Ören, T.I. (2001 - Invited Paper). Software Agents for Experimental Design in Advanced Simulation Environments. In: S.M. Ermakov, Yu.N. Kashtanov, and V. Melas (eds.) Proc. of the 4th St. Petersburg Workshop on Simulation, June 18-23, 2001, pp. 89-95.

Software Agents for Experimental Design in Advanced Simulation Environments

Tuncer Ören, Ph.D., Professor Emeritus
School of Information Technologies
University of Ottawa, Ottawa, Ontario, Canada
oren@site.uottawa.ca

ABSTRACT

In this methodology-oriented article, the use of agent-supported experimental design functionalities in advanced simulation environments is advocated. As a background, references are given on experimental design for simulation and simulation environments to support experimental design; input characteristics of software agents are elaborated on to provide a basis for advanced autonomous systems.

1. INTRODUCTION

Simulation is goal directed experimentation with dynamic models. There are several types of simulation (Ören, 1987). As it is expressed in the literature: *"An unplanned, hit-or-miss course of experimentation with a simulation model can often be frustrating, inefficient, and ultimately unhelpful. On the other hand carefully planned simulation studies can yield valuable information without an undue amount of computational effort or (more imortantly) your time."* (Kelton, 2000). Software agents can assist simulationists in performing design of simulation experiments, in simulation-based optimization, and in analyzing simulation results.

This article aims to explore the benefits of using software agents to support experimental design functionalities in advanced simulation environments. For this purpose due to space limitation, first some references are given on experimental design for simulation and on simulation environments to support experimental design; afterwards, some characteristics of software agents are elaborated on to provide a basis for advanced autonomous systems.

2. EXPERIMENTAL DESIGN FOR SIMULATION

"One of the principal goals of *experimental design* is to estimate how changes in input factors affect the results, or *responses*, of the experiment." (Kelton, 2000, p. 35). Design of experiments is an area known for a long time (Connor and Young, 1961; Box et al., 1978). Several aspects of design of experiments are still investigated (Sanchez, 1994, 2000; Montgomery, 1997).

Experimental design for simulation is a well developed area of knowledge –albeit not very straigthforward for the novice simulationist. Several classical references exist on experimental design for simulation. For example, chapter 12 of Law and Kelton (2000 – the first edition was published in 1982) and Kleijnen (1987). Some other references are Mauro (1982), Biles (1984), and Shaikh et al. (1986). Due to its importance, the topic is still being investigated (Hood and Welch, 1992; Swain and Farrington, 1994; Özdemirel et. al., 1996; Kleijnen, 1998; and Kelton (1999, 2000).

A field related to experimental design for simulation is *simulation-based optimization* which is a very important topic in discrete event simulation (Law and McComas, 2000). In simulation-based optimization, heuristic optimization techniques such as genetic algorithms, simulated annealing, and tabu search are used to have simulation as a basis for optimization. A good survey on simulation-based optimization is given by Swisher et al. (2000). It contains a good list of references including other good surveys. Another reference on the topic is Neddermeijer et al. (2000).

3. SIMULATION ENVIRONMENTS TO SUPPORT EXPERIMENTAL DESIGN

Some references on experimental design and analysis for simulation are Taylor and Hurrion (1988), Park and Mellichamp (1990), Ören (1993), and Tao and Nelson (1994). Experimental design facilities are not yet fully explored in simulation environments. Opprtunities still exist for academic research and industrial development. Recently, two optimization packages are reported for simulation-based optimization (Law and McComas, 2000). It was also asserted that: "Simulation-based optimization is just in its infancy. However, it appears that it will have a considerable impact on the practice of simulation in the future, particularly when computers become significantly faster." (Law and McComas, 2000, p. 49). It appears that currently the time is ripe especially for teams of researchers/implementers with strong background in advanced software environments, software agents, and statistics to implement advanced features in simulation environments.

4. SOFTWARE AGENTS

A reference on the application of artificial intelligence in statistics is Gale (1986). Application of artificial intelligence in software engineering is relatively old (Ören, 1990) and is still a very promising area. Similarly, application of artificial intelligence in simulation is still very important (Ören, 1994). A relatively recent development in artificial intelligence applications is software agents (Huhns and Singh, 1998; Murch and Johnson, 1999), in general and agent-directed simulation, in particular (Ören 2001a, b).

S*oftware agents* are software modules that can work as assistants to users or to other agents. They have cognitive abilities such as autonomy (usually limited for the purpose of not creating havoc in software environments), perception, goal processing and goal-directed knowledge processing; and they can affect their knowledge environment – directly if their environment is purely software, or indirectly through actuators.

Agent-supported simulation is use of agent technology to support simulation activities which 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. Agent-supported simulation is one of the aspects of agent-directed simulation. Two other aspects are agent simulation and agent-based simulation (Ören, 2001a, b).

In considering use of software agents (in any aspect of agent-directed simulation), one should take into account an important characteristic of them that is the ability to accept a variety of inputs (some of these characteristics are not yet fully explored; hence promising possibilities for future research). As seen in Table 1, input to a software module, here to a software agent, can be generated externally or internally.

Externally generated inputs are of two types: passively accepted and actively perceived inputs. *Passively accepted inputs* consist of mostly conventional inputs which can be accepted via coupling, argument passing, message passing, or accessing knowledge in a common area (such as a blackbord). Their nature can be data, facts, forced events, sensations (i.e., input coming from sensors, usually in analog form), and goal(s) imposed to the agents. *Active perception of exogenous inputs* includes *perception*, i.e., interpreted sensory data (which require decoding, selection (filtering), recognition, and regulation). Second type of active perception of exogenous inputs consists of *perceived goals*. A third possibility is evaluated input where either inputs and/or their sources can be scrutinized for their acceptability, reliability, and credibility.

Endogenous inputs, or *internally generated inputs* are also of two types: *Actively perceived endogenous inputs* are based on introspection. They consist of (observed or monitored) perceived internal facts and events as well as lack of them. The second category of endogenous inputs is the *generated endogenous inputs*. They consist of anticipated future data and events (in anticipatory systems) as well as internally generated questions, hypotheses, and goals.

5. ON AGENT-SUPPORTED EXPERIMENTAL DESIGN IN SIMULATION ENVIRONMENTS

Some references of the use of expert systems for the design of experiments are Spiegel, and Lavallee (1988) and Nachtscheim et al. (1996). A reference on the agent-supported experimentation is Wilson et al. (2000). Due to their nature, software agents are good candidates for agent-support in simulation experimentation such as design of experiments, simulation-based optimization, and analysis of simulation results.

An old adage states that: “For a person who holds a hammer in his hand everything look like a nail.” Similarly, having a technology –albeit and advanced one– namely software agents, should not oblige us to convert all computerization applications in it. However, not fully exploring the potentials offered may not be a good idea either.

Table 1. Types of Inputs for Software Modules (including Agents)

Source of input	**Mode of input**	**Type of input**
Exogenous input (externally generated input)	**Passive acceptance** of exogenous input (imposed or forced input)	**Type of access to input**: coupling, argument passing, knowledge in a common area, message passing. **Nature of input**: - ***Data*** *(facts)* - *Forced* ***Events*** - ***Sensation*** (converted sensory data: from analog to digital; single or multi sensor: sensor fusion) - ***External goals*** *(imposed goals)*
	Active perception of exogenous input (perceived input)	- ***Perception*** (interpreted, sensory data and detected events) -- includes: decoding, selection (filtering), recognition, regulation - ***Perceived goals*** - ***Evaluated inputs*** -- evaluation of inputs (acceptability) -- evaluation of source of inputs (reliability, credibility)
Endogenous input (internally generated input)	**Active perception** of endogenous input	- ***Introspection*** (perceived internal facts, events; or realization of lack of them)
	Generation of endogenous input	- ***Anticipated facts*** *and/or* ***events*** (anticipatory systems) - ***Internally generated questions*** - ***Internally generated hypotheses*** by: -- Expectation-driven reasoning (Forward reasoning) (Bottom-up reasoning) (Data-driven reasoning) -- Model-driven reasoning - ***Internal goals*** *(internally generated goals)*

ACKNOWLEDGMENT

I wish to express my appreciation to Prof. V.B. Melas for having extended me an invitation to participate in the Workshop and to Prof. Dr. Jack P.C. Kleijnen who gratiously brought to my attention the four references by the following researchers: Gale, W.A. (ed.) (1986); Nachtscheim, C.J. et. al. (1996); Özdemirel, N.E. et al. (1996); and Tao, Y-H. and B.L. Nelson (1994).

REFERENCES

Biles, E.E. (1984). Design of simulation experiments. Proc. of the 1984 Winter Simulation Conf., 99-104.

Box, G.E.P., W.G. Hunter, and J.S. Hunter (1978). Statistics for experimenters: An introduction to design, data analysis, and model building. New York: John Wiley.

Connor, W.S. and S. Young (1961). Fractional factorial designs for experiments with factors two or three levels. U.S. Department of Commerce, National Bureau of Standards, Applied Mathematics Series 58.

Gale, W.A. (ed.) (1986). Artificial intelligence and statistics. Massachusets: Addison-Wesley.

Hood, S.J. and P.D. Welch (1992). Experimental design in simulation with examples from semiconductor manufacturing. *Proc. of the 1992 Winter Simulation Conf.*, 255-263.

Huhns, M.N. and M.P. Singh (eds.) (1998). Readings in agents. San Francisco, CA: Morgan Kaufman.

Kelton, W.D. (1999). Designing simulation experiments. *Proc. of the 1999 Winter Simulation Conf.*, 33-38.

Kelton, W.D. (2000). Experimental design for simulation. *Proc. of the 2000 Winter Simulation Conf.*, 32-37.

Kleijnen, J.P.C. (1987). Statistical tools for simulation practitioners, New York: Marcel Dekker.

Kleijnen, J.P.C. (1998). Experimental design for sensitivity analysis, optimization, and validation of simulation models. *Handbook of simulation*, J. Banks (ed.), New York: John Wiley, 173-223.

Law, A.M. and M.G. McComas (2000). Simulation-based optimization. *Proc. of the 2000 Winter Simulation Conf.*, 46-49.

Law, A.M. and W.D. Kelton (2000). Simulation modeling and analysis. 4th ed. New York: John Wiley.

Mauro, C.A. and D.E. Smith (1982). Factor screening in computer simulation. *Simulation*, 49-54

Montgomery, D.C. (1997). Design and analysis of experiments. 4th ed. New York: John Wiley.

Murch, R. and T. Johnson (1999). Intelligent software agents. Upper Saddle River, NJ: Prentice Hall. http://www.phptr.com

Nachtscheim, C.J., P.E. Johnson, K.D. Kotnour, R.K. Meyer, and I.A. Zualkern (1996). Expert systems for the design of experiments. *Statistical Design and Analysis of Industrial Experiments*, S. Gosh (ed.). New York: Marcel Dekker.

Neddermeijer, H.G., G.J. van Oortmarssen, N. Piersma, and R. Dekker (2000). A framework for response surface methodology for simulation optimization. *Proc. of the 2000 Winter Simulation Conf.*, 129-136.

Ören, T.I. (1987). Simulation: Taxonomy. *Systems and Control Encyclopedia*, M.G. Singh (ed.), Pergamon Press, Oxford, England, 4411-4414.

Ören, T.I. (1990). A paradigm for artificial intelligence in software engineering. *Advances in Artificial Intelligence in Software Engineering*, T.I. Ören (ed.), JAI Press Inc., Greenwich, Connecticut, 1-55.

Ören, T.I. (1993). Three simulation experimentation environments: SimAd, SimGest, and E/Slam. *Proc. of the European Simulation Symposium*, 627-632.

Ören, T.I. (1994). Artificial intelligence and simulation. *Annals of Operations Research*, vol. 53. 287-319.

Ören, T.I. (2001a). Advances in computer and information sciences: From abacus to holonic agents. Elektrik (Special Issue on Artificial Intelligence, published by TUBITAK). 9:1, 63-70.

Ören, T.I. (2001b). Impact of data on simulation: From early practices to federated and agent-directed simulation. Proc. of the EUROSIM 2001.

Özdemirel, N.E., G.Y. Yurttas, and G. Köksal (1996). Computer-aided planning and design of manufacturing experiments, *Simulation*, 67:3 (Sept.), 171-191.

Park, Y.H. and J.M. Mellichamp (1990). A statistical expert system for simulation analysis. *Proc. of the 1990 Summer Computer Simulation Conf.,* 611-616.

Sanchez, S.M. (1994). Experimental designs for system assessment when noise factors are correlated. *Proc. of the 1994 Winter Simulation Conf.,* 290-296.

Sanchez, S.M. (2000). Robust design: Seeking the best of all possible worlds. *Proc. of the 2000 Winter Simulation Conf.*, 69-76.

Shaikh, M.A., P.J. Knoke, and E.L. Althouse (1986). Testing communication network routing algorithms via statistical design of simulation experiments. *Simulation in the Factory of the Future and Simulation in Traffic Control*, H. Muller, R. Reddy, and S. Takaba (eds.), *Proc. of the European Simulation Multiconference*, 199-203.

Spiegel, J.R. and D.B. Lavallee (1988). Using an expert system to drive a simulation experiment. *AI Papers, 1988*, R.J. Uttamsingh (ed.). *Simulation Series*, 20:1, 108-112, San Diego, CA: SCS.

Swain, J.J. and P.A. Farrington (1994). Designing simulation experiments for evaluating manufacturing systems. *Proc. of the 1994 Winter Simulation Conf.*, 69-76.

Swisher, J.R., P.D. Heyden, S.H. Jacobso and L.W. Schruben (2000). A survey of simulation optimization techniques and procedures. *Proc. of the 2000 Winter Simulation Conf.*, 119-128.

Tao, Y-H. and B.L. Nelson (1994). A framework for computer-assisted simulation experiment design and analysis. *Proc. of the 1994 Winter Simulation Conf.,* 119-128.

Taylor, R.P. and R.D. Hurrion (1988). An expert advisor for simulation experimental design and analysis. *Proc. of the SCS Multiconference on Artificial Intelligence and Simulation: The diversity of applications*, T. Henson (ed.), 238-244.

Wilson, L.F., D. Burroughs, J. Sucharitaves, and A. Kumar. (2000). An agent-based framework for linking distributed simulations. *Proc. of the 2000 Winter Simulation Conf.,* 1713-1721.