Intelligent Computer-Aided Engineering

Kenneth D. Forbus

The goal of intelligent computer-aided engineering (ICAE) is to construct computer programs that capture a significant fraction of an engineer's knowledge. Today, ICAE systems are a goal, not a reality. This article attempts to refine that goal and suggest how to get there. We begin by examining several scenarios of what ICAE systems could be like. Next we describe why ICAE won't evolve directly from current applications of expert system technology to engineering problems. I focus on qualitative physics as a critical area where progress is needed, both in terms of representations and styles of reasoning.

AI applications can provide challenges that drive the field forward. One example is intelligent computer-aided instruction (ICAI) (Sleeman and Brown 1982). ICAI research has spawned many important studies of cognition and learning, leading to intellectual advances as well as providing the potential to revolutionize education. I believe that applications of AI to engineering problems will play a similar role. Engineering is a particularly good domain for AI, for several reasons.

First, the results are testable. Engineering domains typically have a well-tested theoretical basis, and there is a great deal of codified field experience and empirical knowledge to buttress theory. Such domains provide a solid basis for comparing program predictions and performance.

Second, a broad range of competence and several kinds of knowledge are involved. While much can be learned in simple domains, some issues arise only when a large body of different types of knowledge must be acquired and smoothly integrated.

Third, its roots lie in commonsense reasoning. The formal techniques of engineering are grounded in our commonsense models of the world. Thus, understanding the tacit knowledge of engineers is likely to shed light on our commonsense theories of space, time, and quantity.

Conversely, AI is a particularly good technology for engineering. Engineers are tackling projects of increasing complexity and cost, and they can use all the help they can get. The constraints are gettting more complex: There are new materials and processes to exploit and stricter regulations and legal requirements to satisfy. Making products more quickly and efficiently is even more important in these times of increasing economic competition. Engineering applications give AI an important opportunity to repay the investment society is making in it.

Among the first to see that engineering is a valuable domain for AI was Gerald Sussman, who made these same arguments over a decade ago. The work of his (now defunct) engineering problem-solving group at the Massachusetts Institute of Technology (MIT) illustrated the point well. Several important AI ideas grew out of that group's attempts to automate aspects of engineering reasoning, including truth-maintenance systems (Doyle 1979; McAllester 1980), dependency-directed backtracking (Stallman and Sussman 1977), constraint languages (de Kleer and Sussman 1980; Sussman 1977; Steele and Sussman 1980), and qualitative physics (de Kleer 1975; Forbus 1980). Some of this work has already led directly to useful applications (Finegold 1984). Today many groups in this country are exploring various applications of AI to engineering, including academia (MIT, the University of Massachusetts at Amherst, Ohio State University, and the University of Illinois), government (NASA), and industry (Xerox Palo Alto Research Center, Schlumberger, Westinghouse, United Technologies). Engineering applications are the focus of a number of Alvey and Esprit projects in Europe, as well as several AI projects in Japan. A complete listing is impossible, so I will not make the attempt.

The goal of this article is to state what intelligent computer-aided engineering (ICAE) can and should be. First, I describe what I mean by ICAE. Several highly speculative scenarios are introduced for concreteness. Next,

I explain why current expert system technology is not a very good example of ICAE technology and point out some of its limitations. Because I believe that qualitative physics will be central to ICAE, I next outline the issues and open problems in that area that must be addressed for such applications. My hope is to encourage more research in this area.

What ICAE Systems Could Be Like

The goal of ICAE is to construct computer programs that capture and use a significant fraction of the knowledge, both formal and tacit, of engineers. By making programs that understand things the way engineers do, we can make programs that can communicate more easily with human engineers as junior partners. More important, if we can capture the ways in which the best human engineers reason, we can make engineering tools that spread this expertise among a broader community, as suggested for very large scale integration technologies(VLSI) in Stefik and Conway (1982).

There have already been significant AI applications in engineering (see, for example, Sriram and Adey 1986], but while they are important first steps, they are not even close to the capabilities I believe we can eventually achieve. The best way to illustrate the difference is with several scenarios illustrating these capabilities. It is important to note that the programs described in the following three subsections exist only in the imagination; but I believe they are possible, if we are willing to make the necessary effort

ICAE in Design

Scenario One: Suppose we wish to design a wrist watch. Furthermore, we might imagine we are designing it for someone eccentric enough to want to own a mechanical watch. We begin by consulting our able computer assistant. The program informs us that any clock must have an oscillator, a power source, a drive train, and a display. It also provides several initial suggestions about how to implement each functional component from a catalog of typical designs.

Thrilled at the prospect of designing an "original antique," we ignore the computer's initial suggestions and instead begin with a recoil escapement, using a pendulum as the oscillator. But before any specific parameters have been chosen, our assistant warns us that this choice is unsuitable. It explains that a pendulum requires a constant orientation to work because it depends on gravity, while a wrist watch must be able to function at any orientation. Chastised, we retract our choice and accept the computer's suggestion (After all, it has been quite a while since we last designed a mechanical watch!).

As we begin to choose ways to implement the required functions, the computer checks for further inconsistencies. Whenever some choice forces others, this fact is noted. At any stage we can ask why the computer suggested certain choices or even ask it to remind us why we made other choices.

As soon as enough information is specified, the system performs a qualitative simulation to find out how the artifact can behave. For many artifacts this qualitative simulation can be performed much earlier in the design process than any numerical simulation. (For a wrist watch or other mechanical system, we need to choose the shapes of the components.) The envisionment performed by the computer reveals that the desired behavior of the watch is indeed one of the possible behaviors predicted. However, a comparative analysis shows that if the temperature of the environment changes, the period of the oscillator will change. In particular, the period increases monotonically with temperature—as the temperature increases the diameter of the balance wheel rises, slowing the movement, and as the temperature falls, the diameter decreases, causing the watch to gain time. A design change is made to reduce this source of error. Instead of making the wheel out of steel, a brass outer rim is added, and two gaps are cut in the circumference. Since brass expands faster than steel, the two rim pieces swing inward as the temperature rises, thus maintaining the distribution of weight (Harris 1984).

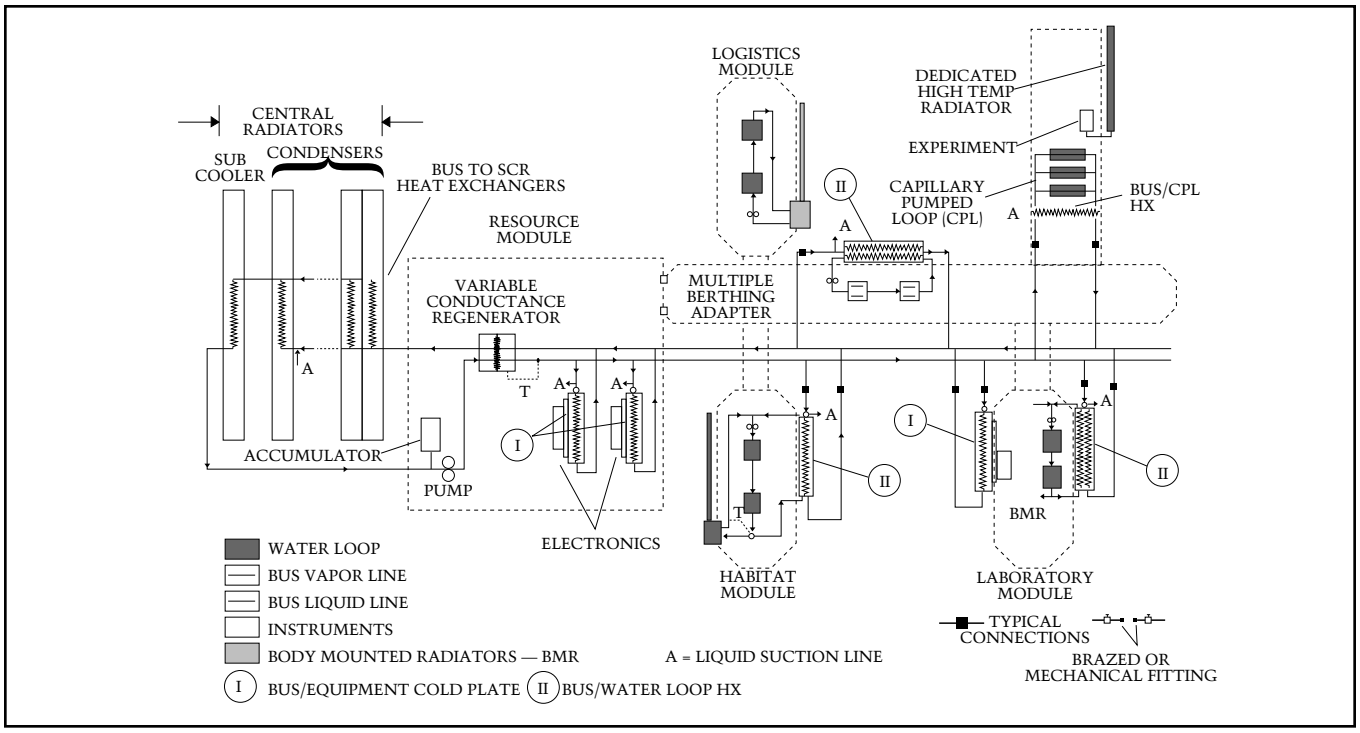
Scenario Two: Alternatively, suppose we are designing a thermal control system for a space station as shown in figure 1 (Malin and Lance 1985a, 1985b; Malin and Harris 1987). We might choose to use ammonia as our working fluid and to use the same heat exchangers for heating and cooling to reduce the total system weight. An important criterion in designing spacecraft systems is "fail operational first, and fail safe next."[1] By representing what can occur with all combinations of system configurations, set points, loads, and failure modes, the computer shows all the dangerous states the system might be placed in by operator error or malfunctions. Knowing the space of potential problems enables the designer to modify the system to minimize their impact. For example, the design can be changed to minimize the effects of operator error and provide more options for temporary reconfiguration to aid failure management.

Once the design is completed, the computer generates detailed manuals of procedures for starting, operating, and maintaining the thermal control system. It also generates a set of rule-based expert systems for monitoring the system's operation, fault diagnosis, and failure management.

ICAE in Plant Operation

Scenario Three: As process plants grow more complex, the cognitive load on the operator grows. An ICAE plant monitor could relieve this load by summarizing the state of the system in easily understandable, qualitative terms. The monitor responds to operator queries using a combination of natural language and graphics. When something goes wrong, the monitor should be able to make suggestions about the source of the problem. More important, the human operator should be able to tell the monitor what is thought to be wrong. The monitor then helps refine this hypothesis, by serving as a "devil's advocate" or "doubting Thomas,"[2] critically evaluating the person's theory and offering tests and measurements that might confirm or disprove it.

The monitor becomes the *computational infrastructure* of the plant. It



(Photo Courtesy, Dr. Jane Malin, NASA Johnson Space Center)

Figure 1. An Abstract Schematic of the Space Station Thermal Control System.

sifts through data about breakdowns and repair times in order to predict when systems are likely to fail and to figure out how maintenance schedules might be improved. It shares its findings with the monitors in other plants, spreading innovative techniques and cautionary tales of things that can go wrong.

Suppose a radical reconfiguration of the plant is needed, either because the product is being changed or a more economical process has been developed. Instead of being thrown away, the monitor becomes an active participant in the restructuring. Using its knowledge of the plant, it suggests various ways that existing stock can be used to carry out the new plan. Once a new configuration has been decided upon, the monitor automatically compiles the appropriate sensory and control programs for its slave computers, which handle the brunt of real-time processing.

ICAE in Computer-Based Training

The complexity of modern engineered systems can be staggering. Training people to operate and maintain such systems is a difficult problem, especially because the trainees often lack the requisite education (Brown, Burton, and de Kleer 1982; Stevens et al. 1981; Hollan, Hutchins, and Weitzman 1984; Woolf et al. 1986). ICAI promises to provide high-quality tutoring that will cost much less than human tutors. But so far it has been hard to deliver on this promise; only a handful of such programs have been developed, and few have seen wide application (Wenger 1987). Part of the problem has been that, until recently, the computing technology available has been woefully inadequate. Even today, many desirable capabilities simply cannot be supported. But most of the trouble is that it is extremely difficult to develop ICAI programs. Years of eliciting knowledge and many cycles of programming are typical. To achieve a high impact on education, we must automate the production of intelligent tutoring systems.

It seems unlikely that a program reasoning directly from first principles will be able to generate fluent real-time explanations of complex systems. People rarely do. For example, consider an expert in, say, automatic combustion controls in U.S. Navy propulsion plants. The expert may have served mainly on ships that use a particular brand of pneumatic component (for example, Bailey controls). When asked to explain an automatic combustion control system made from components from another manufacturer (for example, Hagen), the expert often gets stuck, even though functionally the systems are identical. After studying the Hagen system for a while, however, the expert can explain the new system almost as well as the ones the expert is familiar with. It seems there are compiled explanations that the expert can bring to bear on familiar systems or systems built out of familiar components. Given an unfamiliar system, the expert needs time to extend understanding to include the new kinds of devices.

Scenario Four: Suppose you want to build a computer-based trainer for a newly designed power plant. Imagine a tutor compiler that worked as follows. In an interactive dialog with the compiler, you provide a structural description of the system to be explained, the typical student's prior knowledge, student's likely mental models (Gentner and Stevens 1983), and what class of

questions the tutor must be capable of answering. Notice that I am specifying the tutor much more abstractly than current CAI authoring systems allow—I am not specifying specific questions and specific answers. You might want to specify a lesson plan, but in more general terms: "Make sure they understand the difference between heat and temperature." You also specify the hardware environment the tutor will be operating in—the kinds of computer, displays, and input devices.

At the end of the interactive session, the tutor compiler tells you when to come back and pick up your tutoring program or roughly how long it will be before it can tell you whether the program is possible on the hardware you specified. It then sets to work, generating a special-purpose tutor for the desired system that will run efficiently on the specified hardware.[3] The tutor it generates may be quite dumb, consisting of nothing but canned graphics and text, depending on the hardware involved. While human beings can write such programs, they take months (or years) to do so and often need many cycles of debugging—generating canned lessons is boring! The tutor compiler won't get bored, although it will probably take a long time by the standards we are used to—hours, days, or perhaps even weeks. If the tutor is to be widely used, or the system being taught is sufficiently critical, even months of computer time is a reasonable price for increased reliability.

Decomposing ICAE

Let's look back over these utopian scenarios and identify some of the knowledge and skills involved.

Broad domain models: An ICAE program must know about a broad range of physical domains to reason about real systems. Engineers reason about liquids, gases, chemical reactions, mechanics, elasticity, electricity, and many other domains. Many engineered systems require reasoning with several domains at once.

Layered domain models: Qualitative knowledge is essential for ICAE, but it must be integrated with quantitative knowledge, in the form of equations, empirical relationships, and numerical bounds.

Routine design: An ICAE design system will have a catalog of standard designs, in order to solve routine design problems quickly. These problems include making a small variation in a standard product to meet a special need and performing subtasks in a larger, creative design task (Dixon and Simmons 1984).

Functional descriptions: An ICAE system will need a vocabulary of functional descriptions to analyze existing artifacts, hierarchically design new artifacts, and communicate better with engineers during the initial phases of a design. Examples of such concepts include filter, regulator, oscillator, sensor, and actuator (Forbus 1984a).

Qualitative simulation: Existing computer-aided engineering (CAE) systems often provide numerical simulation facilities for checking designs. However, a numerical simulation is useless in the early stages of design because too little is known. In later stages, numerical simulation can be used, but any serious problems uncovered then are often fixed only at greatly increased expense, both in terms of designer time and computer time. Aside from the ability to catch blunders early, knowing early in the design process what undesirable behaviors might be exhibited can allow constraints to be posted on other choices in the design to prevent such possibilities from occurring.

Procedure generation: Designing a new artifact often results in a second, neglected design problem: developing procedures to operate and maintain the artifact. We can imagine a procedure compiler that automatically identifies what kinds of procedures are needed and generates them as appropriate. Creating procedures requires knowing which aspects of the artifact can be manipulated by the operator, what kinds of conditions are likely to occur, and what conditions constitute unsafe operation. The more the procedure compiler knows about the mental models operators will use to understand the artifact, the better will be the procedures it can construct. If the procedure compiler can be brought into play early enough, operational criteria can influence the design process to create systems that are safer and easier to use.

Failure analysis and fault-tree generation: By including fault models in the qualitative physics used to analyze the design, we can predict, in advance, many possible failure modes of the system. These potential failures could be used in several ways. Fault trees could be generated for calculating the probability of various failure modes. Simple rule-based expert systems could be automatically generated to help monitor, operate, and maintain the system, thus bypassing the traditional knowledge engineering bottleneck (Sembugamoorthy and Chandrasekaran 1984).

Communication skills: People will still be in the loop, and ICAE systems must communicate effectively with them. This communication requires advances in natural language understanding and generation. It also requires advances in direct-manipulation interfaces (Hutchins, Hollan, and Norman 1985); and it requires an understanding of typical mental models, so the program can couch explanations in understandable terms.

Computer individuals: In his 1983 AAAI presidential address, Nils Nilsson advocated the development of AI programs that are never turned off as a way of studying what might be termed "computational ecology." Such architectures will be necessary for ICAE systems that act as the institutional memory of a factory over its operating lifetime.

The Road to ICAE

Despite exciting recent progress in AI, we are still far from building programs that can perform at the level described in these scenarios. No combination of current technology will suffice; we need fundamental advances in the state of the art. This section examines why.

Why Not Rule-Based Expert Systems?

Rule-based expert systems have formed the bulk of AI applications to date. This technology moved from the laboratory in the mid- to late 1970's, and through the microelectronics rev-

olution it has seen widespread industrial application. However, this technology also has well-known limitations, as Davis (1982) and others have pointed out. These problems include the following:

1. Narrowness. Traditional rule-based systems have no common sense. They do not contain a range of solution techniques, using simpler ones to solve simple problems with less work and applying more complicated techniques only when necessary. As de Kleer once said to me, "An intelligent problem solver has to be able to answer stupid questions."

2. Uncertain coverage. Simple rule-based systems are typically generated from particular examples (either by hand-crafting or inductively). Consequently, it is hard to determine just how completely they cover the domain in question.

3. Brittleness. The performance of human experts degrades gracefully in the presence of incomplete or erroneous information. Human experts can tell when they are lost. Neither fact is true for standard rule-based expert systems.

Qualitative physics is crucial because it helps capture the commonsense understanding of the world that is the foundation of engineering knowledge.

Much of today's AI research is aimed at overcoming these limitations. Despite such drawbacks, this technology has proven quite useful. In part, it has been successful because classical rule-based expert systems were the first widespread application of symbolic computing. Even in engineering, where numerical computing is essential, numbers alone are no substitute for symbolic representations. Narrowing the gap between the user's way of thinking and the computer's language resulted in an explosion of enthusiasm. What if we could narrow that gap even more?

The deepest problem with current fielded expert systems technology is that it intermingles domain knowledge with control knowledge about the particular kind of task. Certainly partitioning the program into "knowledge base" and "inference engine" enforces modularity better than simply writing procedures. However, the authors of rules know they are writing the knowledge base for a particular task, and that fact affects the contents of the knowledge base. For example, DENDRAL's predictive rules for mass spectroscopy had to be rewritten to be used for interpretation (Buchanan and Shortliffe 1984). In reexamining MYCIN's knowledge base from the perspective of tutoring rather than diagnosis, Clancy discovered that many control decisions for diagnosis were encoded in the structure of the rules (Buchanan and Shortliffe 1984). This argument is not to say that such modularity cannot in principle be imposed with today's expert system shells. Many expert system tools (KEE, ART, and others) include frame systems and other constructs for such purposes. But I suspect from talking with users of these systems that this distinction is more honored in the breach—it is just too convenient to build in only the distinctions you need for the task. Indeed, this strategy is exactly right for building today's expert systems, but I do not believe it is the right strategy for tomorrow's.

The intermingling of content and intent makes the knowledge-acquisition problem harder than it needs to be. Suppose we want to build a set of ICAE tools that can be used for any engineering problem. Engineering involves a large number of domains and several different styles of reasoning. Encoding this knowledge in today's expert system technology would require representing the knowledge separately for each domain and each kind of reasoning. If there are N domains and M styles of reasoning, we have N * M research problems.

A Methodology for ICAE

Consider a more modular strategy. If one can develop problem-independent representations for domain knowledge and domain-independent theories of the kinds of reasoning involved, then one has only N + M research problems. True, these N + M problems might be harder than the individual N * M problems. But there is already evidence that this is not necessarily the case. Theories of diagnosis (de Kleer and Williams 1986), measurement interpretation (Forbus 1986), and comparative analysis (Weld 1988) are successful examples of domain-independent styles of reasoning (by successful, I do not mean that these problems have been solved once and for all, as the section on reasoning issues makes

clear); and the domain models developed in qualitative physics research (de Kleer and Brown 1984; Forbus 1984b; Kuipers 1984; Williams 1984) tend to be independent of the style of reasoning used.

It is worthwhile to dwell on the notion of a domain model from qualitative physics a bit because it is often misunderstood. When someone decides to develop a qualitative model, there is generally some system or phenomenon being modeled and some purpose in mind for that model. I call the system or situation being modeled the *scenario* and its qualitative model the *scenario model.* The modeling process consists of building that scenario model from something. How does this process work?

Building a specific model for a specific purpose is the simplest path, analogous to the way that expert system applications are built today. There is abundant evidence that the results of this path are useful. But having built one special-purpose model may not provide much help in tackling the next scenario model. For long-range progress, a better path is to start with a domain model that describes a class of related phenomena or systems. Ideally, any scenario models required can be built by composing descriptions from the domain model. Developing a domain model requires more work than developing a model for any specific scenario. The payoffs are that (1) generating a scenario model becomes easy, for some range of scenarios, and (2) ad hoc models are harder to generate, since the constructs of the domain are used in a modular fashion by a variety of examples.

Theories of qualitative physics differ in how they organize domain models. Qualitative process (QP) theory (Forbus 1981b, 1984b, 1984c) organizes domain models around processes, which can be automatically instantiated to form scenario models. Device-centered ontologies (de Kleer and Brown 1984; Williams 1984) provide catalogs of devices, which can be composed to build scenario models. (Kuiper's [1986] QSIM does not provide any abstraction or organizing structure for domain models itself, but one could imagine using it with either ontology.)

It is important to note that there is nothing inherently qualitative in this notion of domain model per se. Domain models could contain arbitrary symbolic or quantitative knowledge, for example. What is important is that (1) the domain model can be used to build scenario models that span a well-defined variety of systems or phenomena and (2) the constructs of the domain model must support several styles of reasoning. For example, a knowledge base for engineering thermodynamics should enable the construction of models for broad classes of refrigerators, power plants, propulsion plants, and so forth. It should support making predictions, interpreting measurements, troubleshooting, and design.

There are significant advantages to this strategy, beyond reducing the amount of research required. Separating the development of the domain models from the styles of reasoning clarifies both. It is easier to see just what phenomena a domain model covers when it is not entangled by considerations of how it is to be used. By freeing the description of a style of reasoning from the language of a single domain, we can understand better what is required to perform it and exactly when it will and will not work.

Qualitative physics is not the only area of AI that is converging on such ideas, although arguably it has carried them farther than most. Many threads in expert systems research point in the same directions. For instance, work on model-based reasoning (Brown, Burton, and de Kleer 1982; Stevens et al. 1981; Hollan, Hutchins, and Weitzman 1984; Davis 1984) overlaps significantly with qualitative physics, and it makes use of more or less the same notion of domain model. Chandrasekaran's (1983, 1985) generic tasks are one codification of styles of reasoning. The creation of expert system shells specialized for particular kinds of tasks, such as EMYCIN, is another. Lenat's CYC project (Lenat, Prakash, and Sheperd 1986) can be viewed as an attempt to generate a high-level domain model for all commonsense reasoning. Various expert system shells, such as IntelliCorp's KEE and Neuron Data's NEXPERT are beginning to provide support for model-based reasoning. And no doubt there are others. What I want to emphasize is that the construction of large knowledge bases and the codification of general styles of reasoning require explicit research; they will not arise as a side effect of adding more rules to the typical systems being applied today.

Qualitative Physics in ICAE

Constructing powerful ICAE systems will require the confluence of several computing ideas and technologies, including advances in symbolic algebra, databases that represent design choices and their dependencies, and heterogeneous processing systems for efficient execution of mixed symbolic and numerical computations. However, this article focuses on one key area: qualitative physics.

Qualitative physics is crucial because it helps capture the commonsense understanding of the world that is the foundation of engineering knowledge. What we know about the physical world is vast compared with what we learn from textbooks in school. The formal knowledge codified in the engineering curriculum must be embedded in this richer framework to achieve humanlike performance. Qualitative physics complements traditional techniques in several ways: (1) it can provide answers even with very little data, (2) it provides an organizational structure for more detailed forms of analysis, and (3) it provides a more natural language for communicating with human engineers. These points have been amply made in the literature (for example, de Kleer 1975; Brown, Burton, and de Kleer 1982; Forbus 1980; Forbus and Stevens 1981), so I will not elaborate on them here.[4]

The qualitative physics enterprise began just over 10 years ago, but only in the last 4 or 5 years has it received intensive effort. While there are some indications of successful applications at the current state of the art (including Dauge, Raiman, and Deves 1987; Hamilton 1988; Garrett, Lee, and LeClair 1987), I believe the real potential lies ahead. The next subsections focus on the research that will be required to bring the field up to the

quality and scope of human engineering.

Modeling Issues

We have reconstructed only a small portion of what people—both expert engineers and people on the street—know about the physical world. We have several notations for expressing time-dependent differential equations and can model simple dynamic systems. We can even model simple spatial systems, although much less progress has been made on that front. I would bet that most of the qualitative physics enterprise lies ahead of us.

Even if only a small number of these (say, five) [domains] were needed by any particular ICAE system, we are still faced with the prospect of 50,000 axioms, and that is just the qualitative knowledge.

Extending the Range of Phenomena. Many phenomena can probably be formalized using existing techniques but simply haven't been yet. For example, small-signal and direct-current analyses of simple electronic circuits have been successfully solved, but to my knowledge, little has been done on alternating-current or radio frequency analyses. Frequency-domain analysis remains unexplored. We can deal with essentially slow liquid flows and some phase changes, but most of thermodynamics, fluid statics, and fluid dynamics remains to be formalized. There has also been some exploratory efforts on materials, conservation laws, and energy (see Forbus 1984b; Davis, R. 1987).

Most existing domain models in qualitative physics are small. How big will the qualitative vocabulary of an ICAE system be? Finding the right metric is difficult, but, following Hayes (1979), a reasonable view for ballpark estimates is to use "axiom-equivalents." Our early QP models of liquids contain roughly 300 axioms.[5] Given our experience in expanding these models to cover more phenomena, and thinking about the phenomena we still have not represented, it seems reasonable that perhaps an increase of a factor between 3 and 10 would suffice. That would be between 1,000 and 10,000 axioms.

Now, that is only one domain. Depending on how one carves up the world, there are easily 10 or 20 domains that would be of widespread use in engineering. Even if only a small number of these (say, five) were needed by any particular ICAE system, we are still faced with the prospect of 50,000 axioms, and that is just the qualitative knowledge. There is still knowledge associated with control, with quantitative knowledge (see following), all the minutiae of specific constants, random facts about the world such as the reliability of suppliers, and so on. ICAE systems will have serious memory requirements!

The important point about the methodology of qualitative physics is that the knowledge should accumulate. If we build ad hoc models for each new system, we are doomed. But if we build powerful domain models, analyzing a new system simply consists of re-applying the concepts we have already formalized and tested.

Ultimately, we would like to see standard libraries of qualitative models, just as there are standard libraries of numerical analysis routines and conventional approximations for complex equations. All the benefits of standardization—certification, proofs of correctness, and so on—should eventually be attainable for qualitative models. Many engineering fields already have established conventions for determining what kinds of approximations to use, so the existence of qualitative model libraries will merely extend the spectrum of available representations.[6] Developing the knowledge base for a new ICAE system should become mainly a process of selecting the right off-the-shelf vocabularies, rather than the laborious knowledge engineering protocol of today's expert systems.

Extension Theories of Quantity. For many engineering purposes, purely qualitative answers are insufficient. The qualitative analysis will serve as a guide to carrying out more detailed analyses, such as algebraic manipulation (for example, de Kleer 1975) or numerical simulation (for example, Forbus 1980, 1981). Some excellent starts have been made. Simmons (1986) has developed utilities for integrating inequalities, intervals, relational arithmetic, and constant elimination. Kuipers and Berleant (1988) use quantitative bounds on functions to constrain qualitative simulation. Several formalizations of order-of-mag-

nitude reasoning (Raiman 1986; Mavrovouniotis and Stephanopolous 1987; Davis, E. 1987) have been developed. The tacit knowledge mathematicians bring to bear is also being studied (Sacks 1985; Yip 1987).

Reasoning with Multiple Ontologies. A full understanding of complex systems, such as thermal and fluid systems, often involves integrating results from more than one point of view. For example, reasoning about the efficiency of a propulsion plant requires thinking about how the properties of the fluid circulating in the system change as it goes through different components. Essentially, one thinks about *little pieces of stuff*, a collection of molecules taken together as an object whose properties are to be reasoned about.

The problem with the little-pieces-of-stuff viewpoint is that, by itself, it is insufficient to figure out what is happening in the plant. Somehow the fact that the liquid is circulating must be ascertained, a fact that is impossible to establish without considering mass properties of the fluid in the various containers. By contrast, the contained-liquid ontology introduced by Hayes (1985) is ideal for establishing this fact. Being able to switch from one ontology to another appears to be essential for solving many problems concerning fluid systems.

A special case of the little-pieces-of-stuff view is handy in engineering thermodynamics. The idea is to consider an infinitesimal piece of fluid, so small that it never splits into two parts when flowing through a system, but large enough to have macroscopic properties. An envisionment using this molecular collection ontology can be computed from a description using contained stuffs. Any particular state in the contained-stuff envisionment describes what processes are active, including flows of mass and heat. A simple set of rules applied to these process descriptions suffices to produce a new envisionment, which describes how this abstract infinitesimal fluid piece moves and what happens to it as it goes from place to place. This description can be used to recognize thermodynamic cycles and heat pumps (see Collins and Forbus 1987).

There are still other ways of individuating fluid stuffs that have yet to be explored, such as macroscopic pieces of stuff. Right now we do not know how many other cases of ontological shifts appear in engineering reasoning, nor what other kinds of relationships between ontologies there might be.

Ultimately, we would like to see stadard libraries of qualitative models, just as there are standard libraries of numerical analysis routines and conventional approximations for complex equations.

Formalizing Modeling Assumptions. Engineers constantly use simplifying assumptions to manage complexity. For example, when trying to figure out how a system containing a heat exchanger works, engineers tend to assume that the fluid in the hot leg is hotter than the fluid in the cold leg. If the system is operating as intended, making this assumption saves effort because the other two alternatives (the temperatures being equal or the cold-leg temperature being higher than the hot-leg temperature) need not be considered. If the system is not operating as intended, then the engineer's predictions will be wrong and the analysis must be redone to consider the other alternatives.

One distinguishing feature of QP theory is that it provides the ability to formalize modeling assumptions. QP theory does this formalizing in two ways. First, the conditions under which a description can be instantiated are formally specified in the domain model. By contrast, the modeling work of choosing which device should be used to model parts of a physical system is done by the user in other theories of qualitative physics (those of de Kleer and Brown, Williams, and so on.). Second, the preconditions field of processes and time-varying relationships provides an interface for information that normally lies outside the realm of QP theory, such as geometric relationships.

However, almost all qualitative models developed to date are flat; that is, they assume the input descriptions are in terms of the theoretical vocabulary, and they make as few assumptions as possible about the operating environment. This is a good strategy for developing the basic representation of a domain, and continuing such studies is crucial. But soon we must also learn how to construct domain models that provide layers of varying approximations, with explicit simplifying assumptions to record the dependence on them for explanation and backtracking. We believe that such layered models can provide a way to gracefully extend more basic models to capture the competence of human engineers.

There are several issues involved in constructing such models. First is mapping from real-world objects to abstract objects. A manifold can typically be modeled as an abstract container. However, this approximation breaks down if the material in it is corrosive, if it is punctured, or if it is placed in thermal contact with some other working fluid.

Second is representing normal behavior. The behaviors of a well-designed system are a small subset of the logically possible behaviors. Analysis is simplified when unlikely behaviors can be ignored. We believe that many of these descriptions can be succinctly represented by the individual views of QP theory.

The third issue is representing faults. Faults represent violations of the mapping from real-world objects to their intended abstractions. Some faults may be represented by retracting assumptions of normal behavior (for example, equal temperatures in both legs of a heat exchanger). Others, such as leaks, may require extra explicit assumptions.

We have begun exploring these issues and have developed a set of conventions for modeling assumptions

that allows us to construct large, multiperspective, multigrain qualitative models (Falkenhainer and Forbus 1988). Our initial test case has been a high-level model of a U.S. Navy propulsion plant, and next we plan to test these ideas on modeling thermal control systems for NASA. However, much research remains.

Spatial Reasoning. Motion has been analyzed for special cases: linear one-dimensional (Forbus 1984b), sliding in one dimension (de Kleer 1975), free motion in two dimensions (Forbus 1980, 1981a), and some mechanisms (Gelsey 1987; Nielsen 1988; Faltings 1987). However, despite its importance, little general progress has been made on qualitative spatial reasoning.

We claim that, unlike dynamics, there is no useful general-purpose, purely qualitative representation of space (the poverty conjecture; see Forbus, Nielsen, and Faltings 1987), and thus all attempts to find one are doomed to failure. The model we propose instead starts with a quantitative, diagrammatic representation (the metric diagram [MD]) and computes an appropriate qualitative representation from it (the place vocabulary [PV]). We have demonstrated the MD/PV model for free motion of point masses in two dimensions (Forbus 1980, 1981a) and reasoning about mechanisms (Forbus, Nielsen, and Faltings 1987; Faltings 1987; Nielsen 1988).

The MD/PV model has allowed us to achieve a milestone in qualitative physics: In February 1988, we successfully completed the first envisionment of a mechanical clock. The input description is a diagram specifying the shapes of the parts and a description of which parts could potentially touch. (The latter was provided only to reduce computing time; the system is capable of determining that two parts can never touch.) The output envisionment includes the normal behavior of the clock, as well as a variety of "stuck" states and odd behaviors. The analysis was carried out entirely from first principles; that is, no special-purpose knowledge about gears or escapements was used. At this writing we are still analyzing the results and testing the system on more examples.

The MD/PV model implies that progress in qualitative spatial reasoning is directly tied to better quantitative representations of shape and space. Several such representations are being developed in vision and robotics—Lozano-Perez's (1983) configuration space formalism, for example, and Ullman's (1985) visual routines. Using these representations to construct qualitative representations appears quite promising. For example, Faltings (1987) describes a theory and implemented system that computes place vocabularies for a mechanical clock using configuration space.

One way that spatial reasoning integrates with dynamics is the notion of spatial derivatives. By this I mean the analysis of systems best described by partial differential equations, with or without time. Examples include the stresses on different parts of a bridge, the flow of wind over an airplane wing, and the distribution of temperature during a heat flow across a metal bar. Developing formal qualitative laws for partial derivatives will probably be straightforward. The difficult part seems to be formalizing the strategies that tell us how to divide the shape under analysis and what directions to use as our frames of reference. Is there a qualitative physics that can naturally encompass field phenomena?

Reasoning Issues

CAE systems will require computational resources that seem prodigious by today's standards. It would not surprise me to find that several orders of magnitude of improvement in speed and memory capacity are required for the utopian scenarios presented earlier. If we are lucky, current technological trends will provide us with such power before too long. Happily, the situation is not an all-or-nothing game; in the meantime, much can be done even with today's computers. (However, we may be seeing an analog of Gresham's law operating in AI computing: Manufacturers are focusing on low-cost "delivery vehicles" in hopes of short-term profit, rather than on developing machines to support extending the state of the art. For instance, several vendors tell me their marketing studies indicate that no one needs larger address spaces. In the long run that cannot be true; to build systems that know more, you must put that knowledge somewhere.) Here we examine some styles of reasoning that use qualitative models and point out several areas where advances are needed.

Prediction. Since qualitative models quantize continuous parameters into a finite symbolic vocabulary, in principle there are only a finite number of qualitative states for any system. Thus, one can generate all possible states of the system and all transitions between these states, which is called *envisioning*, and it is the best-explored qualitative simulation technique.

The envisionment of a system can serve as a knowledge base for a variety of programs. For example, a key step in measurement interpretation (reading gauges) is finding the qualitative states that could explain a particular temporal segment of a system's behavior. If the envisionment for a system is known, then potentially a set of lookup tables could be precomputed that drastically reduce the computational burden of this step (see Forbus 1986).

Exploring all alternatives is critical for many ICAE tasks. Some system behaviors can correspond to uneconomical or undesirable states, and given the knowledge that they might happen we can take steps to see that they don't. A subset of potential behavior not considered by a careless operator or pruned out by a heuristic might contain a potential catastrophe. Envisioning is particularly important in developing new qualitative models because we want our models to predict all and only those behaviors that some physical system might produce. People tend to be unsystematic in testing models; rarely do they set up "stupid" initial conditions and see what happens. People also tend to ignore low-probability states. Yet these precise situations could arise in an application, and bizarre behavioral results will cause some unpleasant surprises. Unfortunately, the computational

complexity of envisioning will require developing new techniques for reasoning with larger models. We discuss this issue in detail in the section Scaling Up.

Any path through an envisionment corresponds to a predicted possible behavior of the physical system, or a potential history of that system (as in Hayes 1979, 1985). However, Kuipers (1986) has demonstrated an algorithm that generates histories directly, rather than indirectly through an envisionment. This history-generation algorithm is yet another kind of qualitative simulation, with some very useful properties. The states in an envisionment are generic in that numerical values are distinguished only with respect to global properties of the domain. The states produced by Kuiper's QSIM program include newly generated landmark values to represent the specific values of quantities at particular times. As Kuipers points out, the landmark values produced in history generation are essential to a detailed analysis of specific behaviors.

Unfortunately, Kuipers has also shown that direct history generation, because of its locality, can lead to spurious predictions about behavior. Worse yet, such spurious predictions can even arise when generating histories indirectly from envisionments. To constrain successive choices of landmark values and avoid predicting physically impossible behaviors, more research is needed to determine and appropriately encode path constraints. Work is proceeding on identifying such constraints (for example, Lee, Chiu, and Kuipers 1987), but much remains to be done. I believe that one useful way to constrain landmark introduction is to integrate it with the more generic descriptions produced by envisioners. For example, the envisionment could be used to easily identify cycles of behavior (corresponding to oscillations), and landmark introduction could be used to analyze the behavior in more detail.

Measurement Interpretation. The ATMI measurement interpretation theory (see Forbus 1986, 1987) produces explanations of observations taken over time in terms of the states of a qualitative model. It has been demonstrated on simple examples where the input measurements were hand-processed. We have begun testing routines for automatic translation of data from numerical simulators. However, at least two critical issues must be addressed before we will have an implementation capable of dealing with real data. First, what are the best strategies for recovering from errors induced by noisy data? We are beginning to explore this question experimentally by introducing noise into our simulators. Second, what are the best ways to represent estimates of rates and durations for heuristic pruning? For example, suppose two different states can explain a temperature rise during a particular temporal segment. If we knew that the rate of rise was larger than could be explained by one of the states, then we would have a unique explanation for that segment.

Comparative Analysis. Suppose we have just designed a control system and discovered that its response time is too slow. What properties or parameters of the system could we change to increase the response time, without degrading other performance criteria? Answering this question requires us to analyze, in depth, the dependencies of the situation and to find the effects of a small perturbation of them. This kind of comparative analysis (Weld 1987) is receiving increasing attention in qualitative physics. In Mohammed and Simmons (1986), such an analysis is used to help debug integrated circuit fabrication lines. The idea is to generate a history for the system with equations linking the various changes that happen across time. By performing a sensitivity analysis on the resulting system of equations, an appropriate change can be found.

The best work in this area is Weld's (1988) recent Ph.D. thesis, which presented a domain-independent theory of comparative analysis, explored both by formal analysis and by implementation. He formalized the variety of differential analysis suggested in Forbus (1984b) (finding and fixing rather signficant flaws), developed an entirely new kind of comparative analysis called *exaggeration,* and analyzed the trade-offs between these two forms of analysis. Exaggeration works by creating extreme cases. For instance, to ascertain how the weight of a block in a simple spring-block system affects the period of oscillation, Weld's system considers what would happen if the mass were infinite. In that case the period is infinite; hence, increasing the mass will increase the period. Weld's theory is bound to be an important tool in developing a variety of ICAE systems.

Planning and Procedure Generation. The view of engineered systems as isolated dynamical systems is inadequate: Real systems usually have controls that are often operated by human beings. They are often subject to all manner of external effects, such as changing loads and temperatures. Qualitative physics must be integrated with models of agency and action to more completely predict the behavior of real systems. By capturing the kinds of things agents can do to a system, the effects of carrying out procedures can be represented and thus provide tools for automatic procedure generation.

My group is currently exploring several ways to integrate physics with planning. One way is to take the physics into the planner. John Hogge (1987) has developed an operator compiler that takes QP domain models and produces rules and operators suitable for a temporal planner. Given a goal such as "cause the water level in this container to increase," his planner can use the knowledge of what it can do, combined with the "operators" representing what the physical world will do, derived from the QP model, to figure out that it should place the container under a faucet and turn on the tap. Currently, his planner is being tested on a variety of examples and domain models to better understand its advantages and limitations.

Another way to think about planning is to move action into the physics. An envisionment consists of all the different states possible given a fixed configuration of objects. Consider augmenting the envisionment in two ways. First, suppose instead of one fixed configuration, the input were a set of configurations representing all the transformations that could be achieved by an agent acting on these

objects. Second, consider adding to the set of state transitions the effects of an agent taking one of the possible actions in that state. We call this description the *action-augmented envisionment* (AE).

AEs could be very useful for procedure generation. An AE explicitly represents every possible execution sequence for all possible procedures. One can find the minimum number of actions that will lead to a disaster, for example, and change a design to maximize this number.

How hard will AEs be to compute, relative to envisionments? That seems to depend on the nature of the operator vocabulary. Some actions will produce new instances of processes (for example, cutting a hole in a vat), and some won't (for example, turning a valve off or on). If the operator vocabulary contains nothing that can create or destroy instances of processes, then the number of states in the AE will be the same as the underlying envisionment. The only difference will be additional transitions. (This condition may seem surprising, but it follows from the fact that in this case either the operators are irrelevant, or they affect the operation of processes by changing their preconditions. All different combinations of preconditions are already represented in the envisionment.) Thus, AEs might be quite reasonable for designing procedures for a process plant. Domains that look more like robot planning appear worse: The number of initial configurations typically skyrockets. Even if AEs turn out to be infeasible to compute for all but the simplest systems, I suspect that the framework they provide will be useful in understanding and characterizing other planning systems. Experiments with a prototype implementation have been encouraging so far (Forbus 1988b).

Developing a full qualitative physics that can capture the breadth of engineering expertise in a wide variety of domains is not a 5-year exercise, and probably not even a 10-year exercise.

Scaling Up. Earlier I argued that domain models that fully capture the tacit knowledge of engineers will probably be orders of magnitude larger than our current models. Given that our current models often strain the most powerful symbolic workstations available, how can qualitative physics possibly be made an important part of ICAE? The answer is not substantially different from that given for any other kind of AI problem: We must organize and use the knowledge more cleverly.

We have had the luxury of simple envisioning for longer than we should have expected. The envisionment can be viewed as a problem space for several classes of problems, and few AI problems are simple enough to allow the problem space to be explicitly generated. We should not give up envisionments as theoretical tools, and even in developing our domain models we should use envisioning as a means of exposing flaws that otherwise might remain hidden. For practical ICAE, we will have to use more sophisticated computational schemes.

Consider the problem of constructing a total envisionment for a nuclear power plant. A typical plant contains thousands of components, many of which have more than one state. A straightforward combinatorial analysis suggests that explicitly storing a total envisionment is infeasible for any memory technology we are likely to have in the foreseeable future.[7] However, as with any other computational problem, there are several techniques that should offer relief from this combinatorial explosion by trading space for time.

The first technique is decomposition. A standard way to stave off a combinatorial explosion is to decompose it into small pieces, solve them, and combine the smaller solutions as needed. The importance of this idea has already been recognized in qualitative physics—the identification of the local evolution problem in Forbus (1984b), for example, and partial-state envisionment in Williams (1986)—but little work has been done to realize it. The notion of p-component in QP theory defines an isolation criterion for domain models; if the boundaries of p-components do not change, then simulation within them can proceed independently. Extending this notion to piece together descriptions of behavior across changes in p-component boundaries potentially offers a compact representation for very complex systems.

The second technique is hierarchy. Another standard way to avoid combinatorial explosions is to use the most abstract description that will suffice to solve a problem. By starting out with

an abstract description and resorting to a detailed description only when necessary, irrelevant problem-solving work can be avoided. It seems likely that as qualitative models evolve they will become hierarchical, and that the choice of what range of models to use in solving an engineering problem will be determined by the required safety factor, just as it is in current engineering practice.

Some progress has been made on using hierarchy in domain models to reduce the complexity of qualitative simulation. Falkenhainer and Forbus (1988) describe an algorithm for solving the problem of selecting the right grain size and aspect of a model in the context of intelligent tutoring systems. However, many open problems remain, including successive refinement of qualitative simulations and the combination of results from multiple perspectives.

Hiearchy can also be exploited at the level of behaviors. For instance, Weld's (1984) work on aggregation provides a dynamic summary of discrete-process or continuous-process models. Kuipers' (1987) time-scale abstraction technique allows the effects of slow components of behaviors to be abstractly represented as functions, under certain assumptions.

The third technique is time-space trade-offs. How much of an envisionment is precomputed rather than constructed on the fly will depend on the kind of problem to be solved and the hardware available. In an ICAE design system, for instance, you might explicitly construct a total envisionment for the most abstract levels of description but switch to dependency-directed search when performing a safety analysis using more detailed models. In an execution monitor, where speed is crucial, you might decompose the system into several state clusters, within which the p-components remain constant. Total envisionments for the p-components could then be computed and used to build lookup tables for measurement interpretation using the ATMI theory.

While the abstract form of the solutions to scaling up seems clear, there are of course many technical problems that must be solved in order to accomplish it. I am convinced that right now we cannot even clearly state what several of these problems are. Consequently, I suspect it will take several Ph.D. theses and a number of master's theses before we will have satisfactory answers.

Conclusions

AI has already been successfully applied to engineering tasks, but these systems are only the beginning. ICAE, as it arrives, has the potential to transform the engineering profession. ICAE research offers a rich source of examples and problems to challenge AI techniques. Qualitative physics, which I have argued is central to ICAE, in particular will benefit from considering such problems. By understanding the complicated reasoning performed by engineers, we should also better understand how our minds work.

Creating the kinds of large-scale knowledge bases we seek will take the efforts of many people. My group, the Qualitative Reasoning Group at the University of Illinois, is focusing on building a knowledge base for engineering thermodynamics that integrates both qualitative and quantitative knowledge. Edward Feigenbaum's group at the Stanford Knowledge Systems Laboratory is focusing on mechanics and electromechanical systems.[8] We hope that others will join us in these efforts.

A warning: Progress will be slow. Human beings are very smart. We know much more than we know how to represent or reason with right now. Developing a full qualitative physics that can capture the breadth of engineering expertise in a wide variety of domains is not a 5-year exercise, and probably not even a 10-year exercise. No single breakthrough will herald the arrival of the kinds of systems I've speculated about, only a lot of dedicated slogging. To make progress we have to commit ourselves over the long haul, but that is true of any worthwhile goal.

There will, of course, be incremental progress, and the successes of current expert systems supplies us with a crucial lesson. Even small advances in computing technology can yield important leverage for real-world applications. This thought is encouraging, since the full capabilities of ICAE as suggested here appear to be far off. Nevertheless, I believe we will find the journey very interesting.

Acknowledgments

This article evolved from a paper of the same title that originally appeared in the Proceedings of the AAAI Workshop on AI in Chemical Engineering, held in March 1987 at Columbia University. My views have evolved as a result of the influence of many people. Among them are Johan de Kleer and Gerald Sussman of the old engineering problem-solving group at MIT; Al Stevens, Bruce Roberts, Jim Hollan, Mike Williams, and Ed Hutchins of the STEAMER crew at Bolt, Beranek, and Newman and the Navy Personnel Research and Development Center; and Pat Hayes, Dedre Gentner, Allan Collins, and John Seely Brown. Dedre Gentner, Paul Nielsen, Brian Falkenhainer, and Jane Malin provided many insightful comments on this article.

The research described here is supported by the National Aeronautics and Space Administration (contract no. NASA NAG 9-137), the Information Sciences Division of the Office of Naval Research (contract no. N00014-85-K-0225), an IBM Faculty Development award, and a National Science Foundation Presidential Young Investigator award.

References

Anderson, J. R., and Skwarecki, E. 1986. The Automated Tutoring of Introductory Programming. *Communications of the ACM* 29(9): 842-849.

Brown, J.; Burton, R.; and de Kleer, J. 1982. Pedagogical, Natural Language, and Knowledge Engineering Techniques in SOPHIE I, II, and III. In *Intelligent Tutoring Systems,* eds. D. Sleeman and J. Brown, 227-282. New York: Academic.

Buchanan, B., and Shortliffe, E., eds. 1984. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, Mass.: Addison-Wesley.

Chandrasekaran, B. 1985. Generic Tasks in Knowledge-Based Reasoning: Characterizing and Designing Expert Systems at the "Right" Level of Abstraction. In Proceedings of the Second Conference on AI Applications, 294-300. New York: IEEE Computer Society.

Chandrasekaran, B. 1983. Towards a Taxonomy of Problem-Solving Types. *AI Magazine* 4(1): 9-17.

Collins, J., and Forbus, K. 1987. Reasoning about Fluids via Molecular Collections. In Proceedings of the Sixth National Conference on Artificial Intelligence, 590-594. Menlo Park, Calif.: American Association for Artificial Intelligence.

Dauge, P.; Raiman, O.; and Deves, P. 1987. Troubleshooting: When Modeling Is the Trouble. In Proceedings of the Sixth National Conference on Artificial Intelligence, 600-605. Menlo Park, Calif.: American Association for Artificial Intelligence.

Davis, E. 1987. Order of Magnitude Reasoning in Qualitative Differential Equations, Technical Report, 312, Computer Science Dept., New York Univ.

Davis, R. 1987. Robustness and Transparency in Intelligent Systems. In *Human Factors in Automated and Robotic Space Systems.*, Technical Report, National Academy of Sciences.

Davis, R. 1984. Diagnostic Reasoning Based on Structure and Function. *Artificial Intelligence* 24: 347-410.

Davis, R. 1982. Expert Systems: Where Are We? And Where Do We Go from Here? *AI Magazine* 3(2): 3-22.

de Kleer, J. 1975. Qualitative and Quantitative Knowledge in Classical Mechanics, Technical Report, 352, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

de Kleer, J., and Brown, J. 1984. A Qualitative Physics Based on Confluences. *Artificial Intelligence* 24: 7-84.

de Kleer, J., and Sussman, G. 1980. Propagation of Constraints Applied to Circuit Synthesis. *Circuit Theory and Applications* 8: 127-144.

de Kleer, J., and Williams, B. 1986. Reasoning about Multiple Faults. In Proceedings of the Fifth National Conference on Artificial Intelligence, 132-139. Menlo Park, Calif.: American Association for Artificial Intelligence.

Dixon, J. R., and Simmons, M. K. 1984. Computers that Design: Expert Systems for Mechanical Engineers. *Computers in Mechanical Engineering,* November.

Doyle, J. 1979. A Truth Maintenance System. *Artificial Intelligence* 12(3): 231-272.

Falkenhainer, B., and Forbus, K. 1988. Setting Up Large-Scale Qualitative Models. In Proceedings of the Seventh National Conference on Artificial Intelligence. Menlo Park, Calif.: American Association for Artificial Intelligence. To appear.

Faltings, B. 1987. Qualitative Kinematics in Mechanisms. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 436-442. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Finegold, A. 1984. The Engineer's Apprentice. In *The AI Business: Commercial Uses of Artificial Intelligence,* eds. P. Winston, and K. Prendergast. Cambridge, Mass.: MIT Press.

Forbus, K. 1988a. Commonsense Physics: A Survey. In *Annual Review of Computer Science* 3. To appear.

Forbus, K. 1988b. Introducing Action into Qualitative Simulation, Technical Report, Dept. of Computer Science, Univ. of Illinois. To appear.

Forbus, K. 1988c. Qualitative Physics: Past, Present, and Future. In *Exploring Artificial Intelligence,* ed. H. Shrobe. Los Altos, Calif.: Morgan Kaufmann. To appear.

Forbus, K. 1987. Interpreting Observations of Physical Systems. *IEEE Transactions on Systems, Man, and Cybernetics* 17(3): 350-359.

Forbus, K. 1986. Interpreting Measurements of Physical Systems. In Proceedings of the Fifth National Conference on Artificial Intelligence, 113-117. Menlo Park, Calif.: American Association for Artificial Intelligence.

Forbus, K. 1984a. An Interactive Laboratory for Teaching Control System Concepts, Technical Report, 5511, Bolt, Beranek, and Newman.

Forbus, K. 1984b. Qualitative Process Theory. *Artificial Intelligence* 24: 85-168.

Forbus, K. 1984c. Qualitative Process Theory, Technical Report, 789, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Forbus, K. 1981a. A Study of Qualitative and Geometric Knowledge in Reasoning about Motion, Technical Report, 615, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Forbus, K. 1981b. Qualitative Reasoning about Physical Processes. In Proceedings of the Seventh International Joint Conference on Artificial Intelligence, 326-330. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Forbus, K. 1980. Spatial and Qualitative Aspects of Reasoning about Motion. In Proceedings of the First National Conference on Artificial Intelligence, 170-173. Menlo Park, Calif.: American Association for Artificial Intelligence.

Forbus, K., and Stevens, A. 1981. Using Qualitative Simulation to Generate Explanations, Technical Report, 4490, Bolt, Beranek, and Newman. Also in Proceedings of the Third Annual Meeting of the Cognitive Science Society, August, 1981, 219-221. Los Altos, Calif.: Morgan Kaufmann.

Forbus, K.; Nielsen, P.; and Faltings, B. 1987. Qualitative Kinematics: A Framework. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 430-435. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Garrett, P.; Lee, C. W.; and LeClair, S. R. 1987. Qualitative Process Automation vs. Quantitative Process Control. In Proceedings of the American Control Conference, Minneapolis, Minn.

Gelsey, A. 1987. Automated Reasoning about Machine Geometry and Kinematics. In Proceedings of the Third IEEE Conference on AI Applications. Washington D.C.: Institute for Electrical and Electronics Engineers.

Gentner, D., and Stevens, A., eds. 1983. *Mental Models.* Hillsdale, N.J.: Lawrence Erlbaum.

Hamilton, T. 1988. HELIX: An Application of Qualitative Physics to Diagnostics in Advanced Helicopters. Paper presented at the First Qualitative Physics Workshop, Urbana, Ill., May 1987. Also to appear in *International Journal of AI in Engineering.*

Harris, H. G. 1984. *Handbook of Watch and Clock Repairs.* New York: Barnes & Noble.

Hayes, P. 1985. Naive Physics 1: Ontology for Liquids. In *Formal Theories of the Commonsense World,* eds. R. Hobbs and R. Moore, 71-107. Norwood, N.J.: Ablex.

Hayes, P. 1979. The Naive Physics Manifesto. In *Expert Systems in the Micro-Electronic Age,* ed. D. Michie. Edinburgh, U.K.: Edinburgh University Press.

Hogge, J. 1987. Compiling Plan Operators from Domains Expressed in Qualitative Process Theory. In Proceedings of the Sixth National Conference on Artificial Intelligence, 229-233. Menlo Park, Calif.: American Association for Artificial Intelligence.

Hutchins, E. L.; Hollan, J. D.; and Norman, D. A. 1985. Direct-Manipulation Interfaces. *Human-Computer Interaction* 1: 311-338.

Hollan, J.; Hutchins, E.; and Weitzman, L. 1984. STEAMER: An Interactive Inspectable Simulation-Based Training System. *AI Magazine* 5(2): 15-27.

Kuipers, B. 1987. Abstraction by Time-Scale in Qualitative Simulation. In Proceedings of the Sixth National Conference on Artificial Intelligence, 621-625. Menlo Park, Calif.: American Association for Artificial Intelligence.

Kuipers, B. 1986. Qualitative Simulation. *Artificial Intelligence* 29(3): 289-338.

Kuipers, B. 1984. Common Sense Causality: Deriving Behavior from Structure. *Artificial Intelligence* 24: 169-204.

Kuipers, B., and Berleant, D. 1988. Using Incomplete Quantitative Knowledge in Qualitative Reasoning. In Proceedings of the Seventh National Conference on Artificial Intelligence. Menlo Park, Calif.: American Association for Artificial Intelligence To appear.

Lee, W. W.; Chiu, C.; and Kuipers, B. J. 1987. Developments Towards Constraining Qualitative Simulation, Technical Report, AI TR87-44, Artificial Intelligence Laboratory, Univ. of Texas at Austin.

Lenat, D.; Prakash, M.; and Sheperd, M.

1986. CYC: Using Commonsense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks. *AI Magazine* 6(4): 65-84.

Lozano-Perez, T. 1983. Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers* 32: 108-120.

McAllester, D. 1980. An Outlook on Truth Maintenance, Technical Report, Memo 551, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Malin, J., and Harris, R. 1987. CONFIG: Qualitative Simulation Tool for Analyzing Behavior of Engineered Devices. Paper presented at the First Annual Workshop on Space Operations Automation and Robotics, Houston, Tex.

Malin, J., and Lance, N. 1985a. An Expert System for Fault Management and Automatic Shutdown Avoidance in a Regenerative Life Support System. In Proceedings of the First Annual Workshop on Robotics and Expert Systems. Instrument Society of America.

Malin, J., and Lance, N. 1985b. Feasibility of Expert Systems to Enhance Space Station Subsystem Controllers. In Proceedings of the Conference on Space Station Automation, vol. 850. Bellingham, Wash.: International Society for Optical Engineering.

Mavrovouniotis, M., and Stephanopolous, G. 1987. Reasoning with Orders of Magnitude and Approximate Relations. In Proceedings of the Sixth National Conference on Artificial Intelligence, 626-630. Menlo Park, Calif.: American Association for Artificial Intelligence.

Mohammed, J., and Simmons, R. 1986. Qualitative Simulation of Semiconductor Fabrication. In Proceedings of the Fifth National Conference on Artificial Intelligence, 794-799. Menlo Park, Calif.: American Association for Artificial Intelligence.

Nielsen, P. 1988. The Qualitative Statics of Rigid Bodies. In Proceedings of the Seventh National Conference on Artificial Intelligence. Menlo Park, Calif.: American Association for Artificial Intelligence. To appear. Also, Technical Report, UIUCDCS-R-87-1354, Dept. of Computer Science, Univ. of Illinois at Urbana-Champaign.

Raiman, O. 1986. Order of Magnitude Reasoning. In Proceedings of the Fifth National Conference on Artificial Intelligence, 100-104. Menlo Park, Calif.: American Association for Artificial Intelligence.

Sacks, E. 1985. Qualitative Mathematical Reasoning. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 137-139. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Sembugamoorthy, V., and Chandrasekaran, B. 1984. Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems, Technical Report, Artificial Intelligence Group, Ohio State Univ.

Simmons, R. 1986. Commonsense Arithmetic Reasoning. In Proceedings of the Fifth National Conference on Artificial Intelligence, 118-124. Menlo Park, Calif.: American Association for Artificial Intelligence.

Sleeman, D., and Brown, J., eds. 1982. *Intelligent Tutoring Systems.* New York: Academic.

Sriram, D., and Adey, R., eds. 1986. *Applications of Artificial Intelligence in Engineering,*vols. 1 and 2. Berlin: Springer Verlag.

Stallman, R., and Sussman, G. 1977. Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis. *Artificial Intelligence* 9: 135-196.

Steele, G., and Sussman, G. 1980. Constraints—A Language for Expressing Almost-Hierarchial Descriptions. *Artificial Intelligence* 14: 1-39.

Stefik, M., and Conway, L. 1982. Towards the Principled Engineering of Knowledge. *AI Magazine* 3(3): 4-16.

Stevens, A.; Roberts, B.; Stead, L.; Forbus, K.; Steinberg, C.; and Smith, B. 1981. STEAMER: Advanced Computer-Aided Instruction in Propulsion Engineering, Technical Report, 4702,Bolt, Beranek, and Newman.

Sussman, G. 1977. SLICES: At the Boundary between Analysis and Synthesis, Technical Report, Memo 433, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Ullman, S. 1985. Visual Routines. In *Visual Cognition,* ed. S. Pinker, 97–59. Cambridge, Mass.: MIT Press.

Weld, D. 1988. Theories of Comparative Analysis. Ph.D. diss., Massachusetts Institute of Technology.

Weld, D. 1987. Comparative Analysis. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 959-965. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Weld, D. 1984. Switching between Discrete and Continuous Process Models to Predict Genetic Activity, Technical Report, 793, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Wenger, E. 1987. *Artificial Intelligence and Tutoring Systems.* Los Altos, Calif.: Morgan Kaufmann.

Williams, B. 1986. Doing Time: Putting Qualitative Reasoning on Firmer Ground. In Proceedings of the Fifth National Conference on Artificial Intelligence, 105-112. Menlo Park, Calif.: American Association for Artificial Intelligence.

Williams, B. 1984. Qualitative Analysis of MOS Circuits. *Artificial Intelligence* 24: 281-346.

Woolf, B.; Blegen, D.; Jansen, J.: and Verloop, A. 1986. Teaching a Complex Industrial Process. In Proceedings of the Fifth National Conference on Artificial Intelligence, 722-728. Menlo Park, Calif.: American Association for Artificial Intelligence.

Yip, K. 1987. Extracting Qualitative Dynamics from Numerical Experiments. In Proceedings of Sixth National Conference on Artificial Intelligence, 665-670. Menlo Park, Calif.: American Association for Artificial Intelligence.

Notes

1. Jane Malin, NASA Johnson Space Center, personal communication, November 1987.

2. This name was suggested by Mike Williams of IntelliCorp.

3. Anderson's group has in fact proposed building a system that, while less ambitious, is similar in spirit to the tutor compiler proposed here (Anderson and Skwarecki 1986). The goal is to run their existing tutors on cheaper hardware, not to automatically generate completely new tutoring programs.

4. I've surveyed the state of the art in qualitative physics in two recent papers incorporating different perspectives; Forbus (1988a) is more tutorial, providing some historical perspective and open problems, while Forbus (1988b) reviews progress in related areas such as temporal reasoning and causal reasoning as well as qualitative physics.

5. The mapping from QP descriptions to axioms is one to many; hence, these numbers are approximate.

6. For example, both civil and nuclear engineering tend to base these decisions on safety factors.

7. Dan Dvorak, University of Texas at Austin, personal communication, November 1986.

8. Ed Feigenbaum, Stanford University, personal communication, April, 1988.