OUTLINE OF REMARKS MADE IN PRESENTATION ON THE NATURE, DESIGN, AND USE OF INTELLIGENT SOFTWARE AGENTS

Yoh-Han Pao

Case Western Reserve University
Cleveland, OH 44106, U.S.A.

Abstract: There is genuine need for computer software systems that can carry out significant cognitive tasks in robust, autonomous and adaptive manner. There is much confusion in the manner in which such activities are discussed. This is unfortunate because exaggeration can lead to loss of insight and of appreciation of opportunities. In this presentation, we delineate some qualities of intelligent software agents and cite some unique characteristics of such entities. We also attempt to give an overall view of the technologies that support such capabilities.

Keywords: Software Agents, Learning, Design, Neural Networks, Parallel Distributed Processing

1 SOME 'NECESSARY' ATTRIBUTES OF INTELLIGENT AGENTS

Agents are entities to whom we can delegate tasks with the expectation that those tasks will be performed in reasonably efficient and effective manner, and that we will not be bothered all the time by requests for further advice and instructions. For problem-solving task-performing agents, the solutions should be timely and preferably highly effective and the costs should be reasonable in a relative manner of speaking.

We are all familiar with the concept of agents. Resourceful well-motivated travel agents or insurance agents can be of great help. Many others are also agents even if they do not use that designation explicitly.

In computer-based automation, agents appear as an aspect of system architecture, to keep the design of the system modular. Many major software products are organized in that manner. But as that practice becomes commonplace we begin to feel the need for higher level of performance by software agents.

And so not only do we need software agents but we need them to be more competent, and more 'intelligent'.

In this discussion we are concerned with problem-solving task-performing software agents and we face the issue of what needs to be done to enable competence and intelligence. [Figure 2]

Recently one feature article in a special issue of the publication IEEE Spectrum tried to peer into what might the next few decades bring. The format was for the feature writer to interview a few visionaries and to elicit some prognostications from them. I was much taken by the wording in one of the replies from Leornardo Chiariglione, a significant leader of research in CSELT, a major Italian information and communications company. Those words are listed in Figure 3.

Incidentally it is easy to get quite caught up with all this and sometimes become quite anthropomorphic as indicated by the next quotation (not from Chariglione) which also caught my interest. It is not as fanciful as it seems to be. It is merely metaphorical.

Murch and Johnson published a highly readable book in 1998 entitled Intelligent Software Agents (Murch. and Johnson. 1999). In that book they list and discuss several prototypical 'intelligent' agents and try to identify features which would justify their being considered to be intelligent. They also refer to compilations of "approved" definitions of agents. A popular view of intelligent agents can be synthesized from all that material and an abbreviated version of that is shown in Figure 4 and 5.

We have also assembled a list of what might be considered to be 'necessary' attributes. These are shown in Figure 6. This is not necessarily a 'sufficient' list and it is not that different from the list of Murch and Johnson; but the emphasis is different. We are concerned with what might be considered absolutely necessary for Intelligent Software Agents. For the purposes of this discussion we will focus on the features of *competent, can design solutions, and can interact and communicate.* Not that the other attributes are not important but these three attributes will serve to provide focus in our discussions.

2 ADAPTIVE PARALLEL DISTRIBUTED PROCESSING

Actually Intelligent Agents have been studied by Artificial Intelligence (AI) researchers for a long time, ever since the inception of that discipline and research community. In retrospect and with hindsight we see that that research explored and established a totally new computing paradigm, that of computing in terms of symbols other than numeric, with emphasis on logic and correctness, and on search. Many new insights on computing were established. In its brilliance, that brand of AI nevertheless tended to concentrate on purely cognitive matters, and not on the sensors or motors which interacted with the 'mind'. More recently, various efforts under the newly named designations of Situated Cognition, Situated Robotics, Artificial Insects, Artificial Life and so on have worked to go quite far a field from what might be called the Good Old Fashioned AI [GOFAI; Haugeland 1985]. In this connection in recent times there has been a large resurgent interest in pattern-based computational paradigms. Some of those, the more distinctive ones, are listed in Figure 7 under the general designation of Parallel Distributed Processing.

One theme of our discussion is that to be seen to be intelligent, one needs to be competent. The PDP technologies are some highly effective enabling technologies in so far as computational competence is concerned.

In the functioning of an Intelligent Software Agent these PDP components work modularly and synergistically. However, *self-organization* addresses principally the tasks of data processing and management. *Neural Networks* come in a variety of forms but are mostly used for recognition, estimation and prediction. *Genetic Algorithms and Evolutionary Programming* can work in guided stochastic search for optimization, for the formation of new strategy and for the inversion of functions. *Visualization* expedite communications between the Agent and humans. Researchers in the community of *Artificial Life* address issues of strategy formation.

We touch briefly on the nature of Artificial Neural-Net Computing in the context of function approximation, prediction and estimation [Figure 8]. It is an efficient and powerful way of inferring and representing an analytical relationship between vector fields. As shown in Figure 12, we might have seven examples of the value of a function that is assumed to exist. We can use a neural-net to build an approximation of what that function might be. And if future data show that we are in error, we can adapt dynamically and correct our error instantly on the run! [Figure 9]

We have been talking about the need to be competent and what technologies help to make that possible. Now we shift to the requirement that an Intelligent Software Agent be able to design its own solution. This requirement can be satisfied at different levels. At minimal level, the scenario is that an Agent may be of a general category but when notified that it is being made responsible for the performance of a certain type of task, it will help its client assemble the modules needed to solve that problem in a standard routine sort of way. At higher levels of performance, the Agent not only will evaluate different options but can and will identify new creative solutions and will evaluate those, and propose the leading candidates or act autonomously on it own within prescribed constraints. [Figure 10]

We can illustrate the idea of design in the case of two different tasks. In one case the task is to provide medical outcome decision support. When provided patient details data and description of proposed treatment, the Agent is able to predict treatment results and expected patient recovery.

In such a case, the Agent would first build an input data space, and an output data space and learn a model of the relationship between input and output. Then in the case of this medical treatment outcome prediction, the task is that of asking 'what if 'questions and mapping the new, never previously encountered, conditions onto the outcome space.

3 AUTONOMOUS OPTIMAL DESIGN

There are also other types of tasks, where we have some firm ideas of what we want to achieve in the way of results and we want to know how to arrive at those desired results.

Under those circumstances, the Agent completes the loop shown in the diagram of the solution using guided stochastic search or some of the other capabilities and basically inverts the learned mapping. Performance constraints and autonomy constraints are both considered and satisfied.

The 'what if' type of service on the part of an Agent can be illustrated schematically with the help of the next few figures showing what happens in the treatment of the condition called 'carpal tunnel' relating to damage sustained through prolonged repetitive inappropriate but seemingly harmless motions or to sustained poor postures. This data was provided by Ed Wilson and he will be discussing that application in detail in one of the WRC presentations. [Figure 14, 15, 16, 17, 18, 19 20, 21, 22 and 23]

To get a feeling for the product or process discovery task, we refer to some research sponsored by the Materials Directorate of the Air Force Research Laboratory, WPAFB. The central focus of the research is the question of how to discover and design new materials which will have certain specified properties.

As an example, we consider the question of whether it would be possible predict whether two or more chemical elements (atomic species) would or would not form solid compounds of definite structure and properties. That is we describe each element in terms of certain features and we have a training set of data relating patterns of input features to compound forming category. We then test the learned mapping using a test set of cases.

Given a few thousand patterns each of (say) 15 features, it is difficult for a human to know what he has. An Agent knows what it has but needs to be able to share that knowledge with the human client. In the next few figures we show that an Agent can display of a body of materials systems using a particular type of dimension reduction scheme. In addition, using color, the Agent can provide information on the compound-forming characteristics of the material systems shown in the display. With this display both the human client and the Agent have a shared view of the meaning of the data. If given the description of a newly proposed materials system, it might not be clear whether the new materials system would be compound-forming or not. However either the Agent or the human can ask for a more local view and sometimes that view can provide much understanding. All this is but to illustrate the need for ability to communicate. [Figure 24 and 25]

We have had very encouraging and useful results in the matter of predicting the properties of materials systems and it seems that we may be able to implement a prototypical materials discovery system using Agent technology. [Figure 26, 27, 28, 29, 30 and 31]

4 LEARNING: TACTICS AND STRATEGY

However in intelligent task performance and intelligent problem solving, one needs to be clear about the need to distinguish between tactics and strategy.

In this discussion we will not talk about the details of such matters. Let us consider two examples. In one case we ask an Agent to evolve a control system which will take care of two objectives at the same time, one objective being that the pendulum needs to be nearly upright and the second objective being that the cart needs to be on the platform. That situation is typical of many control tasks. It turns out that it is relatively a straight forward matter for the Agent to fashion a net which given the system state will try to improve the average of the two deviations and to succeed in its task in this manner sometimes for a very long time. But almost always if it stuck to a certain fixed net it almost always fails eventually. A much more effective strategy consists of varying the relative emphasis on pendulum or on cart depending on the circumstances, i.e. the environment. Given that new strategy, control is efficient and effective and optimal control in the sense of minimum control action can be formulated automatically by the agent. [Figure 32]

As another example, we consider the game of Othello (Rosenbloom, 1982, Billman and Shaman, 1990, Moriarty and Miikkulainen, 1995). The strategies of playing to achieve maximum disc count at every play is not effective by itself. It seems that maximum disc might be considered one objective and mobility might be considered another, and questions of border, corner or stage of play (start, middle, end) are modulating factors of the environment, not unlike the circumstances of the pendulum control situation. [Figure 33, 34 and 35]

In conclusion, we hope that this overview of some aspects of Intelligent Software Agent technology has been of some interest.

REFERENCES

Billman, D. and Shaman, D. (1990). Strategy knowledge and strategy change in skilled

performance: a study of the game Othello. *American Journal of Psychology*, **103**, 145-166.
Moriarty, D.E. and Miikkulainen, R. (1995). Discovering complex Othello strategies through evolutionary neural networks. *Connection Science*, 7, 195-209.
Murch, R. and Johnson, T. (1999). *Intelligent Software Agents*, Prentice-Hall, New Jersey.
Rosenbloom, P.S. (1982). A world-championship-level Othello program. *Artificial Intelligence*, **19**, 279-320.