MACHINE INTELLIGENCE AND THE HUMAN WINDOW

DONALD MICHIE
The Turing Institute
Glasgow, UK

From the time of Aristotle's Lyceum to the present day, the institutions of science have been centers both for teaching and for learning. Hitherto we have been concerned only with human teachers and learners. Today, as foreshadowed more than forty years ago by the father of Artificial Intelligence, Alan Turing, the first industrial-strength machine learning systems are with us. Computer programs begin to behave like pupils, not only learning from their masters but also from their own experience and work. Turing's prophetic words, spoken in 1947 to the London Mathematical Society, were as follows:

> One can imagine that after the machine had been in operation for some time, the instructions would have been altered out of recognition, but nevertheless still be such that one would have to admit that the machine was still doing very worthwhile calculations. Possibly it might still be getting results of the type desired when the machine was first set up, but in a much more efficient manner. In such a case one would have to admit that the progress of the machine had not been foreseen when its original instructions were put in. It would be like a pupil who had learnt much from his master, but had added much more by his own work.

Here we have the theme of computer as learner as indicated in Fig. 1. One further step is needed to unite Turing's vision with the theme of computer as teacher. The intelligent machine, having learned more than its masters of some specialist topic, must transmit the new knowledge back to them. Of course *unintelligent* machines do this today in their untiring discovery of new numerical facts through computer-based calculation and tabulation. But for computers to make themselves useful at the level of concepts rather than brute facts, a common conceptual channel of discourse must be established, a machine-oriented language for expressing human concepts without distorting them.

The transparency or otherwise of computer processes is of clear significance to our technological society. It was first addressed at governmental level ten years ago, namely by Japan, with the formation and promotion of the concept of a Fifth Generation computer (JIPDEC Fifth Generation Computer Committee, 1981).

A theme of the Fifth Generation plan lying rather inconspicuously in the main Japanese planning documents of 1981 is known as inductive learning, that

Applied Artificial Intelligence: 5:1–10, 1991

A. M. Turing (1947) The Automatic Computing Engine

- **Assimilation of knowledge by the machine as a pre-requisite of intelligent behaviour**

- **Role of machine learning, from teacher and trial-and-error experience, as critical to knowledge-acquisition**

Figure 1. Key themes of AI's first milestone: Turing's 1947 Lecture to the London Mathematical Society

is, machine acquisition from data of machine-executable rules. When such rules are recognizably human in style, we say that the machine is forming concepts. By generalizing over examples, whether supplied tutorially by the user or sampled from real-time monitoring equipment, or dredged from large database files, the machine induces descriptive rules and stores them for later execution by the same or other machines or by humans. Given only that such synthetic rules satisfy certain human-like structural constraints so as to qualify them as concepts, then they can serve as contributions not only to our stock of software but *also to our stock of knowledge.*

To induce knowledge, including brand new knowledge, from facts is now an established commercial technology in Europe. It was the Fifth Generation's intention to do it on a massive scale. To date, however, no serious attempts have been reported and the goal has receded into the background. Meanwhile it is moving into the foreground in Europe (Bratko and Lavrac, 1987).

From one point of view there is something puzzling about this. In his keynote address at the 1984 international meeting in Tokyo, Dr. K. Kawanobe clearly identified inductive learning as a critical missing factor (Kawanobe, 1984) (see Fig. 2). Moreover, as early as 1976 such a contribution was already recorded to the credit of the American program Meta-DENDRAL. This is the inductive inference module of the celebrated DENDRAL expert system developed by Feigenbaum and members of the Heuristic Programming Project at Stanford University. By generalizing over mass spectroscopy data for a particular class of compounds, namely the mono- and poly-keto-androstanes, Meta-

DENDRAL was able to infer a set of rules for deducing the molecular structures of these substances from their mass spectra (Buchanan et al., 1976). The fruits of the program's work were in due course published in the chemical literature as an original contribution to knowledge. A pioneering achievement from Michalski's machine learning laboratory at Illinois University was also reported in the same year (Chilausky et al., 1976). Machine synthesized diagnostic rules in Michalski's study of soy-bean data were sufficiently human-transparent to serve as an improved reference text for professional plant pathologists (see Fig. 3).

With machine knowledge systems, transparency is all. In seeking appropriate forms for the machine-oriented representation of knowledge, designers of AI programming languages must not rest content until the programs that their followers write routinely pass not just one but both tests formulated in Fig. 4.

Let us now turn back to the Japanese Fifth Generation idea. In the *Preliminary Report* a diagram is reproduced (Fig. 5). What does it mean? Clearly, there are three subsystems. To the left lies the cognitive machinery of the user, rich in culture, poor in store and processor. To the right is deployed the might of tomorrow's gigaflop processors and gigabyte memories. What then is the role of the subsystem in the middle?

It is not at all easy to work out from this diagram alone just what was being proposed. My own decipherment was aided by confirmation from the diagram's author. Dr. Koichi Furukawa of Japan's ICOT Laboratory. The *general* notion is clear enough, namely, that the central circle is to contain an artificial intelligence charged with a bridging function—to mediate between our richly endowed but

"For the basic high level inference software, attempts will be made to develop high level artificial intelligence including inductive inference, analogy and learning functions"

Kawanobe : INTERMEDIATE STAGE (1985-89)

Figure 2. K. Kawanobe's prescription for the Fifth Generation's "Intermediate Stage."

R. Chilausky, B. Jacobsen and R. S. Michalski (1976) An application of variable-valued logic to inductive learning of plant disease diagnostic rules.

- First inductively synthesised rule-base
- Rule-base performance exceeds human expert's

Figure 3. A milestone in machine learning.

(1) can the user understand the code ?

(2) can he check its operation in his head ?

Figure 4. Two criteria for knowledge-based programming.

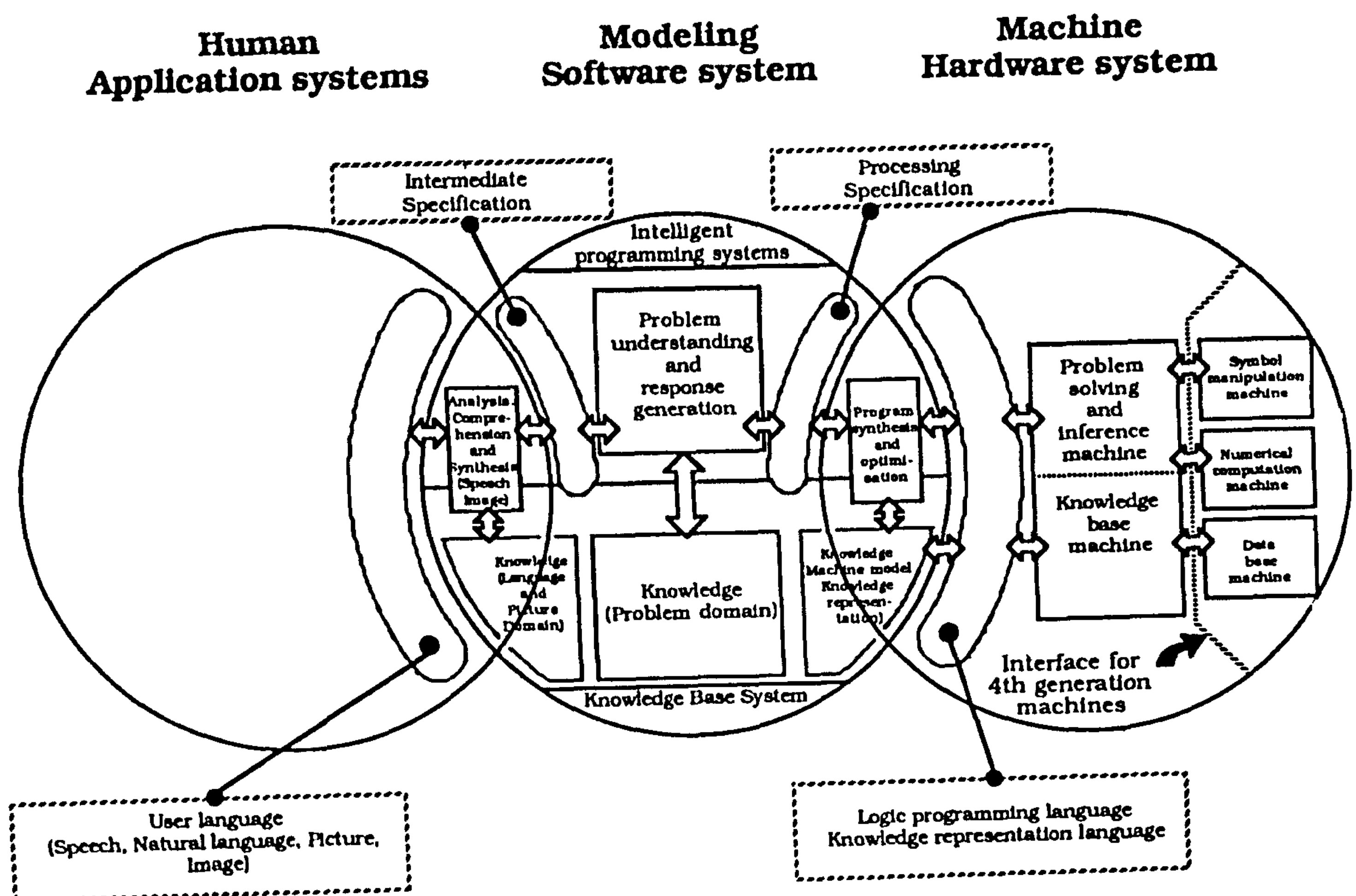

Figure 5. The target product of Japan's Fifth Generation plan for research and development. The central circle encloses a new AI-based technology intended to mediate between two rapidly diverging technical cultures—between human arts, crafts, and sciences on the one hand and computational "brute force" on the other.

slow and muddled human brains and the new race of super-powerful, and super-opaque, computing architectures with which we shall have to live. More scrutiny of the diagram reveals the Fifth Generation notion of an expert system:

- Such a system is able, on the basis of a representation coming in from the left-hand circle, to understand exactly what the problem is that the user wants to have solved.
- It can formulate a solution as a machine program written in an ultra-high-level language, optimized to run efficiently on the knowledge machine with its complex of supercomputer slaves in the right-hand circle.
- It is able to operate in the inverse direction of the above two steps, that is, it can translate the output obtained from the supercomputer complex into a form acceptable by the user as an intelligible answer to his query.

To do all this, the system in the central circle must manipulate models not only of the problem domain itself but also of what goes on inside the two circles on either side of it. The task associated with the central circle is thus new. Previous generations of designers gave us computing systems programmable to construct *solutions* within defined problem areas. The Fifth Generation's task was to design systems able to construct *solution strategies.*

In complex domains the construction of strategies—for concreteness I shall call them "calculative plans—is a quintessential human activity. But whenever such a plan is brought into being by human intelligence, the question presents itself of who or what is to execute it. So can we get computers to construct their own calculative plans? The mechanism of inductive inference, as I shall show, offers a means whereby computers can do just this. Thus the program Meta-DENDRAL was able to infer a set of calculative plans in the form of rules for deducing molecular structures from mass spectra (Buchanan et al., 1976). But first we must ask what sort of calculative plans, or programs, are they liable to produce? Will these necessarily possess the qualities of transparency to man that we should demand? Or are special measures of some kind needed to ensure that these machine products come equipped with a human window?

A recent case from the Space Shuttle program may help to clarify, in miniature, so to speak, the steps by which a computer program may convert a collection of facts into a piece of operational knowledge. NASA's requirement was a decision procedure able to monitor real-time variables relevant to the question "At each given instant should the spacecraft be under the control of the Autolander or under manual control? Moreover, on the basis of measurement, for example, of atmospheric turbulence, wind direction, attitude of craft, stability, etc., the procedure is required to give real-time answers to this critical decision question."

The Autolander's chief designer, Roger Burke, together with members of his design team, attempted the required programming task. At first the team was

frustrated by the difficulty universally experienced by expert practitioners of highly tuned mental skill, namely inability to introspect their own decision rules. The same experts, however, can normally articulate the same skills if allowed to express them in a language of examples. The upper part of Fig. 6 shows a solution of the Shuttle problem expressed in such a language. Burke and his

```
MODULE: shuttle

STATE: one
ACTIONS:
        noauto [advise "Don't use auto land"]
        auto [advise "Use auto land"]

CONDITIONS:
        stab [ask "stable ? " "stab, xstab"] [stab xstab]
        error [ask "errors ? " "XL, LX, MM, SS" ] [XL LX MM SS]
        sign [ask "sign ? " "pp, nn"] [pp nn]
        wind [ask "winds ?" "head, tail"] [head tail]
        mag [ask "magnitude ?" "L, M, S, O"] [L M S O]
        vis [ask "visibility ?" "y, n"] [y n]

EXAMPLES
        -       -       -       -       -       n       =>      (auto, goal)
        xstab   -       -       -       -       y       =>      (noauto, goal)
        stab    LX      -       -       -       y       =>      (noauto, goal)
        stab    XL      -       -       -       y       =>      (noauto, goal)
        stab    MM      nn      tail    -       y       =>      (noauto, goal)
        -       -       -       -       O       y       =>      (noauto, goal)
        stab    SS      -       -       L       y       =>      (auto, goal)
        stab    SS      -       -       M       y       =>      (auto, goal)
        stab    SS      -       -       S       y       =>      (auto, goal)
        stab    MM      pp      head    L       y       =>      (auto, goal)
        stab    MM      pp      head    M       y       =>      (auto, goal)
        stab    MM      pp      tail    L       y       =>      (auto, goal)
        stab    MM      pp      tail    M       y       =>      (auto, goal)
        stab    MM      pp      head    S       y       =>      (noauto, goal)
        stab    MM      pp      tail    S       y       =>      (auto, goal)

MODULE shuttle IS
        STATE: one
           IF (ask "visibility ? " "y, n") IS
              "y" : IF (ask "errors ? " "XL, LX, MM, SS") IS
                    "XL" : (advise "Don't use auto land", goal)
                    "LX" : (advise "Don't use auto land", goal)
                    "MM" : IF (ask "stable ? " "stab, xstab") IS
                           "stab" : IF (ask "sign ? " "pp, nn") IS
                                    "pp" : IF (ask "magnitude ? " "L, M, S, O") IS
                                           "L" : (advise "Use auto land",goal)
                                           "M" : (advise " Use auto land", goal)
                                           "S" : IF (asks "winds ? " "head, tail") IS
                                                 "head" : (advise "Don't use auto land", goal)
                                                 ELSE (advise "Use auto land", goal)
                                           ELSE (advise "Don't use auto land", goal)
                                    ELSE (advise "Don't use auto land", goal)
                           ELSE (advise "Don't use auto land", goal)
                    ELSE IF (ask "stable ? " "stab, xstab) IS
                           "stab" : IF (ask "magnitude ? " "L, M, S, O") IS
                                    "L" : (advise "Use auto land", goal)
                                    "M" : (advise "Use auto land", goal)
                                    "S" : (advise "Use auto land", goal)
                                    ELSE (advise "Don't use auto land", goal)
                           ELSE (advise "Don't use auto land", goal)
           ELSE (advise "Use auto land", goal)

GOAL OF shuttle
```

Figure 6. *Solution of Shuttle problem in a language of examples. The upper part shows the final form of a training set of examples for the Autolander problem. The lower part shows the decision rule.*

team arrived at it in interaction with an induction program known commercially as RuleMaster. The program converts example sets into machine-executable rules in a form illustrated in the figure's lower half. In addition to being a solution to NASA's problem, such a rule can be viewed as a miniature operational theory—a grain of knowledge, so to speak. We call it knowledge rather than just information because as seen by the NASA engineers this grain of machine-synthesized knowledge meets both the criteria of Fig. 4. First, the space engineers can understand the rule, second they can also run it in their heads on test cases.

BEYOND INTUITION

For complex tasks is it true, as the Japanese have concluded, that only rule-based programming disciplines, geared to mimic the representations used by the expert brain, can provide the needed bridge function? Can a human window be defined? Are standard programming styles doomed for some reason to miss this window and so to generate only opaque products, unfriendly and therefore unwelcome and unsafe?

We certainly cannot afford to leave the issue of the human window in the long distance. Realization has been growing that for socially critical tasks *unintelligent* but increasingly powerful computers are becoming actually dangerous.

The first public body to take note of this possibility was the European Community. In 1982 the EEC's FAST program placed a study contract (Kopec and Michie, 1983) with Dr. Daniel Kopec and myself. Fig. 7 lists the four case studies that we selected. The published FAST report drew a disquieting conclusion. Until and unless the top-level monitoring and control software of such installations can be restructured in user-transparent rule-based style, we concluded that such accidents will recur. Indeed we predicted that they will at some stage begin to escalate in frequency as technological complexity grows. Reviewing the scene today, seven years after the FAST report, I see indications that the predicted escalation has started.

It is plain from the planning documents that Japan's Fifth Generation plan was motivated by a similar awareness of the human window imperative. Unfortunately, the suddenness with which Japan's awe-inspiring picture was unveiled blinded the computing world from carefully reading the message. Moreover, continued Japanese omission of the inductive learning tools called for by the original timetable has now undermined their entire objective (Institute for New Generation Computer Technology, 1988). Nonetheless, Japan's Fifth Generation enterprise has made and is making a contribution.

Perhaps more important from a European perspective is the fact that the planning documents, unlikely now to find fulfillment in Japan, constitute an excellently researched characterization of an inspiring objective. European labo-

Contract from the EEC's "FAST" programme

F orecasting and

A ssessment of

S cience and

T echnology

To look at 4 well-documented failures of the man-machine interface in complex (and costly) decision-taking:

Hoogovens steel mill **Three Mile Island** **NORAD military system** **Air traffic control**	**chain of inference** **Understood ?** **can it be (even approximately)** **Checked ?**

Figure 7. Terms of reference of the Kopec-Michie study in 1983.

ratories have meanwhile accumulated the required tools and methods for inductive knowledge acquisition and synthesis. These laboratories and others in Europe are at liberty now to readdress Japan's abandoned plans. With a little help from our friends, we could then even decide to implement them.

REFERENCES

Bratko, I, and Lavrac, N. eds. 1987. *Progress in Machine Learning*. Wilmslow: Sigma Press.

Buchanan, B. G., Smith, D. H., White, W. C., Gritter, R. J., Feigenbaum, E. A., Lederberg, J., and Djerassi, C. 1976. Application of Artificial Intelligence for Chemical Inference XXII. Automatic Rule Formation in Mass Spectroscopy by Means of the Meta-DENDRAL Program. *J. American Chemical Society*, 98:6168–6178.

Chilausky, R., Jacobsen, B., and Michalski, R. S. 1976. An Application of Variable-Valued Logic to Inductive Learning of Plant Disease Diagnostic Rules. In *Proc. 6th Ann. Symp. on Multi-variable Logic*, Utah.

Institute for New Generation Computer Technologies. 1988. *Fifth Generation Computer Systems 1988*. Berlin: Springer.

JIPDEC Fifth Generation Computer Committee. 1981. *Preliminary Report on Study and Research on Fifth-Generation Computers in 1979–1980*. Tokyo: Japan Information Processing Development Centre.

Kawanobe, K. 1984. Current Status and Future Plans of the Fifth Generation Computer Systems Project. In *FGCS 84. Procs. of the Int. Conf. on Fifth Generation Computer Systems 1984*. Tokyo, Japan, November 6–9, 1984, pp. 3–17.

Kopec, D., and Michie, D. 1983. *Mismatch between Machine Representations and Human Concepts: Dangers and Remedies*. FAST Series No. 9. Brussels: Commission of the European Communities.