An AIer's Lament

Jeffrey A. Barnett

Northrop Research and Technology Center, One Research Park, Palos Verdes Peninsula, CA 90274

It is interesting to note that there is no agreed upon definition of artificial intelligence. Why is this interesting? Because government agencies ask for it, software shops claim to provide it, popular magazines and newspapers publish articles about it, dreamers base their fantasies on it, and pragmatists criticize and denounce it. Such a state of affairs has persisted since Newell, Simon, and Shaw wrote their first chess program and proclaimed that in a few years, a computer would be the world champion. Not knowing exactly what we are talking about or expecting is typical of a new field; for example, witness the chaos that centered around program verification of security related aspects of systems a few years ago The details are too grim to recount in mixed company. However, artificial intelligence has been around for nearly 30 years, so one might wonder why our wheels are still spinning.

Below, an attempt is made to answer this question and show why, in a serious sense, artificial intelligence can never demonstrate an outright success within its own discipline. In addition, we will see why the old bromide that "as soon as we understand how to solve a problem, it's no longer artificial intelligence" is necessarily true.

Knowledge in Programs

It goes almost without saying that artificial intelligence has a lot to do with a class of computer programs. Therefore, it is logical that we might learn more about the nature of the beast if we knew something more about the composition of programs themselves. I want to start with an obvious claim: Programs that do something in a domain embody knowledge from and about that domain.

Note: This claim does not say the knowledge is necessarily good, accurate, or sufficient. Rather, the claim is, for example, that a program working on chess problems must contain some knowledge about chess, *i.e.*, what constitutes a legal move or what the goal of the game is A very sophisticated chess program could even contain knowledge of strategies, relative values of the pieces, and an opening book Now consider a payroll package; it too must contain considerable knowledge. Tax laws, company policy for sick leave and vacation, who works for the company, their pay rate, and the like, are examples of the kinds of knowledge that must be included.

A moment's reflection should make this claim obvious. Computer programs do contain knowledge– often lots of it. This is true of any program, so it is not the criterion that differentiates artificial intelligence systems from others To do this, we must look more deeply into the knowledge that programs embody.

Coherence of Knowledge

Now that we have agreed that programs contain knowledge, it is natural to ask for descriptions of that knowledge, perhaps even a taxonomy. However, we do not need to pursue this question so deeply for our purposes here

A categorization of knowledge naturally splits into two topics: Quantity of knowledge and quality of knowledge. As far as quantity is concerned, there is not much to say We can simply observe that some programs contain more than others, and, that artificial intelligence systems seem to contain more rather than less than is typical.

There are many possible ways to describe the quality of knowledge contained in a program: Format, domain, chunking, and organization are a few examples Perhaps the most interesting aspect is coherence. A mathematical equation, or its representation as an algorithm, is a highly coherent representation of the knowledge necessary to solve a class of problems. An example of less coherent knowledge is tables or databases that represent some *ad hoc* information about a domain. If the knowledge can

be represented in this way, there must be a fair degree of coherence and structure, it is not as perspicuous as a formula, but it is reasonably coherent.

In some systems, the knowledge is less coherent. Consider today's batch of expert systems. In most, knowledge about the domain of expertise is coded as a set of antecedent-consequence rules. These rules provide the know-how to solve portions of problems However, as a set, the rules may allow the derivation of inconsistent or contradictory problem solutions. This dilemma is addressed by the control structure that selects the rules to use and by confidence measures associated with rule application

One cannot easily predict the behavior of these systems by examining the rule base. This is true even if one is very familiar with the control mechanism and the calculus for combining confidence measures We have a much better feel for the behavior of programs when the embodied knowledge is in the form of mathematical equations and databases

It is in this sense that the knowledge in rule-based systems is less coherent: We simply don't have an *a priori* sense of how they will perform or even what kind of answers they will produce.

It is possible for domain knowledge to be still more incoherent than in the previous examples, and it is in these domains that we will find an application of artificial intelligence technology. Consider a system that is built to understand stories about world economics This system is given symbolic (ASCII) copies of newspaper and magazine articles, and its output is a synthesis of these. In order to perform this task, an incredible amount of knowledge must be included: Various kinds of linguistic knowledge, economics models, knowledge about the major players in the field, geography, agriculture, and global and national politics to name a few things. Even if we assumed that sufficient knowledge to perform this task could be gathered and represented in the computer, we surely would admit that it would almost certainly be incoherent in our meaning of the word.

It is possible for domain knowledge to be still more incoherent than in the previous examples, and it is in these domains that we will find an pervasive technique that is present in virtually every artificial intelligence system. Search.

Search

When knowledge is well-organized, the method of application is straightforward: Integrate or solve an equation, retrieve the answer from a database, and so forth. On the other hand, if knowledge is not well organized and coherent, there is an issue about how to use it to solve problems When all is said and done, we are reduced to some sort of trial and error paradigm. In computer science, this method of last resort is called search. Search is really a simple idea. At any point in the problem solving process, various pieces of knowledge are applicable. The control mechanism picks one and applies it This application can change the problem-solving state. If a solution has not been found, the cycle is repeated until the system's termination condition is met.

Sometimes the application of a piece of knowledge leads to a dead end or the system does not make reasonable progress The control mechanism, in these cases, can restore the system state to a previous one and try a different set of knowledge applications. This is the search paradigm.

As search has been described here, it is merely straightforward trial-and-error problem-solving. If implemented this way, artificial intelligence systems could not work since they contain large amounts of knowledge: The search space would be prohibitively large and the chance of stumbling onto an acceptable answer vanishingly small.

In real life, search is not at all this blind. General knowledge about search, in addition to knowledge from the domain, is used by the control mechanism to be more efficient. The bag of tricks developed by the artificial intelligence community to control search is perhaps its greatest contribution.

There is another way of viewing all this. Artificial intelligence as a technology is an attack on complexity. When the amount of knowledge that must go into a system is large and that body of knowledge does not have a coherent structure, artificial intelligence tries to tame the beast so that something useful can be done.

Beyond Search

It has been claimed by many people that search is a necessary component of any intelligent system, whether the system be natural or artificial. I think that this claim is valid I don't think, however, that search is the ultimate problem-solving strategy; rather, it is the necessary method of last resort.

Domains tackled by artificial intelligence have an interesting history. Though no examples are cited below, my comments are inspired by systems that worked in a variety of areas such as linguistics, game playing, pattern matching, medicine, equipment diagnostics, and modeling. What appears when a domain is tackled by the artificial intelligence community is an epistemological revolution within that domain. Sometimes this revolution is very quiet and goes by unnoticed, but it occurs nevertheless

The first step happens with the initial attempt to collect sufficient knowledge to build the system. In order to represent that knowledge and place it at the disposal of the system, some organizational principles must be detected and exploited. For this task, artificial intelligence provides a bag of tricks as it does for search. But this is not the main gain.

When the fledgling system first starts to exhibit behavior, it is very likely to perform poorly. At first, the

problems can be traced to lack of knowledge. However, once the system knows enough to solve serious problems, another category of issues surfaces: The system dies due to resource limitations. If you have heard rumors that artificial intelligence systems gobble memory and cycles with unquenchable thirst, you have been told the truth.

The computational complexity encountered is a very serious problem. The problem comes about because these systems contain so much incoherent knowledge. But, as we shall see, the cure for this problem, if one can be found, is the addition of more knowledge.

The extra knowledge needed is not like the initial batch. Rather it is knowledge about how to better organize and use that which we already have. It is obvious that nothing else would help. Over a period of time, the domain knowledge is synthesized, distilled and ultimately compiled into a more efficient and usable form. This is the substance of the real epistemological revolution.

Put another way, the knowledge in the system becomes more coherent. In fact, a successful system is often restricted to a subdomain for which the knowledge can be made coherent. At this point of development, the system no longer relies on search as its major problem-solving technique. And at this point, the system is no longer an artificial intelligence system.

Expert Systems

Many times, the knowledge in these improved systems has taken the form of a set of antecedent-consequence rules. As mentioned previously, this is the most prevalent representation technique employed by expert systems. This representation has been so successful as the first step out of the chaos that some believe it to be compelling and universal. In fact, there is a conviction among many people that this technology works so well that it should be our initial target.

I demur slightly on this point for two reasons. The first is that even expert systems do some search and often the encounter between the domain and artificial intelligence causes the development of highly specialized and efficient search strategies for that domain. The second reason is that artificial intelligence encourages one to tackle bigger domains, and the experience gained may help the eventual expert systems to work on a larger set of problems.

Finally, there are philosophical reasons to consider artificial intelligence. Simply put, there are many interesting and important classes of problems for which the knowledge base is just too large and incoherent to consider any other approach. Our desire to process natural language provides many obvious examples. Also, we must realize that man, the ultimate intelligent engine, resorts to trial and error and *ad hoc*ery very often. There is no reason at this moment to think our machines can do otherwise for the really hard problems.

An AIer's Lament

If one accepts the definition of artificial intelligence as a technology that builds computer systems for domains with large incoherent knowledge bases, then one can understand why I am sad. If my field does a successful piece of work—*i.e.*, if an efficient smart system is invented—then we no longer get credit for it. Our major contribution will have been to help you learn enough to not need us anymore My only lasting successes will be the development of new knowledge representation techniques and search strategies. But, alas, these will only be appreciated by my co-workers and there are not very many of them.

When I embarked on a career in this specialty, I had the same dream as every other young scientist: to do something worthwhile and be famous for it. I think the joke is *on me.*