Int. J. Appl. Engng Ed. Vol. 7, No. 6, pp. 452–455, 1991
Printed in Great Britain.

0742–0269/91 $3.00+0.00

Computer-based Learning, Expert Systems and Software Engineering: Advanced Tools for Engineering Education Now and in 2001

K. L. MODESITT*
Western Kentucky University, Bowling Green, KY 42101, *U.S.A.*

Who will design and build space station Freedom of today and the lunar and Martian crafts and bases of tomorrow? Several advanced tools can address the shortage of qualified future engineers. Expert computer systems permit experts to transmit their expertise to successors. Computer-based learning tools help people learn difficult concepts by interactive dialogues. Software engineering methods enable large programs to be built with a higher degree of reliability and safety.

WHAT IS THE PROBLEM?

THE world has never lacked for problems or 'opportunities'. Moreover, it has also been recognized by at least one authority that problems are not usually solved by the people who create them:

> The world that we have made as a result of the level of thinking we have done this far creates problems we cannot solve at the same level of thinking at which we created them.
>
> Albert Einstein

It does not take a person of Einstein's intellect to perceive that the world is replete with a multitude of significant problems of our own making: the greenhouse effect, acid rain, deforestation, pollution, bloody altercations over boundaries of nations, crime, food supplies to needy areas, ad infinitum.

Some reasons for these are: lack of money, lack of motivation, lack of vision, fear of losing current resources, etc. The single most important lack world-wide, however, is that of technically competent professional individuals who can attack the gaps between 'what is' and 'what should be'.

As a specific example, Fig. 1 shows the dearth of currently technically qualified staff at a major aerospace corporation. The figure demonstrates the wealth of senior knowledge, the promise of new technical staff, and a gap in the productive mid-range. Similar plots would hold nationwide for most engineering-related companies *and* universities!

SOLVING THE PROBLEM

There are a number of approaches to solve the problem of increasing the effectiveness of a resource, i.e. the pool of technically qualified people.

1. Find a new supply of the resource. Increase enrollment efforts for women, minorities, Third World, under-represented populations. Retrain other professionals.
2. Find a new type of resource. Use robots to perform tasks now done by humans. Not viable!
3. Discover ways of making better use of existing resources. Provide powerful tools to augment the expertise of current and future engineers.

The last option is adapted for this paper, the typical 'engineering' approach to a resource allocation problem: that of 'cutting the pie' into smaller slices by more efficient use of the existing resource.

SOFTWARE SOLUTIONS

The virtual explosion of computing within the lifetime of most of us is practically unknown in history. Never has such a powerful tool been adopted so quickly world-wide. Figure 2 gives an indication of how dominant computing has become. In the United States, the amount is rapidly approaching that spent in the entire automotive industry.

We will look at three major software solutions for augmenting human expertise: knowledge-based systems, computer-based learning, and software engineering. A short definition of each will be followed by a historical perspective and future projections. Finally, we suggest specific ways to turn the future into reality.

* Professor in Computer Science, Head of Department of Computer Science, Western Kentucky University.

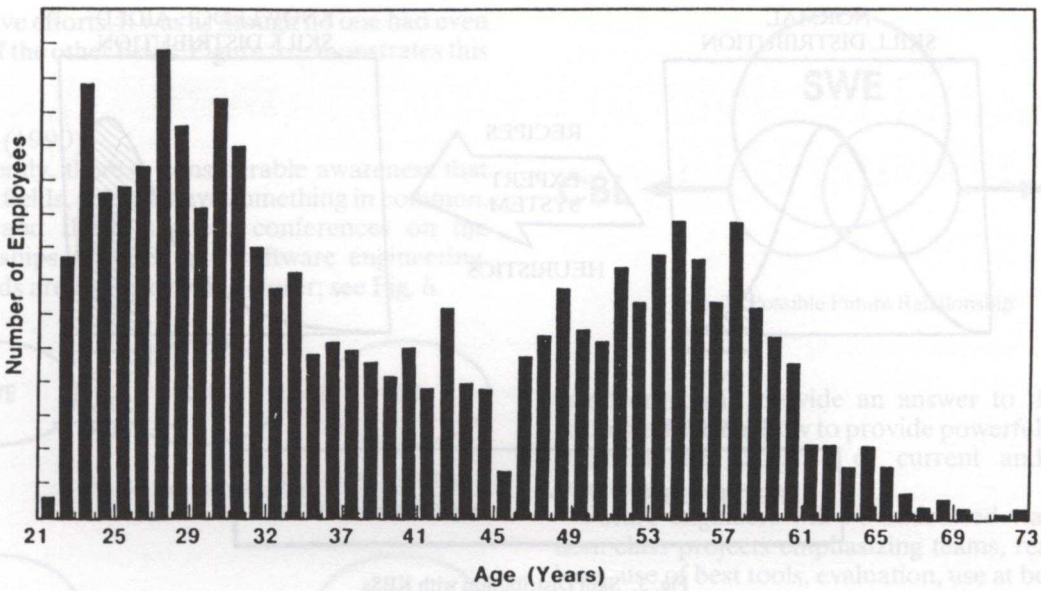

Fig. 1. Distribution of Engineers

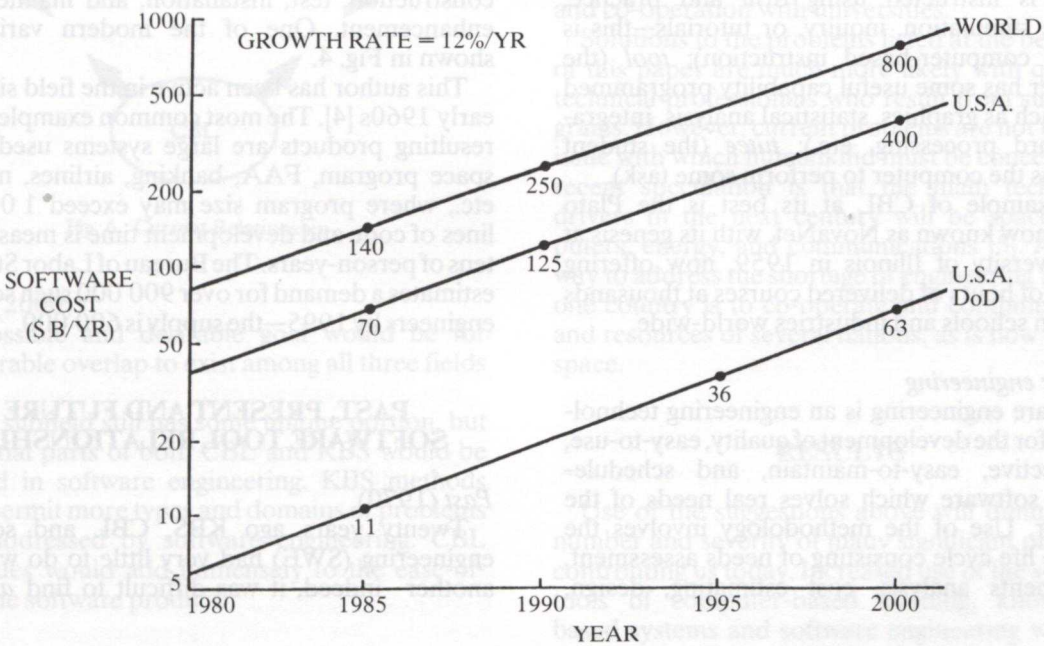

Fig. 2. Software Cost Trends

Knowledge-based systems (KBSs)

The rationale for the existence of KBSs may be paraphrased as 'to enable a computer to behave more intelligently'. These are computer programs which make extensive use of human experimental knowledge, and which exhibit more 'intelligence' than is usually displayed by normal programs. The subset of KBSs known as *expert systems* is able to perform on a par with human experts in very narrow domains, although very fragile at their boundaries. This author has worked in the field a number of years, most recently on the space shuttle main engines [1]. A classical reference concerning KBSs is by the late Donald Waterman [2]. Figure 3 shows the attraction of KBSs.

Computer-based learning (CBL)

The rationale for the existence of CBL may be paraphrased as 'to enable a human being to behave more intelligently'. The use of computers in the learning process, both in education and industry, is nearly as old as the computer itself. A recent paper at the 31st Annual Conference of a leading professional organization noted that ACM, the professional computing society, was only 10 years older [3]. Use of CBL involves several features: *tutor* (the

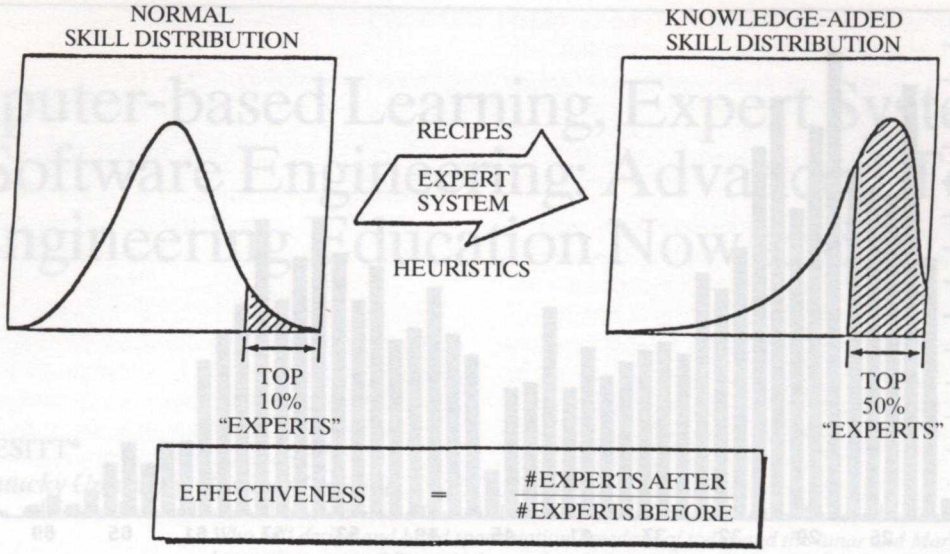

Fig. 3. Skill Distribution with KBSs

student is instructed using drill and practice, dialogue stimulation, inquiry, or tutorials—this is classical computer-based instruction); *tool* (the computer has some useful capability programmed into it such as graphics, statistical analysis, integration, word processing, etc.); *tutee* (the student programs the computer to perform some task).

An example of CBL at its best is the Plato system, now known as NovaNet, with its genesis at the University of Illinois in 1959, now offering millions of hours of delivered courses at thousands of sites in schools and industries world-wide.

Software engineering

Software engineering is an engineering technology tool for the development of quality, easy-to-use, cost-effective, easy-to-maintain, and schedule-meeting software which solves real needs of the customer. Use of the methodology involves the software life cycle consisting of needs assessment, requirements analysis, cost estimating, design, construction, test, installation, and maintenance/enhancement. One of the modern variants is shown in Fig. 4.

This author has been active in the field since the early 1960s [4]. The most common examples of the resulting products are large systems used in the space program, FAA, banking, airlines, military, etc., where program size may exceed 1 000 000 lines of code and development time is measured in tens of person-years. The Bureau of Labor Statistics estimates a demand for over 900 000 such software engineers by 1995—the supply is 600 000.

PAST, PRESENT AND FUTURE SOFTWARE TOOL RELATIONSHIPS

Past (1970)

Twenty years ago KBS, CBL and software engineering (SWE) had very little to do with one another—indeed, it was difficult to find *any* col-

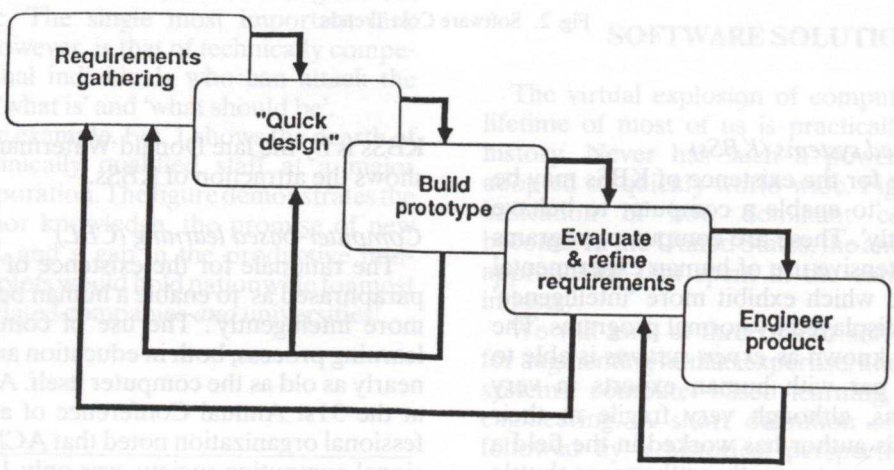

Fig. 4. Modern Software Development

laborative efforts. It was as though no one had even heard of the other fields. Figure 5 demonstrates this gap.

Present (1990)

Currently, there is considerable awareness that pairs of fields, at least, have something in common. ASEE and IEEE sponsor conferences on the relationships of KBSs and software engineering. The fields are finally drawing closer; see Fig. 6.

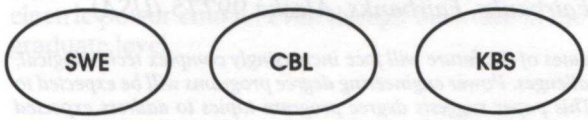

Fig. 5. Past Relationship

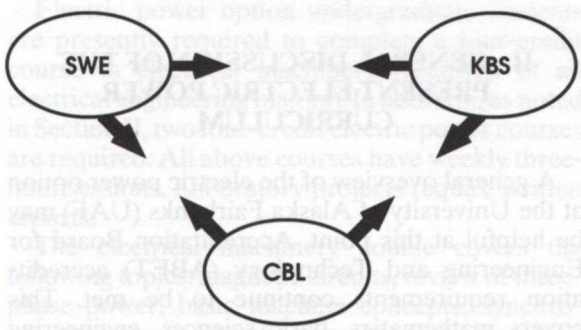

Fig. 6. Current Relationship

Future (2000)

A possible and desirable goal would be for considerable overlap to exist among all three fields (Fig. 7).

Each subfield still has some unique portion, but substantial parts of both CBL and KBS would be included in software engineering. KBS methods would permit more types and domains of problems to be addressed by software engineering. CBL techniques would add immensely to the ease-of-use of the software product.

HOW DO WE GET TO THE FUTURE?

The future goal of strong co-operation among professionals in KBSs, CBL and software engineering will provide an answer to the issue addressed earlier: how to provide powerful tools to augment the expertise of current and future engineers.

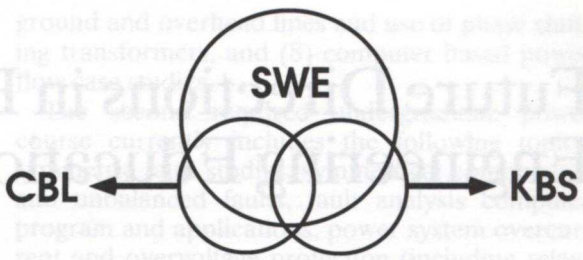

Fig. 7. Possible Future Relationship

Future engineers will be addressed via education: class projects emphasizing teams, real problems, use of best tools, evaluation, use at beginning of academic program and co-operation with industry. Current engineers will be aided by training with teams, real problems, use of best tools, evaluation and co-operation with universities.

Solutions to the problems listed at the beginning of this paper are much more likely with qualified technical professionals who result from such programs. However, current problems are not the only issue with which humankind must be concerned. A recent speculation is that the main technology drivers in the next century will be space, computers, energy, and communications [5]. Another way to address the shortage of engineering talent in one country is to co-operate and combine efforts and resources of several nations, as is now done in space.

RESULTS

Use of the suggestions above will diminish the number and severity of many significant problems confronting us today. Increased use of the software tools of computer-based learning, knowledge-based systems and software engineering will be a positive impact for engineering education now and in 2001.

REFERENCES

1. K. Modesitt, Inductive knowledge acquisition experience with commercial tools for space shuttle main engine testing, *Fifth Conference on Artificial Intelligence for Space Application*, NASA/University of Huntsville, AL (1990).
2. D. Waterman, *A Guide to Expert Systems*, McGraw-Hill, New York (1986).
3. K. Modesitt, Lessons learned in computer-based learning: a personal tale of three decades, *Association for the Development of Computer-based Instructional Systems Conference*, Washington, DC (1989).
4. K. Modesitt, Software engineering training for TI middle management, *ACM National Conference* (1980).
5. G. Piel, *2081*, Prentice-Hall, New York (1981).