Varieties of Operationality

Thomas G. Dietterich
Department of Computer Science
Oregon State University
Corvallis, OR 97331
tgd@cs.orst.edu

James S. Bennett
Coherent Thought Inc.
3350 West Bayshore Road, Suite 205
Palo Alto, CA 94303
jbennett@coherent.com

1 Introduction

In recent papers on machine learning, the term 'operationalization' has been used to describe the purpose of the learning process. In particular, explanation-based learning systems are said to 'operationalize' the given target concept. Unfortunately, the exact meaning of this term has varied from one paper to another, and frequently the term has been used without being precisely defined. In general, the term 'operational' has encompassed many different aspects of problem solving including correctness, efficiency, and effectiveness.

Although one might argue that a precise definition of the term is neither possible nor desirable, this paper takes the opposite position. Without a precise definition of the goals of the learning process, it is difficult to evaluate particular learning systems. The following sections identify several different aspects of 'operational' and propose a collection of more specific terms, such as 'testable', 'achievable', and 'efficient', to clarify this all-too-vague term. These more precisely defined terms can then be applied to evaluate current learning systems and guide future research.

2 A brief history of 'operational'

The notion of 'operational definitions' was first articulated by Bridgman (1927) in his attempt to clarify the status of theoretical terms in science. He took the position that one cannot know the meaning of a theoretical term unless one has an executable (i.e., 'operational') procedure for identifying instances of the term. This doctrine has been very influential in the social sciences, particularly in psychology, where it has encouraged researchers to define carefully such terms as 'intelligence' and 'psychosis' by providing measurement procedures.[1]

[1]Interestingly, Bridgman's 'operationalism' has been largely rejected by philosophers of science who have pointed out that the operational definition of a theoretical term is often only an approximation to that more theoretical—and

The term ‘operationalization’ entered the machine learning literature in a series of papers by Hayes-Roth and Mostow (1979; 1981) where it was employed to describe the process by which advice, such as “avoid taking points,” was translated into an efficient and executable plan of action for approximately achieving the goal of winning at Hearts. This use of the term encompasses three separable aspects: (a) effectiveness (i.e., the player could actually execute the ‘operational’ form of the advice to achieve the goal during the game), (b) efficiency (i.e., the ‘operational’ form could be applied efficiently), and (c) approximate correctness (i.e., the ‘operational’ advice often only approximated the original goal). Only the effectiveness aspect corresponds to Bridgman’s original use of ‘operational’.

More recently, the term ‘operationalization’ has been employed by Mitchell, Keller, and Kedar-Cabelli to describe the process and the results of explanation-based learning (EBL) (Mitchell, Keller, and Kedar-Cabelli, 1986; Keller, 1983, 1987, 1988). Keller (1983) and Mitchell, Keller, and Kedar-Cabelli (1986) emphasize almost exclusively the efficiency aspect of operationalization. Keller (1988), on the other hand, also raises the issue of approximate correctness and lays a foundation for making the tradeoff between correctness and efficiency by explicitly considering the higher-level goals of the agent. Finally, in his dissertation (Keller, 1987), the term ‘operational’ became synonymous with ‘serving the goals of the agent’ (i.e., maximizing some utility measure that stresses efficiency while maintaining effectiveness).

Each of these definitions captures different aspects of ‘operationality’. After introducing a few basic definitions that capture several important relationships among agents, actions, and goals, the paper reviews each of the major senses of ‘operational’, defining each of them precisely in terms of the basic definitions. The paper then considers the process of ‘operationalization’ and provides two different perspectives on this process. Finally, the paper concludes with some prescriptions for future research in this area.

3 Agents and Goals

We take an agent to be a problem solving system that is embedded in some environment in which it confronts problems that it attempts to solve. Let us define a problem to be a 3-tuple $\langle G, E, R\rangle$ consisting of a goal G, an environment E, and a set of resource constraints R. For example, in the game of Hearts, the goal might be to win the game, the environment consists of the physical setting of the players (i.e., each player cannot see the cards in the other players’ hands, cards can neither be created nor destroyed, the rules of the game must be obeyed, etc.), and the resource constraints state that the agent must take no more than 30 seconds to decide which card to play.[2]

In order to define various kinds of operationality, it is convenient to make the following definitions.

Let the relation

$$\text{acts}(Ag, G, E, R, A)$$

mean that when agent Ag is confronted with the problem determined by $\langle G, E, R\rangle$, the agent will perform a sequence of actions A. The agent is given a complete description of G and perhaps of R but not necessarily of E. Instead, the agent is embedded within E. Some of the actions A may, therefore, be actions that gather information about the environment in order to achieve G. This definition simply describes what the agent will do, which we can verify by observing the actions of

non-empirical—definition of the term. See, e.g., Suppe (1977) and Hempel (1952).

[2]In general, one can imagine combining the environment, the goal, and the resource constraints into a single, very complex, goal description. We have chosen to separate these three components so that goals are simple statements concerning the relationship between the agent’s actions and the environment.

the agent. It says nothing, however, about whether the actions performed will achieve the goal (or solve the problem).

Let the relation

$$\text{achieves}(Ag, G, E, R, A)$$

mean that *if* agent Ag performs actions A in environment E, *then* those actions achieve goal G subject to the resource constraints R. This definition, in contrast to the preceding one, describes only the relationship between an agent's actions and a goal. It says nothing about whether the problem-solving agent would choose to perform these actions when confronted with this goal.

Another peculiarity of this second definition is that it makes goal achievement an all-or-nothing affair. It is often the case that degrees of satisfaction can be assigned to the achievement of goals. This can be accomplished by employing the familiar notion of utility.

Let the statement

$$\text{utility}(Ag, G, E, R, A) = u$$

mean that the utility value of having agent Ag perform actions A with respect to problem $\langle G, E, R\rangle$ is u. To actions A that only achieve G approximately, the utility function can assign intermediate values. Actions A that fully achieve G can be given high utility, and actions that fail to achieve G can be given low (or even negative) utility.

In addition to capturing degrees of goal satisfaction, the utility function can also be employed to assess the cost of the selected actions A by making it inversely proportional to the cost of A. Indeed, the utility function even allows us to capture cases in which we prefer that one agent Ag_1 rather than another agent Ag_2 achieve the goal in question.

It is important to note that, while the 'acts' relation is objective (i.e., all observers would agree on the actions that were taken by agent Ag when it was confronted with a problem $\langle G, E, R\rangle$), the 'achieves' and 'utility' relations are subjective. To apply these relations, there must exist another agent called the "observer" (abbreviated A_o). In the case of 'achieves', A_o must decide whether the actions performed by Ag solved the problem $\langle G, E, R\rangle$. In this case of 'utility', it is A_o who knows and applies the utility function to evaluate the actions performed by Ag.

With these definitions introduced, let us review the alternative aspects of operationality found in the machine learning literature.

4 Aspects of 'Operational'

4.1 Achievable and Testable

One of the fundamental uses of 'operational' is to describe goals that can be directly carried out by an agent. This is the sense in which an operational goal is directly achievable. The goal statement, G, can be viewed as directly describing the appropriate action for the agent, Ag, to take. Let us define this sense of 'operational' as follows:

A goal G is 'achievable' for an agent Ag in environment E under resource constraints R if (a) when the agent is presented with the goal (and the resource constraints) it performs actions A and (b) those actions achieve the goal within the resource constraints. Formally,

$$\text{achievable}(G, Ag, E, R) \equiv \exists A \; \text{acts}(Ag, G, E, R, A) \wedge \text{achieves}(Ag, G, E, R, A).$$

The original sense of 'operational' as introduced by Bridgman applies not to goals but to predicates in general. A predicate's definition is operational if that definition can be applied by an agent to test whether the predicate holds in some environment.

We can define this 'testable' sense of 'operational' in terms of 'achievable' by defining a meta-level goal, whether-P, whose purpose is to determine whether definition P holds. If the meta goal is achievable, then the predicate definition P will be said to be 'testable'.

$$\text{testable}(P, Ag, E, R) \;\equiv\; \text{achievable}(\text{whether-}P, Ag, E, R)$$

This definition reduces 'testable' to 'achievable'. As we indicate below, making certain predicates (e.g., operator selection heuristics) testable is one technique for making other goals achievable.

4.2 Efficient and Accurate

Within the machine learning literature, 'operational' often is synonymous with 'efficient'. For example, in explanation-based learning the goal concept is already achievable before learning starts (for sufficiently generous resource constraints). The purpose of learning in such cases is to convert the goal concept into a form that can be applied more efficiently. To capture this aspect of 'operational', we apply the notion of utility, as defined in Section 3. All that is required is to define the utility function so that it assigns higher utility to actions of lower cost.

Historically, accuracy has not been an aspect of operationality. However, a common strategy for making a goal more operational is to sacrifice some accuracy to gain efficiency (e.g., Keller, 1987). Hence, any formalization of 'operational' should be able to represent this tradeoff. This can also be accomplished by defining an appropriate utility function. The function must simply assign higher utility to actions A that more accurately achieve the goal G.

4.3 More Useful

Employing the notion of utility, we say that one goal G_1 is 'more useful' than another goal G_2 if the actions performed by an agent in response to G_1 are "better" (according to some utility function) than the actions performed by another agent in response to G_2. Following Keller (1988), we would say that G_1 is 'more operational' than G_2.

Formally,

$$\begin{aligned}
&\text{more-useful}(G_1, Ag_1, E_1, R_1, G_2, Ag_2, E_2, R_2) \;\equiv \\
&\qquad \text{acts}(Ag_1, G_1, E_1, R_1, A_1) \;\wedge\; \text{acts}(Ag_2, G_2, E_2, R_2, A_2) \wedge \\
&\qquad \text{utility}(Ag_1, G_1, E_1, R_1, A_1) > \text{utility}(Ag_2, G_2, E_2, R_2, A_2).
\end{aligned}$$

In short, the utility function prefers the actions that agent Ag_1 performs in environment E_1 for goal G_1 to the actions that agent Ag_2 performs in environment E_2 for goal G_2. If the utility function rewards accuracy and efficiency, then agent A_1 must be solving its problem $\langle G_1, E_1, R_1\rangle$ more accurately and more efficiently than Ag_2 is solving its problem $\langle G_2, E_2, R_2\rangle$.

There are a number of interesting special cases of this relationship. For example, if we require that $Ag_1 = Ag_2$, $E_1 = E_2$, and $R_1 = R_2$, then more-useful$(G_1, Ag, E, R, G_2, Ag, E, R)$ says that goal G_1 can be achieved (by agent Ag) more efficiently and more accurately than goal G_2. This formulation captures an efficiency/accuracy tradeoff.

Another interesting case arises when $G_1 = G_2$ and $R_1 = R_2$. Under these conditions, more-useful$(G, Ag_1, E_1, R, G, Ag_2, E_2, R)$ asserts that agent Ag_1 in environment E_1 is more efficient and more accurate than agent A_2 in environment E_2 at solving the same goal within the same resource bounds.

This completes our discussion of the various definitions of 'operational'. The reader should notice that only two notions were required to capture all of these different definitions: 'achievable' and 'more-useful'. In the next section, we discuss what it means to "operationalize" a goal according to these two notions.

5 Operationalization

5.1 Making Goals Achievable

There has always been a potential paradox surrounding the notion of 'operationalization' in the sense of 'making goals achievable'. If an agent accepts a goal, "operationalizes" it, and then executes the operational version of the goal to achieve it, then the goal must already have been operational for the agent, because it was already achievable by that agent.

The paradox can be resolved by making a distinction between the agent that is performing the "operationalization"—call it agent Ag_1—and the agent that will achieve the "operational" form of the goal—call it agent Ag_2. Suppose that Ag_2 cannot achieve the goal as originally specified. However, Ag_1 is able to take the original goal G_1 and reformulate it to produce goal G_2, which Ag_2 is able to achieve. The fact that Ag_1 might itself have been able to achieve G_1 is irrelevant. The important thing to note is that the goal has been operationalized for Ag_2.

For example, consider the LEX system (Mitchell, Utgoff, and Banerji, 1983), which seeks to operationalize such concepts as "states in which applying operator OP3 can eventually lead to a solution" (abbreviated "whether-useful-OP3"). Let agent Ag_1 be the entire LEX system (particularly the problem solver, generalizer, and critic), and let agent Ag_2 be the part of LEX that decides whether a given operator should be applied. This agent Ag_2 works by consulting a knowledge base of heuristic rules that examine the current state and determine which operators are 'useful'. Before learning occurs, the goal 'whether-useful-OP3' is not testable by Ag_2, because it has no heuristic rule for OP3. During the operationalization process, Ag_1 converts the original definition of 'whether-useful-OP3' (stated in terms of an exhaustive search for a solution) into a heuristic rule, which it then gives to Ag_2. Because heuristic rules are directly executable by Ag_2, the goal 'whether-useful-OP3' is now achievable by Ag_2.

In general, operationalization can involve more than just reformulating the goal. Because achievable(Ag, G, E, R) has four arguments, any of those arguments can be changed. Another way of viewing LEX is that, rather than reformulating G, it is changing the agent Ag_2 so that the original goal is now achievable for Ag_2. One can also imagine situations in which agent Ag_1 changes the environment or the resource constraints available to agent Ag_2. In general, the task of an operationalizing agent is to search the space of agents, environments, goal formulations, and resource constraints in order to find a combination that can achieve the original goal.

5.2 Making Goals More Useful

In the discussion of operationalization thus far, we have only considered the process of making a goal 'achievable'. By looking at the 'more-useful' predicate, we can expand the discussion to include cases where the operationalizing agent chooses an efficient, but approximate statement of the goal over an inefficient, but exact goal statement. We can also describe cases where the operationalizing agent replaces one approximation by a better one while keeping the overall computational cost constant.

The task of the operationalizing agent Ag_1 in this case is to search the space of agents Ag_2, environments E_2, goal formulations G_2, and resource constraints R_2 in order to find a combination that maximizes utility. To capture approximate satisfaction of a goal, the utility is measured relative to the *original* goal, G_1. In other words, the goal of the operationalizer is to maximize

$$\text{utility}(Ag_2, G_1, E_2, R_2, A)$$

subject to the constraint that

$$\text{acts}(Ag_2, G_2, E_2, R_2, A).$$

Agent Ag_1 can introduce approximation either by giving Ag_2 a goal G_2 that only approximates the original goal G_1 or by setting $G_2 = G_1$ but having Ag_2 perform a set of actions A that only approximately achieves G_2.

In Mostow's (1981; 1983) Hearts system, FOO, the original goal of "avoid taking points" (G_1) is approximated by the reformulated goal "play low cards in the suit led" (G_2). In a particular game (i.e., a particular environment $E_1 = E_2$), this strategy will not achieve G_1, but it will tend to minimize the number of points taken and thereby maximize utility.[3]

As a side note, there are some situations in which it is possible to "operationalize" one goal by "operationalizing" another goal. In LEX, for example, the overall goal of the system is to solve integration problems. The task of the learning subsystem in LEX is to "operationalize" this goal—that is, to reduce the cost of achieving it. It accomplishes this by "operationalizing" goals such as 'whether-useful-OP3', in the sense of converting them into testable form, so that they can be applied as search heuristics during problem solving.

5.3 Operationalization Over an Ensemble

So far, we have only considered problem solving and operationalization with respect to a particular goal, environment, and resource combination. In general, however, an operationalizing agent Ag_1 may be interested in the effectiveness of problem solving over some collection or ensemble of problems. To formalize this situation, let Π be a collection of problems. Each problem $\pi \in \Pi$ has the form $\langle G_i, E_i, R_i \rangle$.

Again we can distinguish two cases, depending on whether we focus on the 'achievable' or 'more-useful' aspects of 'operational'.

If the measure of success is based on 'achievable', then the task of the operationalizing agent, Ag_1 is to search the space of problem solving agents for an agent Ag_2 for which the largest proportion of Π is achievable. Formally, let the set S be the set of all problems in Π that are solved by Ag_2:

$$S = \{\langle G_i, E_i, R_i \rangle \in \Pi \mid \text{achievable}(Ag_2, G_i, E_i, R_i)\}.$$

The task of Ag_1 is to find an Ag_2 that maximizes $|S|$. We call this maximizing the 'breadth' of the problem solver.

On the other hand, if the measure of success is based on maximizing utility, then another degree of freedom is introduced. We have already seen how the utility measure might permit Ag_1 to trade accuracy for efficiency. Over an ensemble of problems, Ag_1 could trade accuracy and efficiency on some problems for greater total accuracy and efficiency over the entire ensemble. In other words, the task of Ag_1 is to find an agent Ag_2 that maximizes

$$\sum_{\langle G_i, E_i, R_i \rangle \in \Pi} \text{utility}(Ag_2, G_i, E_i, R_i, A_i)$$

where A_i is determined by

$$\text{acts}(Ag_2, G_i, E_i, R_i, A_i).$$

This strategy is pursued in MetaLEX (Keller, 1987), where Ag_2 is modified so that it solves most problems very efficiently (and correctly) but loses its ability to solve other problems entirely. Over the ensemble, its performance has improved (at least for the utility functions considered in Keller, 1987).

[3] The semantics of 'avoid' are troublesome here. We have interpreted G_1 to mean "take no points". By choosing a utility function that assigns higher utility to taking fewer points, we capture the connotation that Ag_1 should "do its best" to avoid taking points.

Finally, it is often the case that some problems in the ensemble are more likely to occur than others. To formalize this, let P be a probability distribution over Π that encodes the likelihood that agent Ag_2 will encounter each of the individual problems within Π. Now the task of Ag_1 is to maximize either the *expected* size of S or the *expected* utility over Π.[4] This permits Ag_1 to allow poor performance on the unlikely problems in order to obtain very good performance on the more common problems.

6 Discussion

The history of 'operationalization' in machine learning has been marked by the progressive identification of additional factors that must be considered in assessing the quality of problem solving performance. The initial papers by Mostow and Hayes-Roth stressed primarily the "directly executable" nature of 'operational' statements, and thus remained quite close to Bridgman's original definition. Subsequent researchers have emphasized additional components of operationality: efficiency, approximation, and overall utility. There are three important points to note about this progression.

First, the term 'operational' has now become misleading, because it has lost its original machine learning (and historical) meaning. Consequently, an important goal of this paper is to suggest more precise terms that can take its place. Instead of the phrase "'operationalizing a goal," we advocate the phrase "improving a problem solver." When discussing the improvement of a problem solver, it is important to define what "improvement" means (in terms of a utility function) and to identify what agent is performing the improvement.

Second, as the definitions of 'achievable' and 'utility' show, there is no limit to the factors that can conceivably be considered in assessing the quality of a problem solver. Because the environment is always included as one of the arguments to these predicates, *any* aspect of the environment can potentially influence 'operationality'. Hence, there is no point in publishing claims of the form "X is an important factor in determining operationality."[5]

Future research in this area should focus on the relationship between changes to problem solving agents and the resulting changes in utility functions. In other words, for the 'adapting' agent, Ag_1 to succeed, it must have some knowledge of the utility function being employed to assess problem solving performance. Furthermore, it must be able to reason with the utility function and select a set of changes to Ag_2 that will increase utility. A nice example of this appears in Keller (1987, p. 110) Table 3-2, where he shows the relationship between his TRUIFY and FALSIFY operators and changes in such performance measures as efficiency, breadth, and accuracy. Examples of more detailed analysis include Minton (1988) and Tambe & Newell (1988), who have attempted to identify the conditions under which chunking increases utility. An area that is largely unexplored concerns identifying situations in which changes to the environment yield significant utility improvements.

These remarks reinforce Keller's observations that information about the utility of achieving a set of goals must be made explicit to an operationalizing agent if we (and it) are to make progress in the understanding and construction of agents that can autonomously operationalize a wide range of goals.

---

[4] Of course, we could estimate the expected size of S as the expected utility over Π by employing a utility function that assigns no utility to unachievable problems.

[5] This was crisply articulated by Sridhar Mahadevan during the 1988 AAAI Spring Symposium on Explanation-Based Learning, when he asked the audience if there were any factors that could *not* influence operationality. No answers were forthcoming.

7 Acknowledgments

We thank Jack Mostow, Lou Steinberg, Leslie Kaelbling, Nick Flann, Rich Keller, Hyam Hirsh, Giuseppe Cerbone, and Peter Karp for valuable comments and observations on earlier drafts of this paper.

8 Bibliography

Bridgman, P. W. (1927). *The Logic of Modern Physics.* New York: The Macmillan Company.

Hempel, C. G. (1952). *Fundamentals of Concept Formation in Empirical Science.* Chicago, IL: University of Chicago Press.

Keller, R. M., (1983). Learning by re-expressing concepts for efficient recognition. In *Proceedings of AAAI-83.* Washington, D. C., August, 1983, 182–186.

Keller, R. M. (1987). The role of explicit contextual knowledge in learning concepts to improve performance. Doctoral Dissertation. Rep. No. ML-TR-7, Department of Computer Science, Rutgers University. See also Keller, R. M., 'Concept Learning in Context'. In *Proceedings of the Fouth International Workshop on Machine Learning*, Irvine. Los Altos: Morgan Kaufman, 1987. 91–102.

Keller, R. M. (1988). Defining operationality for explanation-based learning. *Artificial Intelligence*, 35 (2) 227–241.

Mitchell, T.M., Utgoff, P.E., and Banerji, R. (1983). Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics. In Michalski, R. S., Carbonell, J., and Mitchell, T. M., (eds.), *Machine learning: An artificial intelligence approach, Vol. I*, Palo Alto: Tioga Press, 1983.

Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T., 1986. Explanation-based generalization: a unifying view. *Machine Learning*, 1 (1) 47–80.

Minton, S. (1988). Quantitative results concerning the utility of explanation-based learning. In *Proceedings of AAAI-88.* Los Altos, CA: Morgan-Kaufmann.

Mostow, D. J. (1981). Mechanical transformation of task heuristics into operational procedures. Doctoral dissertation. Department of Computer Science, Carnegie-Mellon University.

Mostow, D. J. and Hayes-Roth, R. (1979). Operationalizing Heuristics: Some AI methods for assisting AI programming. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence (IJCAI-79)*, Tokyo. Los Altos: Morgan-Kaufmann. 601–609.

Mostow, D. J. (1983). Learning by being told: Machine transformation of advice into a heuristic search procedure, in Michalski, R. S., Carbonell, J., and Mitchell, T. M., (eds.), *Machine learning: An artificial intelligence approach, Vol. I*, Palo Alto: Tioga Press, 1983.

Suppe, F. (1977). *The Structure of Scientific Theories.* Urbana, IL: University of Illinois Press.

Tambe, M. and Newell, A. (1988). Why some chunks are expensive. Rep. No. CMU-CS-88-103, Department of Computer Science, Carnegie-Mellon University.