the Shimoku Experiments

In a preliminary study, Shimoku, a new one-person game played interactively on the computer, was used to examine the problem-working activities of individuals in two groups of subjects: 20 technically trained professionals and graduate students of design (Group I), and 14 ghetto and barrio youths (Group II). The task was designed to elicit, in a controlled situation, some key aspects of real-life problem working done by planners. It showed (a) how workers' different statements of the "real nature" of the same problem depended on their search for representations; (b) how the representations selected imposed different information-processsing (IP) demands; and (c) how these selections led to different procedures for forming solutions which, in turn, attained different degrees of success.

Subjects could not use all the information available but had to deal with a larger space of possible solutions than they could enumerate and evaluate in the situation. Time and move limits curtailed, drastically, the extensive evaluation needed to play well incrementally and to carefully adapt new actions to existing conditions at each step. Subjects were put to the first of three one-hour plays right after initial instructions were given, making accessible parts of the learning process. Step-by-step game records, traced and kept by the computer, provided the experimentors an objective basis for detailed descriptions of the subjects' actions. In addition, attempts were made to infer IP activities which could have accounted for the game data collected, with questionnaire and interview materials. The assigned goal was so stated that any game in which a subject made as many points as he could was a member of the class of solutions. Highlights of the account are summarized below:

1) Objective game records alone could be used to group the sequences of actions which constituted subjects' solutions into three main types and their subtypes: Incremental (INC), Master Planned (MP), and Adaptive Master Planned (AMP).

2) INC games constituted more than 85% of the total--including all games played in Group II and most in Group I. INC games scored -63 to 110 All master games scored over 110, with MP peaked at 185 and AMP at 241 points. Thus, games describable as generated by different types of procedures scored in different point ranges. The different procedures had detectable consequences in the effectiveness with which elements of the problem situation could be used to attain solutions.

3) The notion of Arrived stage, or mature, games permitted separating the requirements for merely interpreting and executing a previously worked out game prescription from those for playing while learning to make strategy formulations for guiding subsequent play. Three composite, prescriptive, Arrived stage instruction sets were made in substantially the forms, and at the levels of description, that subjects used to

direct themselves in the best games of the three main types, A/INC-h, A/MP-c, and A/AMP-c. IP requirements for a player's interpreting and executing each type of prescription for play were examined.

4) The A/INC-h game could be prescribed by (a) a short, recursive, incremental program and (b) a largely unordered collection of rules of thumb suggesting ways to modify the interpretation and execution of the basic incremental program. Attempts to execute high quality incremental play seriously overloaded highly intelligent players and, in weaker subjects, led to drastic simplifications and disorganization. Attempts to apply instructions for playing a good INC game ran into difficulties that are typical for a human using an incremental framework in a task whose final reward hinges on the quality of complex coordinations achieved in a final state.

5) In contrast, the prescriptions for A/MP-c play directed executing a totally preplanned large structure diagram on the board, according to an easily operationalized program. Little adaptation to board conditions was attempted, and imposition of the master plan could be performed in a nontaxing, very nearly algorithmic fashion. MP-c players, while considering their games optimal, did not score as high as the AM-c player.

6) The prescriptive instructions for A/AMD-c play integrated strong features of both the other types of games: (a) rules of thumb to guide the on-line processing needed to play adaptively and (b) a preplanned--but here flexible--structural scheme. Like the INC-h game instructions those for the AMD-c game brought a heavy load of on-line processing during play. But in contrast to INC-h, they assisted the player in reducing the load.

7) The prescriptive instruction sets for Arrived games of each main type showed different representations of the problem, usable in different ways to guide action. The assigned goal statement was felt by most players to be too under-specified to guide play. But INC players thought a more specified version could not readily be found. In the goal statement that they used, all of the entities and relations necessary to achieve the goal were part of the "givens" of the problem. The INC program formation task--to devise a procedure which would start from the "givens" and arrive at a game result consistent with the INC goal statement--was handled without introducing a novel representation.

8) In contrast, no master range (110 +) game was achieved without introducing a new subproblem: devise a procedure for finding a better goal statement. With one serendipitous partial exception, no improved description was found without conscious search. No sufficiently articulated new representation occurred without a subject's conscious protection of his search for it against time pressures. No search succeeded without postponement of initial intuitive evaluation suggesting that a preplanned solution would fail on economic grounds.

9) The account of steps taken in attaining new representations borrowed Amarel's analysis of the program formation task for derivation (D), formation (F) and quasi-formation (quasi-F) types of problems. It was found that all subjects who had represented the Shimoku task as a D problem with a weakly specified goal statement played INC games. That is, they played by an incremental strategy that impeded the coordination it tended to seek.

10) Subjects concentrating on a strong scoring-pattern-interlock condition and de-emphasizing adaption could represent Shimoku as an F problem--one requiring (a) discovery of a novel representation in a language of large structures not given by the experimenter's problem statement; (b) generation of candidate solutions in that language which could express structurally the problem conditions placed on the solution as a whole; (c) testing and articulating the class of candidate solutions against all the problem conditions (those stated as properties on the whole and those imposed by the "givens") in order to converge toward an acceptable solution; and (d) finding a representation of the program scheme in a control program which could execute it, now transformed into a new D problem, by taking parts of the "givens" into parts of the newly structured solution. MP-c players saw Shimoku as an F type problem.

11) Shimoku could be represented as a quasi-F problem by a subject whose AMP-c game statement retained both strong adaptive and strong scoring-pattern-interlock-maximizing conditions, after setting each aside temporarily during formulation in order to apprehend and to represent the theoretical solution requirements of the other. The quasi-F representation had been achieved by repeated cycles of subject I/KN's modeling his own information-processing capabilities in relation to his playing experience, in order to achieve successive representations of the game.

12) The task of playing while learning to play was considered in light of the IP requirements not only for executing but also for modeling games of the three main types, Not only were INC games harder to execute well, but they were harder to model effectively to foster learning. Learning while playing and learning between plays proceeded best in the presence of conscious modeling of game events as related to a player's own IP capabilities.

13) Several programlike models used by subjects were considered in our attempts to account for more aspects of Shimoku events. Subjects' models of themselves as problem solvers, and of their problem working knowledge and heuristics, contributed in traceable ways to their building of game models and to their modeling of interactions between themselves, with their IP capabilities, and the given task.

14) Difficulties characteristic of weaker players' problem-working activities were summarized. Their vulnerability was discussed in terms of deficits in skills for representing and monitoring the execution of problem-working activities. A question possibly identifying a special subgroup of serious, representation-sensitive ghetto youths was found.

15) A conjoint group discussion was used on a trial basis, with good immediate results, to promote additional learning from the Shimoku experience.

16) The moment-by-moment interweaving of subjective experience during problem working and whatever task-oriented information processing was done by a given subject was shown to be very close, important to understand, and capable of being examined within the same broad IP framework that included the task.

17. Evaluation and critique of the study and suggestions for further research and for practical applications in planning and in education completed the presentation.

Appendix C contains original case material and the questionnaires.

4. PERIOD 3: EMPHASIS ON MAN-MACHINE PARTNERSHIP (by Aiko M. Hormann)

The experience and insights gained from the work performed in Period 2 set the tone in Period 3 for the greater emphasis on adaptive planning and on man-machine interactive, mutually assisting work, aiming toward the stage that can be called man-machine synergy.

However, man-machine cooperation is not needed (nor even desired) in many real-world situations in which man alone or the machine alone is better suited. For practical purposes, therefore, it is important to identify and characterize real-world problem situations in which man-machine techniques are needed or from which substantial benefits can be expected. During this period, many classes of real-world problem situations were examined and characterized, and the types of decision dynamics influenced by these characteristics were identified. Attempts were made to identify interdependencies of man's capabilities and limitations and the machine's potential capabilities and limitations. The design of Gaku, as described in Section 2, has undergone changes to incorporate desirable new features and remedy limiting factors that were identified.[1]

These topics are covered in detail in Hormann (1970), therefore, only highlights of Period 3 are summarized in the following sections.

4.1 CHARACTERISTICS OF REAL-WORLD PROBLEMS AND DECISION DYNAMICS

Many real-world problem situations are complex mixtures of well-defined and ill-defined subproblems with both constant and changing conditions. These subproblems are often interrelated and their solutions interdependent so that treating them in isolation by purely analytical/mechanical methods or by purely intuitive/rule-of-thumb approaches would produce inadequate, misleading, or disastrous results. These situations are characterized as complex, unclear problem boundary, and evolving types. Such types and their ill-defined natures would make it difficult for man to analyze and integrate available information, which is usually a mixture of factual and value-oriented information; qualitative and quantitative measures; and both objective and subjective judgments. Decision makers in such situations typically face:

- Multiple objectives and constraints that may change in time and in context;
- Incompletely known alternatives, requiring an inventive approach to generating alternative courses of action;
- Difficulty of estimating consequences of alternative courses of action because of complex and obscure cause-and-effect relations that often create unexpected side effects; and,

[1] Implementation of the new design of Gaku depends upon the implementation of User Adaptive Language (UAL), the major portion of which has been completed.

- Difficulty in evaluating alternatives in terms of the stated objectives because of the lack of a fully defined utility function that takes all the important factors into consideration and because of complex trade-off implications.

We have glimpsed in the Shimoku environment what man tends to do when complexity exceeds his capacity to cope with it--oversimplification, disorganization, and premature conclusions often result--the principle of "cognitive economy" is in operation. This principle seems to be a function of the degree of complexity, of time pressure for decision/action, and of preparedness of an individual in a given problem situation. Ways to reduce the cognitive load were recognized to be by master planning (especially adaptive master planning) and by delegating to the machine those functions that are specifiable and frequently used and those that require extensive search and/or bookkeeping.

4.2 EMPHASIS ON ADAPTIVE PLANNING

Adaptive planning[1] explicitly provides allowances for adapting to, or capitalizing on, present conditions and possible future changes in the environment (caused by the decision maker or by outside forces); incomplete information; and unexpected leaps of inventive ideas. It also promotes progressive adjustments in the time stream. Techniques used for this adaptive planning are "partial specification"--leaving some unspecified portions to be filled in later--and "aggregate specification"--specifying decision rules (or, at a higher level, the way rules may be generated or modified) by generalizing conditions and processes.

In the man-machine context, these techniques can be used to bridge the gap between the impreciseness of human thinking and the preciseness and completeness required by the computer. Complete formalization is not necessary at the outset. Man can finesse the computer restrictions by partial and aggregate specifications and supply details later as they become available and/or clarified. For the exploratory and conceptual phases of problem solving, discerning "promising" avenues of possibilities and "relevant" criteria for evaluation is in man's province, but the exploration and evaluation phases can be done more efficiently by the machine once they are specified. This intermeshing of the two capabilities can be promoted to the fullest when we understand the many-faceted aspects of cognitive economy and partial and aggregate specifications.

---

[1] As used here, adaptive planning is a general descriptive term and does not refer to the specific term AMP (adaptive master planning) used in 3.2.3.

4.2.1. Hierarchical Nature of Planning

The hierarchical nature of planning processes comes into existence through two related aspects: the evolvement of plans (from the general to the concrete) and the hierarchical subdivision of the task.

Evolvement of Plans from General to Concrete

Planning usually starts with the conceptualization of a given problem situation and, as such, is an aggregated, low-resolution consideration of various factors ranging over the full scope of a situation. As understanding of the problem increases, more details are added and vague or fuzzy concepts are clarified. Then, as more variables become identified and relations among variables become better understood, a reasonable separation may be made between decision variables (those the planner can or chooses to control) and other variables (over which he has little or no control) that also determine the outcomes. This process of progressive articulation and substantiation, including adjustment to changing enivironmental conditions, continues down to the very last step of implementation. It is a continuum of transitions although it is commonly divided into four stages for convenience. These are the conceptual, definitional, developmental, and operational stages (Figure 25).

Subdivision of the Task

For a large-scale, complex situation, it is important that the planner divide the problem into its subparts, each of which is presumably easier to manipulate and anlyze. After the separate subparts are examined in appropriate detail, resulting plans of attack for each are synthesized to furnish insights into the original problem (Eastman, 1969; Green, 1969). This process of subdivision is practiced both in individual problem solving and in group planning.

Typically, in a hierarchical organization, the planning task is broken into a number of subtasks, which are assigned to particular division heads who, in turn, subdivide their assignments and delegate portions to department heads, and so on. Subplans generated are successively channeled back to be coordinated into an integrated plan.

These processes of generating subproblems and attacking them separately are schematically depicted as small pyramids and subpyramids scattered within the outer pyramid shown in Figure 25.

4.2.2. Gaku's Framework: Composite Structure

Gaku's framework is shown in Figure 26. The guiding mechanism for interaction, known as a "user-modifiable feedback loop," is shown in the top "cut"

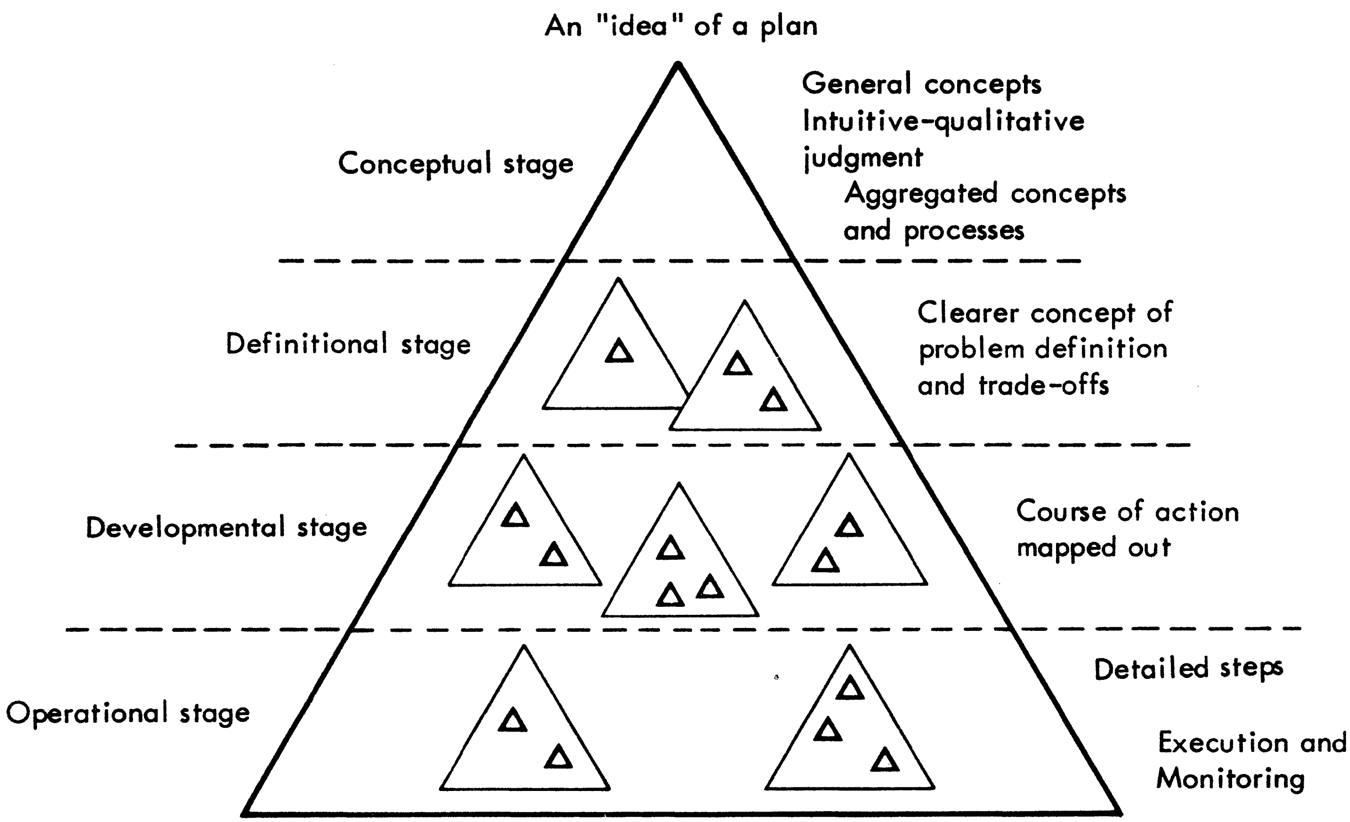


Figure 25. The Hierarchical Nature of Planning

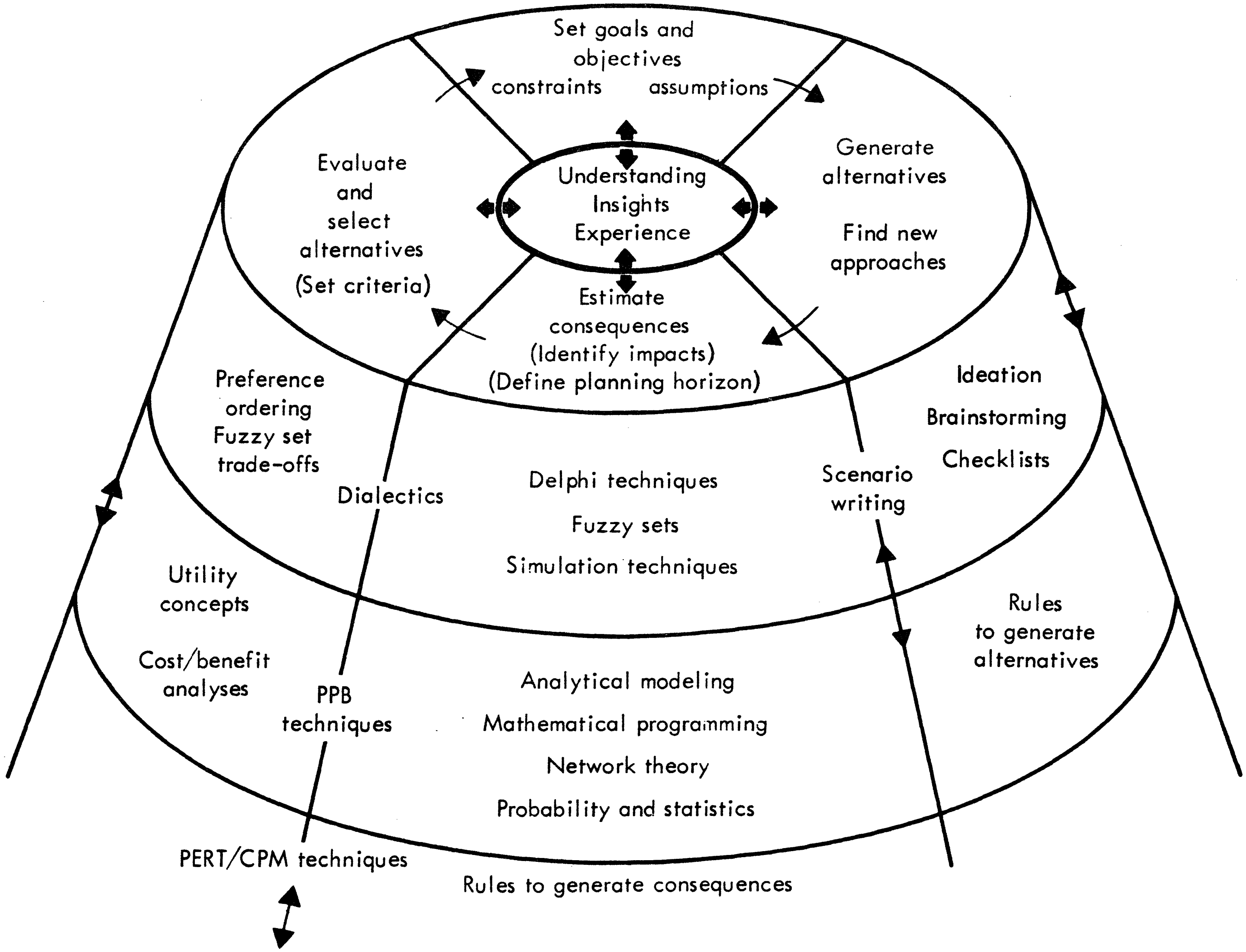


Figure 26. Gaku's Framework

surface of the cone-like structure. This feedback loop has been derived from the following four-step decision cycle commonly used in planning:

- Define the objectives and formulate a utility function (where possible;
- Enumerate possible alternative courses of action;
- Identify the consequences of each alternative;
- Evaluate the consequences in terms of the objectives and choose the alternative which best achieves the objectives.

The four sections at the top with clockwise arrows indicate the usual sequence of steps, but two-way arrows connecting the center circle with each of the four sections indicate that any section may be revisited at any time before completing the visits to all four sections in sequence. The tower-like appearance of the structure suggests the hierarchical nature of planning--the higher on the cone, the nearer to the conceptual stage.

A variety of decision-aiding tools and techniques are shown along the outer wall of the cone, merely to indicate that appropriate decision aids should be made available to the decision maker. These decision-aids must be geared to the needs occurring in the decision step and the stage of the planning hierarchy the decision maker is currently in. A rich assortment of tools, rather than a single powerful tool, appears to be needed. Some of the tools and techniques, such as statistical routines and the Critical Path Method (CPM), have been generalized and are available as "library" programs (Montalbano, 1965; Shure et al, 1967). Other techniques, however, will have to be tailored to fit a particular situation or devised during the actual involvement of man with the system.

Figure 27 depicts the various aspects of subtask handling which are represented by small cones within the main cone. The center "core" of the cone is shown here to indicate that some common resources and techniques are made available to man-machine teams at various levels of interaction. Man-machine teams themselves are included here since their actions and the information they generate are also used as resources and techniques.

If an individual has generated a number of subproblems and is attacking them one at a time, only one mechanism at a time within the outer cone is needed. Moving from the original problem to one of its subproblems is done by recursive use of the guiding mechanism. Upon receiving a new subproblem, the mechanism reapplies itself, using a push-down list to keep track of the various levels of activities in which the man is engaged. On the other hand, if a number of planners are using the system simultaneously, a copy of the guiding mechanism is created as needed (not prestored) for

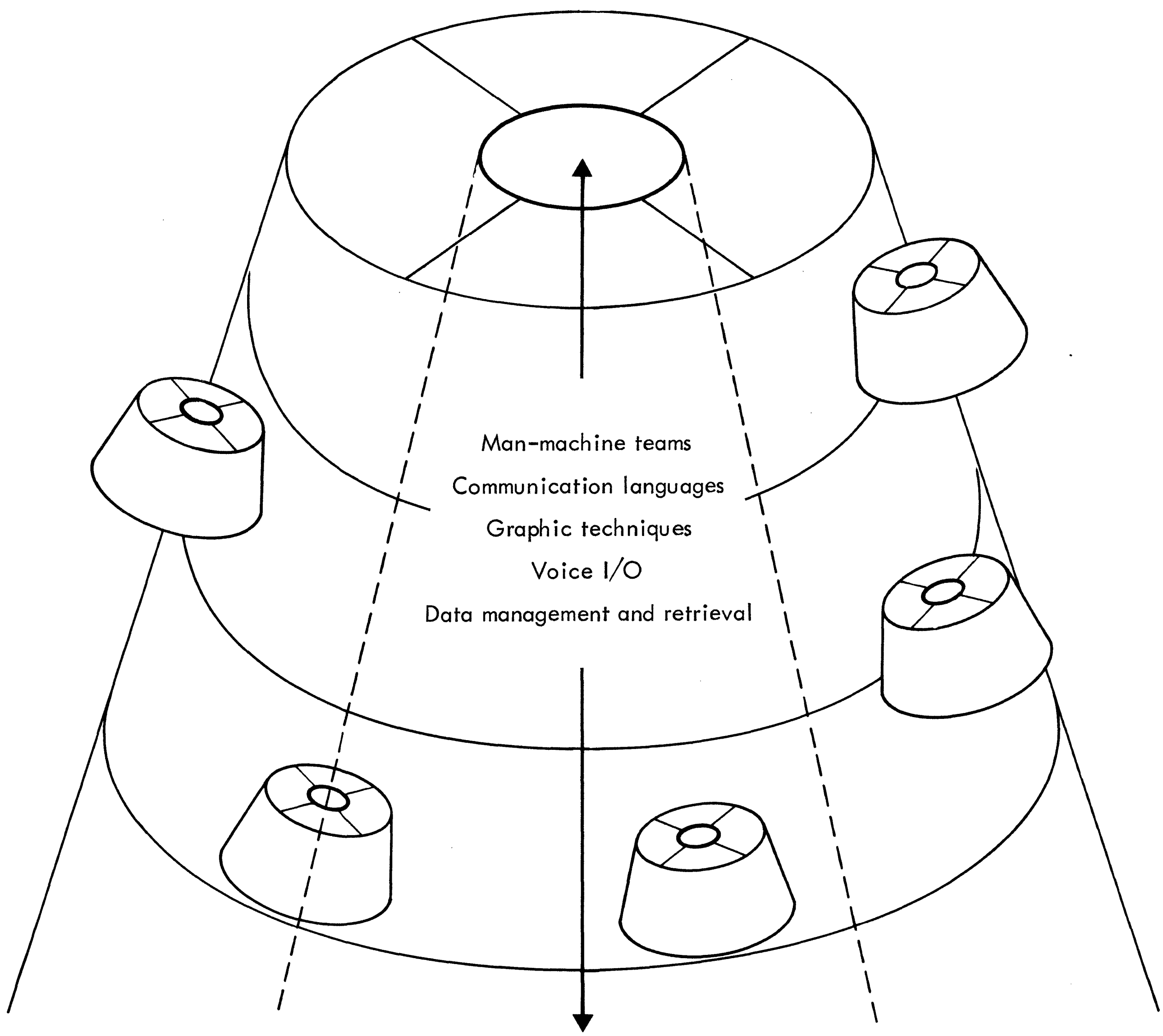


Figure 27. Substructures for Team Planning

each individual, who can generate and handle his own subproblems by the recursive use of his own mechanism. All such guiding mechanisms, one for each individual, can operate simultaneously and independently of each other but can also contact and influence each other through the communication channels. Such channels or links among these mechanisms are usually specified in advance, but new links may be established during planning activities and old ones altered or deleted.

Ideal conditions toward which we strive are those in which information needs are met all levels, and coordination and integration of subplans are easily achieved through effective communication channels throughout the hierarchy. The man-machine system can then allow higher-level planning to be broad and comprehensive while taking advantage of the detailed knowledge and specialized skills available at the lower levels. Few people can be expert in many varied fields, but group planning in the man-machine context can promote multidisciplinary synergism.

Eventually, the man-machine synergistic approach will lead to the faster generation of higher-quality plans and that it will open up new possibilities of dealing with complex, changing situations that have heretofore been inaccessible to computer assistance. In an evolutionary process, we need not look for an exact "solution" once and for all or for a single massive proposal that takes into account all possible events and consequences. It is more important to be able to indicate the relative merits of the various schemes and identify variable critical factors. Incompleteness and impreciseness of first analyses can be corrected and refined iteratively in an adaptive plan generation, which will continue to evolve with changes in the environment, in technology, in human value systems and in our understanding of the problem.

ACKNOWLEDGMENTS

We wish to acknowledge the technical contributions of David C. Crandell and Antonio Leal, System Development Corporation, in implementing the Shimoku design and part of the Gaku design on the AN/FSQ-32 time-sharing system and for their work on the User Adaptive Language. We are also grateful to Erik Andresen, School of Architecture and Urban Planning, University of California at Los Angeles, for his photographic work during the Shimoku experiments.

We wish to thank all the colleagues who have helped us with suggestions, criticisms, and encouragement, and all of those who influenced our thinking through publications beyond those referenced here. They are too numerous to be mentioned individually.

REFERENCES AND RELATED READINGS

Alexander, C. Notes on the Synthesis of Form, Cambridge, Massachusetts: Harvard University Press, 1962.

Amarel, S. "On the Mechanization of Creative Processes," IEEE Spectrum, Volume 3, No. 3 (1966), pp. 112-114.

Amarel, S. "An Approach to Heuristic Problem Solving and Theorem Proving in the Prepositional Calculus," in J. F. Hart and S. Takasu (eds.), Systems and Computer Science, Toronto, Candada: University of Toronto Press, 1967, pp. 125-200.

Amarel, S. On Representations and Modeling in Problem Solving and on Future Directions for Intelligent Systems, Scientific Report No. 2, Radio Corporation of America, Princeton, June 1968.

Amarel, S. "Representation and Modeling in Problems of Program Formation." in Bernard Meltzer and Donald Michie (eds.), Machine Intelligence 6. New York: American Elsevier, 1971, pp. 411-466.

Balzer, R. M. "Search for a Solution: A Case Study," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D. C., 1969, pp. 21-32.

Banerji, E. B. The Theory of Problem Solving: An Approach to Artificial Intelligence, New York: American Elsevier Publishing Company, 1969.

Becker, J. D. "The Modeling of Simple Analogic and Inductive Processes in a Semantic Memory System," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D. C., 1969, pp. 655-668.

Bellman, R., and S. E. Dreyfus. Applied Dynamic Programming, RAND document R-352-PR, The RAND Corporation, Santa Monica, California, 1962.

Berlyne, D. E. Structure and Direction in Thinking, New York: John Wiley and Sons, 1965.

Bobrow, D. G. Natural Communication with Computers, Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., 1967.

Bobrow, D. G. and W. Teitelman. Debugging in an On-Line Interactive LISP, Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., 1967.

Bower, G. H. Analysis of a Mnemonic Device. American Scientists, Volume 58, No. 5 (1970), pp. 496-510.

Bruner, J. S. "The Act of Discovery," Harvard Educ. Rev., Volume 31 (1961), pp. 21-32.

Bruner, J. S., J. J. Goodnow and G. A. Austin. A Study of Thinking, New York: John Wiley and Sons, 1956.

Cady, G. M. Planning Activities and Applicable Techniques, SDC document SP-3481, System Development Corporation, Santa Monica, California, 1970.

Citrenbaum, R. L. The Planning Problem and a Survey of Applicable Techniques from the Field of Artificial Intelligence, SDC document SP-3479, System Development Corporation, Santa Monica, California, 1970.

Colby, K. M. and D. C. Smith. "Dialogues Between Humans and an Artificial Belief System," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D. C., 1969, pp. 319-324.

Colby, K. M., L. Tesler, and H. Enea. "Experiments with a Search Algorithm for the Data Base of a Human Belief System," Proceedings of the International Joint Conference on Artificial Intelligence, 1969, pp. 649-654.

Coles, L. S. "Talking with a Robot in English," Proceedings of the International Joint Conference on Artificial Intelligence, 1969, pp. 587-596.

Culler, G. J. "An Attack on the Problems of Speech Analysis and Synthesis with the Power of an On-Line System," Proceedings of the International Joint Conference on Artificial Intelligence, 1969, pp. 41-48.

Curry, J. E. "A Tablet Input Facility for an Interactive Graphics System," Proceedings of the International Joint Conference on Artificial Intelligence, 1969, pp. 33-40.

Dalkey, N. "An Experimental Study of Group Opinion: The Delphi Method," FUTURES, Volume 1, No. 5 (1969), pp. 408-426.

Davis, M. R. and T. O. Ellis. "The RAND Tablet: A Man-Machine Graphical Communication Device," AFIPS Conference Proceedings, Volume 26, Part I, 1964 Fall Joint Computer Conference, Baltimore, Maryland: Spartan Books, 1964, pp. 325-331.

Davis, R. M. "Man-Machine Communication," in Carlos A. Cuadra (ed.), Annual Review of Information Science and Technology, New York: Interscience, Volume 1, 1966, pp. 221-254.

De Groot, A. Thought and Choice in Chess. The Hague: Mouton, 1965.

De Groot, A. "Perception and Memory Versus Thought: Some Old Ideas and Recent Findings," in Benjamin Kleinmuntz (ed.), Problem Solving: Research, Method and Theory. New York: Wiley, 1966, pp. 19-50.

Eastman, C. M. "Cognitive Processes and Ill-Defined Problems: A Case Study from Design," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D. C., 1969, pp. 669-690

Emery, J. G. "The Planning Process and Its Formalization in Computer Models," Proceedings of the Second Congress of Information Systems Science, 1964, pp. 369-389.

Engelbart, D. C. and W. K. English. "A Research Center for Augmenting Human Intellect," Proceedings of the Fall Joint Computer Conference, 1968, pp. 395-410.

Engelbart, D. C. and Staff of Augmentation Research Center. Advanced Intellect-Augmentation Techniques. Final Report to National Aeronautics and Space Administration. Stanford Research Institute, 1970.

Engelman, C. MATHLAB: A Program for On-Line Machine Assistance in Symbolic Computations, MITRE Report MPT-18, MITRE Corp., Bedford, Massachusetts, 1965.

Evans, T. G. "Machine-Aided Design of Context-Free Grammars," Association for Computing Machinery, Proceedings of the 20th National Conference, 1965, New York: Lewis Winner, 1965, pp. 344-353.

Feigenbaum, E. A. "Information Processing and Memory," in Donald Norman (ed.) Models of Human Memory. New York: Academic, 1970, pp. 451-468.

Feigenbaum, E. A. and J. Feldman (eds.). Computers and Thought. New York: John Wiley and Sons, 1963.

Feurzeig, W., S. Papert, Marjorie Bloom, R. Grant, and Cynthia Solomon. Programming Languages as a Conceptual Framework for Teaching Mathematics. Report No. 1889 to National Science Foundation, Bolt Beranek and Newman, Cambridge, Mass., 1969.

Findler, N. V. and W. R. McKinzie. "On a Tool in Artificial Intelligence Research: An Associative Memory, Parallel Language, AMPPL-II," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D.C., 1969, pp. 259-270.

Frye, C. H., F. D. Bennik, and S. L. Feingold. Interim User's Guide to PLANIT: The Author-Language of the Instructor's Computer Utility, SDC document TM-3055/000/03, System Development Corporation, Santa Monica, California, 1968.

Gordon, W. J. J. Synectics. New York: Harper & Row, 1961.

Green, C. "Application of Theorem Proving to Problem Solving," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D.C. 1969, pp. 219-240.

Greenberger, M., et al. On-Line Computation and Simulation; the OPS-3 System, Cambridge, Massachusetts: M.I.T. Press, 1965.

Helmer, O. "The Delphi Techniques and Educational Innovation," in O. Helmer et al., Social Technology, New York: Basic Books, Inc., 1966.

Hormann, Aiko M. "Programs for Machine Learning, Part I," Information and Control, Vol. 5, No. 4 (1962), pp. 347-367.

Hormann, Aiko M. "Programs for Machine Learning, Part II, "Information and Control, Vol. 7, No. 1 (1964a), pp. 55-77.

Hormann, Aiko M. "How a Computer System Can Learn," IEEE Spectrum, Vol. 1, No. 7 (1964b),pp. 110-119.

Hormann, Aiko M. Three Branches of Artificial Intelligence Research, SDC document SP-1858/000/01, System Development Corporation, Santa Monica, California, (1964c).

Hormann, Aiko M. "Gaku: An Artificial Student," Behavioral Science, Vol. 10, No. 1 (1965a), pp. 88-107.

Hormann, Aiko M. Designing a Machine Partner--Prospects and Problems, SDC document SP-2169/000/01, System Development Corporation, Santa Monica, California, 1965b.

Hormann, Aiko M. A New Task Environment for Gaku Teamed with a Man, SDC document TM-2311/003/00, System Development Corporation, Santa Monica, California, 1966a.

Hormann, Aiko M. Shimoku Posed as an Assignment Task, SDC document TM-2311/004/00, System Development Corporation, Santa Monica, California 1966b.

Hormann, Aiko M. Problem Solving and Learning by Man-Machine Teams--Progress and Planned Investigation, SDC document TM-2311/008/00, System Development Corporation, Santa Monica, California, 1967.

Hormann, Aiko M. Problem Solving and Learning by Man-Machine Teams (Summary of Current and Projected Work), SDC document SP-3336/000/01, System Development Corporation, Santa Monica, California, 1969a.

Hormann, Aiko M. Application Problems of Man-Machine Techniques, SDC document TM-4452, System Development Corporation, Santa Monica, California, 1969b.

Hormann, Aiko M. Planning by Man-Machine Synergism: Characterization of Processes and Environment. SDC document SP-3484, System Development Corporation, Santa Monica, California, 1970 (updated in Hormann, 1971b, and 1971c).

Hormann, Aiko M. Machine-Aided Value Judgements Using Fuzzy-Set Techniques. SDC document SP-3590, System Development Corporation, Santa Monica, California, 1971a.

Hormann, Aiko M. A Man-Machine Synergistic Approach to Planning and Creative Problem Solving: Part I, "International Journal of Man-Machine Studies, Vol. 3, No. 2 (1971b).

Hormann, Aiko M. "A Man-Machine Synergistic Approach to Planning and Creative Problem Solving: Part II," International Journal of Man-Machine Studies, Vol. 3, No. 3 (1971c).

Hormann, Aiko M., A. Leal and D. Crandell. User Adaptive Language (UAL): A Step Toward Man-Machine Synergism. SDC document TM-4539, System Development Corporation, Santa Monica, California, 1970 (revised, 1971).

Hunt, E. G. Concept Learning: An Information Processing Problem. New York: John Wiley and Sons, 1962.

Jefferson, D. K. "An Introduction to the Heuristic Programming System," Proceedings of the International Joint Conference on Artificial Intelligence, Washington D.C., 1967, pp. 253-258.

Jones, M. M. Incremental Simulation on a Time-Shared Computer, (Thesis), Massachusetts Institute of Technology, 1967.

Kahn, H. and A. J. Wiener. The Year 2000, The Macmillan Company, New York, 1967.

Kellogg, C. H. "Natural Language Compiler for On-Line Data Management," Proceedings of Fall Joint Computer Conference, Vol. 33 (1968), pp. 473-494.

Kellogg, C. H. and J. Burger. "Progress in Natural Language Data Management," Proceedings of the Third Hawaii International Conference on System Sciences, 1970, pp. 846-849.

Kelly, J. L., Jr., and O. G. Selfridge. "Sophistication in Computers: A Disagreement," IRE Transactions on Information Theory, IT-8 (1962), pp. 78-80.

Kleine, H. and R. L. Citrenbaum. An On-Line Interactive Hierarchical Organization and Management System for Planning, SDC document SP-3482, System Development Corporation, Santa Monica, California, 1970.

Kleinmuntz, B. (ed.). Problem Solving, New York: John Wiley & Sons, 1966.

Kleinmuntz, B. (ed.). Formal Representation of Human Judgement, New York: John Wiley & Sons, 1968.

Kochen, M. "Adaptive Mechanisms in Digital "Concept" Processing," in Kochen, M. (ed.), The Growth of Knowledge; Readings on Organization and Retrieval of Information, New York: John Wiley and Sons, 1967, pp. 185-202, Discussion by Merrill M. Flood, pp. 202-203. (Reprinted from: Proceedings of the Joint Automatic Computer Conference, American Institute of Electrical Engineers, 1962, pp. 49-59).

Laska, R. M. "MIS: Rx for Local Government Malaise," Computer Decisions, Vol. 2, No. 2 (1970), pp. 28-36.

Lederberg, J. and E. A. Feigenbaum. Mechanization of Inductive Inference in Organic Chemistry, Stanford Artificial Intelligence Project Memo No. 54, Stanford University, Stanford, California, 1967, p. 29.

Licklider, J. C. R. "Man-Computer Partnership," International Science Technology, No. 4 (1965), pp. 18-26.

Licklider, J. C. R. "Interactive Dynamic Modeling," in G. Shapiro and M. Rogers (eds.), Prospects for Simulation and Simulators of Dynamic Systems, New York: Spartan Books, 1967, pp. 281-289.

Licklider, J. C. R. "Man-Computer Communication," in Carlos A. Cuadra (ed.), Annual Review of Information Science and Technology, Vol. 3, Chicago: Encyclopedia Britannica, Inc., 1968. pp. 201-240.

Licklider, J. C. R. and R. Taylor, "The Computer as a Communication Device," International Science Technology, No. 76, (1968), pp. 21-31.

Lientz, B. P. On Time-Dependent Fuzzy Sets, SDC document SP-3485, System Development Corporation, Santa Monica, California, 1970.

McCarthy, J. "Programs with Common Sense," in Mechanization of Thought Processes, Vol. 1 (National Physical Laboratory Symposium No. 10), London: Her Majesty's Stationery Office, 1959, pp. 75-84.

Manheim, M. L. Hierarchical Structure: A Model of Planning and Design Processes, Cambridge, Massachusetts: M.I.T. Press, 1966.

Manheim, M. L. Problem-Solving Processes in Planning and Design, Professional Paper P67-3, Massachusetts Institute of Technology, Department of Civil Engineering, Cambridge, Mass.: M.I.T. Press, 1967.

Miller, G. A. "The Magical Number Seven, Plus or Minus Two," Psychological Review, Vol. 63 (1956), pp. 81-97.

Miller, G. A., E. Galanter, and K. H. Pribram. Plans and the Structure of Behavior, New York: Henry Holt and Co., 1960.

Milne, M. The Design Process, Masters Thesis, University of California at Berkeley, 1965.

Minsky, M. "Steps Toward Artificial Intelligence," Proceedings of the Institute of Radio Engineers, Vol. 49 (1961), pp. 8-30.

Minsky, M. "Matter, Mind, and Models in Information Processing," in Proceedings of IFIP Congress 65, W, A. Kalenich (ed.), Washington, D.C.; Spartan Books, Vol. 1 (1965), pp. 45-49.

Minsky, M. (ed.), Semantic Information Processing, Cambridge, Macc.: M.I.T. Press, 1968.

Newell, A. Learning, Generality, and Problem Solving, RAND document RM-3285-1-PR, The RAND Corporation, Santa Monica, California, 1963.

Newell, A. "On the Representations of Problems," in Computer Science Research Review, 1966, Annual Report, Carnegie Institute of Technology, Pittsburgh, Pa., 1966, pp. 19-23.

Newell, A., Studies in Problem Solving: Subject 3 on the Crypt-Arithmetic Task DONALD + GERALD = ROBERT. Carnegie-Mellon Computer Science Document, 1967. A more readily accessible reference will be: Allen Newell and Herbert Simon. Human Problem Solving. Englewood Cliffs, N.J.: Prentice-Hall forthcoming.

Newell, A. "Remarks on the Relationship between Artificial Intelligence and Cognitive Psychology." in Banerji and Mesarovic (eds.) Theoretical Approaches to Non-Numerical Problem Solving. Berlin: Springer-Verlag, (1970). pp. 363-400.

Newell, A. and G. W. Ernst. "Some Issues of Representation in a General Problem Solver," Proceedings of the Spring Joint Computer Conference, 1967, pp. 583-599.

Newell, A. and G. W. Ernst. GPS: A Case Study in Generality and Problem Solving, New York: Academic Press, 1969.

Newell, A. and H. Simon, "Problem-Solving Machines," International Science and Technology, No. 36 (1964), pp. 46-62.

Newell, A. H., A. Simon and J. C. Shaw. "Elements of A Theory of Human Problem Solving". Psychological Review, Vol. 65 (1958) pp. 157-166.

Nilsson, N. J., "Searching Problem Solving and Game Playing Trees for Minimal Cost Solutions," Proceedings of the IFIP Congress, 1968, pp. H-125 - H-130.

Nilsson, N. J. "A Mobile Automaton: An Application of Artificial Intelligence Techniques," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D.C., 1969, pp. 509-520.

Oettinger, A. G. Technological Aids to Creative Thought (Project TACT), Progress Report, Cambridge, Mass.: Harvard University Computation Laboratory, 1966.

Paige, J. M. and H. A. Simon. "Cognitive Processes in Solving Algebra Word Problems," in Benjamin Kleinmutz (ed.), Problem Solving: Research, Method and Theory. New York: Wiley, 1966, pp. 51-118.

Pike, K. L. Language in Relation to a Unified Theory of the Structure of Human Behavior. The Hague: Mouton, 1966, Part I, pp. 8-12.

Polya, G. How to Solve It, a New Aspect of Mathematical Method (Second Ed.), New York: Doubleday Anchor, 1957. (First published 1945).

Polya, G. Mathematics and Plausible Reasoning. Vol. I: Induction and Analogy in Mathematics. Vol. II: Patterns of Plausible Inference. Princeton: Princeton University Press, 1954.

Polya, G. Mathematical Discovery. On Understanding, Learning, and Teaching Problem Solving. Vols. I and II. New York: Wiley 1962-1965.

Press, L. I., M. S. Rogers, G. H. Shure. "An Interactive Technique for the Analysis of Multivariate Data," Behavioral Science, Vol. 14, No. 5 (1969), pp. 364-370.

Quillian, M. R. and D. G. Bobrow. "Growing a Semantic Memory from English Text," Presented at Fifth Annual Meeting of the Association for Machine Translation and Computational Linguistics, Atlantic City, New Jersey, 1967.

Quinlan, J. R. "A Task-Independent Experience-Gathering Scheme for a Problem Solver," Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D.C., 1969, pp. 193-198.

Raphael, B. SIR: A Computer Program for Semantic Information Retrieval, Ph.D. Dissertation, Cambridge, Massachusetts: M.I.T. Press, 1964.

Reitman, W. R. Cognition and Thought, New York: John Wiley and Sons, 1965.

Reitman, W. R. "The Uses of Experience: Open Statements, Ill-Defined Strategies, and Intelligent Information Processing," in Jerome Hellmuth (ed.), Cognitive Studies, Vol. 1 New York: Brunner/Mazel, 1970a, pp. 210-229.

Reitman, W. R. "What Does it Take to Remember?", in Donald Norman (ed), Models of Human Memory. New York Academic, 1970b, pp. 470-508.

Reitman, W. R, R. B. Roberts, T. W. Sauvain, D. D. Wheeler and W. Linn. "AUTONOTE: A Personal Information Storage and Retrieval System." Proceedings of the 24th National Conference of Association for Computing Machinery. New York: Association for Computing Machinery, 1969, pp. 67-76.

Restle, F. Psychology of Judgement and Choice, New York: John Wiley, and Sons, 1961.

Roberts, L. G. "Graphical Communication and Control Languages," in J. Spiegel and D. E. Walker (eds.), Information System Sciences: Proceedings of the Second Congress, Hot Sprints, Virginia, 1964, Washington, D. C. : Spartan Books, 1965, pp. 211-217.

Rosove, P. E. A Provisional Survey and Evaluation of the Current Forecasting State of the Art for Possible Contributions to Long-Range Educational Policy Making, SDC document TM-3640, System Development Corporation, Santa Monica, California, 1967.

Rovner, P. D. and D. A. Henderson, Jr. "On the Implementation of AMBIT/G: A Graphical Programming Language," Proceedings of the International Joint Conference On Artifical Intelligence Intelligence, Washington, D.C. 1969, pp. 9-20.

Sackman, H. Advanced Research in On-Line Planning: Critique and Recommendations, SDC document SP-3480, System Development Corporation, Santa Monica, California 1970.

Sandewall, E. J. "Concepts and Methods for Heuristic Search," Procedures of the International Joint Conference on Artificial Intelligence, Washington D.C. 1969, pp. 199-218.

Seaborg, G. T. "Birth Pangs of a New World." The FUTURIST. Vol. 4, No. 6, pp. 205-208.

Shaw, J. C. JOSS: Conversations with the JOHNNIAC Open-Shop System, RAND document P-3146, The RAND Corporation, Santa Monica, California, 1965.

Shelly, M. W. and G. L. Bryan (eds.). Human Judgements and Optimality, New York: John Wiley and Sons, 1964.

Shure, F. G. and Robert J. Meeker, "A Computer-Based Experimental Laboratory. Administrative Science Quarterly, Vol. 14, No. 2, 1969, pp. 286-293.

Shure, G. H. and R. J. Meeker and W. H. Moore, Jr. "TRACE--Time Shared Routines for Analysis, Classification and Evaluation," Proceedings of the Spring Joint Computer Conference, 1967, pp. 525-529.

Simmons, R. F. On-Line Interactive Displays in Application to Linguistic Analysis and Information Processing and Retrieval, SDC document SP-2432/001/00, System Development Corporation, Santa Monica, California, 1966.

Simon, H. A. The Sciences of the Artificial, Cambridge, Mass. M.I.T Press 1959.

Slagle, J. R., and J. K. Dixon. "Experiments with Some Programs that Search Game Trees," ACM Journal, Vol. 16, No. 2 (1969), pp. 189-207.

Sutherland, I. E. Sketchpad: A Man-Machine Graphical Communication System, Lincoln Lab Technical Report 296, Massachusetts Institute of Technology, Lexington, Mass. 1963.

Sutherland, I. E. Computer Displays. Scientific American. Vol. 222, No. 6, pp. 56-81. June 1970.

Sutherland, W. R. "Language Structure and Graphical Man-Machine Communication," in Donald E. Walker (ed.) Information System Science and Technology, Washington D.C.: Thompson, 1967, pp. 29-31

Teitelman, W. "Toward a Programming Laboratory," Proceedings of the International Join Conference on Artificial Intelligence, 1969 , pp. 1-8.

Tikhomirov, A. K. " On the Functions of Eye Movements." Soviet Psychology, Vol. 8 Nos. 3-4, (1970), pp. 172-177. Trans. from Psikhologicheskiye issledovaniya, Vol. 1 (1969) pp. 25-30.

Tikhomirov, O. K. and Y. E. Vinogradov. "Emotions in the Functions of Heuristics" Soviet Psychology, Vol. 8 Nos. 3-4 (1970) pp. 190-233. Trans. from Psikhologicheskiye issledovaniya, Vol. 1 (1969), pp. 3-24.

Tikhomirov, O. K. "Problem Solving in Chess." Lecture at Psychology Colloquium, University of California, Los Angeles. February 25, 1971.

Tolman, E. C. "Cognitive Maps in Rats and Men," in Tolman, E. C. (ed.) Behavior and Psychological Man, Essays in Motivation and Learning, Berkeley and Los Angeles: University of California Press, 1958, pp. 241-264.

Uhr, L. and M. Kochen. "MIKROKOSMS and Robots," Proceedings of the International Joint Conference on Artificial Intelligence, 1969, pp. 541-556.

Wagner, H. M. Principles of Operations Research, with Applications to Managerial Decisions, Englewood Cliffs, New Jersey: Prentice Hall Inc. 1969.

Waldinger, R. J. and Richard C. T. Lee. "PROW: A Step Toward Automatic Program Writing," Proceedings of the International Joint Conference on Artificial Intelligence, 1969, pp. 241-252.

Walter, W. G. Brain Responses to Semantic Stimuli. Journal of Psychosomatic Research, Vol. 9 (1965), pp. 51-61.

Wigington, R. L. "Graphics and Speech, Computer Input and Output for Communications with Humans," in F. Gruenberger (ed.) Computer Graphics, Washington D.C.: Thompson, 1967, pp. 81-98.

Willmorth, N. E. Human Factors Experimentation in An Interactive Planning System, SDC document SP-3483, System Development Corporation, Santa Monica, California 1970.

Wittgenstein, L. Philosophical Investigations (Trans. G.E.M. Anscombe). Oxford: Basil Blackwell, 1967.

Zadeh, L. A. "Fuzzy Sets," Information and Control, Vol. 8 (1965), pp. 338-353.

APPENDIX A

GAKU'S PERFORMANCE ON THE TOWER OF HANOI PUZZLE

The Tower of Hanoi puzzle was invented by a French mathematician and sold as a toy in 1833. It was originally described as a simplified version of a mythical Tower of Brahma in a temple in the Indian city of Benares, said to consist of 64 disks of gold stacked on one peg and which are being transferred by the temple priests to one of two other pegs according to a precise ritual. Once the transfer is complete, says the myth, the temple will crumble into dust and the world will vanish in a clap of thunder. The disappearance of the world may be questioned, but there is little doubt about the crumbling of the temple. Assuming the priests work night and day, moving one disk every second, and provided they know the shortest sequence of moves, it will take them about 585 billion years to finish the job.

The Tower of Hanoi puzzle is one of many puzzles which can be described as board-and-counter games for which the required solution is a sequence (often specified to be a shortest sequence possible) of legal moves from the initial state to a desired end state. The problem, displayed in Figure A.1, is to transfer the disks from one peg to either of two empty pegs in the fewest possible moves, moving one disk at a time and never placing a larger disk on top of a smaller one. The upper part of Figure A.1 shows the usual appearance of the puzzle with 8 disks and the lower part shows a schematic diagram of this puzzle for the 3-disk case. A, B, and C represent three pegs (or stack cells of the board) and numbers circled in the figure and underlined in the text 1, 2, and 3 are numbered disks (counters) from the smallest to the largest. A slashed zero (∅) indicates an empty cell.

The puzzle can be varied by altering the number of disks and pegs, thus allowing a training sequence from the simple to the more difficult within the same class of tasks. The puzzle has the important property that the methods for simple cases, with suitable abstraction, do provide some help in solving harder cases in a fairly nontrivial way.

GAKU'S ATTEMPTS AT PUZZLE-SOLVING

The mechanism coordinator receives Gaku's first task, the 3-disk Tower of Hanoi puzzle shown in the lower part of Figure A.1. Conditions for legal moves and "limit-of-effort" restrictions incorporated into the program tell Gaku that if the exploration should go beyond 50 nodes or 10 levels (whichever is reached first) in the tree of intermediate states and attempted moves, that fact should be reported to the trainer who will then feed back appropriate suggestions.

The mechanism coordinator's first task is to activate routines for abstraction and characterization of the given task. (For relative simplicity in this informal description, many programming details and special terms are stated in more general and intuitive expressions.) The abstraction routine compares the initial state and the goal state one part at a time and records its findings.

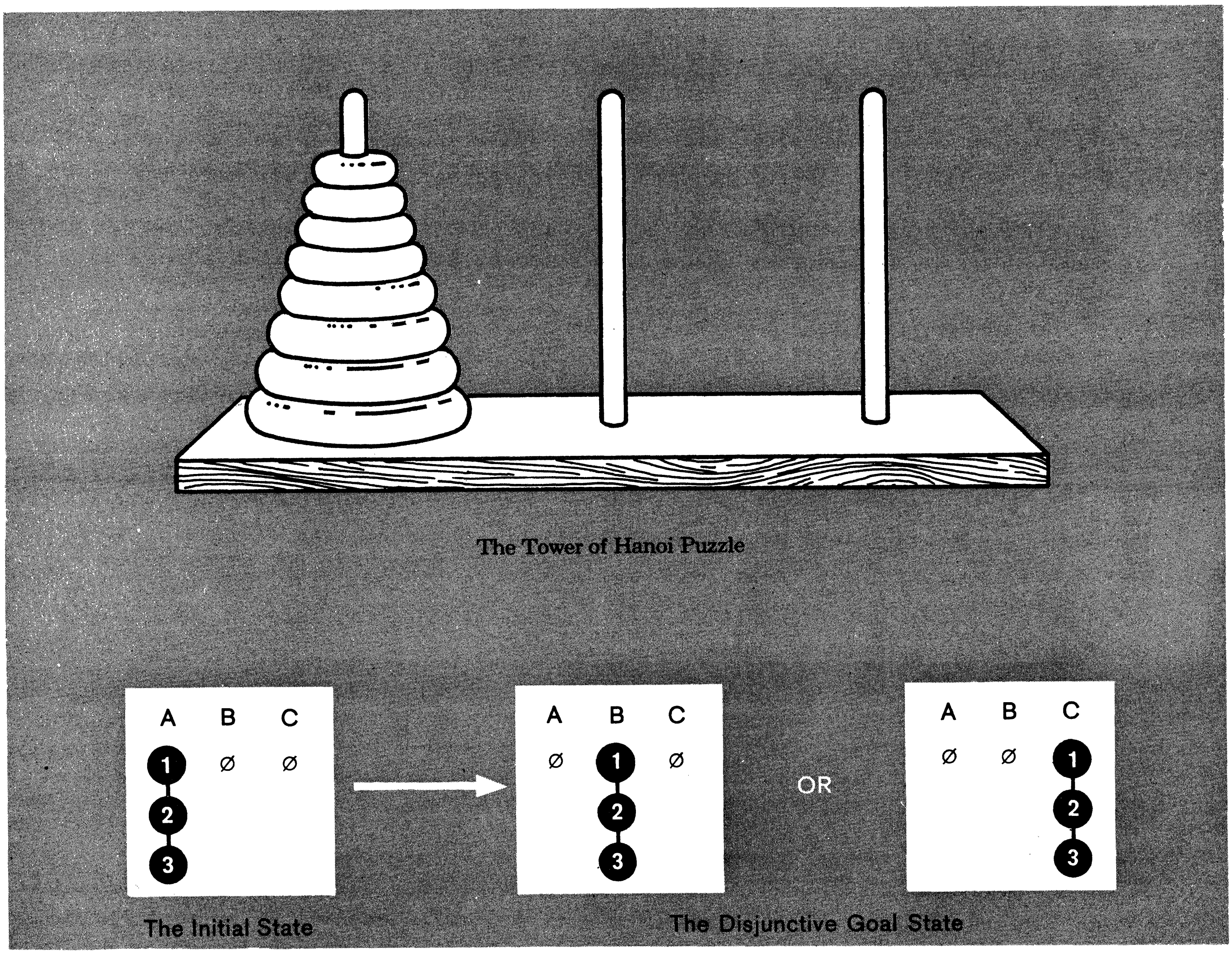


Figure A.1. The Three-Disk Case of the Tower of Hanoi Puzzle

It examines, among other things, the number of counters involved, the positions of the counters, and the contents of cells in the initial state which match those in the goal state, both with and without noting the order of occurrence of the counters. The characterization routine, which helps Gaku create subgoals and preference criteria in selecting moves, then processes the resulting abstracts. The abstracted subgoal indicates that the content of A be made empty (in abstracted form, because B and C contain a symbol which indicates that the content is unspecified). In addition, moves which take these counters out of A are listed separately as preferred moves.

If Gaku had had previous experience with similar puzzles, their characteristics in the past-experience-record tree would indicate possible assistance toward solution by the planning mechanism and the induction mechanism. For this first task, however, there is no record of past experience, so the mechanism activates the problem-oriented mechanism which consists of the task analyzer, the move selector, and the consequence generator.

The task analyzer of the mechanism, having received the information from the mechanism coordinator, creates the first node of a move tree (S_0 in Figure A. 2) which contains the initial state. The move selector then finds two moves, 1-B and 1-C (move counter 1 to B and move counter 1 to C), leading to nodes S_{11} and S_{12}, respectively. Since both moves are equally preferred, the task analyzer chooses one of them randomly, stores it as the new current state, and creates a new node corresponding to S_{11} in the move tree.

The next move chosen is 2-C, because it meets the preference criteria whereas the other legal move, 1-C, does not. Therefore, S_{22} is designated as the current state. At this state neither of the legal moves, 1-A or 1-C, is preferred. Consulting with the preference criteria, the move selector discovers that moving disk 3 either to B or to C is desirable. The move selector checks if it can be done by giving the suggested moves, 3-B and 3-C. The consequence generator, in an attempt to make the suggested moves, checks the legality conditions and finds that disk 3 can be moved only into an empty cell. The current state, S_{22} in Figure A.2, clearly shows that neither move 3-B nor 3-C can be made.

Removal of this impediment is now attempted by a recursive use of the problem-oriented mechanism. The consequence generator, which is itself a part of the problem-oriented mechanism, requests that either B or C be made empty without disturbing disk 3 in A, and calls the problem-oriented mechanism. Thus, the entire cycle of the mechanism becomes involved again at one level lower. The task analyzer automatically takes care of this second-level entrance by a push-down cell, peculiar to the list-processing technique, which permits the second entrance to the mechanism, before exit is made from the first involvement of the mechanism, without destroying the information needed for exit from the first one.

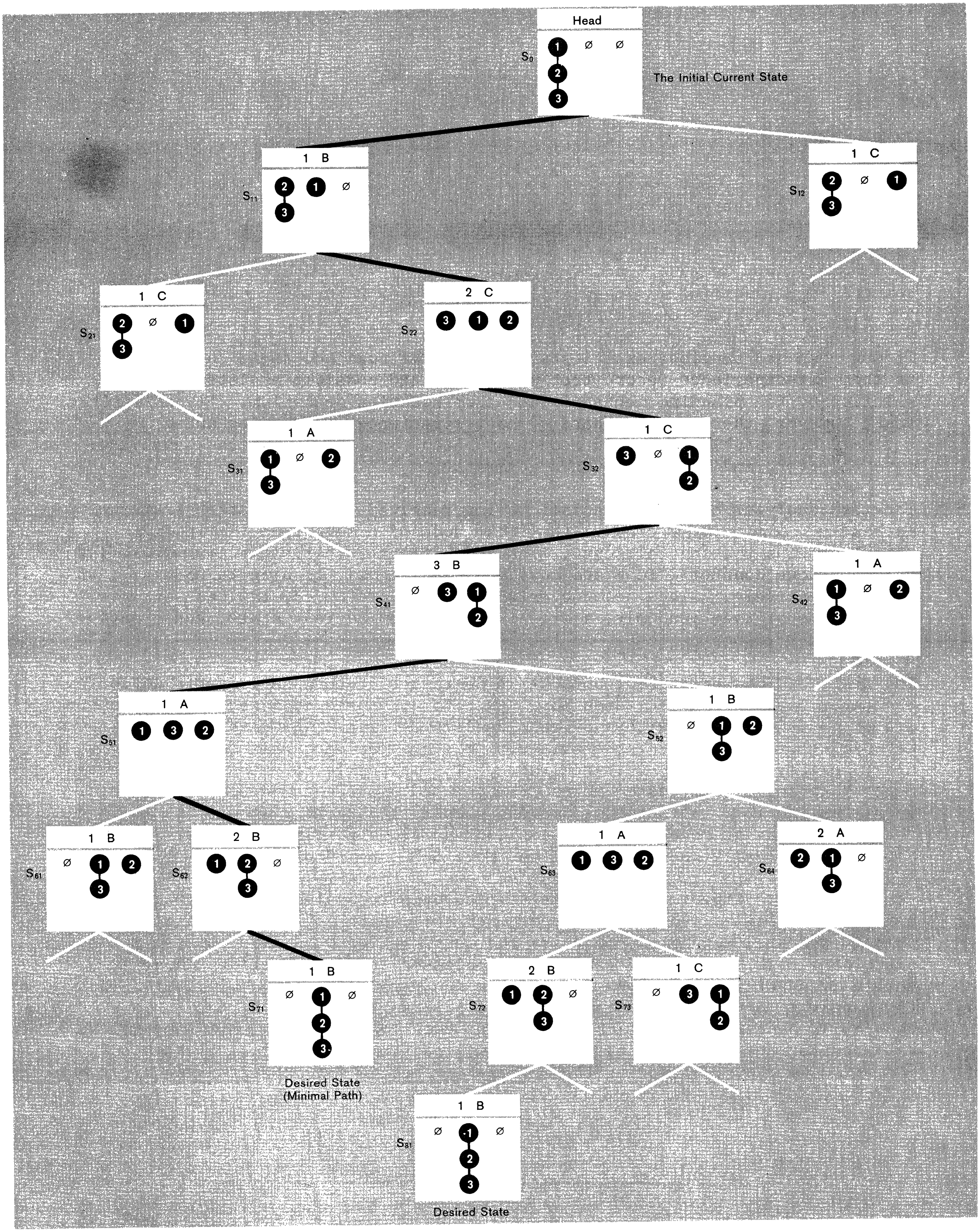


Figure A.2. Part of a Move Tree for the Three-Disk Puzzle

At the lower level, preference criteria, generated by the request to make B or C empty, tell the move selector that those moves taking counters out of B or C are preferred. The move selector generates 1-A and 1-C as legal moves, from which 1-C is selected because 1-A is listed as undesirable. The consequence generator produces this as the new current state, S_{32}. Since the requested condition has been satisfied, the signal "Task accomplished" is sent out and exit from the entire mechanism is made.

When control is returned to the consequence generator which initiated the request, the current state, S_{32}, now has B empty and the move 3-B can now be made to produce the new state, S_{41}.

At state S_{41}, move 1-B is preferred to 1-A because the latter is listed as undesirable (the move does not keep A empty). This leads to the state S_{52}, and more exploration of the move tree is necessary until the goal state S_{81} is reached. Since the path leading to S_{71}, as shown by heavy branches in Figure A.2, is a minimal path, Gaku is instructed to find a sequence shorter than eight steps. More backtracking and exploration of the move tree occurs until finally the goal state S_{71} is reached in seven steps as required.

RECORDING OF THE RESULTS AND CLEAN UP

The mechanism coordinator, after receiving the signal from the trainer, processes the information collected so far in the task environment memory to decide which parts are to be recorded in the past-experience-record tree, and which are to be erased. Additional abstraction and characterization of the record is done so that top levels of the tree (really an inverted tree) will contain more general abstracted information, and lower levels will contain more detailed and specific information. When the available space in the core memory becomes scarce, the mechanism coordinator transfers the lower part of the tree to the auxiliary memory by leaving that information at truncated parts of the tree.

In addition, periodic "contraction" of a body of information is achieved by grouping together separate trees of information and by forming a new node containing an "abstract", to combine the trees. Therefore, the past-experience-record tree will grow not only downward and sideward, but at the top levels. For example, after Gaku experiences many cases of the Tower of Hanoi puzzle, case-history trees can be grouped together to form a new node which will contain general characters and a generalized form of solutions; before this is done, each case is stored as an independent entry.

Returning to the 3-disk case, the mechanism coordinator stores the following information as a subtree in the past-experience-record tree: the list of characteristics of the given task derived by the abstraction and characterization routines; the sequence of moves accepted as the solution; an abstracted form of the the accomplished task which says, in essence, that "moving disk V from A to B was accomplished" (V is a grouping parameter that combines disks 1, 2, and 3

into a single symbol); the name of the list of preferred moves whose performance value contains 4 (number of moves it influenced with no definite record of failure); the name of the move-tree; and copies of the initial task-defining lists together with their identifying names so that a new set of task-defining lists can be compared with their counterparts in the old set.

Memory cells in which intermediate results were handled are erased.

EXTENSIONS OF THE PUZZLE

Next, Gaku is given the 4-disk Tower of Hanoi puzzle (see Figure A.3). This time the mechanism coordinator can utilize its past experience. Gaku exhibits its early stage of learning and shows how a little information on past experience can be used to best advantage by coordinating the functions of the problem-oriented mechanism, the planning mechanism, and the induction mechanism.

To produce a new list of characteristics, the mechanism coordinator activates the abstraction and characterization routines as before. This new list is then compared item by item with the old one. The comparison discloses that the new task is identical to the old one except that the new task involves one more counter, i.e., counter 4. The old record on the 3-disk case is brought back to the working memory. With this much preparation, the mechanism coordinator activates the planning and induction mechanisms to aid the performance of the problem-oriented mechanism. These two mechanisms work simultaneously (in principle, but not in the serial computer we used) and independently of each other, each reporting its suggestions to the mechanism coordinator.

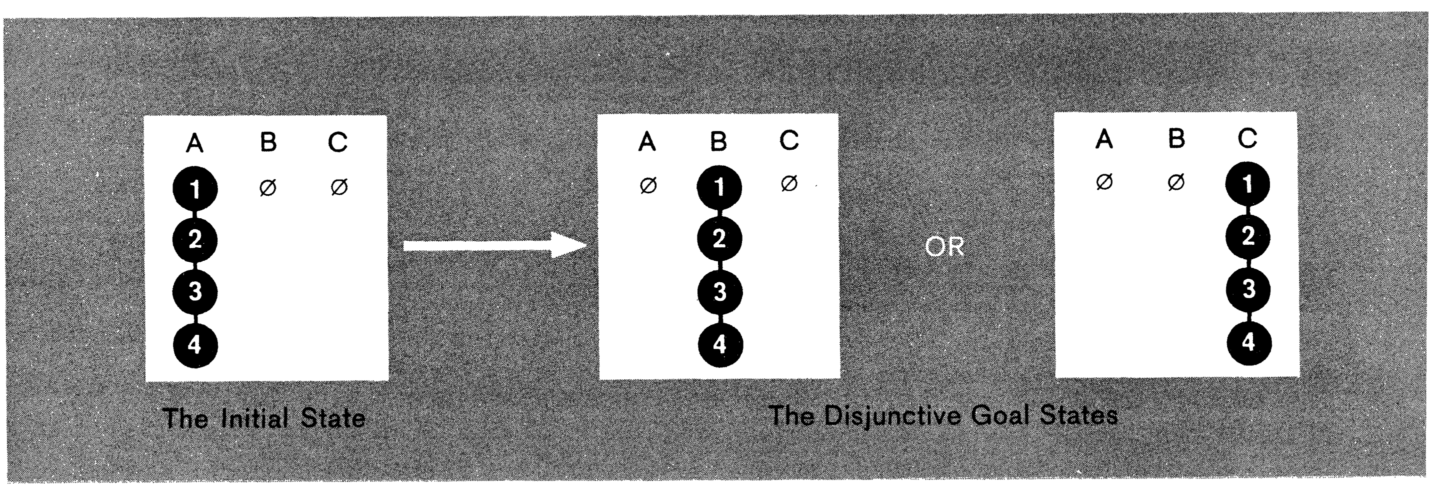


Figure A.3. Four-Disk Puzzle

THE PLANNING MECHANISM

With the information provided by the mechanism coordinator in the working memory, the task analyzer of the planning mechanism uses the abstracted form of the accomplished task to suggest the context in which the subtask provider is to work. As shown in Figure A.4, the subtask provider finds two legal moves, V-B and V-C, from the initial state, but chooses V-B, a move already recorded as being solved. This choice is given to the consequence generator, which then provides the new current state S_{11}.

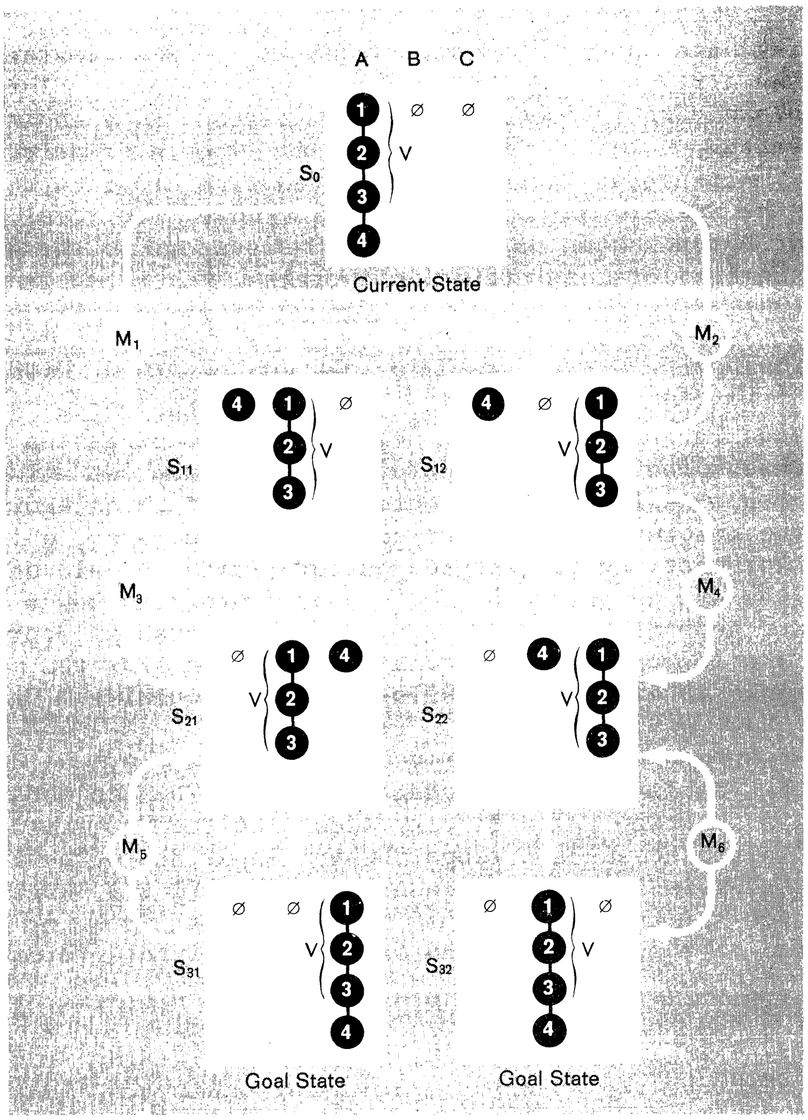


Figure A.4. Subtasks of the Tower of Hanoi Puzzle

It is illegal for the problem-oriented mechanism to move more than one counter at a time, because the mechanism deals only with unit actions defined by the task. However, in the planning mechanism the main objective is to get an over-all picture of the entire problem, and the three disks are moved together as if they were one, a technique analogous to using a previously proved theorem as one step even if the theorem itself required many steps. With 4-C chosen in the preferred list, S_{21} is formed and stored as the third item. The fourth item is the goal state. This list of items (states) is processed by the mechanism coordinator taking a pair of states at a time to form a subtask. The subtasks thus created (M1, M3, M5 in Figure A.4) are given to the problem-oriented mechanism for actual performance.

What if the 4-disk case is given a unique goal state, S_{32}, instead of disjunctive goal states? Then arriving at S_{31} would be an unsuccessful solution. The mechanism has to explore more intermediate states to reach S_{32}, and may have to make an exhaustive search. However, planning of this kind is relatively cheap. It takes only a few examinations at this abstracted level to find the path exhaustively. In contrast, if the individual moves are considered at the level of detail necessary in the problem-oriented mechanism, an exhaustive search for the 4-disk case will require 65,534 examinations of intermediate states.

THE INDUCTION MECHANISM

In the meantime, the induction mechanism attempts to conjecture a general pattern of successful moves beyond the 3-disk case in order to suggest a sequence of moves for the 4-disk case. The task analyzer of the induction mechanism analyzes a given sequence of successful moves to determine a pattern category to be tried out and passes the information onto the conjecture generator, which then produces programs which represent the conjectured pattern, with the aid of its own subunit. The suggested moves are generated by the produced programs under the supervision of the consequence generator, and are given to the mechanism coordinator which, at this point, is able to use the information obtained from the planning and induction mechanisms to guide the problem-oriented mechanism.

For the given sequence of successful moves in this example, the task analyzer decides to try the "cyclic" pattern category first. The conjecture generator then activates its subunit, which is very much like the programming mechanism but has a more limited domain and range for its input and output. The first input to this subunit is the list of counters $\underline{1}\ \underline{2}\ \underline{1}\ \underline{3}\ \underline{1}\ \underline{2}\ \underline{1}$, with the cyclic signal which causes the task analyzer of the subunit to look for the first recurrent position of the first item on the list and finds it to be the third item.

It now takes the first two items $\underline{1}\ \underline{2}$ as defined a cycle phase and asks the programming provider to construct a program that will generate $\underline{1}\ \underline{2}$ repeatedly as $\underline{1}\ \underline{2}\ \underline{1}\ \underline{2}$, etc. The constructed program is executed by the executor-monitor, and the results are checked by the task analyzer, one by one, against the given list until a point of

mismatch, the fourth item, is detected. A new cycle phase is found to be 1 2 1 3, and a program is produced to generate 1 2 1 3 repeatedly. The conjecture generator, which activated the subunit, now receives both a success signal and the generated program.

The same procedure is used for the list of cell names B C C B A B B, and the repeated pattern which is finally accepted turns out to be B C C B A B. The conjecture generator combines these two programs so that together they will produce a sequence of counter-cell pairs as suggested moves, and transmits the resulting program to the consequence generator.

The consequence generator executes the generated program and produces a sequence of moves. It was noted that in the 3-disk case the program was limited to 50 nodes and 10 levels in its exploration. In the 4-disk case, the limit of effort is decided by the experimenter to be 100 nodes and 20 levels. Since the total number of moves for the 4-disk case is unknown to Gaku at this time, the mechanism generates 20 moves (the upper limit for number of levels) and submits these moves to the mechanism coordinator.

The mechanism coordinator uses the total information obtained from both mechanisms to guide the problem-oriented mechanism in its attempts to solve the 4-disk case. Actions of the problem-oriented mechanism are more straightforward because, instead of examining all the legal moves at each state, it uses the suggested move as the first-choice trial. There is, however, some exploration of the move tree, because the suggested sequence contains some wrong moves.

After the complete sequence of moves is found, correct and suggested moves are compared and unmatched elements noted in order to determine whether the pattern-generating programs can be modified, or whether a new pattern must be tried. The suggested and corrected moves are compared below. (Items in italics indicate mismatches.)

Suggested moves:

1	2	1	3	1	2	1	*3*	1	2	1	3	1	2	1
B	C	C	B	A	B	B	C	C	*B*	A	*B*	B	C	C

Correct moves:

1	2	1	3	1	2	1	4	1	2	1	3	1	2	1
B	C	C	B	A	B	B	C	C	A	A	C	B	C	C

The mechanism coordinator, in an attempt to evaluate the merit of the suggested moves by the induction mechanism, makes the following test: Is Number of wrong moves/Total number of moves less than K? If the answer is no, then induction mechanism is to find a different pattern generation program. If the answer is yes, modification of the current program by parameterization (which will be explained next) will be suggested. The current value for K is 0.3. The ratio for the list of counters is 1/15 and that for the list of cells is 2/15, so the

answers are both yes. A signal for the "yes" case is then given to the induction mechanism to start its modification action.

The conjecture generator of the induction mechanism then modifies previously constructed programs by parameterization, i.e., it replaces unmatched symbols with parameters. Special symbols (P_i's) are used to indicate variables, and a sublist containing possible values is created for each P_i. Resulting programs, when executed, will produce a sequence like this:

[1 2 1 P_1] 1 2 1 P_1 1 2 ....
[B C C P_2 A P_3] B C C P_2 ....

(Parts in brackets represent cycle phases. P_1 finds 3 and 4 in its sublist, P_2 finds A and B, and P_3 finds B and C.)

When the 5-disk case is presented, the same procedure is followed--moves are suggested and compared with the successful moves. If a mismatch is found, a subsequent modification is made by parameterization. For this 5-disk case, parameterization yields the following sequence of moves:

[1 2 1 P_1] 1 2 1 P_1 1 ....
[B P_4 C P_2 A P_3] B P_4 C ....

When these new programs are used to suggest moves for the 6-disk case, all of them turn out to be correct. In fact, the parameterized program which has now been constructed will solve all n-disk cases as long as the initial state has n disks in A and the disjunctive goal states are given. Of course, Gaku will never know this unless told by the trainer. However, as Gaku becomes more experienced with the puzzle, and the conjecture program is used successfully, the performance value of the conjecture increases each time so that the mechanism coordinator will direct a straightforward use of the program.

APPENDIX B

A SHORT DISCOURSE ON USER ADAPTIVE LANGUAGE

NEED FOR DYNAMIC EXTENSIBILITY

The following is a summarization of the findings that combine our own observation and the reflective-introspective comments expressed by experimental Shimoku subjects. These findings confirmed our earlier belief that the dynamic extensibility of Gaku capabilities, which depends on dynamic extensibility of the communication language, is one of the essential features in achieving man-machine synergism.

- Even simple bookkeeping functions of the machine can greatly enhance the information-handling capacity of the man, especially in keeping track of interrelated elements in a complex situation. However, the need for such bookkeeping assistance can arise in so many different ways and different situations that a fixed set of predetermined machine functions cannot handle all such needs adequately. Provisions must be made to enable the man to formulate these requests to the machine as the need arises. Some clever ways of utilizing simple machine functions apparently come about dynamically during the interaction when the complexity of the situation exceeds the information-handling capacity of the man. When such clever ideas are not forthcoming or machine aids are not available, the man tends to oversimplify the situation, deliberately (or unconsciously) ignoring many interrelated elements and often leading to a premature decision or conclusion. This is especially true when complexity and time constraints are present simultaneously.

- Before actually executing their decisions, most subjects ask "What if" questions either overtly or covertly in order to estimate the consequences of the tentative decision steps that have been formulated. However, the breadth and depth of such "what if" questions varies greatly with individuals, even with machine assistance. Since exhaustive examination of alternatives in more than three-step depth is infeasible (in the Shimoku environment) even by the machine, very selective "what if" explorations are generated during the interaction. Understanding how such selectivity is formulated dynamically will answer one of the major questions about what separates a good problem solver from a poor one.

- The performance records of the subjects (in terms of the numerical "score") show a dramatic separation between those who attacked the problem incrementally (immediate or short-term payoffs were considered)

and those who had strategic plans for the problem situation as a whole. Those who scored in between the two groups made statements such as: "I had a rather vague plan, but I couldn't follow through with it because so many unexpected possibilities opened up as I played that I got distracted by attractive new prospects and deviated from the plan. Then things got too confusing, so I gave up the plan and played incrementally".

Those who scored highest seemed to have utilized intuitive pattern recognition (both spacial and abstract patterns) to structure the problem space and to make a rough estimate of cause-and-effect relation patterns.

It is our belief that good planning and selectivity in asking "what if" questions are related and that they are highly problem specific. Cleverness in both activities seems to depend on how astutely the subject discerns the idiosyncracies of the problem situations and takes advantage of them in formulating his plans and his "what if" questions. Such characteristics dictate that dynamic extensibility of machine functions is needed beyond the predetermined set of machine functions; many of the relevant questions and much of the selective exploration cannot be formulated at the time of the problem definition. Predetermined machine functions, however, should include any machine assistance that will make it easier for the man to express his tentative ideas and requests for exploration and to delegate certain decision functions to the machine once they are defined and identified as useful.

- The subjects unanimously agreed that visual display of their performance in graphic form and of environmental changes caused by their actions was helpful in assessing previous decisions and formulating new ones. However, more sophisticated techniques of summarizing the current "state of the environment" are needed to display information in a variety of formats and at varying levels of aggregation. At one point, the subject may be interested in the overall relational aspects; at another point, he may be interested in detailed information about one small portion of the environment. Again, the need for dynamic definability of man's ideas and requests became clear.

UAL AS A STEP TOWARD MAN-MACHINE SYNERGISM

UAL has been designed to fulfill the essential need for dynamic extensibility demonstrated by the Shimoku experiments.

An additional advantage can be gained by the use of UAL for the implementation of Gaku. Traditionally, a system is implemented in one language, and another language is specially developed to allow for user-oriented and/or problem-specific expressions. We have, instead, geared the design of UAL and associated techniques to serve both purposes without the compromises that would usually be required.

The basic UAL will be used for initial Gaku implementation and an extended UAL will be used for user-Gaku interaction and also for designer-Gaku interaction in system modification. This will be possible through the use of the extensible features of UAL and of the techniques of building problem-oriented primitives from which higher-level, problem-oriented functions and capabilities can be constructed for the users' convenience. Thus the basic UAL can be used in the same manner as other programming languages but extended UAL can be made into a higher-level, English-like, and/or highly problem-specific language, depending on the users' needs and convenience.

The extensibility of a language refers to its ability to modify itself--that is, its ability to create new primitive terms and functions and to define new infix operators. This becomes important when the problem situation dictates a new notation that the original language does not accept or when new primitive terms and functions are required to reduce complexity. These new terms and functions may not be known at the outset but, through interaction, new ideas may be generated and the need for new terms and procedures realized. Thus, extensibility permits dynamic definition.

With these features, the user can start interacting with Gaku from the initial problem-conceptualization and problem-definition stage and continue to the intuition-guided, hunch-generation-and-testing stage of problem-solving (and perhaps back to the problem-definition stage to repeat the process). The conventional separation between the problem-definition and the problem-solving stages caused by specialists or programmers, or by extra languages, is not necessary. Gaku implemented on UAL can handle user-defined terms and procedures directly, without internal translation or mode changes, since expressions in UAL are the basic items that Gaku can understand, generate, and manipulate. For partially-defined or ill-defined problems, the problem-definition stage cannot be separated from the problem-solving stage because of the iterative nature of the two. The former can be thought of as a model-building process and the latter as model exercising, especially when the model includes a man to provide the behavioral-procedural information that was not known or could not be made explicit at the outset. Gaku can bring these together by allowing the undefined portions to be supplied by the user in the light of new information and insight as supported by his own background knowledge and past experience. Use of the basic UAL in constructing Gaku and also in defining a given problem environment is shown schematically in Figure B.1.

Whenever a new problem environment is considered, the man first introduces a set of primitive terms and functions to UAL in order to establish a semantic link between basic UAL and the problem environment. Once the semantic linkage is established, the Gaku system, which is built on top of UAL, becomes a problem-oriented or special-purpose system. Then, given that the primitives are adequate, the man can communicate any statement or questions about the problem environment and can define any operations in it. In Figure B.1, this set of primitives is represented by the solid box on the right.

In practice, however, it will be easier and more efficient conceptually if the modeling system is equipped with higher-level terms and operations that are more amenable to man's levels of thinking. Representing these is a dotted box labeled "Problem-Oriented Terms and Processes (higher-level)", placed on top of the box of "Problem-Oriented Primitives". The model is built on top of this. Dotted boxes imply that contents are subject to change and expansion.

Intensive research is still required to make the semantic linkage as easy to build as possible so that a user need not feel committed to his first choice of primitives but can start with a "quick-and-dirty" version to experiment with and change it as he discovers its inadequacies. After we have gained experience by building primitives for a number of different problem types, we may be able to extract a set of commonly used steps for primitive building. From this, a set of "primitive-building primitives" may be made available for a quicker and easier definition of new problems.

When model building becomes truly quick and cheap, man may be encouraged to try out different formulations or representations of the same problem situation, thus gaining different viewpoints, one of which may reveal a lead to an answer or to solution methods that could not have been discovered in any other way.

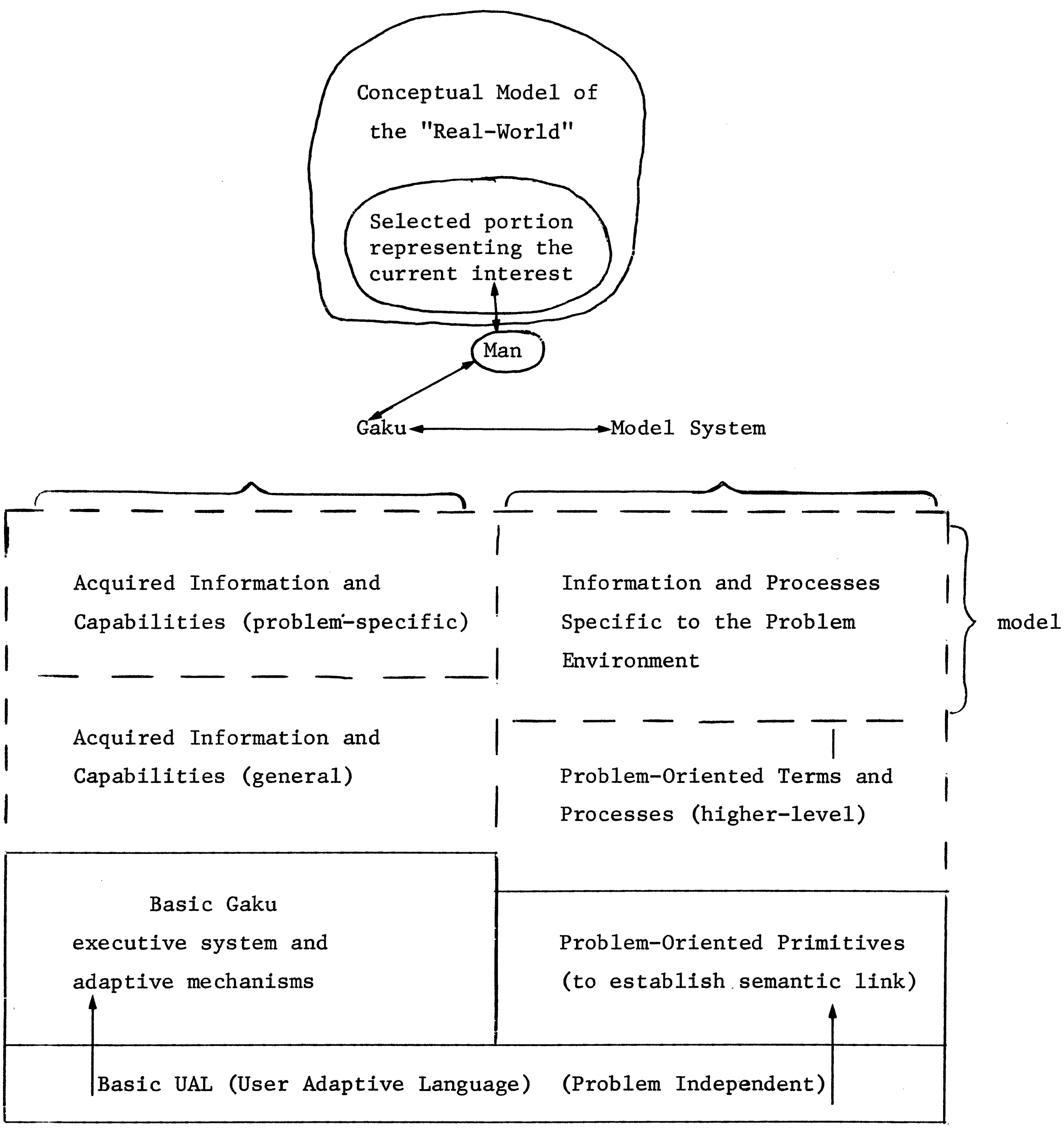


Figure B.1. Schematic Diagram of Man-Machine Cooperative Endeavor, Defining a Problem (or Building a Model) and Solving the Problem (or Experimenting with the Model).

DESCRIPTION OF UAL (USER ADAPTIVE LANGUAGE)

The User Adaptive Language (UAL) is a functionally oriented, extensible, problem-solving language. In addition to many new concepts and ideas, the language incorporates desirable features from existing languages in a unified and consistent manner that makes them easy to use and learn. As well as being user adaptive, the language is user oriented. It is hoped that UAL can be effectively used by those not directly in the computing field.

UAL is designed to be used interactively through a remote terminal so that an immediate response from the computer is received for each and every input. Every expression is evaluated (executed) as soon as it is entered, rather than being stored for evaluation at a later time. However, it is possible to inhibit evaluation if desired. In addition to a number of available predefined functions that aid the user in problem-solving, the language has five different data types: (1) numbers, (2) character strings, (3) quoted expressions, (4) argument maps, and (5) lists. "Character strings" are useful if nonnumerical applications are involved. "Quoted expressions" and "argument maps" provide a convenient, flexible, and powerful means of defining new functions. These functions may be made English-like in use or compact and stylized; they may be general-purpose functions or problem-specific primitives. A "list" is composed of groups of data elements which can be treated as a unit. Sublists and superlists can be formed into structures or networks. These data types, coupled with predefined functions plus the rules for their manipulation, make up the User Adaptive Language.

Some important features of the language are summarized below;

- The basic data structure in UAL is the list. Elements in a list may be numbers, character strings, other lists, or even functional expressions. Thus, procedures may be stored and manipulated in the same manner as simple data items. In addition, two or more lists may share members so that each is "aware" of a change in the other.

- There are two types of assignments in UAL: pointer changing and value changing. Each variable "points to" and is separate from its value. Thus, either the value itself may be changed, or the variable may be caused to point to a new value.

- Arbitrary functions and infix operators may be defined or redefined, depending upon the user's preference. New terms and primitives may be created to fit specific needs in a given problem situation and to express complex ideas in a clear, readable fashion. The user may also make his new functions general and more English-like.

- The language is extensible, which means it is capable of redefining its own parts. The user may change the action of any built-in function or even the way expressions are processed after entry.

- The language is capable of supporting a semantic linkage for a given problem environment by defining a set of problem-oriented primitives. This allows more specialized problem-oriented languages to be built upon it.

- The normal mode of UAL is evaluation. Expressions are evaluated (executed) immediately upon entry. This makes the language naturally suited for interaction. However, a means is provided to suppress evaluation when desired so that expressions may be stored and subsequently manipulated without evaluation.

- A great deal of power and flexibility is provided for functional definition. The user has control over argument evaluation, the exact place where the function name is to appear, the scope of variables, and other features that allow a large variety of functions to be defined.

- The user can specify directly to UAL environmental conditions that describe the context within which he will work. UAL will not allow these conditions to be violated and will warn the user if he attempts to do so.

UAL has been partially implemented at System Development Corporation in Santa Monica, California on the IBM 360/67 ADEPT Time-Sharing System (Linde, et al [1969]) using LISP 1.5 as the source language (Weissman [1967]). For more details of the language, see Hormann, et al (1971).

APPENDIX C

SHIMOKU CASE MATERIAL AND QUESTIONNAIRES

1. SHIMOKU CASE MATERIAL

This section contains a sample of biographical, observational, and raw printout data, with accompanying comments. The case of the subject already most mentioned in the main text has been selected, so that the reader can piece together a deeper sense of events that occurred in the same individual.

A reader wishing to get a good sense of what was going on in action sequences shown in the printout might copy off the main text's Figure 8, the Opening Board Configuration. While progressing through the subject's printout, he might then draw in the moves on the diagram erasing or crossing out relocated tokens where necessary.

Subject I/KN

Subject KN was the Group I urban design student whose games took profitably into account more of the problem conditions than did those of any other player.

Biographical Sketch, Subject KN

Subject KN, 26, came from a working class family in the South of England. His mother was an unqualified elementary school teacher, who had won her job during a teacher shortage. His father "had 900 jobs". The family of mother and six children was fatherless from the time KN, the eldest boy, was 14.

KN's schooling was on a track system. On the basis of the "eleven plus" examination (which first winnows scholars for academic versus practical, vocational training at age 11), he had been admitted into the academic (grammar) school. There, from having been among the best in school, he became one of the worst. He failed 5 out of 7 subjects in the "O level", pre-university exams, taken a year after the family became fatherless. Having failed the O level exams,[(1)]

---

1. Professor Hilda Himmelweit at the London School of Economics has shown a similar fate for a very high proportion of working class boys who are sorted by the eleven plus exam into grammar school, but, lacking cultural and economic support at home, never qualify in the O level examinations for university. KN was extraordinary, compared to her subjects, for getting back in after exclusion. [The School System, Social Class, and Attainment after School, by Hilda T. Himmelweit and Judy Wright (presented at the Annual Conference of the British Psychological Society, 1968). Mimeo.]

he was substantially excluded from good educational opportunities for a number of years.

He apprenticed, then studied on his own and in an unaccredited academy. He worked several jobs. Eventually piecing together some of the requirements, he got to university somewhat illegally, not yet having fully qualified. Still making up deficits, he began to win "distinction" marks for architecture and design. He went on to planning and to urban design, always with high grades and recommendations. In his unorthodox education, he had been accustomed to criticize what he was taught, and to formulate new knowledge independently. By the time he reached Establishment higher education, he was a tough and skeptical student.

Observations of KN in the Shimoku Situation

Two observations are of interest. First, that KN did not fluster under pressure was shown especially well when he played game 1. It was the highest-scoring game 1 played by any subject, and it was the only one played under such potentially stressful circumstances. A full videotape crew was documenting this one game for our records, and everyone was in and out. It was under such display circumstances that he still took nearly the entire first half hour of the game period to analyze the situation. In a first game--without videotaping--no one else had so long tolerated the internal push to move.

We turn to a second observation. In all his interviews, KN showed a very approximate and apparently careless use of language. He appeared to work with very broad approximations of ideas which were not clearly linguistically encoded, and to refine them when, and only when new notions made refinement necessary.(2) KN used almost none of the linguistic terms in which the game instructions had been given, but instead had a very loose terminology of his own. We had frequently to stop and to check our glosses on his suggested meanings, in order to be sure to translate accurately. Although he very much liked sharing his thoughts, he disliked having to go through writing or speaking about what he was thinking, because the encoding of his ideas into understandable language held him back. He said that when a young lady had complained about his conversation, he had explained that speaking for him was merely putting an approximate gloss on

2. At the same time, he seemed to function as if he had tagged a given approximate notion with something like a comment about its credibility, tentativeness, etc. He did not permit an inadequately tested idea to jell surreptitiously into part of a permanent structure, and to function unexamined thereafter.

something two or three ideas behind where his thought was focusing at any particular moment.(3) KN probably would have been an "unsuitable" subject for a think-aloud study--with or without time limits. Weeding subjects in order to use that technique well would have eliminated the player who processed information most effectively in the Shimoku experiment.

Subject I/KN and the AMP-c Game

The third, or Adaptive Master-classic (AMP-c) game of subject KN is the highest-scoring Shimoku game. KN's approach to playing, and the AMP-c game itself are discussed quite extensively, and measures on game KN:3, and the final game board configuration appear in Figure 21 and Table II of the text.(4) This section presents the actual record for KN:3. Left pages show game actions from the computer printout. Right pages show descriptive commentary on actions at corresponding lines on facing left pages.

---

3. Of course, it may not be only speed of thought, visual nature of thought, and so on, that lead KN to show apparent lack of care in use of language to communicate. Basil Bernstein discusses differences in habitual ways of using language to communicate which separate the English working class from the upper (and more educated) classes. Cf. Basil Bernstein, Aspects of Language and Learning in the Genesis of the Social Process. In Dell Hymes, (ed.), Language in Culture and Society, Harper & Row, 1964, pp. 251-260.

Uses of articulated (upper class, more precisely descriptive) and restricted (working class, more emphatic but less precisely descriptive) "codes" are thought to reflect whether a person uses language primarily to articulate carefully the nature of his individual experience, or primarily to gain consensus and support. In the latter case, he is confirming that his experience is substantially the same as that of his group members. Hence, it need not be so much precisely described, as indicated to them. The issue is not restricted to England. But English sociolinguistics may better suit speculation about an English subject.

4. Also in the main text for game is an account of the criteria on which classification was based. Important aspects of the development, across KN's three game series, of the problem representation used in the AMP-c game; aspects of KN's subjective experience during play; aspects of his repeated cycles of modeling self-task interaction; and aspects of his uses of other heuristic devices are discussed in the main text.

TIME	AC#	MV#	ACTN	LOCTN[5]	TOKEN	CST	G'N	NET	SCOST	SG'N	SNET
18:25.8	1	1	MV1	3 3 3	SQ 3						
18:25.9	2	1	EXCH	3 4 1	UP 4	4	0	-4	4	0	-4

5. Cell position naming for Shimoku playing board: first digit refers to grid number. Numbers run left to right across the four grids. The second digit refers to column number on a grid. Column numbers run left to right within a grid. The third digit refers to row number. Row numbers run from bottom to top within a grid. For example, cell 1 1 1 is the leftmost bottom-most cell of the far left grid.

The AC # column of the printout--the Action Number column--gives convenient line numbers for reference in this account.

AC #

0 Game starts, at 18:20.

5.8 minutes pass before subject starts to move.

1 First part of move 1: subject points to SQ 3 in LOCTN 3 3 3.

2 Second part of move 1: KN points to LOCTN 3 4 1, which is occupied by token (t) UP 4. This effects an EXCHange move. It reverses positions of the t's pointed to in AC's 1 and 2. It puts 2 t's into new positions at the cost of 1 legal move.

The CoST is 4 points. The GaiN is 0, since no scoring pattern (sc) has been completed. NET gain is -4, and the SumCOST, SumGaiN and SumNET columns reflect the cumulative status of the game.

The game board at this point shows the exchanged tokens in their new positions, and shows in the table the current MOVE COST, GAIN, NET; and the SUM COST, GAIN, NET for his game so far. (6)

The graph dips below the zero line, showing the cost of the move.

KN makes first parts of an UP diagonal straight flush, and of a SQ diagonal straight flush on grid 3, with middle range numbers of the series 1-8. The UP diagonal will take adaptive advantage of the UP 6 already in place on the initial board (B_i).

> Subject's comment: KN said in interview 2 that he intended game 3 to have a pattern of "4 X's," one each on the crossed diagonals of each board. The t's constituting the X's should work in high scoring-patterns (H-sc's) on their own grid planes, and should work in space at the same time, in as many ways as possible. Thus, tokens in sc's should conform to a condition of high interlock, of sc's using shared t's. At the same time, the making of the main large structure, and of sc's which would be off the main structure, but would interlock with it, should adapt as much as possible to use of t's in their original B_i positions. He had planned to use straight flushes to compose the X's, and to use middle numbers in the straights, to improve opportunities for making extra sc's off the main 4 X's design.

---

6. Hereafter, we shall not mention the detail of automatically computed events, but shall let the reader carry them along with each move.

18:26.0	3	2 MV1	4 3 2	UP 5						
18:26.3	4	2 JUMP	3 4 4	-- 0	3	6	3	7	6	-1

18:26.7	5	3 MV1	2 2 3	RN 4						
18:26.9	6	3 JUMP	4 3 2	-- 0	3	0	-3	10	6	-4

18:27.2	7	4 MV1	0 0 0	.RN 5						
18:27.2	8	4 ADD	4 1 4	-- 0	5	14	9	15	20	5

KN had noticed during interview 2 that the corners of the outer boards, and the center 4 cells of the inner boards were 7-path cells, and had decided to fill them by priority with middle numbers.

He said that to use middle numbers, and carefully chosen shapes of t's going into 7-path cells would give the best chances of finding extra opportunities to interlock sc's with the main design.

In interview 3 KN demonstrated the idea by drawing a diagram showing 3-, 4-, and 7-path cells, marking all positions in the cube 3, 4, or 7.

He planned to conserve moves by using exchanges where possible.

Move 1 fits all the statements of his plan.

3 KN moves UP 5 from 4 3 2 to 3 4 4, a jump costing 3 points.
4

He is continuing to build grid 3's X configuration.

KN earns 6 points, since AC 4 completes a low-scoring sc (L-sc). It is based on a straight run of numbers only.

This (L-sc) does not fulfil the straight flush condition, mentioned in KN's interview.

It will be revised by moving the RN 7 from its B_i position in AC's 58 and 59, after the rest of the main design is in.

5 RN 4 is jumped from 2 2 3 to 4 3 2.
6

This clears a cell on the diagonal of grid 2, and moves RN 4 into position to fit with RN 3 and RN 6, already given on the B_i diagonal in grid 4.

Again, RN 4 puts a middle number in a center position.

Move 3 fits all criteria of KN's plan.

7 RN 5 is purchased from the reservoir to cell 4 1 4. The purchase costs
8 5, and is the most expensive move. The move completes a straight flush, gaining 14 points, for a net of 9.

It builds part of a diagonal which will adapt to the B_i t's RN 3 and RN 6, already in place on grid 4's diagonal.
(7)
Execution of the plan continues.

---

7. Hereafter, we shall skip some remarks about hypothesized events in the subject about relations holding among tokens and between given moves and subject's statements, when their enunciation would be quite obvious or redundant. Because this game is highly and coherently structured according to a well-articulated plan, it is comparatively easy to follow without the full density of commentary that would burden space and the reader. Since KN's games are treated in the main text in some detail, the reader is referred to it to round out the picture given here.

18:27.7	9	5 MV1	0 0 0	.SQ 3							
18:27.8	10	5 ADD	4 4 4	-- 0	5	0	-5	20	20	0

18:28.6	11	6 MV1	1 3 3	UP 3						
18:28.6	12	6 JUMP	1 3 1	-- 0	3	14	11	23	34	11

KN's large structure main design does not stick only to rearrangement of t's given on B_i, but purchases early.

There is already a RN 5 on the board, in cell 1 1 1, which could be moved to 4 1 4 at a cost of only 3 points.

However, it's given B_i position is on a diagonal with RN 2. KN's purchase of RN 5 in the face of an existing B_i RN 5 suggests (in a relatively well organized over-all game)(8) that subject has other plans at this point for the grid 1 B_i tokens--even though there has been no activity yet on grid 1.

Both RN 5's are marked with heavy black felt pen shape outlines on the working diagram KN turned in at the end of his game. We cannot tell when the marks were made in relation to moves. But KN said that he kept carefully up to date.

If so, at the point he found himself wanting a RN 5 for grid 4, he already had the original B_i RN 5 black-marked to indicate its status as "fixed" in location.

The mark could function as a note, to be checked momentarily for protection against disruption of one part of the plan, during search concentrating on finding a way to execute another part.

It allowed him to check a proposed move against an external memory, rather than to have to carry, or repeatedly to regenerate, the values of current positions of needed t's in his head while carrying out processing of new moves.

9 SQ 3 is purchased from the reservoir to cell 4 1 4.
10

It builds part of a diagonal which will use adaptively the B_i t's SQ 2 and SQ 4, already in place.

Again, a purchase occurs despite presence on the board of the same t.

Here, the B_i SQ 3 had been put into a new position during the exchange of move 1, and had presumably been marked off in black.

11 UP 3 is jumped from 1 3 3 to 1 3 1.
12

The move clears a center position on grid 1's diagonal, and completes both a 4-shapes-different, 6 point sc across the bottom row, and another, 4-shapes-same sc on a north-south internal diagonal, scoring 8 points. Gain is again 14, and net gain, 11 points.

---

8. Here we give a sample of the kinds of inferences, and the kinds of qualifications on them, that could be made. Only with a strong and orderly game is our hypothesis likely to be secure: weak players showed enough inconsistencies about purchase in the presence of a B_i token that we would have had less confidence in our hypothesis.

18:29.2	13	7	MV1	0	0	0	.RN 4							
18:29.3	14	7	ADD	1	3	3	-- 0	5	0	-5	28	34	6
18:29.7	15	8	MV1	0	0	0	.RN 3						
18:29.8	16	8	ADD	1	2	2	-- 0	5	14	9	33	48	15
18:29.8	17	9	MV1	1	2	2	.RN 3						
18:29.8	18	9	DPG										
18:34.5	19	9	EXCH	3	3	4	DN 5	4	-14	-18	37	34	-3
18:34.8	20	10	UN										

13 RN 4 is purchased to fill the just-vacated center position on the diagonal
14 of grid 1.

15 RN 3 is purchased to complete the straight flush on grid 1.

16

KN keeps middle range t's in the center positions--but on this grid, it is the corner positions that have the most paths through them. A move has been saved by keeping RN 2 in its B_i position, but RN 2 will be hard to work with there to get extra sc's.

In interview 3, KN repeatedly regretted leaving RN 2 in place, saying that he lost flexibility by it.

17 In a move part 1, KN points to RN 3 in 1 2 2--the token he has just moved into place.

Is he having doubts about the last move? Is he going to unmove it?

18 Immediately (in the same tenth of a minute), he points to the "display graph," "D" command.

KN has been moving steadily since AC 1, never taking more than .8 minutes between actions. Now, 4.7 minutes pass before AC 19.

Presumably, he is re-evaluating the situation in this "thinking time."

19 He points to DN 5 in 3 3 4.

Because he already has registered a move part 1 pointing to RN 3 in AC 17, and has not unmoved it, the new pointing action completes an exchange move. It reverses the positions of DN 5 and RN 3.

Immediately, the score falls by 14 points. The move breaks the straight flush in which RN 3 was participating on grid 1.

Three tenths of a minute passes.

We hypothesize that during the 4.7 minute "think period" between actions 18 and 19, the subject has been evaluating a different part of the board, and has forgotten that the suspended move still had its first part registered. Hence, effecting an exchange later by pointing to DN 5 is probably accidental.

If the breaking move is not observed before another move is completed, he will not be able to unmove it.

20 Subject unmoves, returning the t's to their previous position, and regaining the lost 14 points.

18:35.1	21	9	PTH								
18:35.2	22	9	PT	3 3 4	DN 5						
18:35.9	23	9	PT	1 2 3	-- 0						
18:36.2	24	9	PT	3 3 4	DN 5						
18:36.3	25	9	DPG								
18:36.4	26	9	MV1	3 3 4	DN 5						
18:36.4	27	9	JUMP	1 2 3	-- 0	3	0	-3	36	48	12
18:36.7	28	10	MV1	2 1 3	DN 7						
18:36.8	29	10	JUMP	1 1 4	-- 0	3	14	11	39	62	23
18:37.9	30	11	MV1	0 0 0	.SC 4						
18:38.0	31	11	ADD	3 1 4	-- 0	5	0	-5	44	62	18

After another 0.3 minutes, KN requests paths through a point (P). He gets paths through the cell 3 3 4, occupied by DN 5, then through cell 1 2 3. Again he points through 3 3 4, containing DN 5.

We hypothesize that he is checking for breakable sc's related to DN 5's present and proposed positions.

However, the cell 1 2 3 is not the same as, but is adjacent to the cell containing RN 4. It appears that he pointed to RN 4 by accident, when trying to move DN 5 to the neighboring cell.

> In interview 3, KN said he had been surprised by the break, and had unmoved it.
>
> But he felt it necessary to investigate the source of the difficulty. After checking with P to see that nothing indeed would be broken by moving as he had meant to move (DN 5 to 1 2 3), he concluded that he had turned on the light pen while still moving it, and had caught RN 4 unexpectedly.
>
> This was the first time KN had made that error, and he tracked it down. The behavior is in marked contrast to that of weaker Group II and occasional Group I subjects. They made such errors frequently, but became discombobulated by them, and did not immediately put into play adequate procedures for isolating the defect.

KN displays the graph to remove paths, and goes on to make the previously attempted move of DN 5 to 1 2 3.

At this point he has completed 9 moves, nearly 1/3 of the alloted number, and has but 12 points.

A lot of investment has gone into building structure which will have later payoff, but does not yet show.

DN 7 is jumped to 1 1 4, completing a straight flush.

This move finished the X of grid 1.

SQ 4 is purchased for the corner of grid 3.

KN is building upon the opportunity provided by move 1, the move of SQ 3 to the diagonally opposite corner of the grid. The next move contributes to that same diagonal.

18:38.1	32	12 MV1	0 0 0	.SQ 5							
18:38.2	33	12 ADD	3 3 2	-- 0	5	12	7	49	74	25
18:38.3	34	13 MV1	0 0 0	.SQ 6							
18:38.4	35	13 ADD	3 2 3	-- 0	5	26	21	54	100	46
18:39.5	36	14 MV1	0 0 0	.UP 5							
18:39.6	37	14 ADD	2 4 1	-- 0	5	12	7	59	112	53
18:39.9	38	15 MV1	0 0 0	.UP 4							
18:39.9	39	15 ADD	2 2 3	-- 0	5	24	19	64	136	72
18:40.0	40	16 MV1	0 0 0	.UP 6							
18:40.0	41	16 ADD	2 1 4	-- 0	5	26	21	69	162	93

32 SQ 5 is purchased.
33

12 points are scored by SQ 5's completing a mixed shapes straight going vertically through the cube. The next move is placed to complete the diagonal on grid 3.

34 SQ 6 is purchased.
35

It makes a straight flush, on that grid, as well as a mixed straight going vertically through the cube. The gross gain on the move is 26 points, and net is 21.

36 UP 5 is purchased for 2 4 1.
37

There it starts to build a new straight flush, adaptively using token UP 7 in its B_i position. At the same time, UP 5 now completes a mixed straight for 12 points.

38 UP 4 is purchased for cell 2 2 3, the space earlier vacated by RN 4.
39

It scores 24 points by completing both a mixed straight running vertically through the cube, and a mixed straight running diagonally through the cube, from DN 7 in 1 1 4 to RN 6 in 4 1 1.

KN begins to reap advantage from the middle number mixed straights interlocking with the main design through key middle grid center cells.

40 Another purchase brings UP 6 to 2 1 4.
41

That completes the diagonal straight flush KN has been building on grid 2, for 14 points, and a mixed straight going through the cube vertically, for 12 more points.

16 legal moves have been made, and 93 points gained. Just past the halfway mark, KN has a score in the range attained only by the most competent INC-h players. He has accumulated 81 of the points in just the past 7 moves. Only 1/3 of his time allotment has been used.

KN appears to have been building according to a structural pre-design, and has purchased freely. But the selection of suits and often of individual t's in his main design straights shows extensive adaptation to B_i conditions.

Now 2.9 minutes pass without an action.

18:43.1	42	17	MV1	0	0	0	.DN 4								
18:43.1	43	17	ADD	2	1	1	-- 0	5	0	-5	74	162	88		
18:44.1	44	18	MV1	0	0	0	.DN 6								
18:44.2	45	18	ADD	2	2	2	-- 0	5	18	13	79	180	101		
18:44.4	46	19	PTH												
18:44.4	47	19	PT	2	2	2	.DN 6								
18:44.6	48	19	DPG												
18:44.7	49	19	PTH												
18:44.7	50	19	PT	2	3	3	-- 0								
18:45.4	51	19	DPG												
18:45.5	52	19	MV1	0	0	0	.DN 7								
18:45.5	53	19	ADD	2	3	3	-- 0	5	6	1	84	186	102		

42 Another purchase places DN 4 in cell 2 1 1.
43

Building a straight flush diagonal on grid 2 using DN tokens requires total construction de novo.

KN has worked first with most of the diagonals which at start contained tokens that could readily be adapted to a 4 X's plan.

He has used what B_i tokens he could in their home cells as constraints on the determination of shapes, and as partial constraints on numbers of tokens to be used in effecting the 4 X's.

KN acts as though he considers the remaining uncompleted diagonal on grid 4 to be already constrained to completion with either a SQ 1 or SQ 5, since he has purchased a SQ 3 in move 5. He has left SQ 2 and SQ 4 in their B_i positions, even though earlier, at move 11, he had to buy a SQ 4 from the reservoir to place in 3 1 4. We observe that the least constrained straight flush is approached last.

44 Move 18 brings in a DN 6 from the reservoir, and earns 18 points gross.
45

It completes a shapes-only straight, going vertically through the cube, and a mixed straight, going diagonally through the cube from 1 1 1 to 4 4 4--for 6 and 12 points, respectively.

46 KN requests paths through a point. He points to cell 2 2 2, containing the
47 newly placed DN 6.

48 He displays the graph, which erases the paths. Next, he will re-enter paths mode to probe the cell adjacent to the last path request 2 2 2. Both are on the diagonal where a straight flush is being built.

49 KN again requests paths through a point. He points to get paths through
50 cell 2 3 3.

We hypothesize that KN is exploring extra scoring patterns--penetrating cell 2 3 3--which might be built in the cube, in order to find additional constraints on the DN token which is to be put in this 7-path cell. Numbers suitable could be 3, 5, 7, or, less likely, 8.

51 Displaying the graph erases the paths preparatory to move 19.

52 DN 7 is purchased for cell 2 3 3.
53

While building the diagonal straight flush, it also completes a four shapes different, 6 point sc running on the diagonal from 1 4 4 to 4 1 1.

13:47.9	54	20	MV1	2	4	4	SQ	1							
13:47.9	55	20	JUMP	4	3	3	--	0	3	26	23	87	212	125
13:48.2	56	21	MV1	0	0	0	.DN	5						
13:48.2	57	21	ADD	2	4	4	--	0	5	20	15	92	232	140
18:53.7	58	22	MV1	2	1	1	RN	7						
18:53.7	59	22	JUMP	2	2	1	--	0	3	18	15	95	250	155

Now 2.4 minutes pass.

54 SQ 1 is jumped to 4 4 4.
55

It is cleared from the path of the newly forming diagonal straight flush on grid 2. It completes a diagonal straight flush on grid 4.

The move earns 26 points. In addition to the 14 for the flush just mentioned, it has completed a horizontal 4-shapes-different sc within grid 4, and a 4-shapes-different sc going vertically through the cube--at 6 points each.

56 A DN 5 is purchased and placed in 2 4 4. KN gains 20 points: 14 for the
57 straight flush, and 6 for a 4-shapes-different sc going vertically through the cube.

There are now diagonal straights filling a 4 X's configuration.

KN is now 28.2 minutes into the game and has half the time to go. He has 140 points, and still 9--nearly one third--of his moves left.

Except for one forthcoming revision, he has completed the main design mentioned in interviews 2 and 3.

He takes 5.5 minutes before the next action.

All subsequent actions will have to use what opportunities for adaptive sc-making are provided by the already existing structure. The opportunities have been somewhat systematized by the nature of that structure. There is need for a good deal of generation and on-line assessment of candidate moves.

In the remaining 9 moves he is to gain another 101 points net from rearrangements of B_i tokens, and from new purchases which coordinate adaptively with the preplanned large structure.

58 RN 7 is jumped to 2 2 1, completing a mixed shape straight across the bottom
59 of grid 2, and a second one running vertically in column 2 of the same grid.

It appears the move should have scored 24 points, but instead a gross of 18 is recorded.

The RN 7 used to make this move came out of (broke) an already completed low-scoring diagonal on grid 3. Thus, 6 points have been subtracted.

When it was in cell 3 1 1, RN 7 occupied its B_i position. Located there, it prevented KN's scoring a straight flush, so long as it was of a different shape from the other 3 t's on the same diagonal.

18:54.0	60	23	PTH								
18:54.1	61	23	PT	2 2 1	RN 7						
18:56.0	62	23	DPG								
18:56.1	63	23	MV1	0 0 0	•UP 3						
18:56.2	64	23	ADD	3 1 1	-- 0	5	26	21	100	276	176
19:01.1	65	24	MV1	0 0 0	•RN 8						
19:01.2	66	24	ADD	3 3 1	-- 0	5	14	9	105	290	185
19:02.2	67	25	MV1	1 2 4	RN 1						
19:02.3	68	25	JUMP	3 1 2	-- 0	3	18	15	108	308	200
19:04.8	69	26	PTH								
19:04.8	70	26	PT	1 1 3	SQ 7						
19:09.8	71	26	PT	3 2 1	UP 2						
19:09.9	72	26	DPG								

Now, KN appears purposefully to have broken the low-scoring pattern on grid 3, to make 2 high-scoring patterns on grid 2.

That clears the cell on grid 3 and enables him to upgrade grid 3's diagonal.

60 Paths through a point is requested. The point is used to display lines
61 passing through the newly moved RN 7 in cell 2 2 1. The graph display erases
62 the lines.

63 UP 3 is purchased for cell 3 1 1, just vacated by RN 7.
64

It completes a straight flush on grid 3's diagonal.

This same move completes a mixed straight going vertically through the cube, for a gross total of 26 points.

Now all 4 X's, one on each grid, contain straight flushes, the highest scoring patterns.

Next, 3.9 minutes pass without an action.

65 RN 8 is purchased for cell 3 3 1.
66

There it earns 14 points by completing a straight flush on the edge diagonal running between cells 1 1 1 and 4 4 4.[9]

67 A minute later, RN 1 is jumped from 1 2 4, where it does no good, to 3 1 2,
68 where it earns 18 points.

It completes a north-south mixed straight and an east-west shapes-only pattern on grid 3.

Again, 2.5 minutes pass without an action.

69 Paths through a point is requested, KN pointing to cell 1 1 3.
70

Another 5 minutes passes without an action. KN is still in paths mode.

71 Paths are directed through cell 3 2 1, containing UP 2. The graph display
72 erases paths.

---

9. Note that it makes nothing whatever in the more highly visible grid and straight-vertical-through-cube planes.

19:09.9	73	26	MV1	3	2	1	UP	2						
19:10.0	74	26	JUMP	1	2	4	--	0	3	6	3	111	314	203
19:11.0	75	27	MV1	0	0	0	•UP	2						
19:11.0	76	27	ADD	4	2	1	--	0	5	8	3	116	322	206
19:11.2	77	28	MV1	0	0	0	•DN	1						
19:11.3	78	28	ADD	4	2	4	--	0	5	18	13	121	340	219
19:22.5	79	29	MV1	2	1	2	DN	3						
19:22.6	80	29	JUMP	4	3	1	--	0	3	18	15	124	358	234
19:23.8	81	30	MV1	0	0	0	•RN	6						
19:23.9	82	30	ADD	2	3	4	--	0	5	12	7	129	370	241
19:24.0	83	31	UN											
19:24.2	84	30	MV1	0	0	0	•RN	6						
19:24.2	85	30	ADD	2	3	4	--	0	5	12	7	129	370	241
19:24.8	86	31	TTI											

73 UP 2 is moved from its B_i position in the probed cell to cell 1 2 4, just
74 vacated by RN 1.

It completes a 6 point north-south shapes-only pattern.

75 Another UP 2 is purchased for 4 2 1.
76

It earns 8 points by completing an internal diagonal through the cube from home cell to 1 2 4. It prepares the next move's opportunity to score.

77 DN 1 is purchased for cell 4 2 4.
78

It completes a shapes-only pattern for 6 points, across grid 4. It also finishes a mixed straight in column 2 of grid 4, for a total of 18 points.

6.7 minutes pass without an action.

Next, the time-shared system goes down for 4.5 minutes, during which the scope display is maintained, but no moves are possible. Subject is allowed the study time gratis, his total time from start of game to end being lengthened to 64.5 minutes.

After the system comes back on, subject starts to move.

79 DN is jumped from 2 1 2 to 4 3 1, scoring 18 points gross.
80

It completes 3 low-scoring mixed-shapes patterns: straight vertical through the cube, and north-south and east-west in the grid plane.

KN has made 29 legal moves.

81 For a last move KN adds RN 6 to cell 2 3 4.
82

This move earns 12 points gross by completing 2 low scoring patterns: both are mixed shapes-only sc's going north-south, and east-west in the grid plane.

83 KN unmoves move 30.

He is told time is up.

84 He quickly repeats the actions to reinstate the just-cancelled move 30.
85

86 TTI, teletype input is pointed, ending the light pen input, and stopping the game. KN's scratch working diagrams are collected.

ADD	RMV	SLD	JMP	RPL	EXC	MV1	PH	PT	DG	TT	LK	BK	UN	DN	IC	NA	AC
20	0	0	10	0	2	32	5	8	6	1	0	0	2	0	0	0	0

86 ACTIONS.

Immediately afterward, the B_f, the final game board configuration, is copied from the scope, and the stored game record is printed out by the research assistant.

The printout contains tabulation across the bottom of names and numbers of instances of each type of action in the game. It gives a total for actions, even though some of them have been unmoved. For example, 20 ADD moves (purchases from the reservoir) are tabulated, although one was unmoved.

KN goes to write questionnaire 3 immediately after completing the game. Then he is interviewed about his play. The scratch diagram and his game board diagram are available to the interviewer at the start of conversation. The print-out is brought in to the interviewer during the session with KN.

Examination of the final board shows that KN has left only two t's from the B_i unused in sc's. Eighteen are used in sc's in their Bi positions.

Although no cell which could potentially have 7 paths through it does so, KN has achieved a peak of participation in 5 sc's apiece for 5 t's, in 4 apiece for 10, and in 3 apiece for 19 t's. No other game has more than 2 t's in 5 sc's at once, most have none. Most good games have only a few t's in 4 sc's at once. The degree of coordination KN has achieved is the greatest, for the greatest number of tokens.

2. QUESTIONNAIRES

Here are the written questions presented to each subject after each game. On the printed formats, ample space for answers was left between questions.

2.1 Shimoku Questionnaire, Run 1

1. Do you or did you in the past play with
   3-dimensional puzzles
   board games (chess, checkers, go, etc.)
   paper-and-pencil games (tic tac toe, etc.)
   card games (poker, bridge, etc.)

2. Please give names of any games of those sorts, and some idea of how often
   at what stage in your life
   how well you played them

3. Do you think of yourself as someone who likes to tackle and solve problems?

4. Can you give some examples of types of problems you do <u>not</u> like to work with?

5. Can you give examples of types of problems you <u>do</u> like to work with?

6. Do you build things with your hands? Describe.

7. When thinking about a new or hard problem, do you prefer to work mostly alone, or very closely with someone else?

8. Does your preference depend on the type of problem? If so, explain.

9. Have you worked with a computer before? If so, please describe.

10. Do you have any current use of mathematics beyond school arithmetic in your daily life? If so, please describe.

11. Is it usually easy or hard for you to describe how you're thinking about a problem to someone else?

12. Do you try to go over in your mind the overall ways you're approaching a problem? (usually, occasionally, rarely)

13. Do you go over the detailed steps and how they fit together? (usually, occasionally, rarely)

14. Can you think of any experiences that might have particularly influenced the way you played Shimoku today?

15. Did you feel like / indifferent to / dislike [ ] this first time playing Shimoku? Please explain:

like	/̅ /	
Did you feel indifferent to	/̅ /	this first
dislike	/̅ /	

time playing Shimoku? Please explain:

16. Please describe what you found most interesting about the game.

17. Please describe the way you went about playing the game. Did your approach change as you went further into it?

18. Is there anything you would do differently next time?

19. Have you suggestions to make the game more interesting? Please feel free to comment further.

2.2 Shimoku Questionnaire, Run 2

1. Did you

enjoy	/̅ /	
feel indifferent to	/̅ /	this second play of game?
dislike	/̅ /	

2. Please describe as well as you can how you thought about the game before you started to play today.

3. (a) Did you come today with any particular intention to play one way or another? If so, please describe.

   (b) If you answered "yes" to question 3. (a)., please try to give your reasons or hunches (even if they're vague or only partly formulated--everything is of interest).

4. Did you discover anything new this play

   (a) about your approach to the game?

   (b) about the game itself?

5. Did you think about the game between your first play and today? If so, please describe.

6. How did you feel about your play this time?

7. Did you use the computer about the same way as before?

8. Please describe how you went about playing this time. Were you changing during the play? (All details are welcome!)

9. Would you like to play again?

10. If you were to play again, is there anything you'd do differently? Please describe.

11. Did the interview influence your thinking and approach after last play? If so, how?

2.3 <u>Shimoku Questionnaire, Run 3</u>

1. Did you talk the game over with anyone other than the interviewer
    between games 1 and 2?
    between games 2 and 3?

   If so, please try to remember the main points you discussed and note them here.

   Please say how you think they might have influenced your play each time.

2. Did the interview itself influence your thinking and approach after last play? If so, how?

3. Please describe as well as you can any thinking you may have done about the game <u>after</u> your second play and <u>before</u> starting today.

4. Before starting today did you have any hunches, general ideas, and/or specific plans about what you might do in the third game? (Please give as many details as possible.)

5. Did you
    enjoy /_/
    feel indifferent to /_/
    dislike /_/
   this third play of the game?

6. Did you set yourself any goals for this play?
   If so, please state them.
   How did you arrive at them?

7. Did you play this game pretty much as you had expected?

8. Did you discover anything new this play
    about your approach to the game?
    about the game itself?

9. Did you use the computer assists any differently this time?

10. Please describe how you went about playing this time. Were you changing your approach as the game went on?

11. If you were to play again, is there anything you'd do differently? Please describe.

12. Would you like to play again
    with the game in the same form and with the same moves and time?
    with the game changed in any way? (specify, please)

13. Would you like to play a competitive Shimoku game against another player?

14. Did you become more aware than before of any aspects of <u>your</u> approach to problems and games over the course of your three plays of Shimoku? If so, please describe.

15. Do you think having the questionnaires and interviews made any difference to your awareness of your own approach? If so, how?

16. Would you like to know anything about how others have played? If so, what would you like to know?
    Why would that knowledge be of interest to you?

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
System Development Corporation Santa Monica, California	Unclassified
	2b. GROUP

3. REPORT TITLE
PROBLEM SOLVING AND LEARNING BY MAN-MACHINE TEAMS: Final Technical Summary Report to the Office of Naval Research

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Technical December 1964 to November 1970

5. AUTHOR(S) (First name, middle initial, last name)
Aiko M. Hormann, Sharon Kaufman-Diamond, Carlos Martin Cinto

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 7, 1971		

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N00014-70-C-0221	TM-4771
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	NONE

10. DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This report describes research work in artificial intelligence, human and machine-aided problem solving and planning activities, interactive languages and visual input/output techniques, and man-machine synergy. The chronological development of concepts, techniques, computer implementation, and experimentation is described in terms of three periods that separate and identify shifted emphases in the work and the differences in the computer facilities used. The emphases during these periods were on (1) developing machine capabilities, (2) understanding man's higher metal functions in complex problem-solving and planning activities, and (3) man-machine synergistic teaming and group planning on real-world problems.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Artificial intelligence man-computer interaction problem-solving						

Security Classification

SYSTEM DEVELOPMENT CORPORATION