THE NEW WORLD OF "AI-SM"

(ARTIFICIAL INTELLIGENCE IN SOFTWARE MAINTENANCE)

Girish Parikh
Independent Consultant
1787-B West Touhy
Chicago, IL 60626

(This article is based on the author's report-in-progress: AI-SM (Artificial Intelligence in Software Maintenance): The New Wealth Building Opportunities of the Eighties and Nineties.)

I have coined the new term "AI-SM" for the emerging trend of applying AI (artificial intelligence) in SM (software maintenance). Let me explain.

A recent item in Datamation (Carlyle, 1985) alluded to the trend, however, the three COBOL restructuring products/services mentioned did not seem to use AI. At the end, the article mentioned "an expert systems tool to look into the [maintenance] problem" devised by productivity and maintenance consultant T. Capers Jones.

A recent announcement of IBM's COBOL/SF (COBOL Structuring Facility) that restructures existing COBOL programs claims to use AI techniques (Babcock and Gallant, 1985).

The trend is clear. The association (the outright marriage is out, because both AI and SM are generic) of AI with SM is coming. What are the implications of this affair?

Obviously one (AI) is a highly popular subject, while the other is equally unpopular. One is almost loved while the other is equally hated. Would this affair last? I predict that it will become one of the most passionate (and profitable) relationships in the history of computing. The love-hate relationships usually intensify the involvement, as the legends testify.

Who knows, some of the popularity of AI may rub off SM, making it an exciting field! And who will not be lured by the wealth building opportunities in the field of AI-SM?

(Incidentally, did you notice that I devised a new term ("SM") for Software Maintenance? The world seems to be in search of a new word for the unpopular subject. Emulating the popular term AI, why not use SM for Software Maintenance? To me the term "SM" tastes sweet! I think it will be less sour than the mouthful and already unpopular term Software Maintenance to many.)

Some (AI) gurus say that the term is a misnomer -- and so is the term SM! What a lovely match of terms in AI-SM! Though made up of unreal (or partially real) terms in meanings, "AI-SM" is real! AI, among other applications, has already started spreading her charm on SM.

AI is also a darling of academia. But in the case of SM, academia is over 30 years behind (Parikh, 1985a), and it seems no attempts are being made to catch up. SM came in existence and has been practiced since the first program was written (and maintained) for the first commercial electronic computer introduced in the mid-fifties. (It is also likely that Ada Lovelace, considered to be the world's first programmer, was also the world's first maintenance programmer!) Now SM, hand-holding AI, will sneak into the academic world!

The giant of SM (I need not repeat the alarming statistics) can be wooed by AI, the lady with many charms. I would not go into details in this short article, but as an example, lèt me quote from the chapter "Expert Software Maintenance Systems" in my research report There is a fortune to be made in software Maintenance (Parikh, 1985b).

(Some of my predictions may read like "software fiction." (Or "software maintenance fiction?") (Again I have coined this term. If there is science fiction, why not software fiction? After all, software is an indispensable part of computer science, and maintenance is intrinsic to software.) Also remember that many inventions were once science fiction.)

> EXPERT RESTRUCTURING SYSTEMS (ERS). We may also see expert restructuring systems (ERS) in the future. Such expert systems may also ask for user input for any changes to the existing functions to the program. Input data formats and output data formats of a program to be restructured may also be optionally modified. Some advanced expert systems may have the capability to convert the program to a DBMS-based program, that is, inputs and/or updates/outputs of the new program are DBMS files. The expert system may also provide the option of rewriting the programs into any third or fourth/fifth generation language, of course from any language of the input program. A variety of expert restructuring systems may become available in the market with different capabilities.
>
> Thus expert restructuring systems which will essentially be expert systems will also do "intelligent restructuring," not just mechanical restructuring.
>
> EXPERT MAINTENANCE MACHINES. Now let's discuss some of the features that the expert maintenance machine might have. In particular, let's see what kind of information it will contain.
>
> Of course it will contain the years of experience of maintenance programmers. The machine may also be custom built for particular systems and programs. [It seems to me an awe-inspiring idea.] Since the development machine will have recorded all the development events, it can trace back and find out how to make certain changes in the program. By making changes at one place, will it affect other parts of the program? The expert maintenance program can quickly tell about this. Also, the events of maintenance programming done on a particular program will be preserved by the expert machine.
>
> Expert software maintenance systems seem to be ultimate in software maintenance area, but with such systems there may be human effort necessary in order to perform the maintenance work, for instance, in making decisions. The amount of human effort required

will depend on the kind of software being maintained. If it is well structured according to some standard methodology, the expert systems may handle the maintenance work according to the commands of the user. If the system is not well structured or unstructured, the expert system may have to depend more on the human effort.

It is also possible that the different expert systems may be designed to meet needs of different kinds of software -- systems software, on-line software, real time software, applications software, etc. There may also be given further descriptions of what method, or no method, was used in developing it; for example unstructured, structured according to Warnier's data structured techniques LCP/LCS, structured according to Constantine/Yourdon structured design techniques, etc. Thus, the expert system will have knowledge of maintaining different kinds of software, and software developed according to different methods. Expert software maintenance systems will thus aid humans in maintenance work.

The expert maintenance machine will keep track of all changes, and will automatically update the documentation also.

THE MONEY-MAKING MACHINES

Imagine how much such an expert machine can save. If James Martin's figure of $30 billion worldwide expenditure on maintenance is at all accurate (Martin and McClure, 1983; p. 15), this expense will almost double in next 10 years, that is, it would be $60 billion (due to inflation). It may even exceed that amount, as there will certainly be more software added to maintain, and less software discarded from the old inventory. Imagine how much the expert maintenance machine can save, how rich the person developing such machines can become. They can prove to be money-making machines for him!

The applications of "AI in Software Maintenance" (AI-SM) seem to offer mind-boggling opportunities. I dare you to grab them!

REFERENCES

Babcock, Charles, and John Gallant (1985). "IBM Unveils Tool to Restructure VM, MVS Cobol Code," Computerworld, November 25, p. 10.

Carlyle, Emmett R. (1985). "Can AI save COBOL?" ("News in Perspective" Section), Datamation, September 15, pp. 42-43.

Martin, James, and Carma McClure (1983). Software Maintenance: The Problem and Its Solutions. Prentice-Hall, Englewood Cliffs, NJ.

Parikh, Girish (1985a). "Academia Is Over 30 Years Behind," Communications of the ACM ("ACM Forum" Department), March, Vol. 28, No. 3, p. 240.

Parikh, Girish (1985b). There is a Fortune to be Made in Software Maintenance: Opportunities in the $30 Billion Software AfterMarket, Shetal Enterprises, Chicago.

Parikh, Girish (in preparation). AI-SM (Artificial Intelligence in Software Maintenance): The New Wealth Building Opportunities of the Eighties and Nineties.)