Principles of Procedures Composition

Christopher K. Riesbeck
Yale University

Edwin L. Hutchins
Navy Personnel Research and Development Center

This paper addresses the problem of how to compose procedures that students can easily learn and remember. The ultimate goal of this endeavor is to develop a set of principles to guide the composition of procedures. At present we have built a set of analytic tools and a set of hypotheses about the nature of procedural learning that can be empirically tested. We came to this topic by way of an examination of the instruction in a navy school that teaches students how to solve relative motion problems with a job aid called the maneuvering board. The procedures taught seemed to us to be confusing. We began by attempting to rewrite them and as we did so, we attempted to be specific about our complaints, and about our attempted solutions to the problems we saw. It became clear immediately that English lacks the precision required to unambiguously represent the procedures. In order to provide a notation for the procedures, we developed the Maneuvering Board Emulation Language (MABEL). With MABEL we could be specific about the nature of the steps which comprise the procedure and also about the relations among the steps in the procedure. This specificity permitted us to propose measures on which the alternative procedures for accomplishing a particular task could be compared.

The maneuvering board is a job aid that represents the motions of ships relative to each other in a way that supports computations that predict the consequences of possible future actions (including no action at all) to be taken by the ships. The maneuvering board itself is a sheet of paper printed with a polar coordinate plot (azimuth grid) and various scales that can be used in plotting ranges and bearings. Problems are solved on the maneuvering board by plotting points, and drawing lines and vectors which represent aspects of ships' motions (DMA 1975).

In this paper we will deal with a portion of only one of the many problems that are solved on the maneuvering board, the Closest Point of Approach (CPA) problem. In this procedure, the relative motion of an observed ship is plotted, and the bearing, range, and time of the closest point of approach between the two ships is determined. If it is determined that the ships will pass closer to each other than is desired, actions will have to be taken to ensure a safe separation. Those actions will be based on other computations performed on the maneuvering board.

Creating a representation language

The main issue in designing a language is finding the right "grain" (Moore and Newell 1974), i.e., the right level of detail. A representation language for the maneuvering board that included the pencil coming in contact with paper fiber and depositing carbon granules would be cumbersome and unenlightening, while one at the same level of abstraction as English fails to capture important distinctions.

The language we have designed was built according to the following constraints:

1) it would not include any appeal to the real world or to the goals to be achieved. Thus, there is no operator for "Find closest point of approach." The operators are all within the world of the maneuvering board itself.

2) it would not include any mention of the actual physical tools involved. Thus, there is no mention of pencils, parallel rules or dividers.

We call this language MABEL, for MAneuvering Board Emulation Language. The objects in MABEL include points, several types of lines (scales, segments, rays, vectors), circles, numbers (speeds, distances, times and angles), and turns (left and right). MABEL has only geometric operators. Although some operators involve fairly complex geometric activity (e.g. INTERSECT(line circle), TRANSLATE(line, point)), not all geometric constructions are included.

Task analysis

Dependency analysis

A dependency analysis constructs a graph representing what steps of a procedure depend on other steps. As a trivial example, we can't find the distance from the reference ship to the closest point of approach (CPA) until we first find the location of the CPA point. Hence we say that the distance determining step depends on the CPA plotting step.

The dependency analysis reveals the constraints on step ordering that are imposed by the nature of the task itself. It defines the set of procedures composed of the given steps that can actually produce the desired result. Among the members of this set, some procedures feel more natural or meaningful than others. One property of procedures that makes them meaningful is the organization of goals and actions.

Goal-action analysis

To make the goal structure of a procedure explicit, we do a goal-action analysis. A goal-action analysis creates a tree whose top node is the goal to be satisfied. Under this node are other nodes representing the goals that have to be achieved in order to satisfy the top goal. Finally, attached to each goal are the actions to be done once the subgoals are achieved. A goal analysis is typically more specific and therefore more constraining than the dependency analysis.

Below is a goal-action tree for finding the bearing of the CPA.

Goal: bearing of CPA (BC)
  Goal: Direction of Relative Movement (DRM)

```
Goal: Line of Movement (LOM)
    Goal: M1
    Action: PLOT(B1, R1, GRID)
    Goal: M2
    Action: PLOT(B2, R2, GRID)
  Action: RAY(M1, M2)
Action: TRANSLATE(LOM, P:GC) => L:DRM
        INTERSECT(L:DRM, P:GE) => P:DRM
        READVALUE(P:DRM) => DRM
Action: ADD(DRM, +/- 90)
```

Measuring Goal-Action Sequences

A goal-action sequence is a linearization of a goal-action forest. The sequence specifies when each goal is initiated (i.e., when work on the goal begins) and when each action is executed (i.e., when the action is performed). To generate a sequence from a forest, we select some goal node in some tree to be the first one in the sequence. After that, we can go to any node in the forest and select its goal or action component, subject only to the following constraints:

-The goal of a node must be initiated before the action of that node can be done.

-Lower actions in one tree must be executed before higher actions.

If the goal-action sequences are to be converted into computer programs, then the order in which things are done really doesn't matter, as long as the constraints given are satisfied. But if the sequences are to become instructions for people to read, follow, learn, and so on, then the constraints fail to take into account the limits of the short-term memory or the organization of long-term memory. Intuitively, we can feel that a sequence of instructions that hopped randomly from one subgoal to another would be very confusing and hard to learn.

In the following paragraphs, we will describe a number of measures for sequences. Each measure is concerned with something that we believe makes sequences easy or hard to learn. For the moment, it is just assumed that these measures are the significant ones. By making each measure explicit, we hope to simplify the problems of actually testing the learnability of instructions.

Number of Top-level Goals (NTG), counts how many goals are initiated in the sequence without any higher-level goal preceding them. We assume that the more top-level goals an instruction text presents, the harder that text is to learn. Hence, NTG should be minimized.

Distance From Goal (DFG), counts for each action how many other actions separate it from its goal. For a sequence, we define the overall DFG to be the maximum of the DFGs for its actions. Distance From Goal should be minimized in sequences. The more actions are delayed, the more likely they are to be forgotten or used incorrectly.

Goal Stack Depth (GSD) counts for each goal in a sequence how many unfinished goals precede it. An unfinished goal is one whose action has not been done yet. The Goal Stack Depth for a sequence is defined to be the maximum GSD of the goals in the sequence. Goal Stack Depth should be minimized in sequences. It is related to Distance From Goal in that a sequence of unfinished goals causes the actions that are eventually done to be far away from their goals. A large GSD is even worse than a large DFG because the actions that are pending have to be done in the right order and this order is opposite to the order in which the goals appeared.

Distance To Usage (DTU) is a measure of the distance between the calculation of a result and the first use of that result. For a sequence, the Distance To Usage is defined to be the maximum of the DTUs for its actions. Distance To Usage should be minimized in sequences. The longer usage is put off, the more intervening results there are, and the more likely that the wrong result will be used.

To illustrate the application of these measures we present excerpts from two variants of the CPA procedure. The first comes from the instruction manual used in a navy training course (FCTCPAC, 1980), and the second is one of several alternatives we ahve investigated. In the procedure taught in the school the steps which accomplish the parts of the overall soultion are mixed together. Below is the portion of the goal-action tree for finding the bearing of the CPA according to the school procedure.

	DFG	GSD	DTU
Goal: M1	-	0	-
Act: PLOT(B1, R1, GRID) => M1	0	1	1
Goal: M2	-	0	-
Act: PLOT(B2, R2, GRID) => M2	0	1	0
Goal: Line of Movement (LOM)	-	0	-
Act: RAY(M1, M2) => LOM	0	1	0
Goal: (DRM)	-	0	-
Act:TRANSLATE(LOM, P:GC) => L:DRM	1	1	0
INTERSECT(L:DRM, P:GE) => P:DRM	1	1	0
READVALUE(P:DRM) => DRM	1	1	3
Goal: Relative Distance	-	0	-
Act: READVALUE(COPY(SEGMENT ....))	0	1	1
Goal: Elapsed Time	-	0	-
Act: SUBTRACT(M2-time M1-time)	0	1	0
Goal: Relative Speed	-	0	-
Act: READVALUE(INTERSECT(RAY(PLOT.)))	0	1	4
Goal: Bearing of CPA (BC)	-	0	-
Act: ADD(DRM, +/- 90) => BC	1	1	-

This procedure does well on keeping goal stack depth low and keeping actions near the the goals they satisfy, but it does so at the expense of having a large number of top level goals making it difficult to remember. The problem is actually worse than shown here since the complete solution to the problem has 12 top level goals.

Here is the procedure rewritten with a more top-down organization:

	DFG	GSD	DTU
Goal: Bearing of CPA (BC)	-	0	-
Goal: (DRM)	-	1	-
Goal: Line of Movement (LOM)	-	2	-
Goal: M1	-	3	-
Act: PLOT(B1, R1, GRID) => M1	0	4	1
Goal: M2	-	3	-
Act: PLOT(B2, R2, GRID) => M2	0	4	0
Act: RAY(M1, M2) => LOM	2	3	0
Act:TRANSLATE(LOM, P:GC) => L:DRM	3	2	0
INTERSECT(L:DRM, P:GE) => P:DRM	3	2	0
READVALUE(P:DRM) => DRM	3	2	0
Act: ADD(DRM, +/- 90) => BC	4	1	-

This procedure has greater DFG and a greater GSD, but has only one top level goal. Expanding it to the whole CPA problem, it has only 3 top level goals and the maxima of DFG and GSD do not increase with the wider scope of the problem.

Optimizing Goal-action Sequences

Based on the measures given above we suggest the following techniques for producing goal-action sequences

-Generate from only one tree in a forest at a time to minimize Distances From Goals and Goal Stack Depths.

-Reorder subsequences to minimize Distances to Usages.

-Uproot certain subtrees and generate from them first to minimize Goal Stack Depths.

-Merge trees to reduce the Number of Top-Level Goals.

The degree to which these measures predict the ease or difficulty of procedure learning and use is, of course, an empirical question. There are certainly limits on the ranges of applicability of some measures, and tradeoffs to be maximized among them. Never-the-less an approach of this type promises to be a significant improvement over the current hit-or-miss approach to procedures composition.

References:

DMA (Defense Mapping Agency, Hydrographic Office), H.O. Publication 217, Maneuvering Board Manual. Washington D.C.: Defense Mapping Agency, 1975.

FCTCPAC (Fleet Combat Training Center, Pacific), Maneuvering Board Manual, 1980.

Moore, J. and A. Newell "How can MERLIN understand," in L. Greg [ed.] Knowledge and Cognition Erlbaum Associates, 1974, pp. 253-285.

The views expressed in this paper are those of the authors and do not necessarily represent the position of the Department of the Navy