Intelligent Computer-Aided Engineering

Kenneth D. Forbus
Qualitative Reasoning Group
Department of Computer Science
University of Illinois
1304 W. Springfield Avenue
Urbana, Illinois, 61801
Internet: forbus@p. cs. uiuc. edu
Voice: (217) 333-0193

Abstract

The goal of *Intelligent Computer-Aided Engineering* (ICAE) is to construct computer programs which capture a significant fraction of an engineer's knowledge. Today ICAE systems are a goal, not a reality. This paper 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 directly evolve from current applications of expert system technology to engineering problems. We focus on qualitative physics as a critical area where progress is needed, both in terms of representations and styles of reasoning.

To appear in AI Magazine, Fall, 1988.
Draft of May 13, 1988

1 Introduction

Applications of Artificial Intelligence can provide challenges that drive the field forward. One example is *intelligent computer-aided instruction* (ICAI) [64]. ICAI research has spawned many important studies of cognition and learning, leading to intellectual advances as well as providing the potential for revolutionizing education.

I believe the application of Artificial Intelligence to engineering problems will play a similar role. Engineering is a particularly good domain for Artificial Intelligence for several reasons:

- *The results are testable.* Engineering domains typically have a well-tested theoretical basis, with 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.

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

- *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 common sense 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 can use all the help they can get. The constraints are getting 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 Artificial Intelligence was Gerald Sussman, who made these same arguments over a decade ago. The work of his (now defunct) Engineering Problem Solving group at MIT illustrated the point well. Several important AI ideas grew directly out of that group's attempts to automate aspects of engineering reasoning, including truth-maintenance systems [19,57], dependency-directed backtracking [66], constraint languages [14,70,67], and qualitative physics

[13,23]. Some of this work has already directly led to useful applications [22]. Today many groups in this country are exploring various applications of AI to engineering, including academia (MIT, University of Massachusetts at Amherst, Ohio State University, and University of Illinois), government (NASA), and industry (Xerox PARC, 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 won't even try.

The goal of this paper is to say what *intelligent computer-aided engineering* can and should be. First, I describe what I mean by intelligent computer-aided engineering (henceforth ICAE). Several highly speculative scenarios are introduced for concreteness. Next I'll explain why current expert systems technology is *not* a very good example of ICAE technology, and point out some of its limitations. Since I believe that qualitative physics will be central to ICAE, I next outline the issues and open problems in that area which must be addressed for such applications. My hope in writing this paper is to encourage more research in this area.

2 What ICAE systems could be like

The goal of Intelligent Computer-Aided Engineering is to construct computer programs that capture and use a significant fraction of the knowledge of engineers, both formal and tacit. 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 importantly, if we can capture the ways the best human engineers reason, we can make engineering tools that spread this expertise among a broader community (as suggested for VLSI in [68]).

There have already been signficant AI applications in engineering (see for example [65]), 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 the capabilities I mean. Please note carefully: the programs described in the following three sections exist only in the imagination. But I believe they are possible, if we are willing to put in the necessary effort.

2.1 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 wish 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 make choices for the 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 just why the computer suggested certain choices, or even ask it to remind us why we made other choices.

As soon enough information is specified, the system performs a qualitative simulation to find out how the designed 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 will need to choose the shapes of the components.) The envisionment performed by the computer reveals that the desired behavior of the watch is indeed is 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 temperature increases the diameter of the balance wheel rises, slowing the movement, and as temperature falls, the diameter will decrease, 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 temperature rises, thus maintaining the distribution of weight [40].

Scenario Two: Alternately, suppose we are designing a thermal control system for a Space Station [53,54,55]. We might choose to use ammonia as our working fluid, and use the same heat exchangers for heating and cooling to save total system weight. An important criteria for 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 finalized, the computer generates detailed manuals of procedures for initializing, 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.

2.2 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 importantly, the human operator should be able to tell the monitor what he/she thinks is 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 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 re-configuration 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 re-structuring. 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 com-

[1] Jane Malin, Personal Communication, November, 1987.

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

piles the appropriate sensory and control programs for its slave computers which handle the brunt of real-time processing.

2.3 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 since the trainees often lack higher education [2,69,44,78]. Intelligent Computer-Aided Instruction promises to provide high-quality tutoring at much lower costs than using 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 [72]. Part of the problem has been that, until recently, the available computing technology 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 knowledge elicitation 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: Consider an expert in, say, automatic combustion controls in Navy propulsion plants. The expert may have mainly served on ships that use a particular brand of pneumatic component (e.g., Bailey). When asked to explain an automatic combustion control system made from components from another manufacturer (e.g., Hagen), he 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 he is familiar with. It seems there are "compiled" explanations which the expert can bring to bear on familiar systems, or systems built out of familiar components. Given an unfamiliar system, the expert requires time to extend his understanding to include the new kinds of devices.

Scenario Four: Suppose you wish 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, their likely *mental models* [38], and what class of questions the tutor must be capable of answering. Notice that we are specifying the tutor much more abstractly than current CAI "authoring systems" allow – we are not specifying specific questions and specific answers. We might wish 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 – what kind 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 or not it 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.

2.4 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 quickly solve routine design problems. 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[18].

Functional descriptions: An ICAE system will need a vocabulary of functional descriptions, in order to analyze existing artifacts, hierarchically design new artifacts, and to communicate better with engineers during the initial

[3] Anderson's group has in fact proposed building a system which, while less ambitious, is similar in spirit to the Tutor Compiler proposed above [1]. Their goal is to run their existing tutors on cheaper hardware, not to automatically generate completely new tutoring programs.

phases of a design. Examples of such concepts include *filter*, *regulator*, *oscillator*, *sensor*, and *actuator* [27].

Qualitative simulation: Existing 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 the 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 the posting of constraints on other choices in the design, in order to prevent such possibilities from occurring.

Procedure generation: Designing a new artifact often results in a second, neglected design problem: developing the procedures required to operate and maintain the artifact. We can imagine a *procedure compiler* which 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 in order 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, possible failure modes of the system can be predicted in advance. These identified 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 [62].

Communication skills: People will still be in the loop, and ICAE systems must communicate effectively with them. This requires advances in natural language understanding and generation. Also, it requires advances in *direct-manipulation interfaces*[45]. And finally, it requires an understanding of typical mental models, so the program can couch explanations in understandable terms.

Computer Individuals: Nilsson has 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.

3 The road to ICAE

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

3.1 Why not rule-based expert systems?

Rule-based expert systems have comprised the bulk of AI applications to date. This technology moved from the laboratory in the mid to late 70's, and through the microelectronics revolution has seen widespread industrial application. However, this technology also has well-known limitations, as Davis [10] and others have pointed out. These problems include:

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 puts it, "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.

Much of today's AI research is aimed at overcoming these limitations. Despite such drawbacks, this technology has proven quite useful. In part this success has been because the 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 subsitute 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 can narrow that gap even more?

The deepest problem with current fielded expert systems technology is that they intermingle 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 re-written to be used for interpretation ([3], p 462). When re-examining 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 [3]. This is not to say that such modularity cannot in principle be imposed with today's expert system shells. Many expert system tools (e.g., 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 breech – it is just too convenient to build in only the distinctions you need for the task. Indeed, this is exactly the right strategy to build 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 systems 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.

3.2 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 [17], measurement interpretation [30], and comparative analysis [75] are successful examples of domain-independent styles of reasoning. (By successful, I do not mean these problems have been solved once and for all, as Section 4.2 makes clear.) And the domain models developed in qualitative physics research [16,28,46,76] tend to be independent of the style of reasoning used.

It is worth dwelling on the notion of a *domain model* from qualitative physics a bit, since 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 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 which 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 [25,28] organizes domain models around *processes*, which can be automatically instantiated to form scenario models. Device-centered ontologies [16,76] provide catalogs of devices, which can be composed to build scenario models. (Kuiper's `QSIM` [47] 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 (a) the domain model can be used to build scenario models that span a well-defined variety of systems or phenomena and (b) 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 which 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* [2,69,44,11] overlaps signficantly with qualitative physics, and uses more or less the same notion of domain model. Chandrasekaren's *generic tasks* are one codification of styles of reasoning [4,5]. The creation of expert systems shells specialized for particular kinds of tasks, such as `EMYCIN`, is another. Lenat's CYC project [51] can be viewed as an attempt to generate a high-level domain model for all of 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 wish to emphasize is that the construction of large knowledge bases and codification of general styles of reasoning requires explicit research; it will not arise as a side-effect of "adding more rules" to the typical systems being applied today.

4 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, heterogeneous processing systems for efficient execution of mixed symbolic and numeric computations, and others. However, this paper focuses on one key area: qualitative physics.

Qualitative physics is crucial because it helps capture the commonsense understanding of the world which 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 human-like performance. Qualitative physics complements traditional techniques in several ways:

- It can provide answers even with very little data.
- It provides organizational structure for more detailed forms of analysis.
- It provides a more natural language for communicating with human engineers.

These points have been amply made in the literature (such as [13,2,23,26]), so I will not elaborate them here[4].

The qualitative physics enterprise began just over ten years ago, but only in the last four or five years has it received intensive effort. While there are some indications of successful applications at the current state of the art (including [7,39,36]), I believe the real potential lies ahead. The next sections focus on the research that will be required to bring the field up to the quality and scope of human engineering.

4.1 Modeling Issues

We have only reconstructed 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 dynamical systems. We can even model simple spatial systems, although much less progress has been made on that front. I would bet that most of qualitative physics lies ahead of us.

4.1.1 Extending the range of phenomena

There are many phenomena which can probably be formalized using existing techniques, but which simply haven't been yet. For example, small-signal and DC analyses of simple electronic circuits have been successfully solved, but to my knowledge little has been done on AC or RF 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 remain to be formalized. Little has been done on materials, conservation laws, and energy (but see [28,8]).

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, 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,

[4] I've surveyed the state of the art in qualitative physics in two recent papers incorporating different perspectives; [32] is more tutorial, providing some historical perspective and open problems, while [33] 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 many to one, hence these numbers are approximate.

it seems reasonable to think that perhaps an increase of between a factor of 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 ten or twenty 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 below), all the minutae of specific constants, random facts about the world such as the reliability of suppliers, etc. 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[6], so the existence of qualitative model libraries will merely extend the spectrum of available representations. 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.

4.1.2 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 (e. g. [13]) or numerical simulation (e. g. [23,24]). Some excellent starts have been made. Simmons [63] has developed utilities for integrating inequalities, intervals, relational arithmetic, and constant elimination. Berelant and Kuipers [49] use quantitative bounds on functions to constrain qualitative simulation. Several formalizations of order of magnitude reasoning [60,56,9] have been developed. The

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

tacit knowledge mathematicians bring to bear is also being studied [61,79].

4.1.3 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 which is impossible to establish without considering mass properties of the fluid in the various containers. By contrast, the *contained-liquid* ontology introduced by Hayes [42] 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 [6] for details).

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.

4.1.4 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 (i.e., 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 re-done 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 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 (i.e., those of de Kleer & Brown, Williams, etc.). Second, the *preconditions* field of processes and time-varying relationships provides an interface for information which normally lies outside the realm of QP theory, such as geometric relationships.

However, almost all qualitative models developed to date are "flat" — 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 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, including:

1. *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 it is placed in thermal contact with some other working fluid.

2. *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 many of these descriptions can be succinctly represented by QP's *individual views.*

3. *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 (e.g., equal

temperatures in both legs of a heat exchanger). Others, such as leaks, may require making extra explicit assumptions.

We have begun exploring these issues, and have developed a set of conventions for modeling assumptions that allows the construction of large, multi-perspective, multi-grain qualitative models [20]. Our initial testbed has been a high-level model of a Navy propulsion plant, and next we plan to test these ideas on modeling thermal control systems for NASA. However, much research remains.

4.1.5 Spatial reasoning

Motion has been analyzed for special cases (linear one-dimensional [28], sliding in one dimension [13], free motion in two dimensions [23,24], and some mechanisms [37,59,21]. 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*, for details see [34]), and thus all attempts to find one are doomed to failure. The *MD/PV model* we propose instead starts with a quantitative, diagrammatic representation (the *Metric Diagram*) and computes an appropriate qualitative representation from it (the *Place Vocabulary*). We have demonstrated this model for free motion of point masses in two dimensions [23,24], and reasoning about mechanisms [34,21,59].

The MD/PV model has allowed us to achieve a milestone in qualitative physics: In February of 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 itself.) 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, i.e., 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 (e.g., Lozano-Perez's *configuration space formalism* [52] and Ullman's *visual routines* [71]), and using these to construct qualitative representations appears quite promising. For example, [21] 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?

4.2 Reasoning Issues

ICAE 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; much can be done even with today's computers in the meanwhile. (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.

4.2.1 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. Doing this 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 (i.e., 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 [30] for details).

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 for developing new qualitative models. The reason is that we want our models to predict all and only those behaviors which 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 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 below.

Any path through an envisionment corresponds to a predicted possible behavior of the physical system, or a potential *history* of that system (as in [41,42]). However, Kuipers [47] 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 produces 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 for detailed analysis of specific behaviors.

Unfortunately, Kuipers has also shown that direct history generation, due to its locality, can lead to spurious predictions about behavior. Worse yet, it appears that such spurious predictions can even arise when generating histories indirectly from envisionments. More research is needed to determine and appropriately encode *path constraints*, to constrain successive choices of landmark values and so avoid predicting physically impossible behaviors. Work is proceeding on identifying such constraints (for example, [50]), but much remains to be done. We conjecture 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 used to analyze the behavior in more detail.

4.2.2 Measurement interpretation

The ATMI measurement interpretation theory [30,31] 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 one 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.

4.2.3 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 an in-depth analysis of the dependencies of the situation, and finding the effects of a small perturbation of them. This kind of *comparative analysis* [74] is receiving increasing attention in qualitative physics. In [58], 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 recent PhD 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 [28] (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.

4.2.4 Planning and Procedure generation

The view of engineered systems as isolated dynamical systems is inadequate: Real systems usually have controls, which 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, we can represent the effects of carrying out procedures and thus provide tools for automatic procedure generation.

We are currently exploring several ways to integrate physics with planning. One way is to take the physics into the planner. John Hogge has developed an *operator compiler* that takes QP domain models and produces rules and operators suitable for a temporal planner [43]. Given a goal like "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. We are currently testing his planner 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* (or AE).

AE's could be very useful for procedure generation. An AE explicitly represents every possible execution sequence for all possible procedures. Thus 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 AE's 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 (e.g., cutting a hole in a vat), and some won't (e.g., 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 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 AE's might be quite reasonable for designing procedures for a process plant. Domains which look more like robot planning appear worse: the number of initial configurations will typically skyrocket. Even if AE's turn out to be infeasible to compute for all but the simplest systems, we suspect the framework they provide will be useful for understanding and characterizing other planning systems. Experiments with a prototype implementation have been encouraging so far [35].

4.2.5 Scaling up

Earlier we argued that domain models which 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 we possibly make qualitative physics an important part of ICAE? The answer is not substantially different than 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. But 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[7] suggests that explicitly storing a total envisionment is infeasible for any memory technology we are likely to have in the foreseeable future. However, as with any other computational problem, there are several techniques which should offer relief from this combinatorial explosion by trading space for

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

time:

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 (e.g., the identification of the *local evolution problem* in [28] and *partial-state envisionment* in [77]), but little work has been done to realize it. The notion of *p-component* in QP theory defines an isolation criteria 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.

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. In [20] we 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 combining results from multiple perspectives.

Hiearchy can also be exploited at the level of behaviors. For instance, Weld's work on *aggregation* [73] provides dynamic summarization of discrete process or continuous process models. Kuipers' *time-scale abstraction* technique [48] allows the effects of "slow" components of behaviors to be abstractly represented as functions, under certain assumptions.

Time/Space tradeoffs: How much of an envisionment is precomputed versus constructed "on the fly" will depend on the kind of problem we wish to solve and the available hardware. In an ICAE design system, for instance, we 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, we 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 which 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.

5 Conclusions

AI has already been successfully applied to engineering tasks. This paper argues that these systems are only the beginning. Intelligent Computer-Aided Engineering, as it arrives, has the potential for transforming the engineering profession. ICAE research offers a rich source of examples and problems to challenge AI techniques. Qualitative physics, which we 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. In my group, we are focusing on building a knowledge base for engineering thermodynamics which integrates both qualitative and quantitative knowledge. 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 and know much more than we know how to represent or reason with now or at any time in the near future. Developing a full qualitative physics that can capture the breadth of engineering expertise in a wide variety of domains is not a five-year exercise, and probably not even a ten-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 is encouraging, since the full capabilities of ICAE as suggested above appear to be far off. Nevertheless, I believe we will find the journey very interesting.

[8]Ed Feigenbaum, personal communication, April, 1988

6 Acknowledgements

This paper evolved from a paper of the same title originally appearing in the Proceedings of the AAAI workshop on AI in Chemical Engineering, held in March, 1987 at Columbia University.

My views have evolved due to the influence of many people. Among these 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 BBN and NPRDC; 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 paper.

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 an NSF Presidential Young Investigator award.

References

[1] Anderson, J.R., and Skwarecki, E. "The automated tutoring of introductory programming" *CACM*, 29(9):842-849, 1986

[2] Brown, J., Burton, R., and de Kleer, J. "Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III", in *Intelligent Tutoring Systems*, Sleeman, D. and Brown, J. (Eds.), Academic Press, 1982.

[3] Buchanan, B. and Shortliffe, E. (Eds.) *Rule-Based Expert Systems: The* `MYCIN` *experiments of the Stanford Heuristic Programming Project* Addison-Wesley, Reading, Mass. 1984.

[4] Chandrasekaren, B. "Towards a taxonomy of problem solving types" *AI Magazine*, 4(1): 9-17.

[5] Chandrasekaren, B. "Generic tasks in Knowledge Based Reasoning: Characterizing and designing expert systems at the 'right' level of abstraction" Proceedings of the 2nd Conference on AI Applications, Miami Beach, FL. New York: IEEE Computer Society, 294-300, 1985

[6] Collins, J. and Forbus, K. "Reasoning about fluids via molecular collections", Proceedings of AAAI-87, July, 1987.

[7] Dauge, P., Raiman, O., and Deves, P. "Troubleshooting: When modeling is the trouble", Proceedings of AAAI-87, July, 1987.

[8] Davis, E. "A logical framework for solid object physics" New York University Computer Science Department Technical Report no. 245, October, 1986. To appear in *International Journal of AI in Engineering*, 1988

[9] Davis, E. "Order of magnitude reasoning in qualitative differential equations", New York University Computer Science Department Technical Report no. 312, August, 1987.

[10] Davis, R. "Expert Systems: Where are we? And where do we go from here?" *AI Magazine*, Spring, 1982, pp 3-22.

[11] Davis, R. "Diagnostic reasoning based on structure and function" *Artificial Intelligence*, **24**, 1984

[12] Davis, R. "Robustness and transparency in intelligent systems"

[13] de Kleer, J. "Qualitative and quantitative knowledge in classical mechanics", TR-352, MIT AI Lab, Cambridge, Mass, 1975

[14] de Kleer, J. and Sussman, G., "Propagation of constraints applied to circuit synthesis", *Circuit Theory and Applications* **8** (1980) 127-144.

[15] de Kleer, J. "Causal and teleological reasoning in circuit recognition" MIT AI Lab Technical Report No. 529, September, 1979

[16] de Kleer, J. and Brown, J. "A qualitative physics based on confluences", *Artificial Intelligence*, **24**, 1984

[17] de Kleer, J., Williams, B. "Reasoning about Multiple Faults", Proceedings of AAAI-86, August, 1986.

[18] Dixon, J.R. and Simmons, M.K. "Computers that design: Expert systems for mechanical engineers", *Computers in Mechanical Engineering*, November, 1984

[19] Doyle, J., "A truth maintenance system", *Artificial Intelligence*, **12(3)**:231-272, 1979.

[20] Falkenhainer, B. and Forbus, K. "Setting up large-scale qualitative models" Proceedings of AAAI-88, August, 1988.

[21] Faltings, B. "Qualitative kinematics in mechanisms", To appear in the Proceedings of IJCAI-87, August, 1987.

[22] Finegold, A. "The Engineer's apprentice", in *The AI Business: Commercial uses of Artificial Intelligence*, Winston, P. and Prendergast, K. (Eds.), MIT Press, 1984.

[23] Forbus, K. "Spatial and qualitative aspects of reasoning about motion", AAAI-80, Palo Alto, California, August, 1980.

[24] Forbus, K. "A study of qualitative and geometric knowledge in reasoning about motion", MIT AI Lab Technical Report No. 615, February, 1981

[25] Forbus, K. "Qualitative reasoning about physical processes" IJCAI-7, Vancouver, B.C., August, 1981

[26] Forbus, K. and Stevens, A. "Using qualitative simulation to generate explanations", BBN Technical Report No. 4490, March, 1981. Also, Proceedings of the third annual meeting of the Cognitive Science Society, August, 1981.

[27] Forbus, K. "An interactive laboratory for teaching control system concepts", BBN Technical Report No. 5511, January, 1984.

[28] Forbus, K. "Qualitative process theory" *Artificial Intelligence*, **24**, 1984

[29] Forbus, K. "Qualitative Process theory" MIT AI Lab Technical report No. 789, July, 1984.

[30] Forbus, K. "Interpreting measurements of physical systems", Proceedings of AAAI-86, August, 1986.

[31] Forbus, K. "Interpreting observations of physical systems", *IEEE transactions on systems, man, and cybernetics*, Vol. SMC-17, No. 3, May/June 1987.

[32] Forbus, K. "Qualitative physics: Past, present, and future" to appear in *Exploring Artificial Intelligence*, H. Shrobe (ed), Morgan-Kaufmann, 1988.

[33] Forbus, K. "Commonsense physics: A survey" to appear in *Annual Review of Computer Science*, Vol. 3, 1988.

[34] Forbus, K., Nielsen, P., and Faltings, B. "Qualitative Kinematics: A Framework", Proceedings of IJCAI-87, August, 1987.

[35] Forbus, K. "Introducing action into qualitative simulation" University of Illinois Department of Computer Science Technical Report, in preparation, 1988.

[36] Garrett, P., Lee, C.W., and LeClair, S. "Qualitative process automation vs. quantitative process control"

[37] Gelsey, A. "Automated reasoning about machine geometry and kinematics" To appear in the third IEEE conference on AI applications, Orlando, Florida, February, 1987

[38] Gentner, D., Stevens, A. (Eds.), *Mental Models*, LEA Associates, Inc., 1983.

[39] Hamilton, T. "HELIX: An application of Qualitative Physics to diagnostics in advanced helicopters", Presentation at the First Qualitative Physics Workshop, Urbana, Illinois, May, 1987. To appear in *International Journal of AI in Engineering*, 1988.

[40] Harris, H.G. *Handbook of Watch & Clock Repairs*, revised edition, Barnes & Nobel books, New York, 1984.

[41] Hayes, P. "The naive physics manifesto" in *Expert systems in the microelectronic age*, D. Michie (Ed.), Edinburgh University Press, 1979

[42] Hayes, P. "Naive Physics 1: Ontology for liquids" in Hobbs, R., Moore, R. (Eds.), *Formal Theories of the Commonsense World*, Ablex Publishing Corporation, Norwood, New Jersey, 1985

[43] Hogge, J. "Compiling plan operators from domains expressed in Qualitative Process theory", Proceedings of AAAI-87, July, 1987.

[44] Hollan, J., Hutchins, E., and Weitzman, L., "STEAMER: An interactive inspectable simulation-based training system", AI Magazine, Summer, 1984.

[45] Hollan, J., Hutchins, E. "Direct-Manipulation Interfaces", Technical Report, Institute for Cognitive Science, UCSD.

[46] Kuipers, B. "Common sense Causality: Deriving behavior from Structure" *Artificial Intelligence*, **24**, 1984

[47] Kuipers, B. "Qualitative Simulation", *Artificial Intelligence*, **29**, September, 1986.

[48] Kuipers, B. "Abstraction by time-scale in qualitative simulation", Proceedings of AAAI-87, July, 1987.

[49] Kuipers, B. and Berleant, D. "Using incomplete quantitative knowledge in qualitative reasoning" Proceedings of AAAI-88, August, 1988.

[50] Lee, W.W., Chiu, C., and Kuipers, B.J. "Developments towards constraining qualitative simulation", University of Texas at Austin Artificial Intelligence Laboratory Technical Report No. AI TR87-44, January, 1987.

[51] Lenat, D. Prakash, M. and Sheperd, M. "CYC: Using commonsense knowledge to overcome brittleness and knowledge acquisition bottlenecks", *AI Magazine* 6(4) 65-84, Winter 1986

[52] Lozano-Perez, T. "Spatial planning: A configuration space approach", *IEEE Transactions on Computers* **C-32**, February, 1983

[53] Malin, J. and Lance, N. "Feasibility of expert systems to enhance Space Station subsystem controllers" Proceedings of SPIE Conference on Space Station Automation, Vol. 850, 1985

[54] Malin, J. and Lance, N. "An expert system for fault management and automatic shutdown avoidance in a regenerative life support system" Proc. First Annual Workshop on Robotics and Expert Systems, Instrument Society of America, June, 1985.

[55] Malin, J. and Harris, R. "CONFIG: Qualitative simulation tool for analyzing behavior of engineered devices", First annual workshop on Space Operations Automation and Robotics, Houston, Texas, August, 1987.

[56] Mavrovouniotis, M. and Stephanopolous, G., "Reasoning with orders of magnitude and approximate relations", Proceedings of AAAI-87, July, 1987.

[57] McAllester, D. "An outlook on truth maintenance" MIT AI Lab Memo No. 551, August, 1980.

[58] Mohammed, J., and Simmons, R. "Qualitative Simulation of Semiconductor Fabrication", proceedings of AAAI-86, August, 1986.

[59] Nielsen, P. "The qualitative statics of rigid bodies", University of Illinois at Urbana-Champaign, Department of Computer Science Technical Report No. UIUCDCS-R-87-1354, July, 1987. Also in Proceedings of AAAI-88.

[60] Raiman, O. "Order of magnitude reasoning", Proceedings of AAAI-86, August, 1986.

[61] Sacks, E. "Qualitative Mathematical Reasoning", Proceedings of IJCAI-85, August, 1985.

[62] Sembugamoorthy, V., Chandrasekaran, B. "Functional representation of devices and compilation of diagnostic problem-solving systems" Technical paper, AI Group, Ohio State University, 1984.

[63] Simmons, R. "Commonsense arithmetic reasoning" AAAI-86, Philadelphia, Pennsylvania, August, 1986

[64] Sleeman, D. and Brown, J., (Eds.),*Intelligent Tutoring Systems*, Academic Press, 1982.

[65] Sriram, D. and Adey, R. (Editors), *Applications of Artificial Intelligence in Engineering Vols I & II*, Springer Verlag, UK, April 1986.

[66] Stallman, R., and Sussman, G. "Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis", *Artificial Intelligence*, **(9)**, October, 1977, pp. 135-196

[67] Steele, G., and Sussman, G. "Constraints – a language for expressing almost-hierarchial descriptions" *Artificial Intelligence* **14** (1980) 1-39.

[68] Stefik, M., Conway, L. "Towards the principled engineering of knowledge", *AI Magazine*, 3(3), 4-16, 1982.

[69] Stevens, A., Roberts, B., Stead, L. Forbus, K., Steinberg, C., Smith, B. "STEAMER: Advanced computer-aided instruction in propulsion engineering", BBN Technical report, July, 1981

[70] Sussman, G. "SLICES: At the boundary between analysis and synthesis", Memo 433, MIT AI Lab, Cambridge, Mass, 1977.

[71] Ullman, S., "Visual Routines", in Pinker, S. (Ed.) *Visual Cognition*, MIT Press, Cambridge, MA, 1985

[72] Wenger, E. *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann, Los Altos, CA, 1987.

[73] Weld, D. "Switching between discrete and continuous process models to predict genetic activity", MIT Artificial Intelligence Lab TR-793, October, 1984

[74] Weld, D. "Comparative Analysis", Proceedings of IJCAI-87, August, 1987.

[75] Weld, D. "Theories of Comparative Analysis" unpublished PhD dissertation, MIT, May 1988.

[76] Williams, B. "Qualitative analysis of MOS circuits", *Artificial Intelligence*, **24**, 1984

[77] Williams, B. "Doing time: Putting qualitative reasoning on firmer ground", Proceedings of AAAI-86, August, 1986

[78] Woolf, B., Blegen, D., Jansen, J., and Verloop, A. "Teaching a complex industrial process", Proceedings of AAAI-86, August, 1986

[79] Yip, K. "Extracting qualitative dynamics from numerical experiments" Proceedings of AAAI-87, July, 1987.

Contents