An Architecture for Holistic Problem Solving

FJ Garlick
Department of Information Science
The University of Portsmouth
Milton Campus
Locksway Road
Portsmouth
Hampshire, UK

W Wynn
School of Mathematical Studies
The University of Portsmouth
Mercantile House
Hampshire Terrace
Portsmouth
Hampshire, UK

Abstract

This paper discusses the nature of generalized problem solving and its algorithmic-like properties. In the systems literature problem solving is usually discussed in relation to its methodological setting - for example, SSM may legitimately be regarded as a problem solving scheme. This paper explores what we believe to be the five basic cognitive elements or strategies involved in problem solving. An examination of these five strategies then suggests a way of understanding why particular methodologies have powerful problem solving power, and why explicit use of these five strategies within a methodology will result in an increased problem solving potential.

Some of the ideas discussed here arose from studies into how knowledge engineers solved the problem of knowledge elicitation and representation. These studies were illuminating since the most common situation seemed to be that no real underlying strategy was employed and that the activity in essence was based on chance plus experience. In other words when practitioners were asked what strategy they were using the most common answer was that they *did not know but they could do it anyway*. This form of knowledge is often called *Tacit Knowledge* - that is the sort of knowledge where we know what to do, but have no clear idea as to explaining how we do it.

An Architecture for Holistic Problem Solving

INTRODUCTION

This paper discusses the nature of generalized problem solving and its algorithmic-like properties. In the systems literature problem solving is usually discussed in relation to its methodological setting - for example, SSM may legitimately be regarded as a problem solving scheme. This paper explores what we believe to be the five basic cognitive elements or strategies involved in problem solving. An examination of these five strategies then suggests a way of understanding why particular methodologies have powerful problem solving power, and why explicit use of these five strategies within a methodology will result in an increased problem solving potential.

Some of the ideas discussed here arose from studies into how knowledge engineers solved the problem of knowledge elicitation and representation. These studies were illuminating since the most common situation seemed to be that no real underlying strategy was employed and that the activity in essence was based on chance plus experience. In other words when practitioners were asked what strategy they were using the most common answer was that they *did not know but they could do it anyway*. This form of knowledge is often called *Tacit Knowledge* - that is the sort of knowledge where we know what to do, but have no clear idea as to explaining how we do it.

What is a Problem?

This question has always been difficult to answer since the notion of a *problem* is hard to pin down. Bryant (1989) most helpfully points out that examination of dictionaries leads to such definitions as *a problem is something to be solved* but then with a tautological unhelpfulness, cross reference the reader to a definition that states that *a solution is the answer to a problem*. Neither of the above explanations is helpful and we might take one or more of the following definitions as more realistic.

(Ackoff 1962) problems are situation related to development or evaluation of the control process.

(Reitman 1965) proposed a categorisation of problems based on how well one could specify each of two terminal states.

(Simon 1985) proposed that a human being is confronted with a problem when he has accepted a task but does not know how to carry it out.

(Bryant 1989) problems are situations where circumstances confound action and doubt clouds decision

A Difference in States?

A common and perhaps more useful notion for us is that of a problem being a difference in states (Mayer 1983). Simply expressed, this means that we have some known facts and a desired outcome - in this sense problem solving may be considered as finding an algorithm or transformation that moves us from one state to the other. In practice, solving real-world problems may be difficult because:

The problem definition may be difficult to formulate

The problem definition may be expressible in many equivalent ways

A given problem may have one, many or no solutions

If a solution can be found it may be unsatisfactory

If a solution is found we may have no way of showing that it is correct or even optimal

Sometime we accept a solution because it works, not because we know it to be correct or optimal

Sometimes we are given the solution and need to find a satisfactory route to it, whilst in other situations we may only have the problem definition and will need to find the route and solution.

The above are essentially describing problem outcomes and it should always be remembered that problems exist in a context. Ackoff (1962) outlined several possible contexts. In this paper we shall not be very much concerned with problem context, rather we will be asking like Jackson and Keys (1984)

are there methodologies available which are capable of helping with problem solving in ... different kinds problem context

Not only will we be asking the above question but additionally we ask, if methodologies are available, are they powerful in terms of problem solving capabilities?

PROBLEM SOLVING ALGORITHMS

An algorithm is a recipe for some activity. If we follow the steps, and we know how to do each step, we will get a predictable outcome. However, Harel (1987), in his book Algorithmics, made a very perceptive comment when talking about algorithmic methods:

It must be noted, however, that there are no good recipes for devising recipes. Each new algorithm is a challenge for the algorithm designer. Some problems are straightforward, some are complicated and even others are tantalizing.

Knuth (1973) attempted to specify the properties that an algorithm must have if it is to actually work. They were; finiteness, definiteness, input, output and effectiveness. Briefly:

Finiteness implies that the algorithm terminates.

Definiteness implies that each step in the procedure is unambiguously defined.

Input/Output means simply that we need zero or more inputs and we must have at least one output.

Effectiveness implies that each step in the algorithm can, in principle, be done exactly and in a finite time.

Arguably, the most important property of an algorithm is its procedural nature. By this we mean that we can tell someone how to do each step unambiguously (*Definiteness*) and that the steps themselves are sufficiently basic that they can actually be done (*Effectiveness*). It is legitimate therefore, to speculate on whether these metrics hold for more general problem solving methods. If they do, then the implication is that we can become better problem solvers, and teach others to solve problems by explicitly applying the algorithmic-like properties inherent in any problem solving architecture.

Heuristic Algorithms

Clearly, the strategies we employ to solve many of our every-day problems are not algorithms in terms of Knuth's metrics, however, it is also clear that human problem solvers in a real sense solve complex problems every day but cannot necessarily say how they achieved their solutions. Their methods might be better described as *heuristic* because, of necessity, they work with rough, perhaps intuitive rules of thumb. Now, these heuristics or *rules of thumb* are often extremely powerful but they are often problem dependent. For example, there are many intractable chess problems but particular problems often have simplifying heuristics. However, it is hard to see how this kind of heuristic might become general and therefore useful in other, distinct, fields.

Generic Heuristic Algorithms

Analysis of the solutions to a number of problems leads to the conclusion that there are five ways in which one solves problems, some of which correspond to a surface approach (Gibbs 1990) whilst others, are representative of a deeper strategy. As one looks at the five strategies one sees that they are comprehensive and in this sense they represent an holistic approach to problem solving since one could use them singly or in combination. The five strategies described below have been represented in other terms (Garlick and Leonard 1993), however, the following framework is a simple and more direct description. The letters following the name of the strategy are used later in the paper to give a more precise mathematical structure to problem solving generally.

Trial and Error (T) this form may be seen in the search for possible solutions but with no clear idea as to where to look. In such a scheme there is a high degree of chance as to whether a suitable answer emerges or not

Use of Generic Ideas (G) this is usually the most productive method since ones thinking will be very much in terms of how you or someone else solved similar problems.

Top Down (D) here we break any given problem down into smaller, more manageable units.

Taking Different Viewpoints (V) in this form one is encouraged to look for other, perhaps novel ways of seeing the same problem.

Identifying Relationships (R) this is a difficult concept in practice and it implies that we search for how elements in the problem domain are related to and affected by each other.

The above are algorithmic in nature since there is a real sense in which we know how to do them. Clearly, they are heuristic but nevertheless, learning to explicitly use these strategies is worthwhile. The other feature of importance is that some of the above strategies only require a surface approach whilst others require the problem solver to go much deeper. A contention based on this observation is that for full understanding of a problem, its context and solution, a problem solver needs to use all levels or composites of them if the approach is not to be *one dimensional.*

A PRECISE DEFINITION?

What follows is a mathematical expression of the meaning of the above problem ossibilities. The formulation is simple and yet revealing since it illustrates the complexity associated with real-world problem solving and points the way to a pbetter understanding of how one might construct a powerful problem solving architecture.

$$P = \{problems\} = \{P_1, P_2, \ldots, P_n\}$$

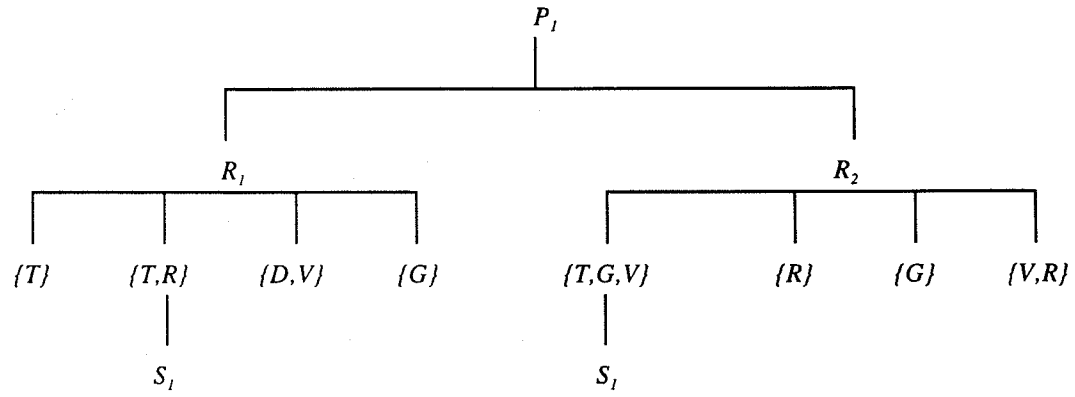
In practice any P_n may have many equivalent classes or formulations as follows:

$P_1 \equiv P_2 \equiv P_3$ *or* $P_4 \equiv P_7$ *and so on.*

Thus, it is clear we may represent any given problem by a set of equivalent reformulations (R), of the original problem.

P_n	=	$\{problems\} = \{R_1, R_2, ..., R_n\}$
PSS	=	$\{problem\_solving\_strategies\} = \{T, G, D, V, R\}$
$P\ X\ PSS$	=	$\{(problem, strategy)\} \mid \{problem:P, strategy:PSS\}$
S	=	$\{solutions\} = \{S_1, S_2, ..., S_n\}$ *for a given problem* P

In diagrammatic form this might be displayed as follows where the letters T, G, D, V and R will represent the five basic heuristic algorithms.



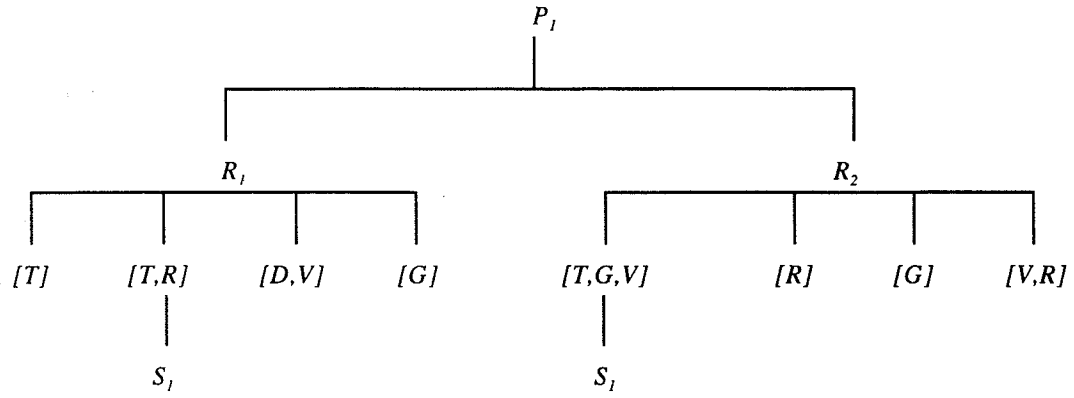
The elements in the above diagram mean that P_1 can be reformulated as R_1 or R_2. From this we see that S_1 arises out of an application of the strategies $\{T,R\}$. Similar meanings can be attached to the route P_1 and reformulation R_2. Now the notation $\{T,R\}$, implies a kind of fusion of the methods *Trial and Error* with *Generic Ideas* where the problem solver is simultaneously using T and G. Clearly this is a difficult process and leads to the conclusion that solving a particular problem may be achieved using just one strategy or a particular fusion of several used simultaneously. Mathematically we might express this powerful idea of a fusion of strategies as a power set where *problem_solving* is the partial function.

Problem_solving: P X $\mathbf{R}$*(PSS)*

In most cases the fusion of methods may not be easy and problem solvers would revert to using strategies singly or in sequence. Mathematically using the strategies in sequence would be represented by the partial function.

Problem_solving: P X Seq(PSS)

In diagrammatic form this might be displayed as:



As before, the elements in the diagram mean that P_1 can be reformulated as R_1 or R_2. From this we see that S_1 arises out of an application of the strategies *[T,R]*. Similar meanings can be attached to the route P_1 and reformulation R_2. However, in this case the notation *[T,R]*, implies that the method *Trial and Error* is applied first then followed by the method *Generic Ideas*.

Summarising these two notations and their associated meanings we find that:

{T,G,V} means use *T*, *G*, and *V* simultaneously or in fusion

[T,G,V] means use *T* then *G* then *V* in sequence

PROBLEM SOLVING ARCHITECTURES

Now that we have the basic ideas we can search for an architecture - a scheme which embodies these ideas in a coherent manner. For example, we might legitimately regard Soft Systems Methodology (Checkland 1990) as a problem solving architecture because it has all the tools and techniques needed for a thorough analysis of a given human activity system.

Orientation - Solution or Problem?

Before looking at the application of these ideas in Systems Science and Systems Dynamics it is worth considering whether a solution or problem oriented approach is best. For example, in an organisational setting where the ideology of the company may have to undergo many changes in responses to rapidly changing environmental factors it is not easy to decide whether a solution rather than problem oriented outlook is more suitable. Brunsson (1988) expressed this point of view in a most interesting and provocative manner as:

> *.. to have problems is often a vital solution for an organisation, whereas solutions can often be a serious problem.*

The point at issue is that a sharp focus on solutions may in effect lead one to solve the wrong problem as well as the more serious consequence of stifling debate and innovation. This does not imply that the solutions are undesirable or always poor or do not address some issue but rather it is a matter of focus.

Problem Solving Power

It has always been difficult to compare problem solving architectures because they are all essentially heuristically defined. However, we could use the five basic strategies outlined earlier and look in detail at the architecture to see if these natural heuristic algorithms, either used singly, in sequence or

perhaps more powerfully, in fusion, are an essential part of the method. If they are not then we might contend that the method is weak because it does not *force* a user to work other than at superficial levels. The next section of this paper will briefly examine two system methodologies and ask are they powerful problem solving architectures in terms of the heuristic algorithms outlined earlier.

Soft System Methodology - A Powerful Problem Solving Architecture?

This is a fair question and no matter what the pedigree of the methodology, good system thinking would always require us to examine and re-examine all we do in the light of experience and perhaps new ideas and insights. It is not possible here to give a full analysis of all stages of SSM but in order to demonstrate the ideas we will look at one particular, and important technique within the general scheme. The technique is called finding *Relevant Systems* and it is a crucial step in the Methodology since it is the first time within the scheme that one suggests a possible resolution of a perceived issue or primary task.

Taking Different Viewpoints (V) - Two choices or views of relevant systems are encouraged. The first view being that the suggested system is visible in the real-world and the second view is that the system is a conceptualisation of something that might exist in order to resolve some perceived difficulty. In addition, analysts would write down many different possibilities for the relevant system, and it is clear that each one represents another view of the problem setting and therefore may give further insight into its nature.

Use of Generic Ideas (G) - It is interesting to speculate on how one finds a particular relevant system and it is highly likely that one uses ones own past experience of similar settings. This is a very powerful mechanism and so long as one is fairly open-minded, can often lead to real innovation and creativity.

Identifying Relationships (R) - As one writes down the various possible relevant systems, most good analysts will be mentally seeing links between the various ideas and recognising various themes emerging.

One final point here is that as one works through this element of SSM it is very likely that one starts off in a linear fashion trying the strategies *{V,G,R}* one after another but as this proceeds it does seem as if a fusion takes place and one moves into the composite strategy *{V,G,R}*. From this brief example one can conclude that SSM does have a powerful problem solving architecture because it requires practitioners to use all the natural heuristic algorithms.

System Dynamics Approach - A Powerful Problem Solving Architecture?

This approach is well known and for purposes of this exercise we shall look at that aspect of the methodology known as Qualitative Systems Dynamics (Wolstenholme 1990) which uses a feedback model as its basis.

Taking Different Viewpoints (V) - Systems dynamics models are typically constructed in two distinct ways. Firstly, by identifying feedback loops which are deemed to be responsible for a systems behaviour. Secondly, by identifying all the various processes associated with a particular systems performance. Here we have two separate viewpoints but in both cases they are driven by the twin needs of understanding how a system works and what its underlying purpose is supposed to be.

Use of Generic Ideas (G) - Wolstenholme (1990) helpfully points out that certain types of simple feedback loop imply certain types of system behaviour. It follows logically from this that if existing reference modes of behaviour are known then it should be possible to infer the types of loop of which the system under investigation is itself composed.

Top Down (D) - At a later stage in analysis one would start to construct resource flows that are related to identified states and their associated processes. One would presume that a top down approach would be used here to separate out the various parts of the system for this detailed level of investigation.

Here one might conclude that Systems Dynamics is less well structured in terms of problem solving power since it tends to be solution oriented and the possible system views seem to be somewhat limited. However, the analysis phase which requires inference of feedback types from observations of system behaviour does seem to be a powerful fusion of two heuristics, namely *Use of Generic Ideas* and *Identifying Relationships {G,R}*.

CONCLUSIONS

Our main conclusions are that better problem solving skills result when the five strategies (heuristics) outlined above are used explicitly. Further, we conclude that any methodology that embodies all the above strategies will be inherently more stable and better adapted to real world problems than one that is based on only a partial set of the above five elements. Finally, it does seem useful to take the five strategies, either singly, in sequence or in fusion, and then use them to examine in detail all methodological steps as a mechanism for deciding what exactly is going on in a particular analysts mind and to decide whether the method as a whole is powerful or not.

References

Ackoff, R.L. (1962). *Scientific Method*, John Wiley, New York

Bryant, J. (1989). *Problem Management*, John Wiley, Chichester

Brunsson, N. (1988). *The Organisation of Hypocrisy*, John Wiley, Chichester

Checkland, P.B., and Scholes J. (1990). *Soft Systems Methodology in Action*, John Wiley, Chichester

Garlick, F.J. and Leonard, G.L. (1993). *The Algorithmic Nature of Problem Solving*, UKSS 3rd International Conference, University of Paisley, Plenum

Gibbs G (1990), *Improving Student Learning*, The Oxford Centre for Staff Development.

Harel D (1987), *Algorithmics - The Spirit of Computing*, Addison-Wesley, Wokingham

Jackson, M.C., and Keys P. (1984). *Towards a System of Systems Methodologies*, Journal of the Operational Research Society, Volume 35

Knuth D (1979), *Fundamental Algorithms 2ed*, Addison-Wesly, Massachusetts

Mayer RE (1983), *Thinking, Problem Solving, Cognition*, WH Freeman

Reitman WR (1956), *Cognition and Thought*, Wiley

Simon HA (1985), *Information Processing Theory of Human Problem-Solving*, Issues in Cognitive Modelling

Wolstenholme, E.F. (1990). *Systems Enquiry*, John Wiley, Chichester