Integration of Techniques for Problem Formulation and Solution

Ivan Bratt

*Industrial Laboraties, The Swedish Match Company*

Follow this and additional works at: [https://commons.erau.edu/space-congress-proceedings](https://commons.erau.edu/space-congress-proceedings)

**Scholarly Commons Citation**


[https://commons.erau.edu/space-congress-proceedings/proceedings-1971-8th/session-6/3](https://commons.erau.edu/space-congress-proceedings/proceedings-1971-8th/session-6/3)
ABSTRACT

The problem solving process covering problem identification until implementation of a solution should be regarded as a multiple feedback process. This paper suggests a technique for problem formulation which can be put into a computer for evaluation with a minimum of programming. The technique is designed to facilitate evaluation of a restructured problem to encourage optimization of systems by iteratively exchanging parts of the system.

INTRODUCTION

The solution of a problem may not likely be better than the problem formulation, perhaps worse. In the problem solving process use of computer has a tendency to create a borderline between the man who formulates the problem and the computer man. Few problems are initially formulated sufficiently well and easy man-computer interaction should increase possibility of feeding back knowledge gained during the evaluation into the problem formulation. Such feedback may mean changing some parameters which is simple, but may also mean changing the system, the performance of which is the problem. As long as we stick to a certain area of knowledge, i.e. electronic circuits, production planning, plant design, etc., it is likely that there is a very limited number of types of subsystems, i.e. resistors, lathes or reactors from which a system is put together with different component values and varying structure.

In order to encourage optimization of systems by changing the system itself we have tried to develop a technique where the problem formulation is given a form which we have found

1. facilitates the problem formulation by ensuring concatenation and completeness
2. in a surveyable manner contains all information necessary to evaluate the problem on a computer
3. makes the computer respond to input data with a comprehensive description of the problem structure and an input data form which corresponds to the structure thus reducing chance of input data error
4. allows writing the necessary software in any reasonably capable computer language
5. is reasonably economical with computer core space.

For the discussion of the technique it might be convenient to ask the question

WHAT IS A PROBLEM?

A problem may exist even if we are not aware of it but appears when we experience that something does not work as we expected it to do.

We talk about a mathematical problem but we generally mean the problem of how to arrive at a solution of a mathematical task which has been set to us. A technical problem is generally closely connected to the cause-effect relationship of a certain object while an economic problem is related to the economic output generated by operating the object. A social problem is related to the utility we get from one social system.

I would like to go a little deeper into what I think a problem is as we cannot discuss problem formulation and solution until we have arrived at a workable concept of a problem.

SYSTEMS (Figure 1)

Let us call the "something", the "object" a system by which we mean any delimited part of reality. Every such system may be subdivided arbitrarily - into subsystems. These subsystems are systems.

Our system forms together with environmental systems a supersystem which also is a system. In fact we arrive at a hierarchical system structure which is

infinitely large and
infinitely detailed and
completely useless.
To make the systems concept work we must delimit our system and specify what interactions between, our system and its environment we know of. (Exogenous variables.) We should also specify from which aspects we look upon the system to avoid unnecessary misunderstandings when the solution is presented to those who meet with another set of problems while, looking upon the same system from another aspect.

We find from this picture that the only possible way of arriving at a common concept of the system is to assume that all spectators describe what they see correctly. It then may be possible to find a way of describing the system which suits all of them until a new spectator looks upon it from a new angle finding new features.

The way real life is we may assume that it is impossible to make a complete and correct description of a system. All we can do is to make a model which at its best contains all presently available knowledge on the system. The inescapable trouble is that although we plan future events according to our models we are subject to the cause - effect relationship of the systems.

Let us put it this way. Our system is subject to known and unknown causes and has known and unknown effects on its environment. Unknown causes contribute to known effects. As our perception of the way the system works is based upon observations of known causes and known effects our knowledge of the system is limited. So every time we estimate the future behaviour of our system we use a model of it.

When does a system not perform up to our expectations? When our expectations took shape they were formed by our model so either the causes - known or unknown - which affect our system deviate from those assumed affecting the model or the model is too inaccurate.

And how do we formulate our expectations? Either we expect the system to produce a certain concrete result or to achieve this at less than a certain sacrifice or both.

In short:
A problem appears when the (utility of the) output of our system is less than results (utility) estimated by our model.

In order to identify a problem we have to specify
1. the system concerned
2. what we mean by output and utility
3. how to measure it
4. what output or utility we expect from the system.

We then should be rather close to knowing what kind of problem we have to deal with.

We will not try to change the system until we have proved with reasonable reliability that the problem will be solved by a specific change of the system and a certain set of parameters. So we have to perform experiments with a model.

Solving a problem should then mean (Figure 4)
1. to improve our model until it fits observations sufficiently well for the purpose
2. test new sets of parameters in the model thus modifying the behaviour of the model until at least sufficiently good result (utility) may be expected from the system
3. modify the system according to parameter set found, and
4. check result (utility) achieved.

In doing so optimization is a two stage process.

1. Find the optimal set of parameters of a certain systems while controlling the system such that a specified goal is achieved.
2. Modify the system and optimize according to 1.

Performing parameter optimization is necessary at every modification of the model of the system or we may be completely mislead.

By comparing utility of the model of the old system and the new not knowing where optimal parameter value is we may be inclined to select a parameter set in the vicinity of the optimal set of the old system. By being that much careful and conservative we may compare a reasonably true optimum of the old system with a set of parameters which are very unfavourable for the new system.

To be able to carry the two-stage optimization of system's models I would like to emphasize integration of problem formulation and solution where man's creativity designs system's models, and the computer evaluates the expected effect and utility performing optimization according to suitable algorithms when such is available.

As a problem formulation which is complete must contain all information necessary for a computer to carry through evaluation it should be possible to write computer programs which translates the problem formulation into instructions for the computer at a suitable language level.

Optimization by modifying the model of the system generally means exchanging models of subsystems by other models and so modification of a system's model should be carried through with the least possible effort and chance of mistakes.
The problem formulation technique

The operator

When we formulate a problem we put a system's model into an environment which provides the model with input data and contains necessary external feedback loops.

We use the term operator for a model which describes input-output relationship such that the operator may be used in different places in a system's model and be supplied with data specific for each place. It consists of

1. a body containing an algorithm which converts inputs to outputs according to
2. a set of parameters which describes the properties of the body actually used
3. one or several output vectors, and
4. none or several connections to input vectors.

Figure 6.
The properties of a body as seen from its environment, may be specified by a name or number (operator type number), the place where it fits into the model (operator number), the number of connectors and lowest connector number.

The problem structure

describes the way the various models are interconnected according to the perception we have of the cause-effect flow in our system. Figure 7.

Information on structure may be described by specifying which input connector (negative number) is transferring data from which output vector to the actual operator body.

The steps in which the problem has to be specified is:

Figure 8.
1. for each operator type: an operator type number, the number of connectors, the lowest connector number and the data structure of each vector
2. for each input connector the operator and vector number it connects to, and for each output vector information on lengths of data arrays.

When feeding this information into a computer it responds by generating an input data form

3. where starting values of all data are noted. These data are fed back into the computer which stores them according to certain rules designed to simplify retrieval of data from any operator in the system.

4. At this stage we might want to modify a number of input data before we start computing.

5. So far it has not been necessary to specify the actual models of the subsystems - algorithms in the operator body - only their external connections. In fact we can exchange one operator body by another as long as attached vector structures fits which simplifies optimization by exchange of subsystems.

6. Additional data controlling the kind of computer run we want, for example:

a/ evaluation of a static equilibrium
b/ simulation of system's dynamics
c/ simulation of the optimal dynamic behaviour of the system when it is forced to achieve specified goals.

7. Feed structural and input data into a computer program containing subroutines according to the algorithms of the different operator bodies, and

8. run the job.

A special kind of problem appears when we try to find optimum utility while simultaneously achieving a specified set of concrete practical goals.

If in a problem there are \( N_p \) parameters and endogenous variables forming the vector \( h \) and \( N_g \) goals, direction and size of a step towards a better solution is determined by

\[
\frac{\partial G}{\partial h} \Delta h = G(h) - G(h_0)
\]

\( G \) is a goal vector and

\[
\frac{\partial^2 U}{\partial h^2} \Delta h = - \frac{\partial U}{\partial h}
\]

\( U \) is utility value.

We thus have \( N_g + N_p \) equations and \( N_p \) variables.

One way of dealing with this problem is to skip \( N_g \) equations from the \( N_p \) equations governing the optimization process and we have

\[
\binom{N_p}{N_g}
\]

different ways to do this. It is unlikely that these optima are identical and so we have \( \binom{N_p}{N_g} \)

different strategies by which we can optimize the system.
The technique the outlines of which I have presented is based upon the assumption that we would gain much by regarding the problem solving process as an interactive feedback process where impulses and ideas may appear from results during evaluating the effects of a set of causes when transferred by a certain model thought to represent the system the performance of which is the problem.

As the computer is just a logical machine it should be correct to assume that the problem formulation must contain all information necessary to make it possible for the computer to perform evaluation.

Improvement of quality of the problem formulation - system's optimization - is a creative act, but the increase of quality depends upon the possibility of evaluation. Converting an idea into a problem formulation accepted by a computer is facilitated when major programming effort is avoided, and speed of recirculation is improved which lowers cost of reformulating a problem and would contribute to improvement of quality of the solution.

We have applied the technique presented to a number of dynamic problems:

1. Linear and nonlinear electronic filters.
2. Mechanical nonlinear discontinuous oscillating circuits.
3. Simulation of change of particle size distribution in a chemical process.
4. Simulation of dissipation of random disturbances and cost in a plant consisting of a mixture of batch and continuous processes with the aim of locating profitable modifications.
5. Improvement of a chemical plant utilizing a new chemical process.

We have developed an optimization algorithm which can be incorporated in the technique on a computer of reasonable size and perform

6. Continuously or discontinuously optimizing the tactic by which a process achieves a specified goal.

ACKNOWLEDGEMENT

Initialization of this development work was sponsored by Swedish Board for Technical Development which made possible approaching the stage of practical application. The main part of support, however, comes from The Swedish Match Company.
SYSTEMS, SUBSYSTEMS AND SUPERSYSTEMS

ENVIRONMENTAL

SUBSYSTEM

SYSTEM

SYSTEM

BORDER OF SUPERSYSTEM

FIGURE 1.
ASPECTS

A RECTANGLE
RIGHT HALF BLACK

A DISC
WITH TWO BLACK SECTORS

A TRIANGLE
LOWER HALF WHITE APEX LEFT

A TRIANGLE
LOWER HALF BLACK APEX RIGHT

A WHITE CIRCULAR DISC

A RECTANGLE
RIGHT HALF BLACK

EACH CONCEPTION OF THE SYSTEM - THE PERSONAL MODEL - MAY VARY WITH ASPECT. AS MODELS ARE CORRECT WE CAN DESIGN A COMMON MODEL UPON WHICH ALL CAN AGREE.

FIGURE 2
THE UNKNOWN IN CAUSE-EFFECT RELATIONSHIP

KNOWN

CAUSES

UNKNOWN

KNOWN

EFFECTS

Figure 3
THE PROBLEM SOLVING PROCESS

SPECIFY THE SYSTEM

SPECIFY OUTPUT AND UTILITY CONCEPTS

SPECIFY OUR EXPECTATIONS

MAKE A MODEL

IMPROVE MODEL

SUFFICIENTLY

ACCURATE

OPTIMIZE UTILITY

SUFFICIENTLY

HIGH UTILITY

IMPLEMENT "OPTIMAL" SOLUTION

MEASURE UTILITY

SUFFICIENTLY

HIGH UTILITY

YES

NO

FIGURE 4
COMPARABILITY OF UTILITY OF SYSTEMS

UTILITY

NEW SYSTEM

OLD SYSTEM

PARAMETER VECTOR

FIGURE 5
OPERATOR

VECTOR STRUCTURE

VECTOR NUMBER
NUMBER OF DATA VECTORS
DATATYPE (INTEGER, REAL, STRING)
NUMBER OF DATA
IN DATAVECTOR

CONNECTORS

BODY

PARAMETER VECTOR

OUTPUT VECTORS

FIGURE 6
SYSTEM AND MODEL

CALCULATE UTILITY
OPTIMIZE

CHAIN A

CHAIN B

FIGURE 7

ELIMINATE MISSMATCHING EFFECTS FROM A AND B
REACH GOAL IF POSSIBLE
INFORMATION FLOW IN THE TECHNIQUE

OPERATOR TYPE

OPNR 2 OF TYPE 2 CONNECTS BY VECTOR-1 TO OPNR 1 VECTOR 1

DATA FORM

OPNR 1
  VECTOR #0
  DATAVECTOR #1
  INTEGER
  -198. -33.

OPNR 2
  VECTOR #1

OPNR 3
  VECTOR #0
  DATAVECTOR #1
  REAL
  .52 .32 .33 .34

INPUT DATA MODIFICATIONS

ALGORITHMS OF OPERATOR BODIES

RUN

FIGURE 8