The Influence of Decision Time on Performance in Use of a Business Simulator

Andreas Größler
Industrieseminar der Universität Mannheim
D - 68131 Mannheim, Germany
Phone: (+49 621) 292-31 40
Fax: (+49 621) 292-52 59
E-mail: agroe@is.bwl.uni-mannheim.de
http://iswww.bwl.uni-mannheim.de/agroe/

Abstract

Most business simulators in use today are event-driven. That means, the natural proceeding of time cannot be experienced with these simulation tools. A time-driven business simulator is presented. An experiment is described which should clarify the influence of time pressure on users' performance.

With the help of business simulators, users should be able to gain insights into the dynamic behavior of real world problems. Although their validity has not been proven yet, there is some evidence that these tools promote learning. One precondition for a learning transfer to take place is a homomorphic relation between the real world domain and the formal model underlying these simulation tools. A characteristic of reality which has no correspondence in the artificial world of most business simulators, however, is the proceeding of time.

For some purposes a self-proceeding business simulator can be a better choice than a regular simulator. After a certain time such a self-proceeding simulator finalizes the current simulation period by calculating its results and switches to the next period in time, even if the user has not finished all of his or her input. This calculation could be based on the last valid input, on some kind of mean values, or on a generic pattern of input.

The effects of restricted decision times on game performance have not been explored so far. Therefore, an experiment is conducted that systematically varies decision time for subjects using a business simulator. The main hypotheses is that subjects' performance decreases when decision time is shorter.

The paper discusses the relationship between business simulators and time. Various areas of the usage of self-proceeding simulators are identified. Different technical aspects of the implementation of a self-proceeding feature in business simulators are described. Finally, results of the above mentioned experiment are presented.

The Paradoxical Handling of Time in Business Simulators

The development and simulation of formal models are widely supposed to have a positive effect on learning about nearly any social system (e. g., Milling 1991, p. 21, de Geus 1988, p. 73, Morecroft 1994, p. 4). To give people an easier and more user friendly access to formal models, pre-build simulations with graphical user interfaces are used (e. g., Bakken, Gould and Kim 1994). Thus, the user does not have to go through the difficult and time consuming process of model development himself (compare the seven levels of system dynamics competence formulated by Meadows 1989, p. 636). With the help of these business simulators (for a definition of this and related terms, see Maier and Größler 1998), users should be able to gain insights into the dynamic structures of a problem. Although their validity has not been proven yet, there are some hints that these programs promote learning (Milling 1995, p. 106).

One precondition for a learning transfer to take place is a homomorphic relation between the real world and the formal model that is used in the simulation tool. In particular, models based on the system dynamics approach are supposed to have a great similarity to the real system (Forrester 1961, pp. 54–55). In this case, the business simulator can become a transitional object as mentioned by Papert (1980) that makes a learning transfer possible.

A characteristic of reality which has no correspondence in the artificial world of most business simulators is the proceeding of time. This sounds paradoxically: these tools, which are built to improve knowledge about dynamic, time dependent aspects of reality, do only show dynamics when asked to go ahead. Only when the user has provided all necessary input, the simulation calculates the results for the next period in time. These kind of simulations are called event-driven: when the user has finished data input he or she presses a button and generates an event which causes the simulation to proceed. Kluwe (1993, p. 404) calls this a "questionable feature, especially with regard to the ecological validity claimed for such systems."

Event- vs. Clock-driven Simulators

Of course, the above mentioned issue is only a paradox at first sight. One has to keep in mind the aims of business simulators: users should gain insights in the dynamic structure of the domain. He or she is supposed to learn. This often cannot take place in real-time and in fixed time intervals. Time has to be slowed down and made "flexible". Users need to have their time to analyze the situation, make decisions, etc. This streching of time is as important as the acceleration of time (decision-reaction cycles) which is solely discussed by most authors. (But compare Kim 1989, p. 327, who speaks of "compression and expansion of time", and Kim and Senge 1994, p. 286.) Learning Environments have to provide „opportunities for reflection“ (Senge and Sterman 1992, p. 147), in order to not let users fall victim to the „video game“ syndrome. This aspect differentiates business simulators and flight simulators—„Management Flight Simulator“ therefore is perhaps not an appropriate term for one-person business simulation games that have their focus on the induction of learning (Maier and Größler 1998). See Figure 1 for a symbolic depiction of different time spans in reality and in business simulators (following Craft's [1967, p. 274] definition of terms).

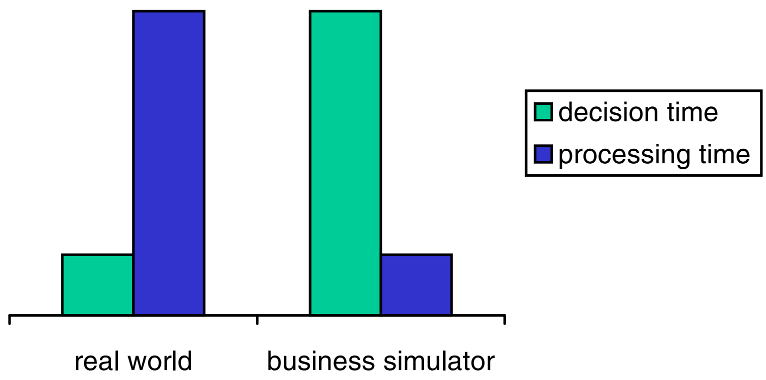

Figure 1: Decision and processing times in real and virtual environments (symbolic illustration)

In reality, time is an independent, continuous variable, which certainly cannot be influenced by the user. Thus, the fidelity of business simulators would be higher if they used a kind of real-time mechanism. Note that this can be implemented in two ways (Buchner 1995, p. 55):

Firstly, by introducing time pressure. Users have to decide and complete a given tasks within a certain temporal limit which is externally set by the game facilitator. This method is often used by experimenters to make results of different subjects comparable or if competitional aspects of simulation gaming prevail. Also, the introduction of time pressure seems "to keep participants in their role" (Tansey and Unwin 1969, p. 5). Because (1) a certain time limitation is set to nearly all uses of business simulators out of practical reasons and (2) users are still able to divide the time available in arbitrary decision intervals, this method will not be considered as a clock-driven or a real-time simulation in the remaining part of this paper.

Secondly, the system changes autonomously its state. Here, time pressure is not externally but inherent in the task. This offers interesting perspectives on possible usage, task difficulty, and learning chances of business simulators. One can call these type of simulation systems clock-driven or self-proceeding; sometimes they are just called „dynamic decision tasks" (Brehmer and Allard 1991, p. 319). In this definition, autonomous state changes and real-time decision making become an important characteristic of a simulator per se. In the rest of this paper, only this kind of autonomously changing simulators are considered.

Figure 2 helps to clarify terms. In all three diagrams, the upper scale depicts time within the simulation game. The lower scale represents time in reality. ΔS_j represents the time interval from one simulator state to the next; ΔT_j stands for the time of one decision/processing cycle.[1] In all three cases depicted, simulated time intervals remain constant. That is, $\Delta S_k = \Delta S_l = const, \ \forall\ k, l \in \{1,...,n\}$. Nevertheless, changing simulation intervals for simulators are conceivable. This case is not depicted here out of simplification purposes. The occurrence of changing simulation intervals does not affect the definition of terms provided here.

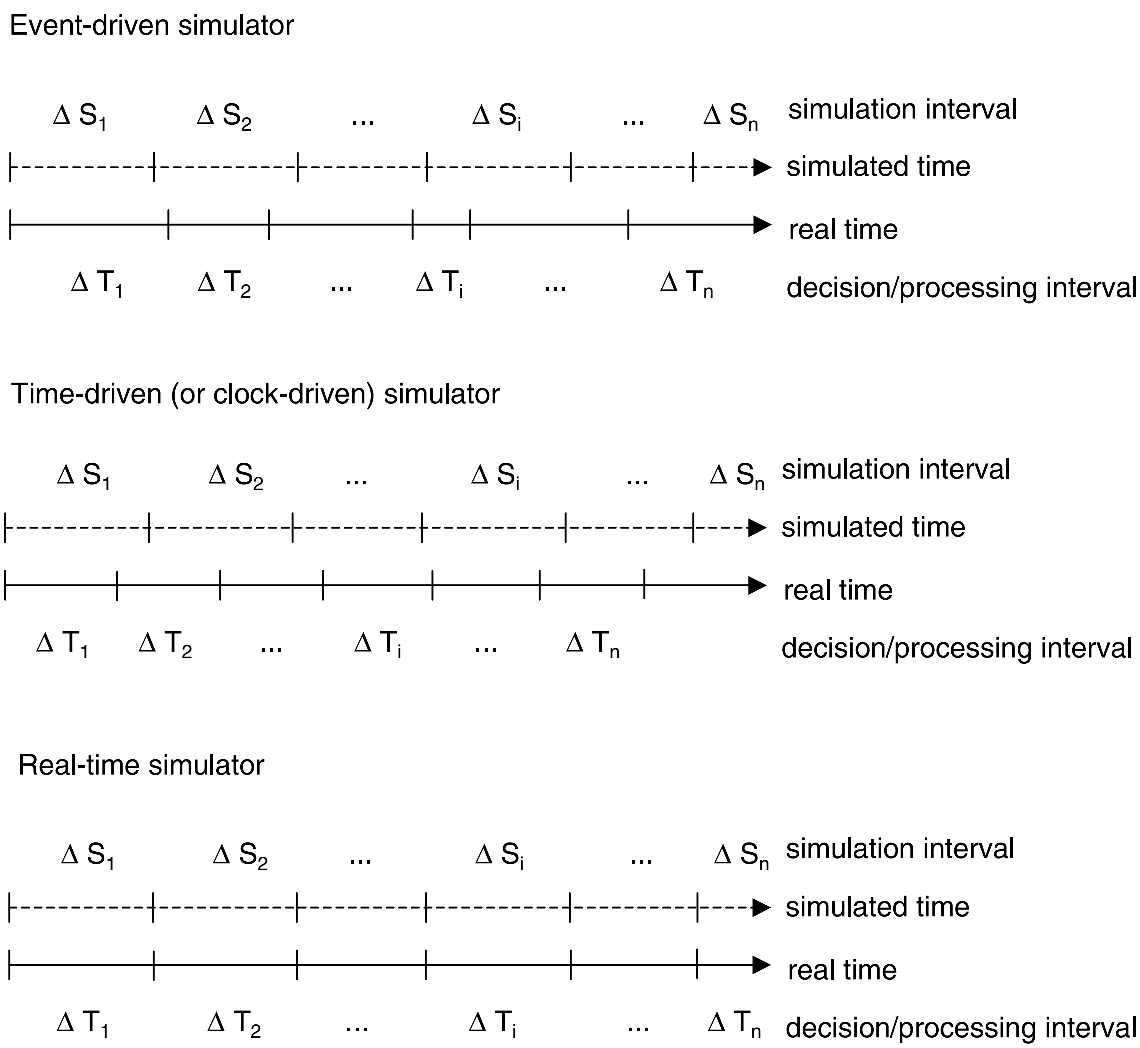

Figure 2: Different types of simulators (inspired by Niemeyer and Reidelhuber 1979, p. 123)

In the case of event-driven simulators, decision time is variable. One interval is finalized, whenever the user decides to finish it. Regularly, $\Delta T_k \neq \Delta T_l,\ \forall\, k, l \in \{1,\dots,n\}$ and $\Delta S_m \neq \Delta T_m,\ \forall\, m \in \{1,\dots,n\}$. Although exceptions to these equations are conceivable they occur by mere chance.

For time-driven simulators, still simulation intervals and decision/processing times are unequal, that is $\Delta S_m \neq \Delta T_m,\ \forall\, m \in \{1,\dots,n\}$. However, the duration of decision/processing intervals is now the same for all intervals and so $\Delta T_k = \Delta T_l = const,\ \forall\, k, l \in \{1,\dots,n\}$ because an internal clock finalizes these intervals, not the users' input. Like simulated time, decision/processing intervals could also change according to a predefined rule. In literature, however, no example of such an application could be found.

In real-time simulators, the length of simulation intervals and decision/processing intervals is equal as well as all decision/processing intervals have the same duration, that is $\Delta S_m = \Delta T_m = const,\ \forall\, m \in \{1,\dots,n\}$ and $\Delta T_k = \Delta T_l = const,\ \forall\, k, l \in \{1,\dots,n\}$. Again, changing interval lengths are conceivable but do regularly not occur.

If both, ΔS_m and ΔT_m converge against zero, $\forall\, m \in \{1,\dots,n\}$ (no mater if $\Delta S_m = \Delta T_m$ or not), the familiar concept of decision-simulation periods in business simulators becomes an unnecessary artefact (even though, from a technical point of view, one always has time steps in digital computing): continuous simulations can run independently from discrete decision-simulation periods. Decisions show effects based on current data. Like in reality, the state of the simulation changes autonomously. These changes in the environment require reactions of the users and have to be considered when crafting a strategy. Thus, „users have to anticipate the system's inherent changes due to the Eigendynamik" (Funke 1995, p. 258). See also the Figure 3 for another presentation of the different approaches of simulation tools.

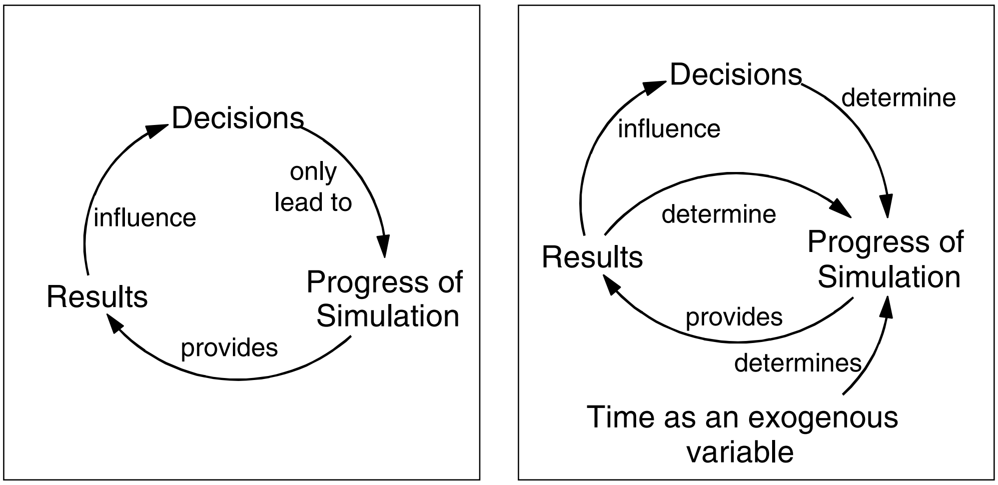

Figure 3: Progress of simulation in event-driven versus clock-driven business simulators

Because real-time simulators (for instance, flight simulators) do neither provide a compression nor a stretching of time, their teaching value seems rather limited. Thus, unless not explicitly stated, they are not considered any more in the rest of this paper.

Effectiveness and Areas of Use of Clock-Driven Simulation Games

In the literature, some statements about the effects of a self-proceeding mechanism can be found. The task of running a clock-driven business simulation game successfully becomes more difficult because mental processes involved have to terminate earlier than in event-driven simulations. Furthermore, autonomous state changes are probably different from ones which are initiated by the user. The user has to distinguish between those system states caused by his or her decisions and those that are caused by autonomous changes of the simulation model (Brehmer 1995, p. 104). And in fact, control performance is severely affected when a simulation system autonomously changes its state as a function of time rather than as a result of user input. (See Funke 1993, pp. 322–324, who also reports that knowledge acquisition is not affected.) However, there seems to be no qualitative difference on the (counterintuitive) effects of delays when an event-driven or a clock-driven simulation is used (Brehmer 1995, p. 125). Nevertheless, clock-driven decision making is in many cases "inherently stressful" (Brehmer 1990,p. 263). A user working with such a simulation system needs (1) the ability to deal with this stress, (2) the ability to act process-oriented, and must (3) consider various time scales, for instance, time in reality and time in simulation (Brehmer 1992, p. 213).

Thus, although decision making in fixed time intervals is more congruent to reality it is probably not the best choice for every learning objective. One has to imagine situations where not learning about the real world domain modeled is the main goal of a business simulator. Therefore, self-proceeding simulations seem to be an alternative to “conventional“ business simulation games in the following situations:

- For personnel selection one wants to know about the subject’s behavior under a close to reality stress situation.
- In group development training the primary goal is to induce group dynamic processes which are more likely under external pressure.
- If domain independent behavior of dynamic systems is a learning objective time-driven simulators might represent this more accurately.
- Students maybe want to prove their newly acquired skills in a time pressure situation after exploring scenarios without time pressure.
- The simulation of real-time control tasks, for instance in production processes, is useless unless clock-driven simulations are used. That means that the task is too simple provided that enough decision time is given.

Time-driven business simulators provide the users with extended strategic possibilities. Action and reaction at the right time are more important than in “conventional“ business simulators. Possible learning effects are shifting from knowledge about the structure of the underlying model to decision making itself. Thus, even they might not be suited as a learning tool, they can (in addition to the possible applications mentioned above) serve well as a tool to conduct research about decision making.

Although until now business simulators (that is, single user games) were in the focus of discussion the results can be easily transferred to multi user games, like corporate planning games (Garp 1995, p. 259). The rest of the paper, however, is limited on effects of clock-driven mechanisms in business simulators only.

An Exploratory Study of the Effects of Different Time Spans on Decision Making in a Simulation Game

To explore effects of different time spans for decision making in a simulation game, Größler (1997) reports of an explorative study using a rather simple predator-prey model. The model was extended with an user interface in order to make it a simulation game. This interface allows to set different time spans before the simulation goes ahead. Subjects had two decisions to make: shooting the predator or shooting the prey. The aim of the game was to stabilize both populations on a high level.

In this study only game performance of two groups was compared: in one group one simulation period lasted 1.6 seconds, in the other 0.8 seconds. Surprisingly, the group with a shorter decision period outperformed the other group, although not in a statistically significant way. Both groups reported no particular stress.

Both of these findings are in contrast to reports about decision making under time pressure in the literature (e. g. Brehmer 1995, p. 122). There are at least two explanations for this:

1. Because this was not a rigorous experiment there could be other factors which influenced performance (for instance, differences in group composition).

2. Maybe the postulated relation between time pressure and performance (the higher time pressure the more performance decreases) is not so simple as supposed. Dependent on task characteristics (in particular, task difficulty) there could be a range of time pressure which leads to better performance than other ranges. A reason for this could be that subjects simply become bored with an easy task when decision periods last too long.

Technical Aspects of Self-Proceeding Simulations

To conduct a more rigorous experiment with limited decision time a business simulator is needed which provides a self-proceeding functionality. After a certain time such a clock-driven simulator finalizes the current simulation period through calculating the results and switches to the next period in time, no matter if the user has finished all his or her input. This calculation could be based on different values of the input variables. If the user has changed them since the last simulation step the current values are valid. If they are unchanged, there are some possibilities how their new values can be calculated:

- the last valid input is used again,
- some kind of mean values are used from former simulation periods,
- a generic pattern of input is used.

The first mode seems to be the technically easiest and most intuitive one. In real-time simulators, it is the only applicable behavior, if the logic of immediate effectiveness of decisions should be conserved.

Of the popular system dynamics software packages, Powersim and iThink offer the possibility of advancing the simulation of a model after a certain time. This mechanism can be used to create the impression of a real-time simulation. However, in this study a different approach was used. A Vensim model was simulated using Vensim DLL. The complete user interface and control program of the simulation was written in Delphi. Thus, it was easy to build a time-proceeding mechanism in the Delphi control program of the business simulator.[2]

The task of the given business simulator is to manage the start-up of a copy shop: copy machines have to be bought, prices for copies must be laid down, advertisement budgets have to be decided on, etc. Reporting functions show a simplified balance sheet of the copy shop and the behavior of the most important variables over time in graphical form. The screenshot in Figure 4 depicts a typical scene while using this business simulator, which is an application in German language.

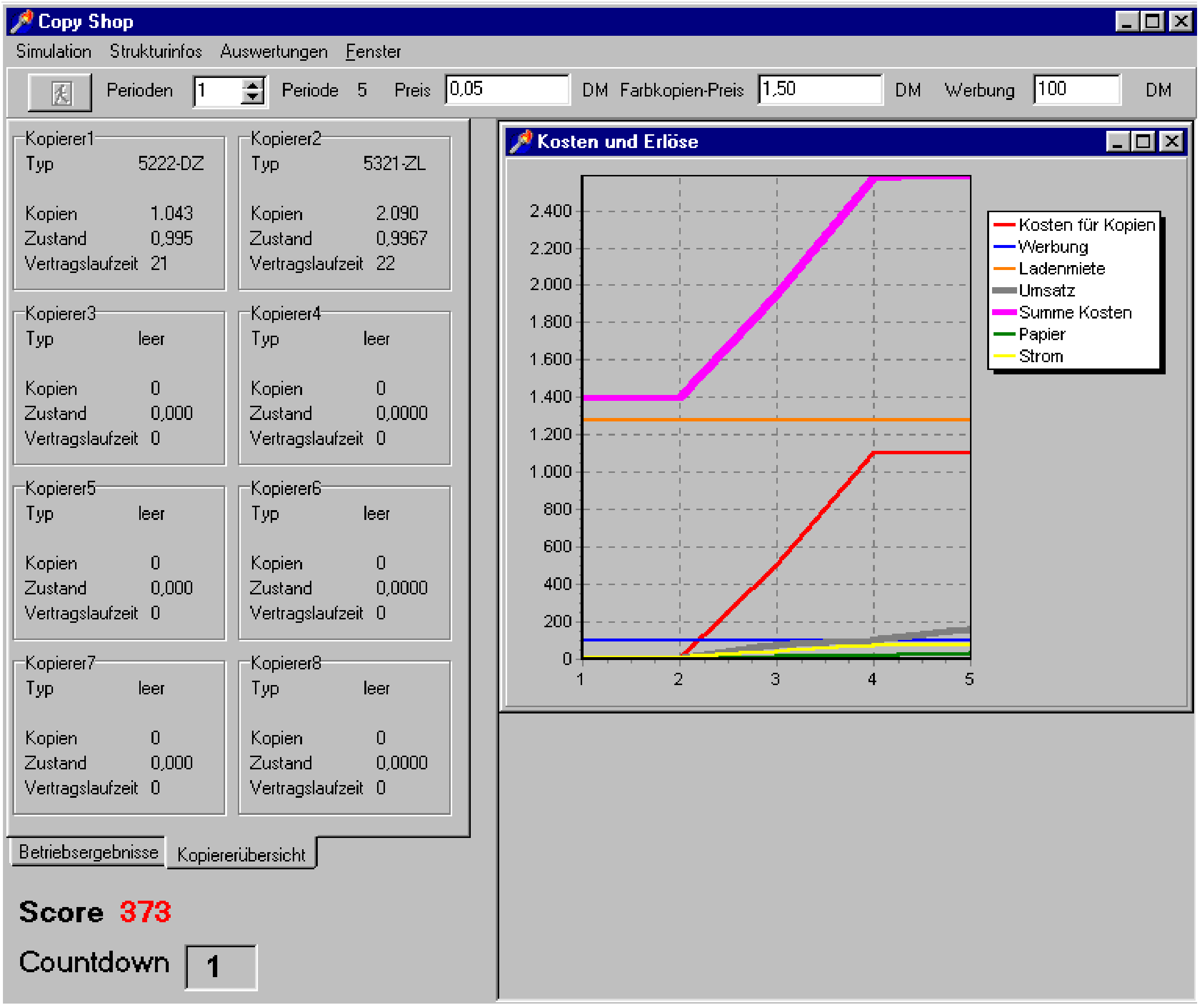

Figure 4: Screenshot of CopyShop business simulator (in German)

The screenshot in Figure 4 shows the input variables in the top row of the screen just underneath the main menu of the application. On the left hand side, the control field of the copy machines can be seen. Double clicking on an empty "slot" allows to buy a new machine. On the right hand side, costs and revenues are displayed in graphical form. Other reporting tools are available but not shown in this figure (balance sheet, costs and revenues per copy machine, report of decisions made in the past, etc.).

An Experiment to Study the Influence of Limited Decision Time on Control Performance and Learning

The CopyShop business simulator with its built-in self-proceeding mechanism was used to conduct an experiment to find out more about the relationship between decision time on one side and control performance and learning on the other. The main hypotheses are:

1. the shorter decision time the worse is subjects' game performance, and
2. the shorter decision time the less subjects learn.

These hypotheses are congruent to literature (see above). However, to take into account the controversial findings of the explorative study reported above, different levels of time pressure (different time intervals) were used as a means to distinguish between the experimental groups. Figure 5 shows the groups used in the experiment. The experimental design is a typical control group design with pre and post test.

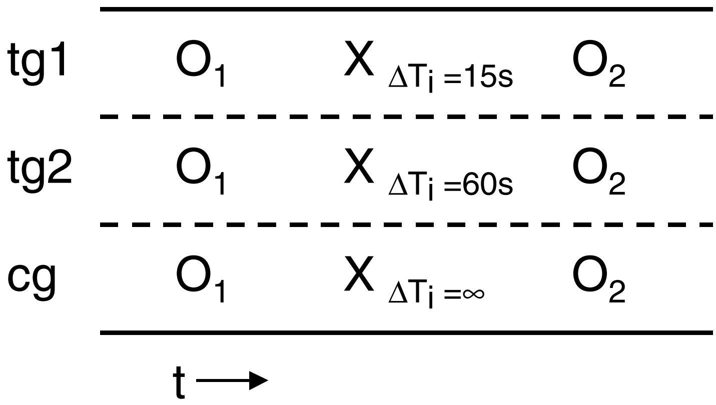

Figure 5: Experimental groups

O_1 and O_2 denote a pre resp. post knowledge test developed for CopyShop in order to measure learning effects. *X* denotes the use of the simulator with the maximum decision time per simulation period shown as subscript ΔT_i. To obtain more reliable data subjects had to play three game runs with 15 simulation periods each. Strictly spoken, the time interval for the control group was not indefinite because their time to play CopyShop had to be limited to 30 minutes per game session (L = 30min). However, it was secured that this time span was sufficient to finish 15 game periods without time pressure.[3] As a measure for control performance the accumulated money on the bank account of the simulated copy shop was used.

In order to guarantee comparable groups, subjects had to fill out a biographic questionnaire in the beginning. In addition to that, general business and economic knowledge was assessed in the beginning as well. Then the above depicted experiment took place. After the post knowledge test subjects had to fill out a brief questionnaire which allowed them to rate the stress they had experienced while using the simulator.

The results of the experiment will be presented at the 1999 System Dynamics Conference in Wellington, New Zealand.
Or check http://iswww.bwl.uni-mannheim/agroe/research.htm for the newest version of this paper.

References

Bakken, B., Gould, J. and Kim, D. H. (1994). Experimentation in Learning Organizations: A Management Flight Simulator Approach. In Morecroft, J. D. W. and Sterman, J. D. (eds.), *Modeling for Learning Organizations*, Portland, OR, pp. 243–266.

Brehmer, B. (1990). Strategies in Real-Time, Dynamic Decision Making. In Hogarth, R. M. (ed.), *Insights in Decision Making*, Chicago/London, pp. 262–279.

Brehmer, B. (1992). Dynamic Decision Making: Human Control of Complex Systems. *Acta Psychologica* 81, 211–241.

Brehmer, B. (1995). Feedback Delays in Complex Dynamic Decision Tasks. In Frensch, P. A. and Funke, J. (eds.), *Complex Problem Solving – The European Perspective*, Hillsdale, NJ, pp. 103–130.

Brehmer, B. and Allard, R. (1991). Dynamic Decision Making: The Effects of Task Complexity and Feedback Delay. In Rasmussen, J., Brehmer, B. and Leplat, J. (eds.), *Distributed Decision Making: Cognitive Models of Cooperative Work*, New York, pp. 319–334.

Buchner, A. (1995). Basic Topics and Approaches to the Study of Complex Problem Solving. In Frensch, P. A. and Funke, J. (eds.), *Complex Problem Solving – The European Perspective*, Hillsdale, NJ, pp 27–63.

Craft, C. J. (1967). Management Games. In Craig, R. L. and Bittel, L. R. (eds.), *Training and Development Handbook*, New York et al., pp. 267–284.

de Geus, A. P. (1988). Planning as Learning. *Harvard Business Review* 66(2), 70–74.

Forrester, J. W. (1961). *Industrial Dynamics*, Cambridge, Massachusetts.

Funke, J. (1993). Microworlds Based on Linear Equation Systems: A New Approach to Complex Problem Solving and Experimental Results. In Strube, G. and Wender, K. F. (eds.), *The Cognitive Psychology of Knowledge*, Amsterdam, pp. 313–330.

Funke, J. (1995). Experimental Research on Complex Problem Solving. In Frensch, P. A. and Funke, J. (eds.), *Complex Problem Solving – The European Perspective*, Hillsdale, NJ, pp. 243–268.

Garp, R. (1995). Computermodell für Unternehmensspiele [Computer Models for Corporate Planning Games]. In Geilhardt, T. and Mühlbradt, T. (eds.), *Planspiele im Personal- und Organisationsmanagement*, Göttingen, pp. 249–260.

Kim, D. H. (1989). Learning Laboratories: Designing a Reflective Learning Environment. In Milling, P. M. and Zahn, E. O. K. (eds.), *Computer-Based Management of Complex Systems: Collected Papers from the 1989 International System Dynamics Conference*, Berlin et al., pp. 327–334.

Kim, D. H. and Senge, P. M. (1994). Putting Systems Thinking into Practice. *System Dynamics Review* 10(2–3), 277–290.

Kluwe, R. H. (1993). Knowledge and Performance in Complex Problem Solving. In Strube, G. and Wender, K. F. (eds.), *The Cognitive Psychology of Knowledge*, Amsterdam, pp. 401–423.

Kreuzig, H. W. and Schlotthauer, J. A. (1991). Ein Computer-Simulations-Verfahren in der Praxis: Offene Fragen – empirische Antworten [A Computer-Simulation in Practice: Open Questions – Empirical Answers]. In Schuler, H. and Funke, U. (eds.), *Eignungsdiagnostik in Forschung und Praxis*, Stuttgart, pp. 106–109.

Maier, F. H. and Größler, A. (1998). A Taxonomy for Computer Simulations to Support Learning about Socio-Economic Systems. In System Dynamics Society (ed.), *Proceedings of the Sixteenth International Conference of the System Dynamics Society*, Québec City, pp. 57–58.

Meadows, D. (1989). Gaming to Implement System Dynamics Models. In Milling, P. M. and Zahn, E. O. K. (eds.), *Computer-Based Management of Complex Systems: Collected Papers from the 1989 International System Dynamics Conference*, Berlin et al., pp. 635–640.

Milling, P. (1991). Strategische Planungs- und Kontrollsysteme zur Unterstützung betrieblicher Lernprozesse [Strategic Planning and Control Systems for the Support of Organizational Learning Processes]. In Milling, P. (ed.), *Systemmanagement und Managementsysteme*, Berlin, pp. 11–31.

Milling, P. (1995). Organisationales Lernen und seine Unterstützung durch Managementsimulatoren [Organizational Learning and its Support through Management Simulations]. *Zeitschrift für Betriebswirtschaft, Ergänzungsheft 3/95: Lernende Unternehmen*, 93–112.

Morecroft, J. D. W. (1994). Executive Knowledge, Models, and Learning. In Morecroft, J. D. W. and Sterman, J. D. (eds.), *Modeling for Learning Organizations*, Portland, OR, pp. 3–28.

Niemeyer, K. and Reidelhuber, O. (1979). Das Planspiel als Methode der Verteidigungsplanung [Planning Games as a Method of Defense Planning]. In Huber, R. K., Niemeyer, K. and Hofmann, H. W. (eds.), *Operationsanalytische Spiele für die Verteidigung*, München/Wien, pp. 99–136.

Papert, S. (1980). *Mindstorms*, New York.

Senge, P. M. and Sterman, J. D. (1992). Systems Thinking and Organizational Learning: Acting Locally and Thinking Globally in the Organization of the Future. *European Journal of Operations Research* 59, 137–150.

Tansey, P. J. and Unwin, D. (1969). *Simulation and Gaming in Education*, London.

Notes

1. Strictly, ΔT_j consists of two components: time of decision making and time to process decisions with $td_j + tp_j = \Delta T_j$. However, with the advent of high performance simulation hard- and software, tp_j converges against zero. Therefore, the two components are not treated separately here. Additionally, in most cases over-all gaming time is limited, that is $\Sigma \Delta T_j \leq L$, for a certain time limit L.
2. A Delphi component of type `TTimer` was defined in the main module of the application which was set to a certain time interval. (A time interval of zero meant no automatic progression, which equals an event-driven simulation.) Whenever the timer is due a function is called which causes Vensim DLL to compute the next simulation period with the current values of the decision variables.
3. As results of Kreuzig and Schlotthauer (1991, p. 108) suggest, game performance is relatively insensitive to reasonable changes of the total time available for finishing a simulation.