- SDACQUIRE -
A Tutorial for Imparting the Methodical Foundations of System Dynamics

Guido Kaupe

Mannheim University
Department of Policy Management in Industry
Schloss, P.O.Box 10 34 62
D-6800 Mannheim, Germany

ABSTRACT

In today's "Information Age", the steadily rising requirements for communication and assimilation of knowledge present us with the constant need to confront new knowledge acquisition methods. One of these methods is Computer Assisted Learning or Computer Based Training. Because of the clear advantages evident in this type of instruction, the German-language computer-based tutorial SDACQUIRE was developed in the Department of Policy Management in Industry at Mannheim University. The goal of this program is to impart to the user the methodological fundamentals of System Dynamics.

This paper presents a description of various aspects of the program's development, implementation and application. Following a brief exposition of the evolution and benefits of Computer Assisted Learning, a description is given of important aspects of the modular development of SDACQUIRE. A discussion of the components of individual lessons is then provided, leading into a final section on the interactive communication connected with the program's use.

The Evolution and Benefits of Computer Assisted Learning

Computer Assisted Learning (CAL) or Computer Based Training (CBT) represents a new chapter in the area of educational technology. The first CAL projects appeared in the 1950s, as computer firms discovered that their machines offered excellent media for internal corporate training. At that time, the idea was an extension of the concept of "programmed instruction", based on the learning theories of B.F.Skinner -- with the goal of achieving independence of the learner from instructor, locale and time *(Seidel, 1989)*. The following decade witnessed the advent of many different CAL projects, developed in cooperation with university researchers and other computer specialists. Major breakthroughs in the field were not forthcoming, however, largely due to the limited availability and capabilities of the mainframe computers of the time. With limited terminal access for learners and no capacity for presentation graphics, the hardware backdrop for these projects was a hindrance to their success.

The advent of personal computers, the rapid advances in microelectronic technology and the subsequent drop in prices for this technology had great effect on the further development of the field. This progress was functionally extended through the ability to produce CAL programs that were no longer simply "electronic page turners." They presented the real capabilities of individual, interactive instruction, which was truly independent of the locale and the time *(Karrer, 1989).*

The advantages offered by the use of interactive communications media are manifold. As mentioned above, use of CAL programs provides for the locational and temporal independence of the learner from the instructor. This form of training or instruction has the effect of no longer requiring the simultaneous presence of course participants in a pre designated classroom. The course contents can be assimilated by the participants equally well at the school, the workplace or in the home. For necessary professional training courses, the use of CAL often enables employers to save the costs of travel and lodging (which out-of-town seminars would otherwise necessitate) and to avoid the problem of replacing absent employees.

A further advantage of computer assisted learning comes with the fact that, in a traditional course setting, participants must operate under the level and speed of instruction determined during the class meetings. In such courses, it is a rare luxury when individual attention can be afforded all participants. Consequently, the general effectiveness of the teaching is inversely proportional to the number of students. CAL offers a student the option of setting the speed of instruction independently, at his or her own pace. The student can choose to concentrate on the areas or topics that are individually the most appropriate.

In addition, through frequent exercises, computer assisted learning can enhance the level of attention given by the student to the material. This constant opportunity for self-correction leads to increased success in the learning process. Through the choices presented by menus, CAL programs provide not only a firm guidance through a well-defined series of lessons, but also a flexible opportunity for students to follow an individualized course of study, determined by their own strengths and weaknesses *(Goetz, 1991).* This creates the options for beginners to concentrate on the standard order of lesson presentation, while the advanced learner is free to select individual lessons for repetition or deeper engagement.

The intentions behind the development of the system dynamics tutorial program were based on the above advantages, together with the fact that the primary tool for implementing the fundamentals of system dynamics is a computer simulation language (DYNAMO). It was felt that the impact of learning a computer language upon the non-computer literate students might be lessened, if the background knowledge for the task were also provided though the use of a computer. A synergistic effect was anticipated in the learning experience. Based on these thoughts, the German language program SDACQUIRE was developed over seven months of the 1990/91 academic year in the Department of Policy Management in Industry at Mannheim University.

In order to build the SDACQUIRE tutorial, it was necessary to select an appropriate programming environment. Here, a choice in favor of a standard, procedural language, with unlimited flexibility and design capabilities, would incur the costs of a complicated programming task. The use of an authoring system, on the other hand, with an authoring system language, brings to the process many of the advantages of fourth generation (non-procedural) software tools. The consistent use of repeated structures in the building of a series of lessons can be accomplished with the use of relatively few functions and constructions. This has an enormous effect on the development time for tutorial programs (Bodendorf, 1990).

The SDACQUIRE tutorial program was developed using an authoring system, with an underlying authoring system language. The authoring system's language offers a large variety of functions, through which the designer can customize the product. Possibilities for screen design (format and colors), interactive dialog, and hooks to externally generated graphics provide the designer with excellent opportunities to build user-friendliness into the tutorial. Additionally, the authoring system avails the designer of opportunities to invoke other (DOS- compatible) application programs from within the environment -- in fact, such hooks can also be built into the tutorial.

Description of the Modular Development of SDACQUIRE

The fundamental structure of the SDACQUIRE tutorial program is based on the modular setup of the authoring system. One module corresponds to one learning unit and a tutorial program consists of several of these modules. Depending on the sizes of the various lessons, several modules may be combined. (For the authoring system used, no module may contain more than 100 lines.) Figure 1 shows the complete set of SDACQUIRE modules and their internal connections (denoted by lines).

The *START* module is the highest level module and provides the initial interface with the program. It is invoked upon execution of the 'SD.BAT' batch file and remains active thoughout the user's engagement with the tutorial. The function of the START module is that of a command module, from which all relevant instructions for the tutorial emanate. After presenting a general introduction to the SDACQUIRE program, the START module calls the MENUE module.

This module contains (as its name suggests) the main menu of the SDACQUIRE tutorial. Here, the user is offered the option of choosing from the 12 different lessons available, with no restrictions as to the order in which they are followed. Depending on the user's selection, the MENUE module will branch to module LEKTIONA (the first lesson), LEKTIONM (the twelfth lesson), or one of the modules in between (LEKTIONB - LEKTIONL).

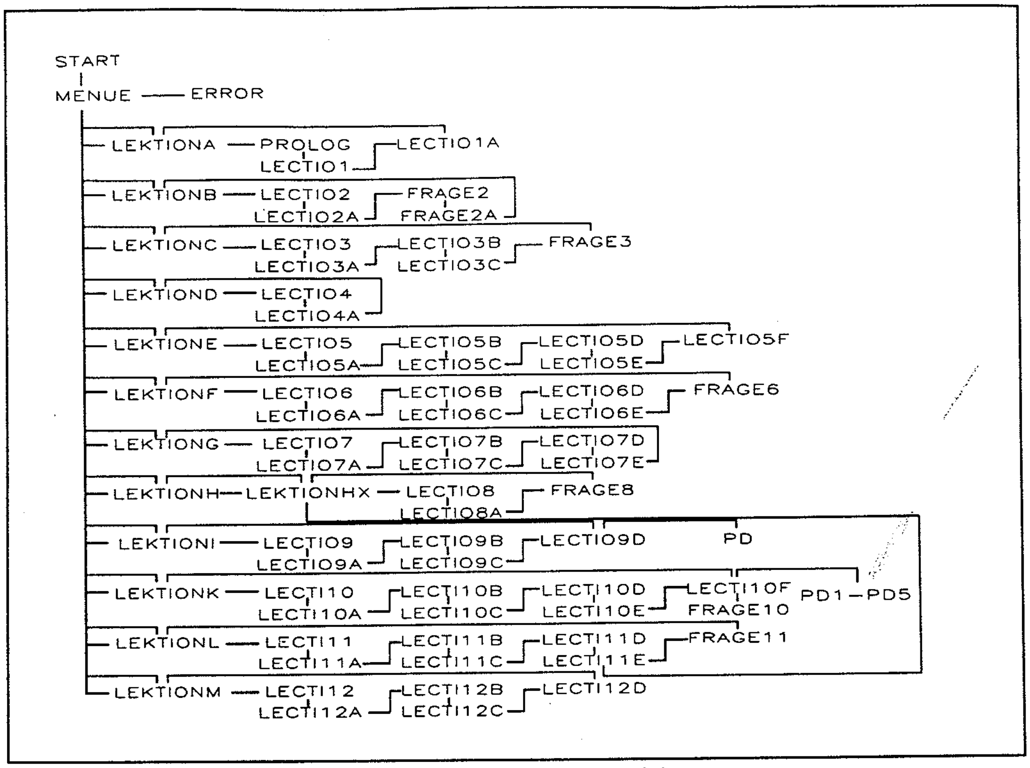

Figure 1: Internal modular structure of the SDACQUIRE tutorial program

After the user selects a lesson, by entering a number from 1 to 12, SDACQUIRE will execute the corresponding module to its last command. As shown in Figure 1, the LEKTIONC module, for example, contains calls to modules LECTIO3, LECTIO3A, LECTIO3B and FRAGE3. Control is passed in this example as follows: From the LEKTIONC module (the command module for lesson 3) the knowledge module LECTIO3 is called. This is executed to its last instruction, after which control returns to the command module, LEKTIONC. This module immediately calls the next knowledge module (LECTIO3A), which is executed similarly. In this fashion, all the subordinate modules of lesson 3 are executed -- including (as the last one) the exercise module, FRAGE3. When LEKTIONC recovers control, after completion of its final sub-module, an "End" instruction signals the conclusion of the lesson. This brings about the immediate return to the next highest module in the structured hierarchy -- MENUE. The program executes the other lessons analogously. The only variations come with the different number of sub-modules within the individual lessons.

An expanded feature within this context has been built into the ninth and tenth lessons (LEKTIONI and LEKTIONK). Here, the capability is provided for a branch to the DYNAMO program from the end of any sub-module. By pressing a particular "hot key", the user can cause the immediate execution of one of six specially designed modules, shown as PD, PD1, ... , PD5 in Figure 1. These modules contain calls to the Professional Dynamo Plus program, as well as distinctive DYNAMO commands, which are intended to establish (in the

DYNAMO environment) illustrations of the concepts being learned. The DYNAMO program retains control until the user enters the "Quit" command. At that point, the tutorial will continue its execution exactly where it left off (when DYNAMO was called).

An additional internal connection has been added to the SDACQUIRE tutorial modules, which addresses the intimate relationship between the contents of Lessons 8 and 11. As part of Lesson 8, the module LEKTIONHX offers a user the opportunity to repeat the knowledge portion of the lesson, without having to repeat its exercises. This module can be called from the LECTI11E module in Lesson 11, whose contents may render appropriate the review of Lesson 8. When the user presses a special "hot key" during the progression of Lesson 11, the tutorial's control will pass to LEKTIONHX, repeating the execution of Lesson 8's knowledge modules. Upon reaching the "End" statement, the program returns control to the LECTI11E module.

In light of the modular design and internal connections explained above, the following section discusses the contents of the individual lessons in the SDACQUIRE tutorial program.

Knowledge Components of the Individual Lessons

Rather than to carry out a detailed description of the particular features of each learning component of the SDACQUIRE program, the goal of this section is to indicate the knowledge areas of the system dynamics field addressed by the different lessons. As explained above, the tutorial consists of 12 lessons and is principally oriented to the system dynamics model-building process. Below is given a topical, lesson-by-lesson description of the knowledge contents of the tutorial.

In Lesson 1, the user of the SDACQUIRE program is presented with a brief introduction to the possibilities for and potential benefits of the application of system dynamics. This introduction is given in general terms, since the purpose is not to establish a definition of system dynamics, but rather to motivate the user and to stimulate his or her interest in the tutorial itself. After the brief introduction, Lesson 1 proceeds to outline the basic concepts of systems research. In particular, the concepts of model and systems analysis are addressed.

Lesson 2 is concerned with the historical development of the field of system dynamics. First, the fundamentals of cybernetics and bionics are described, with analogies between organisms and technical processes as central themes. Then, a discussion is presented of the early history of the system dynamics field, based on the extension of cybernetic principles to economic, ecological and social considerations. At the end of this lesson, a listing is given of the foundational publications of Jay W. Forrester *(1961, 1968, 1971)* and Dennis Meadows *(1972)*, which appeared in the beginning years.

Lesson 3 directly addresses the development of the systems theory of system dynamics. Here, a central theme is the collection of results from four areas of research, which subsequently had a positive effect on the field's evolution -- results from control theory, simulation theory, computer science and decision theory.

Lesson 4 concerns itself with the goals and objectives of the theory of system dynamics. Here, the functions are illustrated, which accompany the application of systems dynamics - the description, explanation, prediction and informational functions. At the end of the lesson, an exposition of the basic axioms of system dynamics application is given.

The contents of Lesson 5 consist of a presentation of the four-level hierarchy that forms the basis of the structural theory of system dynamics. Closed system boundaries, feedback loops, level and rate variables, and policy sub substructures define the four planes of the hierarchy. On the basis of the second plane, the theoretical foundations of causal connections and their feedback relationships are explored.

Lesson 6 describes the model-building process that underlies the modeling and analysis concepts of system dynamics. Instruction is then presented in the design of causal loop and flow diagrams. The graphic symbols that are necessary components of flow diagrams have been built into the tutorial (and can be displayed on the screen) thanks to the graphics hooks in the program.

In Lesson 7, the user is provided with an explanation of the theoretical fundamentals of the DYNAMO simulation language. By the end of this learning unit, the user should have sufficient background to produce an executable DYNAMO listing of equations for a given flow diagram.

The central theme of Lesson 8 is the computation interval length 'DT'. In addition to an explanation of the sense and the functional role of this value, the user is given certain rules of thumb that help in the task of setting its value for a particular model run. The compromise between the desirable aspects of computational precision and the costs associated with computation time and storage requirements is specially addressed in this connection.

Lesson 9 contains a discussion of the procedural fundamentals of the Professional Dynamo Plus simulation language. In this lesson, the user of the tutorial will obtain a basic orientation to handling the DYNAMO program. The individual sections of this lesson are centered around the four main commands of Professional Dynamo Plus's main menu: 'EDIT', 'COMPILE', 'SIMULATE' and 'VIEW'. Here, the user is offered the opportunity, after each section, to practice with the acquired knowledge.

Lesson 10 is concerned with functions that are built into the DYNAMO programming language (as macros). These functions are broken down into five classes: table functions, trigonometric functions, logical functions, time functions, and random value generator functions. The graphical symbols for flow diagrams are also presented, as well as several example diagrams, which are available as running models in Professional Dynamo Plus at the touch of a special "hot key."

Lesson 11 explains the design and functioning of delays. Here, particular delay macros are derived and built-in delays are described. Delays are distinguished in this lesson according to their types and orders. An explanation is then offered of their various behaviors. A representation of the effects of delays on the time factor, in consideration of the choice of interval length ('DT'), is then provided. Since the choice in this respect is not only dependent on these considerations, but also is affected by those described in Lesson 8, the above mentioned branch to Lesson 8 is offered to the user.

In Lesson 12, validation concepts are discussed, which are central to the application of system dynamics. The preparation of the knowledge contents for this module was oriented to the three phases of the validation process: validation of the model structure, validation of the parameter values and validation of the model behavior.

Interactive Communication with the Tutorial SDACQUIRE

The greatest advantages of computer assisted learning programs are realized through the interactive capabilities provided. Indeed, compared with traditional methods of teaching, the dialog features for communication between user and machine, are significant. For this reason, the particular realization of the interactive capabilities present in the SDACQUIRE tutorial program are described in some detail in the current section.

The user of the tutorial is challenged at the end of most lessons, with a series of exercises that relate to the knowledge contents covered. The learner receives three chances to provide a correct answer to each question posed. With failed attempts at answering, the user is given hints that may be helpful to another try at answering the question. If the third attempt is unsuccessful, the correct answer is displayed and the user is presented with the next step. For example, in response to a wrong answer to a request for the name of the book "The Limits to Growth" (by Dennis Meadows), a hint is displayed which mentions the investigation surrounding a world model. If a second answer is also unsuccessful, a second hint mentions the financial backing of the Volkswagen Foundation and the Club of Rome. If a correct response is still beyond the grasp of the user, the answer is provided by the program.

A further option offered the user is in the choice of lesson order. This option is presented in the program's introduction, should the learner wish to follow a lesson order which deviates from the numerical sequence. The introduction's guidance represents recommendation, not restriction. A user who favors a different path is fully free to follow it. This is particularly significant for learners who already possess some degree of knowledge in the field and wish only to repeat or deepen their engagement with individual topics.

Furthermore, the user is offered additional branching possibilities within the SDACQUIRE tutorial program. A jump from Lesson 11 (Effects of the Choice of 'DT' on the Behavior of Delays) to the eighth lesson (General Rules for the Determination of 'DT') is one option. Also, the ability to jump directly into the execution of Professional Dynamo Plus, from Lessons 9 and 10, provides the possibility for immediate application and illustration of acquired knowledge. In particular lessons, the user can also choose to repeat individual parts of the learning units, without being forced to sit through the entire lesson, in order to address certain complex issues in a more concentrated way.

As a final note, it should be mentioned that limited experimental testing has been conducted with a prototype of this program. Preliminary results have indicated that the motivations behind its creation may indeed have been justified. Reactions have shown there to be potential for effective learning of systems dynamics fundamentals in this context.

References

Bodendorf, Freimut: Computer in der fachlichen und universitaeren Ausbildung - Handbuch der Informatik, Bd. 15.1, Munich 1990.

Forrester, Jay W.: Industrial Dynamics, Cambridge, Mass. 1961.

Forrester, Jay W.: Urban Dynamics, Cambridge, Mass. 1968.

Forrester, Jay W.: World Dynamics, Cambridge, Mass. 1971.

Goetz, Klaus; Haefner, Peter: Computergestuetztes Lernen in der Aus- und Weiterbildung, Weinheim 1991.

Karrer, Urs: Computer-Assisted Learning: Toward the Development and Use of Quality Courseware, New York 1989.

Meadows, Dennis L.: The Limits to Growth, New York 1972.

Seidel, Christoph: Psychologische Gesichtspunkte beim CUU, in: Kueffner, Helmuth; Seidel, Christoph (Ed.): Computerlernen und Autorensysteme, Stuttgart 1989, P. 32 - 44.