Int. J. Engng Ed. Vol. 15, No. 3, pp. 168–190, 1999
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Towards Dynamic Modeling of a Teaching/Learning Process Part 3: The Simulation Model*

N. EFTEKHAR and D. R. STRONG
Mechanical & Industrial Engineering Department, University of Manitoba, Winnipeg, Manitoba, Canada. E-mail: strong@cc.umanitoba.ca

In this part of the study, first the STELLA language software, as an operational manifestation of the system dynamics approach, is introduced and discussed. Then the general model of the teaching/learning process is translated into the STELLA stock and flow equation of a system dynamics model. Although the gist of the teaching/learning model is common for both form-oriented and function-oriented learners, because of their totally different approaches to learning, two separate base models could be built and run concurrently. The rest of the paper deals with the description, analysis and results of the base models and the implementation of the different policies to improve the behavior of these two systems.

INTRODUCTION

IN THE PREVIOUS PARTS of this study, in line with step 1 and step 2 of the system dynamics approach, a system for the teaching/learning process was defined [1]. The building blocks of the system, their constituent parts, and their relationships were theorized and described. Then, the main parts of the system were converted into a unified diagrammatic representation. Further analysis of the literature search, uncovered new knowledge about student's learning and resulted in development of a new theory and the introduction of two different types of learners with two distinctive approaches to learning: form-oriented learners and function-oriented learners (Form-Function Theory of Types [2]).

The proposed form-function theory of types, provides a new ground for analyzing, understanding, and re-engineering of the teaching/learning processes. This theory is based on a 'systems as cause' thinking approach and, hence, looks at the systems or processes as the cause of their performance as opposed to their performance being merely determined by outside forces. This theory, principally, demands this study to develop a separate model of the teaching/learning process for each type of learner.

To investigate how a learner and a teacher really work and interact with each other over time, a simulation model should be constructed and run accordingly. In fact, the model should be based on the interaction between three major sets of components in the system:

- the learner's learning abilities and motivation;
- the teaching system's characteristics, and
- the nature and quality of the subject matter.

The answers to two fundamental questions raised from the interaction of the above forces, namely: *what* should be taught to *whom*; and *how* should it be taught should be found in dealing with such a model. This is achieved by using a system dynamics approach and employing a computer simulation program (STELLA Research Software). Thus, STELLA software will be introduced and discussed briefly. Then, two different basic structures for the *form* and *function* learners are constructed in the STELLA language. The base models are run and the results are compared with observed realities to validate the models. A number of policy variables are used to improve and to enhance the situation. For instance, it will be shown that the teaching/learning process may be enhanced by the careful choice of the learning material (subjective, objective, and procedural) that the teaching system presents.

The results of this experimentation indicate the power and effectiveness of using industrial engineering modeling techniques in the field of non-physical (non-rigorous) variables of education. Moreover, and more importantly, the proposed Form-Function Theory of Types introduced by this study, facilitates the better understanding of the mechanism of *learning* from one side and *teaching* from the other side.

STELLA LANGUAGE SOFTWARE

The STELLA software language is built around a progression of structures. *Stocks* and *flows* are at the lowest level and are the fundamental building blocks of the structure. *Infrastructures*, which vary

* Accepted 21 May 1998.

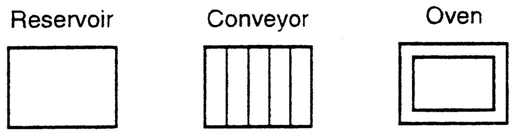


Fig. 1. Stock types: (a) a reservoir; (b) a conveyor; (c) an oven.

in size and complexity, are the next step in progression. They are built up from various combinations of stocks and flows. *Feedback loops* are the final step in the progression and are the relationships that link stocks to flows in various ways. In so doing, they enable infrastructures to exhibit interesting dynamic behavior [3].

This section provides an overview of each step in the progression of the fore-mentioned structures. In fact, this overview will prepare the ground for understanding what the 'structure' looks like at each level, how each structure behaves, and overall, how a STELLA model works. To be more efficient, in discussing each step, this study uses examples from the non-physical variables that, one way or another, are parts of a teaching/learning process in real life [3].

The building blocks

Components, or the building blocks, of the system are the first progression of the structure in the STELLA software language. There are four basic building blocks in the system: the stocks, the flows, the converters, and the connectors. A concise description of each of these components follows [4].

1. Stocks. Stocks are basically accumulations. They collect whatever flows into and out of them. The default stock type in STELLA is the 'reservoir.' A reservoir passively accumulates its inflow, minus its outflows. Any units, which flow into a reservoir, will lose their individual identity. Reservoirs mix together all units into an undifferentiated mass as they accumulate. In a teaching/learning process, for instance, the student knowledge is an accumulation that varies as the process of teaching/learning proceeds.

Three other stock types are available in the STELLA software, but only two of them; 'conveyors' and 'ovens' are used in this study. A conveyor can be thought of as a moving sidewalk or a conveyor belt. Stuff gets on the conveyor, rides for a period of time, and then gets off. The transit time for a conveyor can be either constant or variable. Both capacity and inflow limit can constrain entry to a conveyor.

On the other hand, an oven may be thought as a processor of discrete batches of stuff. The oven opens its doors; fills (either to capacity or until it is time to close the door); bakes its contents for a time (as defined by its outflow logic); then unloads in an instant. By contrast, stuff that enter these two stocks (conveyor and oven) do retain both their magnitude and time-of-arrival identity.

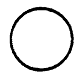


Fig. 3. A converter.

Stocks, in general, can be referred to as system state variables. Figures 1(a), (b) and (c) show a reservoir, a conveyor, and an oven type of stocks respectively.

2. Flows. The task of flows is to fill and drain accumulations. Mathematically, they are the instantaneous rates of flows that represent the means by which the system is controlled and represent activity points in the system. In fact, without flows, no change in the magnitude of stocks could occur. So, stocks and flows are inseparable components. They form the minimum set of structural elements needed to describe the dynamics of a system. Figure 2 exhibits two types of flows that are used in the STELLA program; uniflows and biflows. In Fig. 2(a), the unfilled arrow head on the flow pipe indicates the direction of the uni-directional flow. Clouds represent infinite sources or sinks for flows as illustrated in the diagram. Also, Fig. 2(b) shows a bi-directional flow (biflow), which is used to transport things both into and out of an accumulation. The second, shaded arrow head on this flow points the direction of outflow. Uniflows will assume only non-negative (i.e., inflow) values, but biflows can take on any value.

3. Converters. Converters are auxiliary functions and serve a utilitarian role in the software. They hold values for constants, define external inputs to the model, calculate algebraic relationships, and serve as the repository for graphical functions. In general, they represent the decision processes in the system. They are called 'converters' since they convert system states to system activities (or inputs to outputs). Figure 3 shows the symbol that represents converters in the STELLA mapping.

4. Connectors. As their names suggest, the job of connectors is to connect model elements. In fact, connectors are links that connect all of the components to each other. In so doing, they eventually form arcs that influence the flows (which regulate the system). The only restriction of connectors is that one cannot drag it into a stock. The only way to change the magnitude of a stock is through a flow. Figure 4 shows how a connector looks in STELLA software.

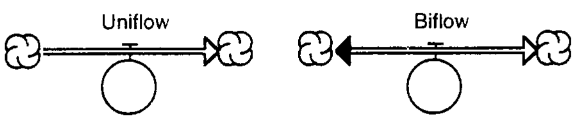


Fig 2. Flow types: (a) a flow (uniflow); (b) a biflow.

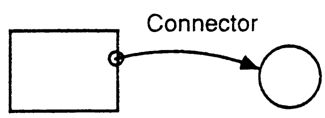


Fig. 4. A connector.

Infrastructures

As stated earlier, stocks and flows are the principal building blocks of the STELLA software language. However, irrespective of how many flows are attached to it, a single stock system can self-generate only a very limited set of dynamic behaviors. In order to produce a more complex dynamic pattern, it is essential to assemble sets of stock/flow combinations. These sets of combinations are called *infrastructures.* They exist in an essentially infinite variety. For the purpose of this study's modeling, it is important to recognize that infrastructures will generally define the range of characteristic behavior patterns that a model of teaching/learning will be capable of exhibiting.

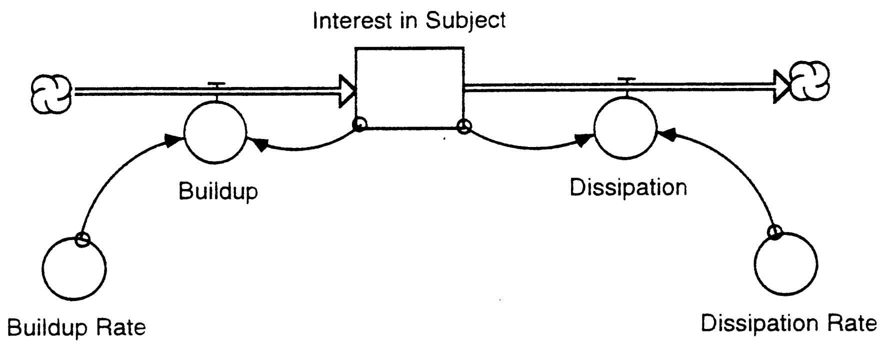
In practice, infrastructures typically appear in a limited number of generic forms. Each generic form has certain dynamic behavior. Five main generic forms are recognized in STELLA software as follows:

1. First-order linear infrastructure: a simple combination of a compounding and a draining process (Fig. 5). (Note: In a compounding process, the stock serves as the basis for producing its own inflow while in a draining process, the stock serves as the basis for generating its own outflow.)
2. S-shaped: a self-reinforcing growth process that eventually is under control by some growth constraint.
3. Overshoot and collapse: accumulations do not make a smooth transition from growth to steady-state. Instead, they grow rapidly, reach their maximum, and then decline to a new steady-state value.
4. Oscillation: an oscillatory behavior produced by a minimum of two stocks while each serves as a catalyst for producing the other stock's flow.
5. Main chain: represents a sequence of stages through which stuff flows while the specific nature of the flows varies, depending on the specific situations being modeled.

Taken as a whole, these generic processes will help this study to operationally specify the teaching/learning processes that it seeks to represent with the software. A model of a teaching/learning process will generally employ a combination of all of the above types. An example of the generic structure of the first-order linear infrastructure is shown in Fig. 5. The system is called 'first-order' since only one stock is involved. Also, it is 'linear' since the constant proportionality between the stock and its flows gives rise to the term linear—which refers to the algebraic form of the flow equation.

As the diagram shows, the stock is fed by a compounding process (as defined and formulated in the figure). It is depleted by a draining process. Both the compounding fraction and the loss fraction are constant, which means that both compounding and draining flows are proportional to the amount of the stock.

A first-order linear infrastructure can exhibit three distinct behavior patterns, depending upon the relationship between the compounding and loss fractions. When the two fractions are constant, and the compounding fraction is greater than the loss fraction, the infrastructure exhibits exponential growth—the compounding process will dominate the behavior. In each cycle of the process, more will be added to the stock than will be taken away. As the stock builds, both inflow and outflow will grow larger. In relative terms, however, the inflow will always be greater than the outflow. The net rate of growth in the stock is simply the difference between the compounding and the loss fractions. On the other hand, when the compounding fraction is less than the loss fraction in this infrastructure and both are constant, the net rate of decline is the difference between the loss and the compounding fractions. Finally, when the



KEY EQUATIONS:
Buildup = Interest_in_Subject * Buildup_Rate
Dissipation = Interest_in_Subject * Dissipation Rate

Fig. 5. Example of first-order linear infrastructure.

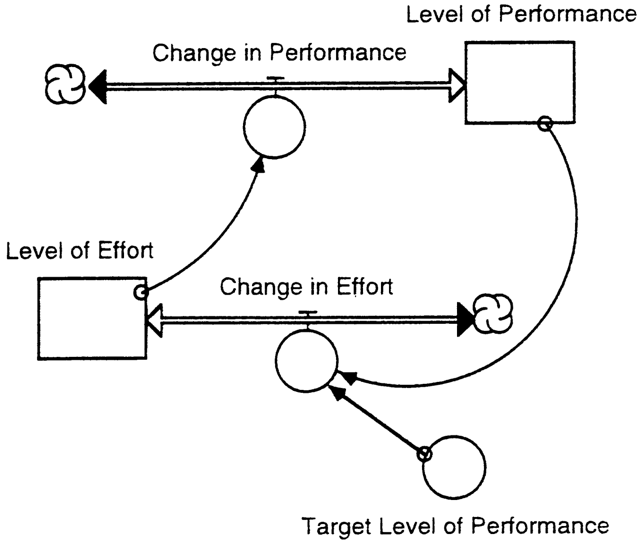


Fig. 6. A simple counteracting loop.

two fractions are equal, the stock will remain constant. The draining flow is equal to the compounding flow, so no change will occur in the stock.

Feedback loops

While infrastructures define the range of dynamic behavior patterns that a model is capable of exhibiting, the particular kind of feedback relationships that exist within the infrastructure will determine which of these patterns is realized. A feedback relationship is a closed-loop circle of cause-and-effect. Feedback loop cause-and-effect always includes at least one stock and one flow. This is because stocks are conditions that give rise to actions (or flows of activity) that in turn change conditions. However, it is really the current state of conditions, relative to some target level for the condition, that inspires conditions to change. Thus, feedback loops could be viewed as relationships that generate goal-seeking behavior. Goal seeking is a fundamental activity in which all dynamic systems engage. In fact, goal seeking is what enables conditions within a system to remain on course. When deviation occurs, feedback relationships inspire and direct corrective actions to bring conditions back in line.

There are two types of feedback relationships: negative (counteracting) and positive (reinforcing) feedback loops. When any variable in a negative loop is changed, then the loop causes that variable to readjust in the opposite direction. The negative loop produces self-regulating change (controlling and restorative behavior). Figure 6 illustrates a common counteracting feedback process. In the loop, the `level_of_effort` is being used to regulate the `level_of_performance`. If performance falls below the level that the student has set as his or her target, then effort should go up. A higher `level_of_effort` leads to an increased `level_of_performance`. So, an initial decrease in performance propagates a signal around the loop, which leads to an increase in performance. The loop thus acts to counteract the initial change.

It should be noted that the loop also could counteract change in the other direction. That is, if performance rises above target levels, effort will be scaled back so as to return performance target levels.

By contrast, positive (reinforcing) feedback processes compound change rather than counteract it. When any variable in a positive loop changes, the resulting interactions cause that variable to change further in the same direction. The positive loop, in other words, characteristically produces self-reinforcing change (unrestrained growth).

Figure 7 is an illustration of how a typical reinforcing feedback process works. The better a student performs, the more confident she or he feels. Subsequently, the more confident s/he feels, the better s/he performs. However, as mentioned in the counteracting feedback relationship, the loop also may change conversely. That is, the less confident one feels, the worse one performs and subsequently, the worse one performs, the less confident one feels.

As the diagram indicates, in the case of a reinforcing feedback loop, the goal or `target_level_of_performance` is linked to the level of `self_confidence`. The link means that when `self_confidence` rises, the target for `level_of_performance` follows suit and vice versa. Then, as performance adjusts to the new target level, self-confidence responds accordingly.

Combining counteracting and reinforcing feedback loops

In fact, it is the interaction and shifting dominance between the two types of feedback relationships that generates the dynamic character of a system. Figure 8 is an effort to combine the two

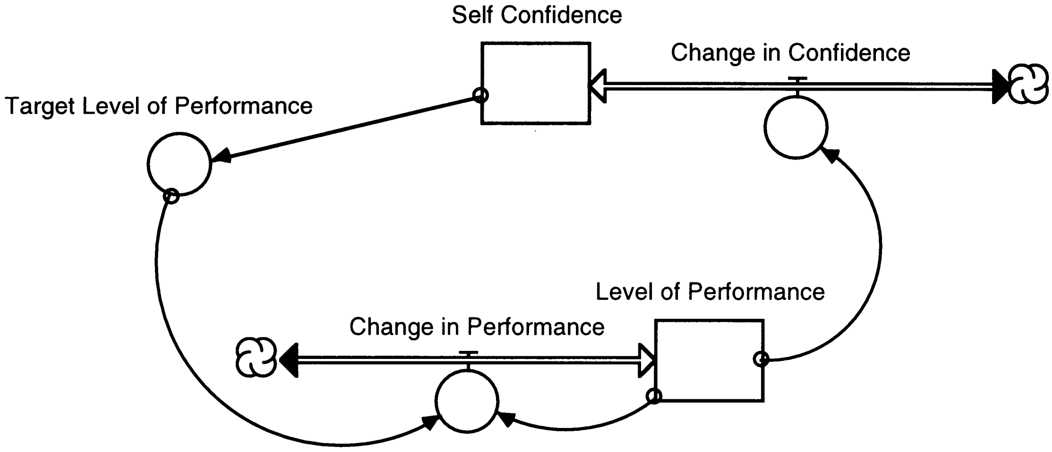


Fig. 7. A simple reinforcing loop.

previous examples and to show the way that the resulting system behaves. As the diagram indicates, now it is `self_confidence` that sets `target_level_of_performance` and `target_level_of_effort`. That is, how confident a student feels, determines both how well s/he thinks s/he should be able to perform as well as how much effort s/he puts out in order to achieve that level of performance. `Level_of_performance` feeds back to determine `self_confidence`, and `level_of_effort` feeds back to determine `level_of_performance`.

Now, if this set of relationships is allowed to operate with the STELLA software, the behavior of the system will depend on the initial levels of confidence, performance and effort as well as the strength of the relationships between `self_confidence` and the two targets. For example, if a decline in `self_confidence` causes a larger decline in the target for effort than it does for performance, then the system accelerates downward.

However a decrease in `self_confidence` has only a minimal effect on the `target_level_of_effort`, the counteracting feedback loop which ties `level_of_performance` back to `change_in_effort` will have a chance to operate. This loop will act to boost the `level_of_effort` which, in turn, will increase the `level_of_performance`. An increase in performance, then will inspire a rise in the level of `self_confidence` accordingly.

Worth mentioning is that there are a lot of 'ifs' in these scenarios. The 'ifs' depend on the relative strengths of the feedback relationships that are involved. This simple example emphasizes the fact that it is difficult to make accurate predictions about the performance of systems involving extensive webs of feedback relationships.

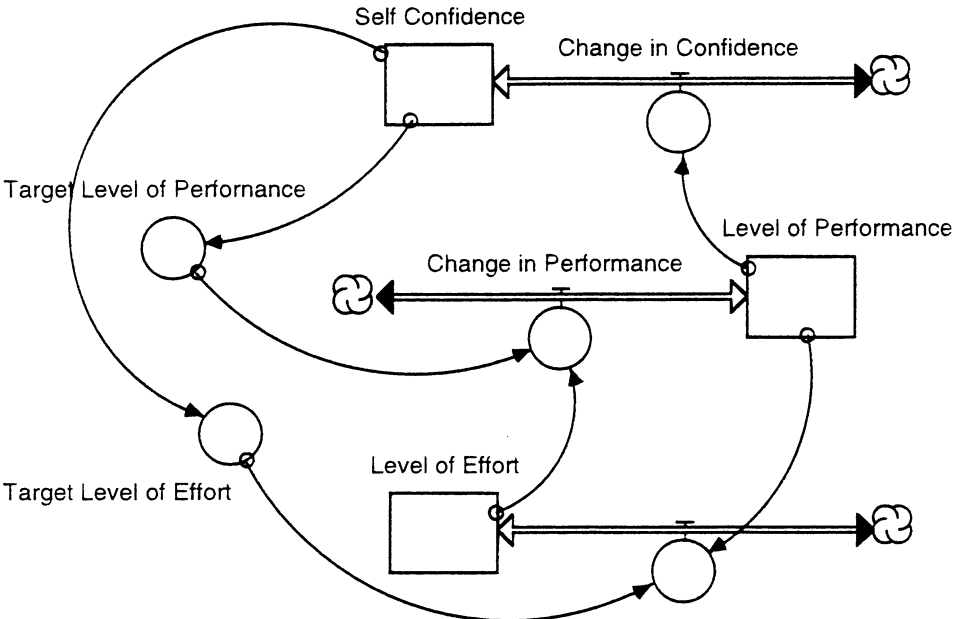


Fig. 8. Combining reinforcing and counteracting feedback relationships.

MODELING SOFT VARIABLES IN A TEACHING/LEARNING PROCESS

Modeling of a teaching/learning process requires total involvement with variables that are internal to both learners and teachers. Variables like student's abilities and motivation or quality of teaching are not entities that can be measured or computed. In fact, since they are non-physical (soft) variables, they could not get numeric or precise values. Despite this reality, however technically, there is a mechanism for tackling such a problem. The mechanism could be found in the fundamental distinction that exists between measurement and quantification [5].

Measurement, by definition, means 'assessing the magnitude of'. The result of the assessment is often expressed numerically. All physical quantities or 'hard' variables like height, volume and weight have their pre-defined units-of-measures. On the other hand, *quantification* means, 'assigning a numerical index to'. While assigning a quantitative index usually is a pre-condition to measuring something, the two activities are not the same. The interesting point is that one can quantify anything.

Fortunately, in the case of a teaching/learning process, it is not necessary to measure all of the soft variables in order to be able to use them in the simulation model. That is, the study will assign a numerical index to each of the non-physical entities that are involved in the system. For instance, to quantify student motivation, the research will assume that 0 represents the complete lack of motivation and 100 represent as much motivation as is possible for a student to have. A similar quantitative index would work equally well for the effort that students put into a learning task or the interest they have in the subject matter. Likewise, to quantify the rate of knowledge acquisition, the research will assume 0 represents the complete absence of effort to learn and 100 represents as much knowledge as is possible for a student to acquire for a given period of time.

Doing this will cause this study to act in a rigorous manner about the relationship each variable bears to other variables in the teaching/learning system. Hence, the more this study tries to quantify, the better the desired model resembles the real one. In addition, this will enable the study to solidify all the soft variables and simulate them to examine their role in the dynamics of a teaching/learning process.

This study, based on the discussions and findings in the previous parts [3, 4], proposes two separate models for the components involved in a teaching/learning process. That is, a system for a form-oriented learner and a separate system for a

Table 1. Defined variables for the proposed models of teaching/learning process

Proposed Model	Stocks	Flows	Converters
Form-oriented learners	hooks_under_development hooks_for_repetition hooks_in_memory quantity_of_info interest_in_subject expectancy_ level_of_effort level_of_performance	taking_ (information) completing_ (the hooks) repeating_ (hooks of information) lecturing_ change_in_quantity_of_info change_in_expectancy change_in_effort change_in_performance waste_	type_of_info: memory_info, relrote_info, relreal_info, procedure_real, procedure_rote learning_reinforcers: testing_ : (rote_type, clsd_prbm_slvg, opn_prbm_slvg), quality_of_teaching, other_reinforcers interest_in_use, interest_in_grade, impact_of_other_values availability_, student's_perceived_availability, forecast_adjustment prior_knowledge, amount_learned, productivity_, coding_. waste_fraction, learning_style_compatibility, constraints_. willingness_, mark of desire_, perceived_assessment target_info, adjustment_fraction allocated_time_factor
Function-oriented learners	relationships_under_study structures_to_form structures_in_memory quantity_of_info interest_in_subject expectancy_ level_of_effort level_of_performance	taking_ (information) evaluating_ (information) finishing_ (the structures) lecturing_ change_in_quantity_of_info change_in_expectancy change_in_effort change_in_performance waste_	Same as above

function-oriented learner. It is noteworthy that the gist of the main structure for both models is the same and only the main chain within each model is different.

Table 1 shows the list of the soft variables that have been defined and considered within the two models. Each soft variable is represented by one of the main building blocks of the STELLA software (stock, flow, or converter). As the table depicts, each model is composed of exactly fifty soft variables with most of them being common in both models. Note that this is a good number for a STELLA model. In fact, a model with this number of variables is neither too complicated to be unmanageable nor too simple to be unacceptable as the representation of the reality.

Each model uses eight stocks and nine flow rates. The remaining variables have been defined by the converters. All of the stocks except two are reservoir types. These two stocks; `hooks_for_repetition` and `structures_to_form`, are Conveyor and Oven type stocks respectively.

On the other hand, four flow rates (out of total nine flows) are biflow types. These flows represent variables that can 'change' values in either direction (e.g., `change_in_quantity_of_info` and `change_in_effort`).

LEARNING MODEL OF A FORM LEARNER

In this section, the STELLA model of a teaching/learning system for a form-oriented learner during a short time period (such as a class lecture) is presented. The role of each variable in the model is highlighted, the nature of the interactions between different variables within the entire system (feedback loops) is described, and finally, the behavior of the system will be discussed.

Description of the base model for a form learner

The system flow diagram for the learning process of a form-oriented learner is as shown in Fig. 9. The diagram has been constructed by STELLA simulation language software and represents all the variables presented earlier in Table 1. Basically, the variables within the system can be recognized in three sets of components: components of the teaching system, components of the learning side (the form learner) and the components of the subject matter. Note that the components of both the teaching system and subject matter are at the top and the right end of the diagram. The remainder of the diagram includes the components that represent the characteristics of the learner (learning, motives and performance). The definition and description of each variable have been given in the List of Equations (Appendix). Worth mentioning is that the STELLA equations created from Fig. 9 in the List of Equations are two types. The first types are stock level equations (which are generated by the software directly from the diagram) and their associated initial conditions. The remainder are the flow and converter equations that are generated by the modeler.

Referring to Fig. 9, the main chain infrastructure at the center of the diagram represents a sequence of stages through which the information flows in the learning side of the system. Apparently, the specific nature of flows vary, depending on the specific situation of each type of information. The chain is fed by a single flow (`taking_`). The cloud on the left-hand side of the flow of `taking_` depicts the boundary of the model. It represents an infinite source for the `taking_ flow`, as shown. (For the purpose of this model, it does not matter what is in the cloud.) The flow of `lecturing_` (teaching system) is governed by two main variables: `type_of_info` and `change_in_quantity_of_info`. `Type_of_info` is composed of five different types of incoming info (as introduced and discussed in Part 2 of this study [2]). In fact, the composition of the `type_of_info` determines the type of the teaching system or the teacher. If more weight is given to `memory_info` or `relrote_info` (relationship rote oriented information), the teacher is most likely a form-oriented teacher. Other potential possibilities can be created and used in the model by changing the composition of `type_of_info`. The governing effect of type_of info on the flow of `lecturing_` has a subsequent impact on the `learning_style_incompatibility` (which has its relevant impact on the learning side as will be discussed later).

In the present base model, the flow of `taking_` is capturing each piece of issuing information from the teaching system (flow of `lecturing_`) and placing it into the mind of learner (stock of `hooks_under_development`). Therefore, the flow of `taking_` depends upon the flow of lecturing_ from one side, and `coding_` (suitability to be categorized and connected to the hooks already stored in the `amount_learned`), and the learner's own `forecast_adjustment` from the other side. The flow of `waste_ drains` the stock of `hooks_ under_development` at a rate that is determined by the level of the stock itself and the `waste_ fraction`. `Waste_fraction` is the product of `constraints_` (that include all impeding factors, whether internal or external to the student, that lead to waste in the process of the knowledge acquisition) and `learning_style_incompatibility.`

Flow of `completing_` takes the hook of information from the stock of `hooks_under_development` and places them in the stock of `hooks_for_repetition`. This flow is under the influence of four variables: the level of the direct upstream stock, the learner's `interest_in_subject`, `interest_in_grade`, and `interest_in_use` [1]. The learner's `interest_in_subject` is represented by a stock and flow combination while the other two types of interest are shown by converters. It is assumed that the magnitude of the learner's interest_in_subject may vary during a lecture period while this is not so for the other two.

As shown in the diagram (Fig. 9), a conveyor type stock represents the state of the hooks_for_repetition in the mind of a form learner. Interestingly, by assigning different inflow limits and capacities to the conveyor, different types of form-oriented minds can be detailed and modeled. Subsequently, the flow of repeating_ gets the repeated hooks off the conveyor and stores them in the stock of hooks_in_memory. The flow rate of repeating_ is adjusted by the level_of_effort that the learner puts into the task. The impact_of_other_values (all the remaining task values) reinforces the flow of repeating_. Finally, the sum of total number of hooks_in_memory and prior_knowledge are represented by the amount_learned. Needless to say, all of this entities have been already defined in the first part of the study [1].

The other five stocks (shown as boxes in the diagram) are quantity_of_info (given by the teaching system), interest_in_subject, level_of_effort, expectancy_, and level_of_performance. Each of these stocks allows these parts of the system to have initial values. These stocks change in value according to the amount they receive or lose since their bi-directional flows can get both positive and negative values.

To simplify the model, only two most important learning_reinforcers namely quality_of_teaching (shown on the top right of Fig. 9) and testing_ (shown on the left end of Fig. 9) have been defined in the model. Learning_reinforcers is modeled by a graphical function of type_of_info and quantity_of_info. On the other hand, testing_ comprises of rote_type, clsd_prbm_slvg, and opn_prbm_slvg components [2]. In fact the other less important types of external reinforcers have been represented by a single converter as other_reinforcers [1].

Also, to have a more solid model, two new components, productivity_ and availability_, have been conceptualized and introduced in the model. Note that the learner's productivity indicates the level of his or her learning effectiveness in the acquisition of new information. It is defined as the ratio of amount_learned to prior_knowledge (output to input) and, as pointed out, influences the learner's performance during the learning process. Availability_, on the other hand, has an important inter-relational role between the learner's *available* knowledge and his or her productivity_ from one side, and the impact of learning_reinforcers on availability_ (of

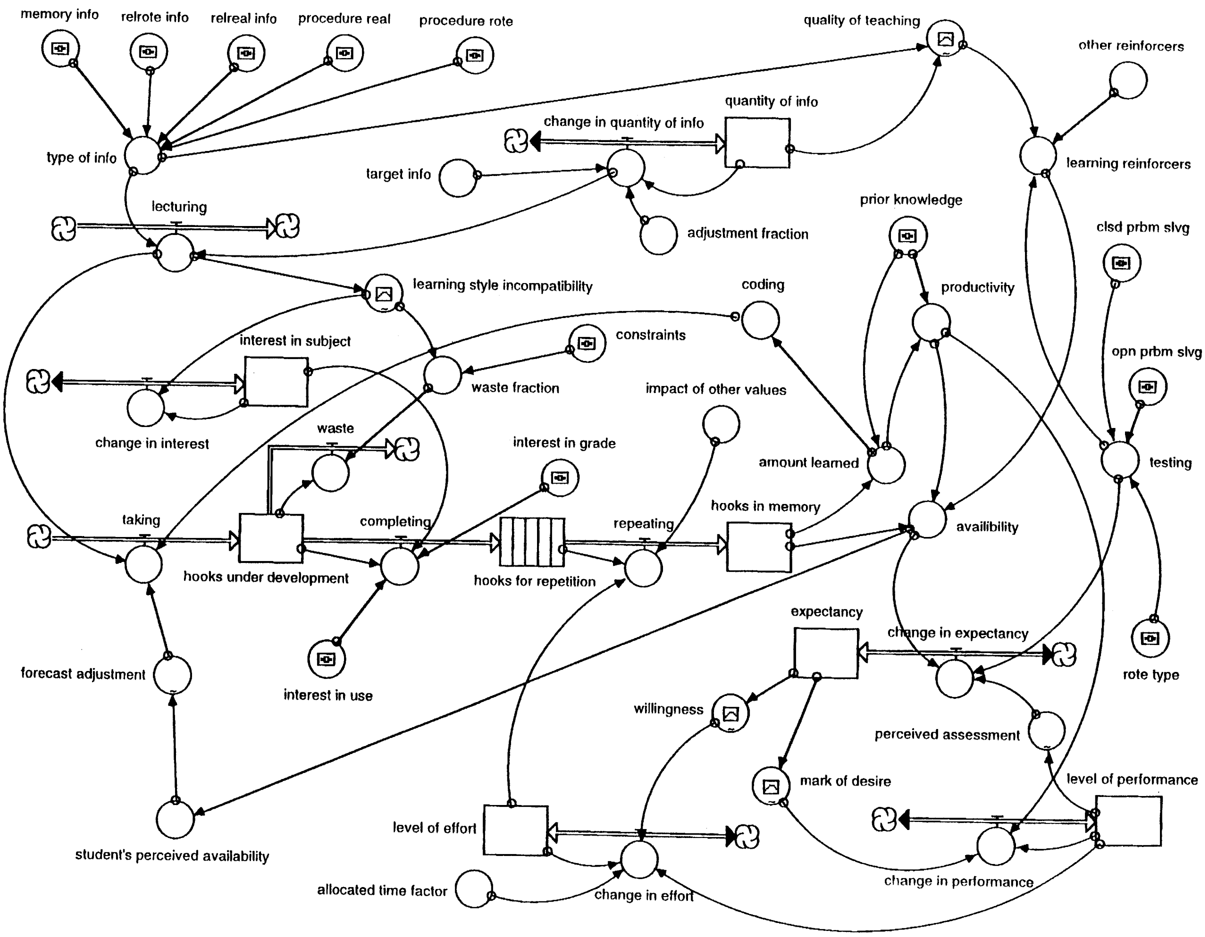


Fig. 9. System flow diagram for a form-oriented learner.

the current knowledge) from the other side. Availability_ connects the main chain of the student' learning abilities to three important dimensions of expectancy_, level_of_effort, and level_of_performance.

The remaining components, willingness_, mark_of_desire, perceived_assessment, student's_ perceived_availability, forecast_ adjustment, allocated_time_ factor, target_ info and adjustment_fraction, represent the other important characteristics within a teaching/ learning system and are defined and detailed in the List of Equations (Appendix) based on the previous discussions made in Part 1 and Part 2 of the study [1, 2].

FEEDBACK MECHANISMS

Several feedback mechanisms are included in the model (Fig. 9). Four of these loops have a determining effect on the resulting behavior of the system. Two loops are acting merely in the learner's ability (to learn) side. One loop is acting in the student's performance side (that demonstrates the impact of subject matter). And the last loop, which is the largest loop is acting in the teaching system side.

The learner's ability (to learn) mechanisms

The first mechanism acts along the main chain running from the stock of hooks_under_development to the stock of hooks_in_memory, and from there to the amount_learned and to the coding_ and finally back to the flow of taking_. This linkage closes a feedback loop in which as the amount of incoming information (lecturing_) increases, the form-oriented learner will take more and make more hooks_under_development. This leads to a higher rate of completing_ (the hooks and strings of information) and subsequently, more hooks_for_repetition. A higher number of hooks_for_repetition, inevitably increases the rate of repeating_ and the amount of hooks_in_memory respectively. Amount_ learned will increase and accordingly causes an increase in the categorizing and coding_ ability of the form learner. This, in return, will facilitate the flow rate of taking_. The reinforcement of taking_ is one of the feedback mechanisms included in the model for responding to changes in amount of incoming information. Thus, the feedback loop starts with an increase in the amount of hooks_under_ development and feedback to taking_ makes it increase more. This phenomenon is the characteristic of a positive feedback loop that tries to reinforce the process. As stated in the previous sections, when any variable in a positive loop changes, the resulting interactions cause that variable to change further in the same direction. The positive loop, in other words, characteristically produces self-reinforcing change (unrestrained growth).

The second mechanism acts in parallel to the first one, keeping the same track but diverts from hooks_in_memory to availability_. Thus, an increase in hooks_under_development, ultimately increases the hooks_in_memory. The result is an increase in availability_ (of the information) that gives rise to a higher student's_perceived_ availability. A higher perceived_availability, subsequently, decreases the student's forecast_adjustment. This, in return leads to a negative impact on the flow of taking_ and a subsequent decrease in the hooks_under_ development.

Summing up, the loop starts with an increase in the hooks_under_development and the feedback to hooks_under_development makes it decrease. This phenomenon is typical behavior for a negative feedback loop. As mentioned in the previous sections, when any variable in a negative loop is changed, then the loop causes that variable to readjust in the opposite direction. The negative loop produces self-regulating changes (controlling and restorative behavior). And so, an initial increase in the number of hooks_under_development propagates a signal around the loop, which leads to an eventual decrease in the level of this stock. The loop thus acts to counteract the initial change.

In summary, it is obvious that the overall behavior of the learning ability of the learner is almost the result of the interaction between these two feedback loops—a positive feedback loop that acts in the 'coding_' side and a negative feedback loop that acts in the 'availability_' side.

The student's performance mechanism

The other feedback loop that has a major effect on the behavior of the overall learning system acts along the student's 'performance' side (down to the right of the diagram). This loop may either act as a negative or a positive feedback loop. The way it works depends upon the direction (or the resulting direction) of changes in the involving biflow rates.

This loop starts with the stock of hooks_in_ memory. Note that an increase in hooks_in_ memory, concurrently, increases the amount_ learned (by the student). This, in turn, increases the student learning productivity_ and availability_ (of the knowledge) respectively. The result is reinforcement in the positive direction of change_in_expectancy which subsequently leads to an increase in student's expectancy_ for a higher achievement. The increase in expectancy_ has a direct impact on the student's expected mark_of_desire, and at the same time, on the willingness_ [1]. However, if change_in_ performance and change_in_effort biflows tend to be in the positive directions, then they result in subsequent increases in level_of_performance and level_of_effort respectively.

On the other hand, mark_of_desire and willingness_ are the target levels for level_of_ performance and level_of_effort respectively.

Therefore, the rate of change in each biflow depends directly on the difference between the value of each reservoir and the value of the corresponding target level. Eventhough, an increase in level_of_performance reinforces the positive direction of `change_in_effort` which subsequently increases `level_of_effort`. The increase in `level_of_effort`, strengthens the rate of `repeating_` and generates more `hooks_in_memory`. Needless to say, the resulting effect is a typical behavior of a positive feedback loop. As mentioned earlier, when any variable in a positive loop changes, the resulting interactions cause that variable to change further in the same direction.

Two points are worth mentioning. First, the biflow of `change_in_effort` is under the influence of `allocated_time_factor`. This converter represents the time dimension of the effort that a student puts into different learning tasks. Second, the biflow of `change_in_expectancy` is under the influence of `testing_`. Note that `testing_`, as discussed in the previous part of the study [2], comprises of three major types of questions:

(1) `rote_type` (true/false, multiple choice and short/long answer questions),
(2) `clsd_prbm_slvg` (closed problem solving type questions) and
(3) `opn_prbm_slvg` (open problem solving type questions).

The value of `testing_` in the model is defined by the following relation:

$$\textit{Value of } \texttt{testing\_} = \\ = \frac{w_r \times r + w_{cps} \times cps + w_{ops} \times ops}{w_r + w_{cps} + w_{ops}}$$

where

w_r = weightpercent of rote type questions
w_{cps} = weight percent of close problem solving type questions
w_{ops} = weight percent of open problem solving type questions
r = value of the rote type questions in the teacher's view
cps = value of closed problem solving questions in the teacher's view
ops = value of open problem solving questions in the teacher's view

In the base case of the model, it is assumed that a form-oriented teacher represents the teaching system. It is obvious that, form-oriented teachers normally intend to ask questions or take tests with higher weight percentages of the types of questions that they prefer the most (i.e., more rote type and less open-problem solving type). Conversely, function-oriented teachers normally intend to ask questions and take tests while they give more weight to the open problem solving and closed problem solving type questions.

In the meantime, each of these types has its value in the teacher's view: form-oriented teachers assign higher values to rote type questions while function-oriented teachers assign higher values to the problem solving types questions. By assumption, the following values (out of 5) have been considered for each type of question in a form and function's view respectively and may be used in the base model:

	Rote type questions	Closed-problem solving	Open-problem solving
Form teacher	5	3	0
Function teacher	0.5	3	5

Since both the form teacher and the form learner prefer the first two types of questions, the better a `testing_` represents these two types of question, the higher the `expectancy_` of a form-oriented student, and the higher s/he sets his or her `mark_of_desire`. Subsequently, the higher s/he sets his or her `mark_of_desire`, the better s/he performs. But as it was mentioned in the counteracting feedback relationship, the loop also may change conversely. That is, the less `testing_` represents the form student's preferred question types (if say, for instance, the teacher is a function-oriented individual), the lower his or her `expectancy_` for a better achievement and subsequently, the less his or her `mark_of_desire` and the worse s/he would perform.

The teaching side mechanism

The largest feedback loop mechanism acts along the '`teaching_`' side (top left) of the model. This loop, again, may either act as a negative or a positive feedback loop. The way it works depends upon the direction (or the resulting direction) of changes in the involving biflows. This feedback loop could be tracked as described below.

The '`teaching_`' loop starts with the stock of `quantity_of_info`. The level of this stock (the total accumulation of information presented at any time) is controlled by rate of `change_in_quantity_of_info`. The biflow of `change_in_quantity_of_info` may change its direction in either positive or negative side to regulate the level of the stock, based on the amount of `target_info` (as is preset by the teaching system for each lecture, here for instance in the base model, say it is set as 200 pieces of information) and `adjustment_fraction` (as is adjusted by the teaching system based on the feedback received from the student's `level_of_performance`).

As the level of `quantity_of_info` increases, provided that `type_of_info` is at its appropriate value for a form learner, the `quality_of_teaching` increases. The increase in `quality_of_teaching` has its reinforcing effect on the student's `learning_reinforcers` and subsequently on the student's `availability_` (of knowledge). The more the `availability_`, the greater the expectancy_, the higher the mark_of_desire, the better the performance, and finally, the larger is the adjustment_fraction. This means a higher flow rate of information to the stock of quantity_of_info. (Note that for the sake of having a neat diagram, the connector that links level_of_performance to the adjustment_fraction is not shown in Fig. 9).

It is worthwhile to notice the role of type_of_info in the teaching_ side of the model. In fact, it represents *what* is flowing, via lecturing_, from the teaching system to the learning side (learner). The different types of information within a lecture were recognized and classified in two sets; 'main set' and 'testing (auxiliary) set,' [2]. At this side, the model shows the types of information that include in the 'main set' namely memory_info (memorizing type—rote info), relrote_info (relationship type info—rote oriented), relreal_info (relationship type info—real), procedure_real (procedure type info—real) and procedure_rote (procedure type info—rote oriented). The four types of the 'testing set' have been reduced to three types and are shown as constituents of testing_ in the right end of the diagram (Fig. 9). They are part of learning_reinforcers like the quality_of_teaching_ and other_reinforcers that were dealt with earlier. The value of type_of_info is determined by the following formula:

Value of type_of_info =

$$= \frac{a \times w_a + b \times w_b + c \times w_c + d \times w_d + e \times w_e}{w_a + w_b + w_c + w_d + w_d}.$$

where

w = weight percent of each type of information in the lecture presented by the teaching system
a = memorizing type information
b = relationship type info—rote oriented
c = relationship type info—real
d = procedure type info—real
e = procedure type info—rote

In the base case of the model, it is assumed that a form-oriented teacher represents the teaching system. It is obvious that form-oriented teachers normally are giving lectures with higher weight percentages of the types of information that they prefer the most (i.e., memory type, relrote, and procedure-rote). Conversely, function-oriented teachers prefer these three types of information the less and are presenting lectures with higher weight percentages of relreal and procedure-real. In the meantime, each of these types has its value in a teacher's view. Form-oriented teachers assign higher values to a, b and e while function-oriented teachers assign higher values to c and d. By assumption, the following values (out of 5) could be considered for each type of information in a form and function's view respectively and may be used in the base model:

type_of_info	Form teacher	Function teacher
memory_type	5	1
relrote	4	2
relreal	1	5
procedure_real	1	5
procedure_rote	5	1

In general, the last two feedback loops described above act on two sides of the model (teaching and learning) and due to the bi-directional effect of their biflow rates, seek eventually either goal maintaining or a growing pattern. At the same time, some smaller loops exist in the model that behave locally and generate their limited effect on the system. Consequently, the overall behavior of the teaching/learning process is the resulting behavior produced by all of the mentioned loops.

BEHAVIOR OF THE SYSTEM

Referring to Fig. 9, and List of Equations (Appendix), it can be seen that the simulation starts with an initial stock of hooks_under_development at 0 hooks, a prior_knowledge of 10 hooks about the subject matter, an initial interest_in_subject of 0.01 (1%) and an initial quanity_of_info of 0 hooks. The other initialization values are as defined and assumed in the List of Equations (Appendix). The time horizon for the model is assumed to be 45 normal minutes, namely the length of a regular lecture.

As the lecture starts, the flow of lecturing_ sends the desired quantity_of_info to the mind of the learner. The form learner begins to receive the information at the rate of taking_. At this stage, as each piece of information moves to the stock of hooks_under_development, it will be classified, coded, and adjusted as well. The stock of hooks_ under_development represents the learner's short-term memory (STM) in real life. To be consistent with the reality, the model assumes that some information is leaked from the stock and has gotten lost during the information taking process at the flow rate of waste_. As discussed earlier, the rate of waste_ is controlled by waste_fraction. Again, as mentioned earlier, the value of waste_fraction is determined by two factors: learning_style_incompatibility and the amount of constraints_ . Note that the lower learning_style_incompatibility (say, for instance, both teacher and learner are form-oriented individuals), the smaller the waste_fraction, the lower the rate of waste_ and the higher is the level of hooks_under_development.

Therefore, as lecturing_ proceeds, the flow of taking_ sends more hooks to the stock of hooks_under_development. The flow of completing_ takes the information from the upstream stock and places them as the completed hooks in the stock of hooks_for_repetition. The rate of completing_ is reinforced by three factors: the level of student's interest_in_subject (shown as a stock), and the amount of both interest_in_grade, and interest_in_use (shown as converters). If the type of information received by the form-oriented learner is compatible with his or her preferences (almost rote types), then learning_style_ incompatibility between the teaching system and the learner would be at its minimum. Again, the lower the value of learning_style_ incompatibility, the lower the waste_fraction, and the less is the variation in the level of student's interest_in_subject.

The stock of hook_for_repetition is represented by a conveyor type stock. The time that it takes that each piece of information finds its location in the episodic memory of the learner's long-term memory (LTM) has been considered as a variable. The in-flow capacity and transit time for the conveyor vary for different form-oriented learners. The maximum inflow capacity for the hooks_for_repetition in the base model is assumed to be at most 20 hooks of information. Also, the transit time (the time it takes that each piece of information gets off the conveyor) for the conveyor is assumed to vary. (Refer to the List of Equations in Appendix)

The flow of repeating_ takes each hook of information from its upstream stock and after required repetition implants it into the student's LTM (hooks_in_memory) as a permanent trace. The rate of repeating_ is regulated by the stock of student's level_of_effort and is reinforced by the impact_of_other_values. Note that three of the student's perception of task value (interest_in_subject, interest_in_grade, and interest_in_use) have been already defined and modeled in the diagram (Fig. 9). The impact of the remaining six values (pride in future profession, self-worth, security in future job, social obligation, bandwagon effect and association with something one likes—as discussed in Part 1 of the study [1]) are represented by a single converter for the sake of simplicity.

Maximum level of both hooks_under_development and hooks_for_repetition happen between minute 8 and 9. This can be found from Fig. 10 (graph of base run for a form-oriented learner). This fits the reality quite well, especially when one notices the large amount of new information that is usually presented by the teacher right at the beginning of each lecture. Normally, as the lecture proceeds, the rate of the presentation of new information decreases and the content of the lecture, more or less, is focused around the expansion of the topics that are presented in the beginning of the lecture.

The maximum level of hooks_in_memory happen at the end of the simulation period and is approximately about 15 hooks. The maximum interest_in_subject is about 0.7 (out of 1.0) and again happens at the end of the lecture. Also, the level_of_performance reaches its maximum; 60 out of 100, at the end of the period. (See Fig. 10.)

The simulation may be run for the analysis of other variables as well. Note that Fig. 11 demonstrates a second graph of base run for the other five major variables. As shown in the graph, the flow rate of waste_ reaches its maximum at minute 7 and then keeps descending at an almost uniform rate as the lecture proceeds to the end. The reason why the rate of waste_ is at its maximum at minute 7 may be found in the large amount of new information that is normally delivered by the teaching system in the beginning of the lecture. The sharp ascending pattern of the graph for the first 7 minutes fits nicely with what is happening in

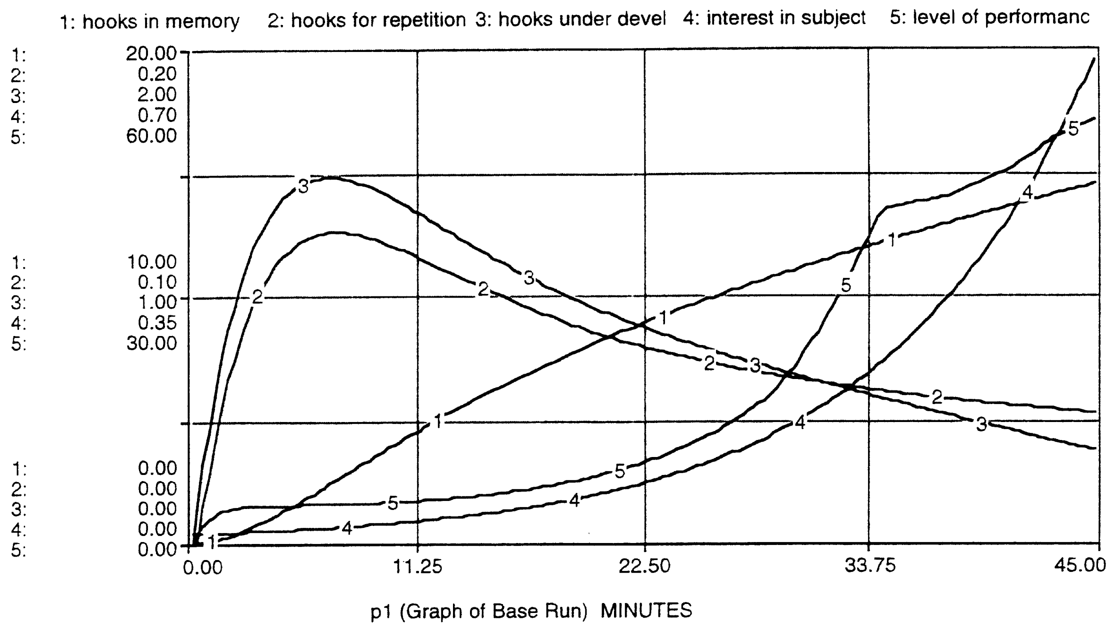


Fig. 10. Graph of base run for form-oriented learner (Part I).

the reality. Interestingly, as the teacher gives more explanation about the topics (learning objectives), which this usually happens after first 5–7 minutes, the rate of `waste_` then decreases.

Two variables, `amount_learned` and `productivity_` follow a common track. Both variables show a continuous increasing trend. These two variables have been defined, simply by using the cognitive algebra concept, as below:

`amount_learned` = `prior_knowledge` + hooks_in_memory

`productivity_` = `amount_learned`/ a prior_knowledge

Note that the simulation begins with `prior_knowledge` of 10 and `hooks_in_memory` of 0 hooks. This means that the initial `productivity_` is equal to one. At the end of lecture, the form-oriented learner acquires 14.5 more hooks and consequently the `productivity_` ratio increase to 2.45.

Also, the flow of `taking_` starts at an initial rate of 0.7 hooks per minute and ends at a rate of 0.2 hooks per minute. In the mean time, the `quantity_of_info` delivered by the teaching system starts at 0 and is accumulated at the final level of 200 pieces of information by the end of the lecture. The patterns of behavior of these two variables over time seem very promising. As the lecture proceeds, the learner gets more detailed information about the topics at hand, becomes more familiar with the subject matter, and consequently adjusts (reduces) his or her rate of `taking_` accordingly.

As mentioned earlier, the behavior of any other variable can be simulated and tracked on the similar graphs (or tables). The graph of base run in Fig. 10 shows the behavior of all of the variables involved in the simulation model over 45 minutes of a typical lecture period. The interested reader can refer to this table and observe how each variable within the base model changes value minute after minute.

EXPERIMENTAL RUNS

A number of experiments were carried out with the simulation model in line with the step 4 and 5 of system dynamics method [1]. The intention was to examine different policy alternatives and determine which policies show the greatest promise. The alternatives were chosen mainly from the experience of the analyst and also from intuitive insights generated during the first three stages of the system dynamics. Although, in a complex system like teaching/learning, there would be many competing criteria for defining failure or success, nevertheless different scenarios of favorable performance might be identified. In addition, the better alternative behaviors would often come from changing the system base structure.

To be concise, only four experiments are discussed in this section. These simulation experiments were carried out to gauge the effects of `prior_knowledge`, `memory_info`, and `rote_type` (questions) as policy variables on the learning behavior of the form-oriented learner. All of these variables have been chosen intentionally. `Prior_knowledge` represents one of the student's trait variables in the model, while `memory_info` and `rote_type` represent the characteristics of subject matter and teaching system respectively.

- **Experiment 1:** A sensitivity analysis was made of the student `prior_knowledge` for 10, 20 and 30

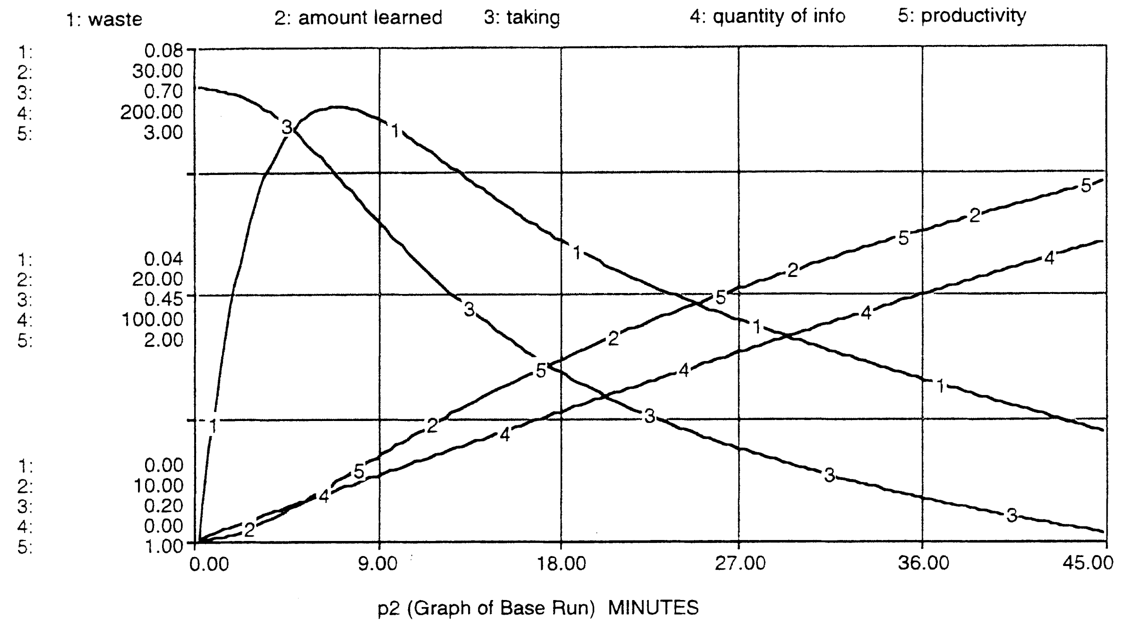


Fig. 11. Graph of base run for form-oriented learner (Part 2).

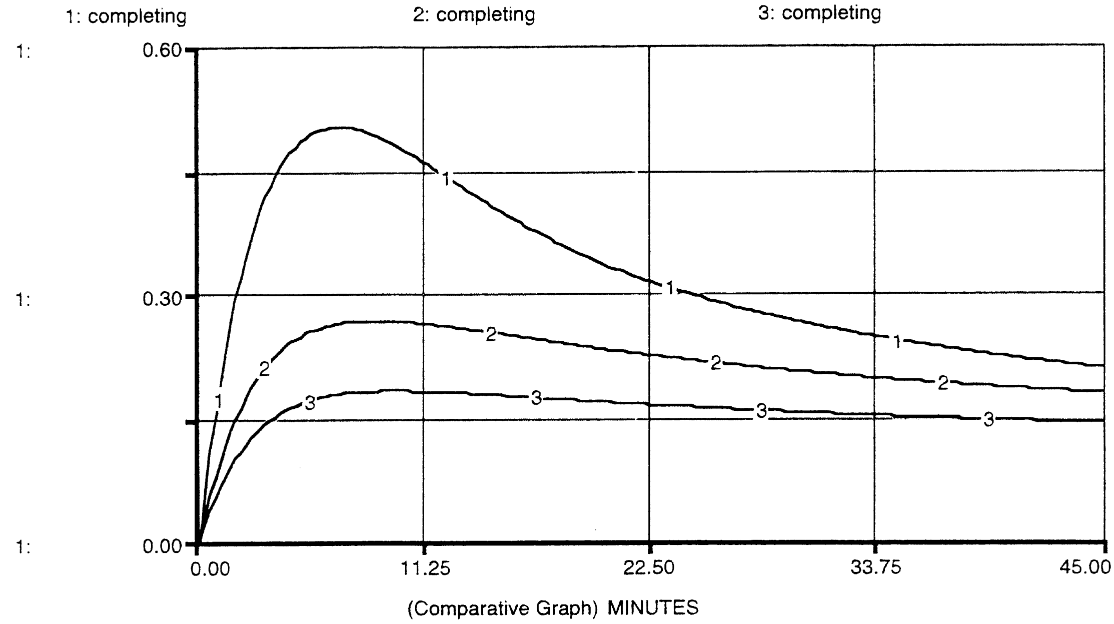


Fig. 12. Comparative graph for experiment 1.

hooks of information to gauge its effect on the rate of `completing_` (Fig. 12).

- **Experiment 2:** A sensitivity analysis was made of the `memory_info` for the weight factor of 5, 10, and 15 to gauge its effect on the `productivity_` of the form-oriented learner (Fig. 13).
- **Experiment 3:** A sensitivity analysis was made of the `memory_info` for the weight factor of 0, 5, and 10 to gauge its effect on the `level_of_perormance` of the form-oriented learner (Fig. 14).
- **Experiment 4:** A sensitivity analysis was made of the `rote_type` (questions) for the weight factor of 0, 5, and 10 to gauge its effect on the student's `expectancy_` for a `mark_of_desire` (Fig. 15).

Effects of other policy variables like `interest_in_subject`, `learning_style` incompatibility, `quality_of_teaching`, `level_of_effort`, and `other_reinforcers` have been investigated as well. However, the discussion of results of the above four experiments (next section) would suffice and serve the purpose of this study.

Results of experiments

To examine the results of the above experiments, the data at four points of interest (minutes 11.25, 22.5, 33.75 and 45.0) were extracted from Figs 12, 13, 14 and 15 respectively, and tabulated as shown in Table 2. The similar results of the base run are also included, so then the changes in the learning

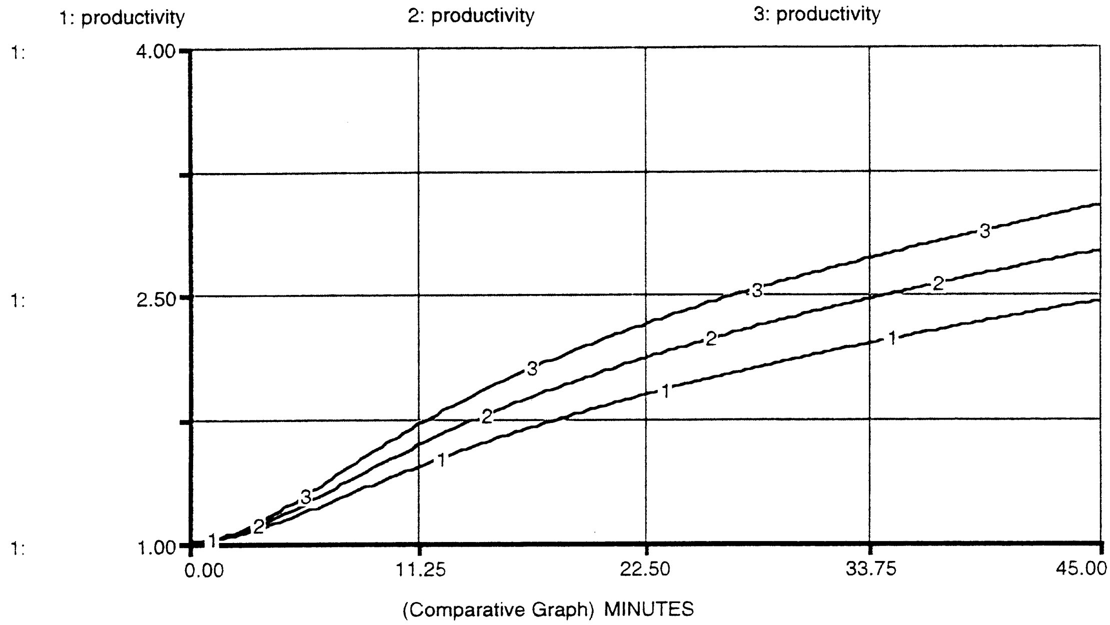


Fig. 13. Comparative graph for experiment 2.

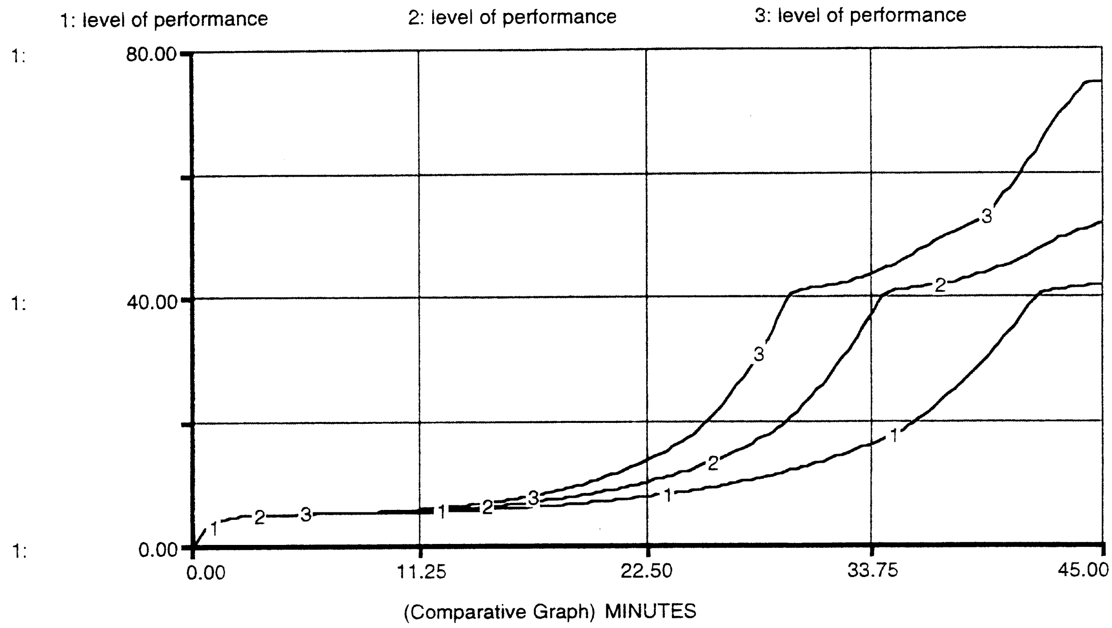


Fig. 14. Comparative graph for experiment 3.

'behavior' of the form-oriented learner would be more obvious.

Rate of `completing_`, `productivity_`, `level_of_performance`, and `expectancy_` are taken as the measures of change in the behavior of the system. These choices look reasonable as everything runs on the rate of acquisition of knowledge (here, in this case, on the rate of `completing_` new hooks of information in the memory). Besides, the level of the acquisition of knowledge could be evaluated based on the `productivity_`, `level_of_performance`, and the `expectancy_` of the form-oriented learner (for his or her `mark_of_desire`).

The results of Experiment 1 indicate that, the higher the initial level of prior-knowledge about the subject, the lower the student's rate of `completing_` would be. This means that a form-oriented learner with more `prior_knowledge` about the subject at the beginning of the lecture, is more 'efficient' in absorbing the new incoming pieces of information and hence, is more relaxed in processing the information (here, read it as: slower in the rate of `completing_` hooks of information in his or her memory).

The difference in the rate of `completing_` is more evident at the beginning of the lecture. As shown in Fig. 12, for all of three runs, the rate of

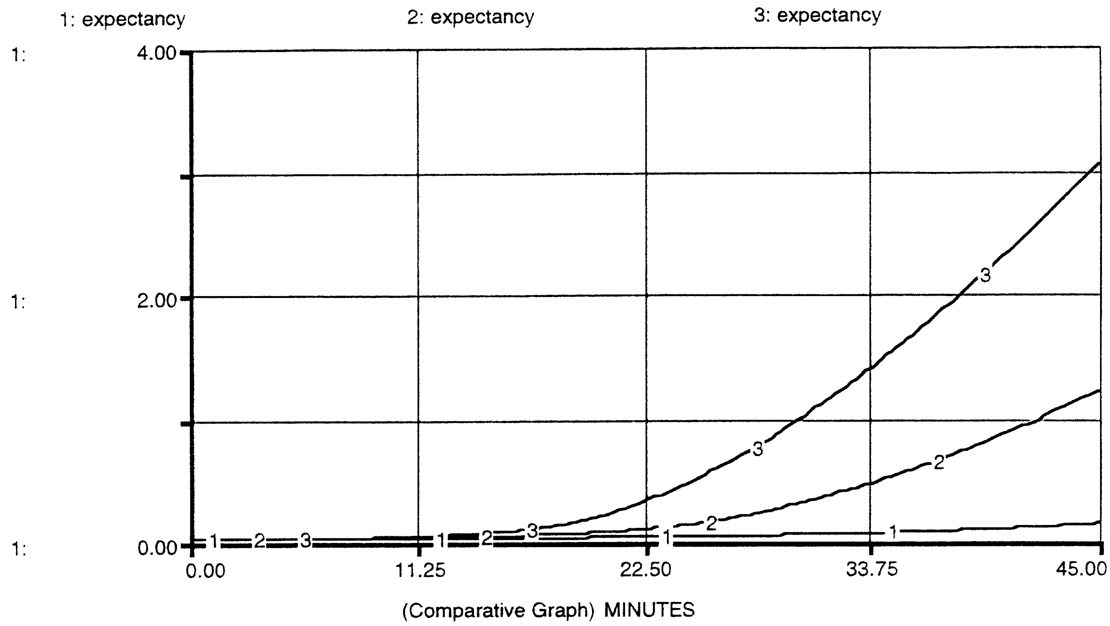


Fig. 15. Comparative graph for experiment 4.

Table 2. Results of sensitivity analysis

Measure of behavior	Minute 11.25	Minute 22.5	Minute 33.75	Minute 45
Base run (prior_knowledge = 10.0)				
completing_	0.46	0.31	0.24	0.21
productivity_	1.45	1.885	2.19	2.45
level_of_performance	4.97	9.60	38.38	51.44
expectancy_	0.01	0.05	0.23	0.66
Experiment 1 completing_	0.46	0.31	0.24	0.21
prior_knowledge: Run 1 = 10.00	0.26	0.23	0.21	0.18
Run 2 = 20.00	0.18	0.17	0.16	0.15
Run 3 = 30.00				
Experiment 2 productivity_	1.46	1.90	2.20	2.45
memory_info: Run 1 = 5.00	1.65	2.12	2.48	2.75
Run 2 = 10.00	1.73	2.32	2.73	2.98
Run 3 = 15.00				
Experiment 3 level_of_peformance	4.90	7.70	16.0	42.0
memory_info: Run 1 = 0.00	4.97	9.60	37.33	51.44
Run 2 = 5.00	5.05	14.0	44.0	75.0
Run 3 = 10.0				
Experiment 4 expectancy_	0.08	0.10	0.15	0.022
type_of_info: Run 1 = 0.00	0.09	0.15	0.50	1.30
Run 2 = 5.00	0.10	0.38	1.40	3.12
Run 3 = 10.00				

completing_ reaches its maximum in the first ten minutes of the lecture and then keeps decreasing for the rest of the lecture. Note that this pattern of behavior is quite consistent with what happens in reality in the teaching/learning environments. Upon beginning a lecture, the teacher usually starts with the presentation of the topics and introduction of the learning objectives. Then, s/he uses the rest of the time of class, to expand around each topic and to go into the details. The student, on the other side, knows well that s/he must build more hooks of information in the beginning to hang the other information incoming later onto them. (Refer to Part 2 of this study, Figs 3(a) and (b) and see the example short lecture experiment that shows how a form-oriented learner treats incoming information in the beginning of the lecture [2].)

Comparing the values of completing_ at minutes 11.25, 22.5, 37.75, and 45 implies another finding. With a prior_knowledge of 10 hooks (base run), a student has a harder job to do in contrast with a student with a prior_knowledge of 20 or 30 hooks. In fact, as the prior_knowledge about the subject increases, the rate of completing_ shows a more promising pattern of behavior. For instance, the rate of completing_ for a student with prior knowledge of 10 (Run 1 in Experiment 1) varies in a larger span than of the student with a prior_knowledge of 20 (Run 2 in Experiment 1) or 30 (Run 3 in Experiment 1). This can be seen in the diagram of Fig. 12. The rate of completing_ for Run 1, starts at 0 in the beginning, reaches its maximum (0.50) at minute 7, and ends up with 0.21 hooks per minute at minute 45. Compare these values with the values of completing in Run 3. Here, the rate of completing_ starts at 0 in the beginning, reaches its maximum (0.18) at minute 8, and ends up with 0.15 hooks per minute at minute 45. What are the differences? The form-oriented learner in Run 3 has an average completing_ rate of about 0.15—0.18 hooks per minute over the whole period of the lecture except for the first few minutes, (which is normally expected). This, of course, may be interpreted as less pressure on the student's mind and more stable behavior in the process of knowledge acquisition. Run 2 shows a similar pattern to Run 3.

On the other hand, according to the results of Experiment 2, as the teacher puts more value on the memory_info and delivers a lecture with a higher content of memory type information, the form-oriented learner's productivity_ would be higher. As shown in Fig. 13, doubling the memory_ info content (from 5.00 to 10.00) would result in, more or less, about 10% increase in the student's productivity_. Even another increase (this time 50%) would give a better productivity_ (Run 3 in Table 2). Again, this fits very well with the reality if one notices that a form-oriented learner is highly productive when s/he receives information in his or her type of preference. Worth mentioning is that if the teacher uses other rote type information like relrote_info and procedure_rote in his or her lecture, the pattern of the behavior would be similar to that shown in Fig. 13. Conversely, if the teacher gives a lecture with more stress on relreal and procedure_real, the form-learner definitely will be in trouble and his or her productivity_ will decrease.

Experiment 3 is to gauge the effect of the same input variable (memory_info) on the student's level_of_performance. As shown in Fig. 14, if the teacher uses no memory_type_info in his or her lecture, the form-learner's achievement falls below the 'passing zone' and is about 42%. In such cases, the teacher most likely is a function-oriented individual and the form-oriented student will be at risk. In contrast, as the teacher uses memory_type info with 5 or 10 weight factors, the student's level_of_performance increases to 51.44% and 75% respectively. Interestingly, the doubling of memory_info content (from 5 in Run 2 to 10 in Run 3) results in about 50% increase in the student's level_of_performance. The model assumes that the continuous assessment of the student's achievement during a lecture is feasible and practical.

Experiment 4 is complementary to Experiment 3 and demonstrates the effect of different types of testing_ on the student's expectancy_. Testing_ could be occasional short oral questions during the lecture or a written short quiz. The important issue is the type (orientation) of testing_. If the teacher asks no rote_type questions (memory oriented yes/no, true/false, short/long answers qestions) in his or her testing_, the form-oriented learner presumes a lower level of expectancy_ for success (in getting a passing grade). In this case, the teacher most likely is a function-oriented individual and hence, the form-oriented student would be definitely at risk. Apparently, as the teacher uses rote_type questions in his testing_, say for instance at a weight factor of 5 (Run 2 of Fig. 15), then the student's expectancy_ rises considerably and results in a higher mark_of_desire accordingly. Furthermore, if the teacher doubles the amount of rote_type questions (Run 3 of Fig. 15), the student's expectancy_, more or less, increases by 150% (Table 2). Once more, one can observe the role of different types of issuing information by a teacher; whether they are part of 'main set' or 'auxiliary' (testing) set (as defined in Part 2 of this study [2]), on the student's achievement.

LEARNING MODEL OF A FUNCTION LEARNER

Description of the base model for a function learner

As mentioned in the beginning, the system flow diagram for the learning process of a function-oriented learner is the same as Fig. 9 except for the main chain of the model. (Refer to Table 1: Defined variables for the proposed models of teaching/learning process.) To be brief and to prevent mentioning repetitive material, in this section, only the main differences are discussed.

The main chain of the system flow diagram for the learning process of a function-oriented learner is shown in Fig. 16. The definition of each variable except for the main chain is the same as described in the List of Equations (Appendix). The definition of each variable in the main chain is given as the following.

Referring to Fig. 16, the main chain infrastructure represents a sequence of stages through which the information flows in the mind of a function-oriented learner. Note that the specific nature of flows varies, depending on the specific situation of each stock. The chain is fed by a single flow (taking_). A non-conserved system is demonstrated by the stock of relationships_under_study. The cloud on the left hand side of the flow of taking depicts the boundary of the model. It represents an infinite source for the taking flow, as shown. (Again, for the purpose of this model, it does not matter what is in the cloud.)

In this model, flow or taking_ is receiving each piece of the incoming information from the teaching system and placing it into the mind of the function-oriented learner (stock of relationships_under_study). The flow of waste_ drains the stock of relationships_under_study at a rate that is controlled by the level of the stock itself and the waste_fraction (not shown in the diagram).

As shown in the diagram, an oven type stock represents the state of structures_to_form in the mind of a function-oriented learner. Two interesting points are worth mentioning. First, by assigning different capacities and fill time to the oven,

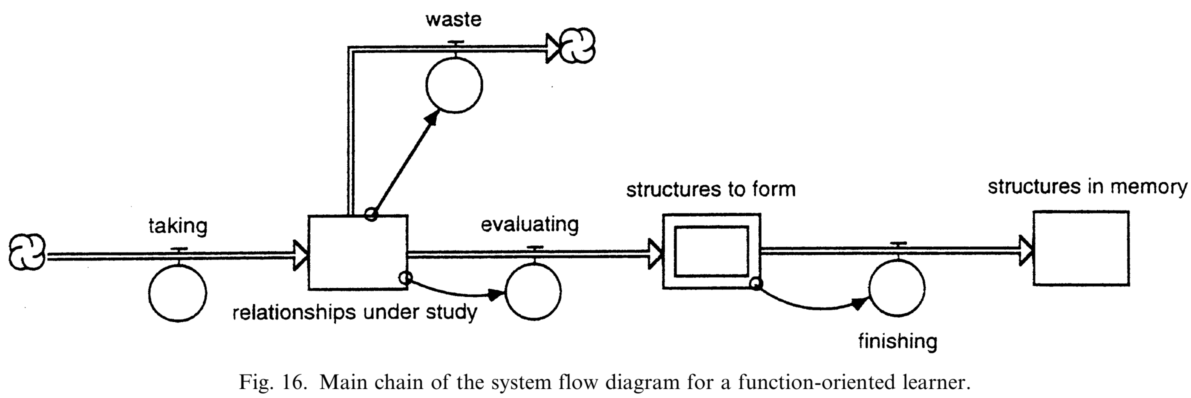


Fig. 16. Main chain of the system flow diagram for a function-oriented learner.

different types of function-oriented minds can be detailed and modeled. Capacity tells how much information the oven can hold. The oven will close its doors and begin processing its contents when capacity is reached, or fill time is expired—whichever comes first. Note that both capacity and fill time may be assigned small or large values, and so, make the analyst's job easy or difficult. Second, oven cook time (processing time) can be set to a constant or it can be made variable. In so doing, different situations can be defined for a function-oriented learner. For example, if the teacher is a form-oriented individual and delivers memory-type information rather than relationship-type information (which is the student's more preferred type of information), the oven can be set to use long fill times and small capacity to represent capacity constrained situations.

Finally, the flow of `finishing_` completes the learning task by taking each would-be-structure from the stock of `structures_to_form` and placing it into the stock of `structures_in_memory`. The flow of `finishing_` is under the direct influence of the amount of the `structures_to_form` and `level_of_effort` (not shown in the diagram). In other word, acquisition of knowledge is viewed as student effort-based activity as well. The function-oriented type student acquires knowledge when s/he puts effort into the learning task over a certain period of time (as defined).

Feedback mechanisms

The same number of feedback mechanisms as was discussed for the model of form-oriented learner are included in the model. Similarly, two of these loops have a major effect on the resulting behavior of the learner. Also, as discussed earlier, both of these loops act in parallel along the main chain, running from the stock of `relationships_under_study` to the stock of `structure_in_memory`, and from there each diverts in a different direction. For the sake of brevity, the material will not be repeated here.

Behavior of the system

The behavior of the system is the same as discussed in the previous section for the form-oriented learner. By assigning initial values to each of the stocks and converters, the simulation model may be run. The initialization values could be defined and assumed in the same way as described in the List of Equations for the form-oriented learner (Appendix). The time horizon for the model is assumed to be 45 minutes, that is, the length of a regular class lecture.

The only main difference between the two models is the nature of the pieces of information that is flowing through the main chain. In the case of a form-oriented learner, the nature of knowledge was based on the hooks of information. In contrast, in the case of a function-oriented learner, the nature is based on the structures of information. The STELLA model has properly taken care of this difference. The use of a conveyor-type stock in the form-oriented case and an oven-type stock in the function-oriented case, account for this major difference.

ASSUMPTIONS AND SIMPLIFICATIONS

The proposed form-function model of teaching/learning in this study is the first attempt in a chain of models that may evolve from this study later. The structure of the present model provides a principal backbone for the future models that necessarily will have more complicated components and linkages. However, while the present model includes all the major variables of a teaching/learning process, it is founded on a few assumptions to maintain its simplicity at this stage. These assumptions as well as simplifications are as follows:

1. Because of the imprecise non-physical nature of a teaching/learning process, any attempt to model this process in a quantitative manner must be influenced by the subjective experiences, backgrounds, and beliefs of the modeler. Therefore, in the system dynamics model presented here, one must expect a degree of subjectiveness in the selection of variable values used in the equations. The values are based on the 'best judgment' of the authors. Clearly, any other researcher might end up with a different set of values. This in no way invalidates this work.
2. Since modeling is an emerging process, any 'model' represents only one of a sequence of models, that provide insight to the situation and form a basis for continued evolution. The model worked on in this study is presented in this spirit. This model is to be viewed as a vehicle that can be used to identify the important dimension of form-function orientation for implementing policies and tracing the resulting behavior of a teaching/learning process.
3. The focus in the proposed model is mainly on the learner or learning side of the system. The reason can be seen in the fact that the two other sides of the system (teaching system and subject matter) have complementary roles in a teaching/learning system and serve the learning side. Hence, the characteristics of the teacher system and the subject matter are not defined and detailed like the learning side in the proposed model. Each of these sides should be detailed and worked out in a sub-model with its characteristics' constituents.
4. The obtained results are valid only for the particular student under the conditions and limitations defined in the boundary of the system. Each individual student, whether form-oriented or function-oriented, has his or her particular traits that in similar situations

may give or not give rise to identical pattern of behavior.

5. The only student's task value that is represented by a stock-flow combination in the proposed model is `interest_in_subject`. The two other major student's task values namely, `interest_in_use` and `interest_in_grade` have been introduced by simple converters. Although, it is assumed that these two values remain constant during the time of study (i.e., during a 45 minutes normal lecture), nevertheless, this assumption is not so far from reality. Considering their negligible change in short-term, these values can not vary much to any extent during a limited lecture period.
6. Also, all of the other values (pride in future profession, self-worth, security in future job, social obligation, association of the task with something one likes, choice of subject, and bandwagon effect) are presented with a single converter (`other_values`). Each of these values is a complex variable that demands to be defined separately and be assigned an appropriate weight factor. Needless to say, some of these variables have reciprocatory inter-relationships with each other.
7. `Willingness_`, despite its complexity, also has been represented by a single entity. This major learning driver should be demonstrated in its own stock-flow combination. To reduce the weight of this inadequacy in the present model, willingness_ is defined as a graphical function and is represented by a graph that is a function of changes in the student's `expectancy_`.
8. Only two external reinforcement factors, `quality_of_teaching` and `testing_` (method of assessment) have been defined in the model. The other five factors (institutional factors, nature and content of the task, feedback from the teacher, satisfaction with the university, and interpersonal relations) are represented by a single converter (`other_reinforcers`). In a more complete model of teaching/learning, these factors should be represented by separate entities. Also, the interactions between the reinforcement factors themselves have not been shown (i.e., effect of `quality_of_teaching` on `testing_` and vice versa). However, the effect of these interactions in the short term (during a lecture period) is minor and may be neglected.
9. 'Knowledge' is the sole content of all of the flows and stocks that are located on the main chain of the model. It is assumed that the unit of knowledge taking, knowledge processing, and knowledge storing for a form-oriented learner is well represented by 'hooks of information.' On the other hand, the unit of knowledge taking, knowledge processing, and knowledge storing for a function-oriented learner is assumed to be well represented by 'structures.'

Other miscellaneous assumptions and simplifications that have been made include but are not limited to: absence of some environmental variables in the model; continuity of `testing_`; introducing `productivity_` (that seems somehow in conflict with the basic concept of productivity); `availability_` (that does not seem to be a perfect term for the concept it represents); simple approach to the definition of `amount_learned` and some other minor items.

SUMMARY AND CONCLUSION

The modeling effort made on the learning process in this study is a unique combination of educational metrics and engineering simulation programs. On the one side, the work consists largely of inferences drawn from available educational experience and viewpoints with an absence of a defensible, universal mechanism. On the other side, it heavily relies on a series of activities drawn from a methodology of system dynamics to build a solid engineering framework for the reinforcement and improvement of the process.

The sensitivity analyses discussed earlier in this study, demonstrate that the behavior of the proposed model seems quite persuasive and promising. At the same time, the four example experiments attest the strength of the system dynamics approach in predicting changes in behavior of a learner due to using different policy actions.

Two dynamic models of teaching/learning worked out by this study are based on continuous phase-type movement of information from the issuing origins (different types of teaching systems) to the receiving destinations (different types of learners). The main result from the models reveals that the advance knowledge about the types of teachers and learners (form-function orientations) warrants an efficient re-engineering of the teaching/learning system. In other word, the types of learners and teachers, whether they are form-oriented or function-oriented, has a major impact on the performance of the system.

The important characteristic of the methodology used here is its power to show the insight of the system or the understanding of what is happening in the system. As one can observe, unlike methodologies that focus only on an ideal future condition for a system, system dynamics reveals the way one arrives at the present and then, in a later step, the path that leads to improvement.

The simulation tests described in this study, determine which policies show the greatest promise and how the study can work toward a consensus for implementation of the policies. Influence of a combination of two or more policies on the behavior of the system can be examined as well. For instance, in the model of a form-oriented student, a learner at prior-knowledge of 20 interacting with a `type_of_info` of 10 (issued from the teaching system) can be taken as an alternative option. In general, by comparing the resulting behavior of the learning system under different options, the most appropriate policy or course of action can be identified. This step would eventually direct the study to the last step of system dynamics [1]. In fact, this study is now at a position that can make a conclusive statement related to the results of the experiments. The conclusive statement will clarify the standpoint of this study on how one can implement changes in the policies and structure of a teaching/learning system for the purpose of its improvement.

One can thus conclude that: the base model for a teaching/learning system and all of the experiments performed by the study on the base model, were strongly under the influence of the *form-function dimension* of the learners and the teachers. This dimension is so powerful that it has a primary role in analyzing any teaching/learning process.

The intensive literature search by this study bears witness to the many dimensions and aspects that exist in the field of teaching and learning. However, what this study has done is to highlight a dimension that by itself takes into account all of the other dimensions (i.e., lecturing, discussion, demonstration, etc.).

Implications

This study can help decision-makers select a performance parameter that will optimize a given policy variable in a teaching/learning process. The effect of system configuration (form-function orientation of individuals) on the performance of the system can be used to influence the design of the system before the planning stage is implemented. This information is critical in developing an efficient system in both academia and industry.

For technical learning it is extremely important that the learning structure emphasizes function-learning orientation. The required degree or intensity of the function-learning orientation for each technical discipline may be considered as an interesting area of research. The field of engineering education has the necessary and sufficient capacity for further investigation in this area. Besides, in other fields, analysts and researchers have to find a similar measure (or measures) regarding to what extent the learning structure should emphasize on form-orientation.

Lastly, with the discovery that this study has made, it is predictable that the education and industry sectors, in general, develop a more cost-effective human resources strategy on the one side, and higher quality products on the other side. The challenge is that each organization starts from its own employees and examines their form-function orientations to specify whether they fit the nature and requirement of their particular jobs or not.

REFERENCES

1. N. Eftekhar, Strong D., and O. Hawaleshka, Towards dynamic modeling of a teaching/learning process: 1—the unified theory, *Int. J. Eng. Educ*., **12**, (6), 1997.
2. N. Eftekhar and D. S. Strong, Towards dynamic modeling of a teaching/learning system 2: forms and functions, *Int. J. Eng. Educ.*, **14**, (6), 1998.
3. Barry Richmond et al., *An Introduction to System Thinking: STELLA Research*. HPS Inc., NH (1996).
4. S. Peterson and B. Richmond, *STELLA Technical Documentation*, HPS, Inc., NH (1996).
5. N. Eftekhar and D. S. Strong, Dynamic modeling of a learning process, *Int. J. of Eng. Educ.* (Full paper is available on: http://www.ijee.dit.ie/articles/999995/article.htm)

APPENDIX

List of equations

□ expectancy(t) = expectancy(t - dt) + (change_in_expectancy) * dt
INIT expectancy = 0.01

DOCUMENT: [1] Expectancy stems from the learner's self confidence which in fact sets his/her mark_of_desire (target level of performance) and willingness (target level of effort). How confident student feels is determining both how well student thinks he should be able to perform as well as how much effort he will put out in order to achieve the level of performance.

INFLOWS:

change_in_expectancy = availability*testing*perceived_assessment/100
DOCUMENT: [1/min] Rate of change in expectancy (in the well-doing of the task) or the rate of change in self-confidence

hooks_for_repetition(t) = hooks_for_repetition(t - dt) + (completing - repeating) * dt
INIT hooks_for_repetition = 0
TRANSIT TIME = varies
INFLOW LIMIT = 20
CAPACITY = 20

DOCUMENT: [H] stock of hooks ready for repetition at current completing rate

INFLOWS:

completing = hooks_under_development*(interest_in_subject+interest_in_use+interest_in_grade)/3
DOCUMENT: [H/min] assuming it takes 3 seconds or 1/20 minute to finish a total new hook or extend a pre-exited hook.

OUTFLOWS:

repeating = CONVEYOR OUTFLOW
TRANSIT TIME = hooks_for_repetition*level_of_effort*impact_of_other_values
DOCUMENT: [H/min] Assuming rate of repeating the available hooks for implanting in the episodic memory.

□ hooks_in_memory(t) = hooks_in_memory(t - dt) + (repeating) * dt
INIT hooks_in_memory = 0

DOCUMENT: [H] total number of active hooks in episodic memory at this time

INFLOWS:

repeating = CONVEYOR OUTFLOW
TRANSIT TIME = hooks_for_repetition*level_of_effort*impact_of_other_values
DOCUMENT: [H/min] Assuming rate of repeating the available hooks for implanting in the episodic memory.

□ hooks_under_development(t) = hooks_under_development(t - dt) + (taking - completing - waste) * dt
INIT hooks_under_development = 0

DOCUMENT: [H] Number of hooks of information under construction or extension in the student's mind.

INFLOWS:

taking = forecast_adjustment*lecturing/coding
DOCUMENT: [H/min] assuming the starting rate of building new hooks of info or extending the existing hooks of info.

OUTFLOWS:

completing = hooks_under_development*(interest_in_subject+interest_in_use+interest_in_grade)/3
DOCUMENT: [H/min] assuming it takes 3 seconds or 1/20 minute to finish a total new hook or extend a pre-exited hook.

waste = hooks_under_development*waste_fraction
DOCUMENT: [H/min] Rate of waste of information due to the impact of waste fraction.
Waste fraction is controlled by the compatibility of learning style

□ interest_in_subject(t) = interest_in_subject(t - dt) + (change_in_interest) * dt
INIT interest_in_subject = 0.01

DOCUMENT: [1] Assuming the level of interest in subject may vary per incoming info.

INFLOWS:

change_in_interest = interest_in_subject*learning_style_incompatibility
DOCUMENT: [1/min] Assuming rate of change in interest (in subject matter) at any time is directly proportional to the rate of lecturing and the level of student's performance.

□ level_of_effort(t) = level_of_effort(t - dt) + (change_in_effort) * dt
INIT level_of_effort = 0

DOCUMENT: [1] Level_of_effort feeds back to determine level of hooks_in_memory.

INFLOWS:

change_in_effort = (willingness-level_of_effort)*(level_of_performance)*(allocated_time_factor)
DOCUMENT: [1/min] Rate of change in the amount of effort that the rote learner puts in the task of memorizing.

□ level_of_performance(t) = level_of_performance(t - dt) + (change_in_performance) * dt
INIT level_of_performance = 0

DOCUMENT: [%] Level_of_performance feeds back to determine the learner's effort .

INFLOWS:

change_in_performance = (mark_of_desire-level_of_performance)*productivity
DOCUMENT: [1/m] Rate of change in performance is a function of the amount of effort and time that the rote learner puts into the task (difference of the mark of desire and current level of performance] at any time.

□ quantity_of_info(t) = quantity_of_info(t - dt) + (change_in_quantity_of_info) * dt
INIT quantity_of_info = 0

DOCUMENT: [H] Assuming the stock level of quantity of info given by the teaching system

INFLOWS:

change_in_quantity_of_info = (target_info-quantity_of_info)*adjustment_fraction
DOCUMENT: [H/min] Rate of change in the quantity of info (number of hooks) given by the teaching system.
Teaching system is a form-oriented source so the flow rate of info is based on the number of hooks issued.

UNATTACHED:

lecturing = type_of_info*change_in_quantity_of_info
DOCUMENT: [H/min] Assuming the rate of hooks given by the teaching system per minute.

○ adjustment_fraction = 0.005
DOCUMENT: [1/min] Assuming it is the required time fraction for the teaching system to adjust any change in the rate of given info.

○ allocated_time_factor = 0.05
DOCUMENT: [1/min] Assuming time fraction it takes for the rote learner to change his/her effort rate.

○ amount_learned = prior_knowledge+hooks_in_memory
DOCUMENT: [H] total number of active and unactive hooks

○ availability = hooks_in_memory*productivity*learning_reinforcers
DOCUMENT: [H] Assuming the theoretical amount of hooks of knowledge available to the learner. Amount of availability is directly related to the level of hooks in the memory and learning reinforcers and indirectly related to the level of starting knowledge(through productivity).

○ clsd_prbm_slvg = 3
DOCUMENT: [1] assuming the value of a closed-problem solving question assigned by the teacher.

Form teacher = 3
Function teacher = 3

○ coding = SMTH1(amount_learned, 3)
DOCUMENT: [1] Assuming the average ability to make decision for either starting new hooks or extending old hooks over the past 3 lecture minutes. This indicates the form learner's ability to find the related hooks compatible to the incoming information as per coding system in his/her memory (fit/unfit].

○ constraints = 0.5
DOCUMENT: [1] Constraints include all impeding factors, whether internal or external to the studentl, that lead to waste in the acquisition of knowledge.

○ impact_of_other_values = 0.3
DOCUMENT: [1] Assuming average quantified impact of other values:
pride in future prof'n, self-worth, security in future job, social obligation, bandwagon effect, and association with something one likes.

○ interest_in_grade = 0.5
DOCUMENT: [1] Assuming the amount of interest in the grade for this course.

○ interest_in_use = 0.5
DOCUMENT: [1] Assuming the amount of nterest in use of the task

○ learning_reinforcers = (testing+other_reinforcers+quality_of_teaching)/3
DOCUMENT: [1] Assuming the quantified average impact of learning reinforcers

○ memory_info = 5
DOCUMENT: [1] assuming the value of "only memorizing (rote) type of information" in the teacher's view. The weight is different and depends on the type of the mind a teacher possesses.
Assume: Form teacher = 5
Function teacher = 1

○ opn_prbm_slvg = 0
DOCUMENT: [1] assuming the valuet of an open problem solving question assigned by a teacher.
Form teacher = 0
Function teacher = 3

○ other_reinforcers = 0.3
DOCUMENT: [1] Assuming the quantified average impact of other learning reinforcers (e.g., institutional variables, interpersonal relations, satisfaction with the university)

○ prior_knowledge = 10
DOCUMENT: [H] total number of active hooks of information in the student's memory.

○ procedure_real = 1
DOCUMENT: [1] assuming the value of the "real procedure type of information" in the teacher's view.
Form teacher = 1
Function teacher = 5

○ procedure_rote = 5
DOCUMENT: [1] assuming the value of the "rote procedure type information" in the teacher's view.

Form teacher = 5
Function teacher = 1

○ productivity = amount_learned/prior_knowledge
DOCUMENT: [1] Productivity is the ratio of the output (amount_learned) to the input (prior_knowledge).

○ relreal_info = 1
DOCUMENT: [1] assuming the value of the "real relationship type information" in a teacher's view.
Form teacher = 1
Function teacher = 5

○ relrote_info = 4
DOCUMENT: [1] assuming the value of the "rote-oriented relationship type information" in the teacher's view.

Form teacher = 4
Function teacher = 2

○ rote_type = 5
DOCUMENT: [1] assuming the value assigned for a rote-oriented type question (e.g. true/false, multiple choice, and short/long answers) by the teacher.

Form teacher = 5
Function teacher = 0.5

○ student's_perceived_availability = SMTH1(availability,2)
DOCUMENT: [H] Assuming the average perceived availability of hooks of information by the student is 2 minutes.

○ target_info = 600
DOCUMENT: [1] Assuming giving each piece of information takes, in average, 3 seconds or 3/60 = 0.05 minutes. Also assuming two-third of time of each 45 minutes-class is spent by pure lecturing, thus:
30 minutes of pure lecture / 0.05 = 600 could be an average target for the number of total pieces of information which is given by a teaching system during a lecture hour.

○ testing = (rote_type+clsd_prbm_slvg+opn_prbm_slvg)/9
DOCUMENT: [1] testing = [1/(w1+w2+w3] * [a * w1 + b * w2 + c * w3]
where
"w" is the weight of each type of question in the teacher's view.
a= No. of only rote type questions
b= No. of only closed-problem solving
c= No. of only Problem Solving

○ type_of_info = (memory_info+relrote_info+relreal_info+procedure_real+procedure_rote)/25
DOCUMENT: [1] value of type of info =
[1/(Wa+ Wb + Wc + Wd +We)] * [a* Wa + b* Wb + c* Wc + d* Wd + e * We]

a= only memorizing info	d= procedure type -real
b= relationship type - rote	e= procedure type-rote
c= relationship type - real	W = Weight %

○ waste_fraction = learning_style_incompatibility*constraints
⊘ forecast_adjustment = GRAPH(student's_perceived_availability)
(0.00, 3.44), (11.1, 3.34), (22.2, 3.14), (33.3, 2.98), (44.4, 2.66), (55.6, 2.34), (66.7, 2.06), (77.8, 1.86), (88.9, 1.62), (100, 0.72)
DOCUMENT: [1] Impact of the amount of available hooks(in memory) on the rate of taking a new piece of incoming info.
⊘ learning_style_incompatibility = GRAPH(lecturing)
(0.00, 0.31), (0.1, 0.245), (0.2, 0.195), (0.3, 0.185), (0.4, 0.165), (0.5, 0.165), (0.6, 0.165), (0.7, 0.15), (0.8, 0.135), (0.9, 0.125), (1, 0.095)
DOCUMENT: [1] Learning style incompatibility is minimum when the lecture is issued by a teacher whose type of mind is similar to the student's type of mind (e.g., both are either function types or form types). It increases as incompatibility tends to increase.
⊘ mark_of_desire = GRAPH(expectancy)
(0.00, 3.00), (0.111, 18.5), (0.222, 40.5), (0.333, 42.0), (0.444, 45.0), (0.556, 49.5), (0.667, 52.0), (0.778, 59.5), (0.889, 68.5), (1.00, 74.5)
DOCUMENT: [1] Assuming Mark of Desire is the rote learner's target level of performance and it is a function of his/her level of expectancy.
⊘ perceived_assessment = GRAPH(level_of_performance)
(0.00, 0.00), (16.7, 0.185), (33.3, 0.3), (50.0, 0.39), (66.7, 0.415), (83.3, 0.435), (100, 0.445)
DOCUMENT: [1/min] Perceived assessment is the perception of rote learner re his/her level of performance at any time.
⊘ quality_of_teaching = GRAPH(quantity_of_info*type_of_info)
(0.00, 0.275), (0.1, 0.285), (0.2, 0.295), (0.3, 0.335), (0.4, 0.365), (0.5, 0.395), (0.6, 0.4), (0.7, 0.43), (0.8, 0.505), (0.9, 0.565), (1, 0.63)
DOCUMENT: [1] Quality_of_teaching depends on the combination of the type and the quantity of presentation.
⊘ willingness = GRAPH(expectancy)
(0.00, 0.29), (0.1, 0.295), (0.2, 0.305), (0.3, 0.32), (0.4, 0.345), (0.5, 0.37), (0.6, 0.41), (0.7, 0.46), (0.8, 0.545), (0.9, 0.63), (1, 0.69)
DOCUMENT: [1] Assuming willingness is inversely proportional to the level of expectancy and is an indicator of the target that the rote learner sets for his/her level of effort at any time.

Nassereddin Eftekhar is a professional engineer who received his Ph.D. from the Mechanical and Industrial Department at the University of Manitoba in Winnipeg, Canada. He holds a B.Sc. in Chemical Engineering from Abadan Institute of Technology, Iran, and an M.Sc. in Industrial Engineering form Sharif University of Technology, Tehran. His main interest is in systems analysis (system modeling simulation).

Douglas R. Strong is a professional engineer and is a professor in the Mechanical and Industrial Engineering Department at the University of Manitoba, Winnipeg, Canada. He received his Ph.D. from the University of Toronto in 1976. His primary research interest is in the analysis of manufacturing processes using modeling and simulation techniques. His current research work is in the field of cost analysis of the manufacturing and service industries which is supported by several industries.