Turk J Elec Engin, VOL.9, NO.1 2001, © TÜBİTAK

Advances in Computer and Information Sciences: From Abacus to Holonic Agents

Tuncer ÖREN
School of Information Technologies, Univ. of Ottawa, Canada and
Information Technologies Research Institute of
TUBITAK Marmara Research Center
Gebze-Kocaeli-TURKEY

Abstract

The shift of paradigm emphasizing the importance of the ability to process knowledge rather then being knowledgeable is stressed. A taxonomy of knowledge processing tools, machines, or systems is offered. Major possibilities for future achievements that the author would like to underline are listed with emphasis on agent technology, agent-directed simulation, holonic agents, holonic-agent simulation, and contribution of system theories for cognitive abilities such as understanding, learning, adaptation, and anticipation in computerization.

Key Words: *shift of paradigm, taxonomy, system theories, agents, holons, agent-directed simulation, holonic-agent simulation*

1. Knowledge, Knowledge Processing, and the Shift of Paradigm

For a long time in the history of civilization, being knowledgeable was an important asset. Information age realities provide tools to store and interactively access a vast amount of knowledge. Hence, they challenge the value of being knowledgeable (in the sense of storing in human brain a vast amount of facts, alone).

For example, almost half of the books I have in my private library are on a single CD-ROM (Corel 1995). It contains over 3500 classical books that are searchable interactively. 32 volumes of Encyclopedia Britannica are on two CD-ROMs (EB). Recent announcements of some encyclopedia include, for example, Encyclopédie Hachette 2001 which comes in three CD-ROMs or one single DVD-ROM (Hachette 2000). Similarly, Encyclopedia Universalis, planned for release at the end of October 2000, will come in five CD-ROMs or in one single DVD-ROM (Universalis 2000). In the beginning of the advent of notebook computers, we had a transition period during which the volume of the documentation of the software loaded on the computer was much larger than the volume of the computer itself. Nowadays, the documentation resides on the hard disk. Similarly, all the knowledge we get through formal education can reside on a single CD-ROM where the knowledge can be stored for interactive seearch without any loss. Libraries were places to work as repositories of human knowledge. Now, information technology shrinks drastically the storage requirements and offers interactive search capabilities. Furthermore with Internet, geography became history;

with a single search command we can collect information from sources that we do not need to know their geographic locations.

Independent of the medium on which knowledge resides-paper, CD-ROM, DVD-ROM, or hard disk-information age has brought its own shift of paradigm: knowledge processing ability became more important than knowledge itself. Indeed, knowledge is necessary but not sufficient to solve problems. For example, no library or no CD-ROM can solve a problem. Therefore, knowledge processing and especially cognitive knowledge processing have to be explored to get the benefits of computerization. A taxonomy of over 500 types of knowledge and knowledge processing knowledge was given by Ören (1990).

2. Knowledge Processing Everywhere

Knowledge processing is done by two types of machines or systems: machines for knowledge processing and machines with knowledge processing abilities. Each group can further be divided into three categories, namely, fixed-wired tools or machines, variably-wired tools or machines, and stored-program tools or machines.

2.1. Machines for Knowledge Processing

These types of tools or machines are built for the sole reason of knowledge processing.

Fixed-wired tools or machines for knowledge processing have existed for a long time and the archtype, the abacus still exists. The relationship of the elements are fixed. As an abacus would inspire, they are indeed fixed-wired tools. Some other examples of fixed-wired knowledge processing tools are: astrolabe (Nasr 1976, Bott 1983), Al-Biruni's gear calendar computer and hodometer (Price 1984), and bar-linkage computers (Svoboda 1965).

Variably-wired tools or machines for knowledge processing include unit record machines (also called punched card machines), analog computers, and hybrid computers (de Beauclair 1968, Fröschl et al. 1993). When I started to work for a computer company in Turkey in 1963, unit record machines were in use as there was only one computer in the country at the time.

Stored-program tools or machines for knowledge processing are basically digital computers with all the variants: personal computers, notebook computers, digital assistants, palm computers, and wearable computers. Paraphrasing Kay (1984) who stated "Computers are to computing as instruments are to music," we can define a computer as an instrument to execute programs. Computers are already used extensively. But it seems this is only the beginning. (Ören 1990, Denning and Metcalfe 1997).

2.2. Machines with Knowledge Processing Abilities

Primary goal for machines with knowledge processing abilities is not knowledge processing; however, with their knowledge processing ability they can perform their task much better.

Fixed-wired tools or machines with knowledge processing abilities include several types of historic automata (al-Jazari 1205). Akman (1976) re-introduced Turks to the works of al-Jazari.

The archtypical example to *variably-wired tools or machines with knowledge processing abilities* is the Jacquard loom. The machine under the control of punched 'cards' could weave different patterns. Furthermore, the punched cards used by the Jacquard looms were the inspiration for the punched card knowledge processing machines: first the unit records and afterwards the punched card computers.

Stored-program tools or machines with knowledge processing abilities are the most important applications. They can be computer-embedded machines (CEM) or computer-embedded systems (CES). When the emphasis is on the computer, they can be referred to as embedded systems. CEMs or CESs are the essence of intelligent machines (Kurtzweil 1990) and can automate functions at different degree of sophistication:

1. In some systems, parameters and some other values can be set based on some automatically measured/computed values (e.g., in a camera, to set film speed, to measure and set distance, to measure light and to set the lens aperture and shutter speed, and to automatically fire the flash); another example is reprogrammable pacemakers that existed since a long time as a forerunner of implantable computers.

2. Intelligent cars, utilities, and buildings can have several functions performed by the embedded computers.

3. Optimizing systems such as a tracking missile can perform its mission with a high degree of effectiveness.

4. Knowledge-based, rule-based, or agent-directed systems can benefit from their advanced knowledge processing abilities.

5. Simulative systems can evaluate, via embedded simulation ability, the outcome of different alternatives and can automatically select most desirable one.

3. Energy Transducers, Programs, and Their Synergy

Perceiving the similarities of energy transducers and computer programs may offer new vistas and may facilitate the comprehension of their synergy (Ören 1990).

3.1. Energy Transducers

An energy transducer, commonly used in engineering, is a device which can perform three possible types of function. In all the functions, the input to an energy transducer is energy. An energy transducer can perform one of the following functions:

1. It can convert one type of energy into another type. For example, a piezoelectric cristal can convert pressure into electric current.

2. It can provide knowledge about the input energy. Some measuring devices use this feature. For example, a tire gauge can accept as input pressure in the tire to provide a read-out of the pressure.

3. It can process an input signal based on a secondary signal to perform one of the above mentioned two functions (to either convert the signal into another one with different characteristics or to provide knowledge about it).

3.2. Knowledge Transducers

Similarly, a computer program-as a knowledge transducer-can accept (different types of) knowledge as input and can perform one of the three functions:

1. It can convert the input knowledge into output knowledge. For example, a translator can transform a program written in a high level language into a program written in a lower level language.

2. It can provide knowledge about the input knowledge. For example, a compiler can generate data dictionary of an input program.

3. Based on some knowledge, another body of knowledge can be processed. For example, based on a query, a data base can be searched to provide either the documents and/or knowledge about them.

A cognitive program (an AI program) may have three types of input and two types of output (Ören 1990): The inputs can be forced input, actively perceived input, and endogenous input.

1. *Forced input* is customarily called "input" in conventional programming.

2. *Actively perceived input* is basically knowledge actively perceived, filtered, and accepted as input by the knowledge processing system.

3. Endogenous input is generated by the knowledge processing system and is accepted as stimulus or input to trigger knowledge processing. Examples of endogenous input can be based on anticipation of future, generation of questions, and formulation of hypotheses.

Outputs of a knowledge transducer can be primary and auxiliary outputs.

1. *Primary output* is generated based on the knowledge processing goal of the system.

2. *Auxiliary outputs* can provide guidance, advice, explanation, and certification.

3.3. Synergy of Energy and Knowledge Transducers

The second characteristic of energy transducers make them a good source of data. Therefore, energy transducers in sensors, can provide data to knowledge processing systems. Sensor fusion refers to multi-channel input when there are more then one type of sensor. Especially, systems/machines with knowledge processing abilities can have on-line input from their environment.

4. What's Next?

Computers are still very young. For example, Konrad Zuse's computer dates back 1936 (Dorsch 1989). We can expect to have advancements on many fronts. An excellent book prepared on the 50th anniversary of the field of computing provides a review of past achievements and future projections (Denning and Metcalfe 1997). Some major possibilities for future achievements that I would like to emphasize are:

1. Software *agents* provide a solid computational paradigm to implement software assistants working (quasi-) autonomously and having perception abilities to observe the existence or lack of some characteristics or events and other abilities to affect their environments. Furthermore, they can process goals and can perform goal-directed knowledge processing. Types of agents and agent-related terms are given in appendices 1 and 2, respectively, to show the many aspects of the field. To appreciate agents, one can consider different software engineering paradigms: batch processing software, interactive software, event-based software, and agent-based software. In agent-based software, agents can trigger events to perform knowledge processing on behalf of the user.

2. *Mobile agents* and distributed computing extend the concept of computational platform to whole or part of the net on intranets and on the Internet.

3. System theories provide strong backgrounds for cognitive, i.e., intelligent, computerization. For example, systems with *understanding* abilities (Ören 2000a), systems with *learning* abilities (Osherson et al. 1986), systems with adaptation abilities, and systems with *anticipation* abilities (Dubois 2000) would provide bases for cognitive knowledge processing. Agents are natural candidates for the implementation of systems with cognitive abilities.

4. *Cooperation* is becoming an important paradigm for both civilian and military applications. Holonic systems are excellent candidates to conceive, model, control, and manage dynamically organizing cooperative systems. A *holonic* system is composed of autonomous entities (called *holons*) that can deliberately reduce their autonomy, when need arise, to collectively achieve a goal. A *holonic* agent is a multi-agent system where each agent (called a holon) acts with deliberately reduced autonomy to assure harmony in its cooperation in order to collectively achieve a common goal.

5. *Agent-directed simulation* is very promising and consists of agent simulation, agent-based simulation, and agent-supported simulation. *Agent simulation* allows simulation of natural or engineered entities with cognitive abilities. Therefore, agent simulation is very appropriate for the simulation of intelligent entities. *Agent-based simulation* is use of agent technology to generate behavior of models. (Parallels with AI-based simulation are knowledge-based simulation, qualitative simulation, and rule-based simulation.) *Agent-supported simulation* is use of agent technology to support simulation activities; they comprise front-end and back-end activities of a modelling and simulation environment, agent-supported validation and verification, as well as agent-supported program generation, program integration (as it would be the case in the formation of federations using HLA), and program understanding for documentation and/or maintenance purposes.

6. *Holonic agent simulation or holon simulation*, in short, is an important type of agent simulation where agents represent holons. Some military application include use of simulation for preparedness for conflict management including conflict avoidance, conflict resolution, and conflict detererence. Civilian applications include modelling and simulation of cooperation of different business entities.

Acknowledgement

An early version of this article was presented at the opening session of the 15th annual ISCIS (International Symposium on Computer and Information Sciences), held in Istanbul, Turkey, October 8-10, 2000 (Ören 2000b). I would like to express my thanks and appreciation to Prof. Dr. Y. Karsligil, the Program Chair of the Symposium who, by his kind invitation, motivated me to concentrate on the article.

References

[1] Akman, T. (1976). First Turkish Cybernetician: Eb-Ül-İz (In Turkish: İlk Türk Sibernetik Bilgini Eb-Ül-İz). Bilim ve Teknik, Nb: 103.

[2] Al-Jazari, Ibn al-Razzaz (1205). The Book of Knowledge of Ingenious Mechanical Devices (translated and annotated by D.R. Hill). Boston: Dorrecht, 1974.

[3] Bott, G. (1983). Treasures of Astronomy. German National Museum, Nurnberg, Germany.

[4] Corel (1995). World's Geatest Classic Books. CD-ROM. Ottawa, ON, Canada: Corel, http://www.corel.com

[5] de Beauclair, W. (1968). Rechnen mit Machinen. Braunschweig: Friedrich Vieweg & Sohn.

[6] Denning, P.J. and R.M. Metcalfe (1997). Beyond Calculation-The Next Fifty Years of Computing. New York: Copernicus/Springer-Verlag.

[7] Dorsch, H. (1989). Der 1. Computer. Berlin Science and Technology Museum, Berlin, Germany.

[8] Dubois, D.M. (ed.) 2000. Computing Anticipatory Systems. Proc. of CASYS'99-3rd International Conference, Liège, Belgium, August 9-14, 1999. Publ. by AIP (American Institute of Physics) Conference Proceedings 517.

[9] EB. Encyclopedia Britannica, http://www.eb.com

[10] Fröschl, K., S. Mattl, and H. Werthner. (1993). Symbol verarbeitende Machinen-Eine Archeologie. Verein Museum Arbeitswelt, Steyr, Austria.

[11] Hachette (2000). http://www.encyclopedies.hachette-multimedia.fr

[12] Kay, A. (1984). Computer Software. Quoted on the Cover of ACM Computing Reviews 25(10).

[13] Kurtzweil, R. (1990). The Age of Intelligent Machines. Cambridge, MA: MIT Press.

[14] Nasr, S. (1976). Islamic Science-An Illustrated Study. London: Festival.

[15] Ören, T.I. (1990). A Paradigm for Artificial Intelligence in Software Engineering. In: Advances in Artificial Intelligence in Software Engineering-Vol. 1, T.I. Ören (ed.), Greenwich, CT: JAI Press, pp. 1-55.

[16] Ören, T.I. (2000a - Invited Paper). Understanding: A Taxonomy and Performance Factors. In: D.Thiel (ed.) Proc. of FOODSIM'2000, June 26-27, 2000, Nantes, France. San Diego, CA: SCSI, pp. 3-10.

[17] Ören, T.I. (2000b - Invited Opening Paper). Advances in Computer and Information Sciences. Proc. of 15th annual ISCIS (International Symposium on Computer and Information Sciences), held in Istanbul, Turkey, October 11-13, 2000 (Y. Karsligil et al., eds.), pp. XIII-XVII.

[18] Osherson, D.N., M. Stob, and S. Weinstein (1986). Systems that Learn. Cambridge, MA: MIT Press.

[19] Price, D. De S. (1984). "A History of Calculating Machines." IEEE Micro 4(1), 22-52.

[20] Svoboda, A. (1965). Computing Mechanisms and Linkages. New York: Dover.

[21] Universalis (2000). http://www.universalis.fr

APPENDIX 1 - Types of Agents

adaptive agent
animated agent
antagonistic agent
anticipatory agent
application agent
application suite agent
authorized agent
autistic agent
autodidactic agent
autonomous agent
autoprogrammable agent
believable agent
bot
broker
client agent
cognitive agent
co-located agent
communication agent
competent agent
competitive agent
complete agent
computational agent
computer interface agent
computer-controlled bot
contractee agent
contractor agent
conventional agent
conventional software agent
cookie
co-operating agent
co-ordinator agent
coupled multi-agents
deliberative agent
desktop agent
diagnosis agent
digital agent
dispatched agent
dispatched mobile agent
distant agent
distinguished agent
domain-specific agent
emotional agent
endomorphic agent
errant agent
ethical agent
fixed agent
fuzzy agent
global agent
goal-directed agent
goal-oriented agent
holonic agent
independent agent
individual agent
information agent
information filtering agent
information gathering agent
information spider
intelligent agent
interface agent
Internet agent
itinerant agent
knowledge-based agent
learning agent
local agent
long-lived agent
loosely coupled multi-agents
mail agent
message transfer agent
messaging agent
mobile agent
mobile agent
model-based agent
multiple agent
multiple mobile agent
network agent
neural net agent
notification agent
offline delivery agent
pedagogical agent
persistent cookie
personal agent
personal digital agent
personal software agent
pro-active agent
purposeful agent
rational agent
reactive agent
reliable agent
remote agent
resident agent
retrieval agent
root agent
rule-based agent
scriptable agent
search agent
self-motivated agent
self-replicating agent
semi-autonomous agent
service agent
sociable agent
software agent
spider
stationary agent
system latency agent
task-specific agent
teachable agent
temporary cookie
tightly coupled multi-agents
tracking cookie
transient agent
transportable information agent
trusted agent
trustworthy agent
unauthorized agent
understanding agent
uniform resource agent
user agent
user interface agent
user-programmed agent
vivid agent
wanderer
Web search agent
Web site agent

APPENDIX 2 - List of Agent-related Terms

absolute autonomy
agency
agent architecture
agent autonomy
agent behavior
agent class
agent cloning
agent code
agent communication
agent communication language
agent completeness
agent efficiency
agent implementation
agent interactivity
agent language
agent security
agent software
agent system
agent user
agent/place interface
agent/user interface
agent-based
agent-based holon
agent-based software
agent-based software engineering
agent-based software provider
agent-directed
agent-enabled
agent-enabled feature
agent-oriented
agent-oriented CASE tool
agent-oriented problem solving
agent-oriented programming
agent-oriented requirements engineering
agent-oriented tool
animated agent technology
anticookie software
community
cookie management
design autonomy
design-system for multi-agent
ethics for agents
execution autonomy
holonic agent simulation
inter-agent communication
inter-agent communication language
inter-agent knowledge processing
interface autonomy
message-based agent communication
mobile agent paradigm
mobile code
mobile object
multi-agent architecture
multi-agent design-system
multi-agent learning system
multi-agent learning technique
multiagent software
multi-agent system
multi-agent understanding system
remote procedure
remote programming
social autonomy