SIMULATION OF SYSTEMS ARCHETYPES

Anne M. Dowling
Roderick H. MacDonald
George P. Richardson

Rockefeller College of Public Affairs and Policy
University at Albany, Milne 308, Albany, New York 12222

Abstract

In recent years an important component of the research agenda in the field of system dynamics has focused on the definition and use of archetypal structures. Although the primary objective of such rsearch is to develop an intrinsic set of system structures that can be used to categorize insights in dynamic systems, the ultimate goal is to provide an effective mechanism by which information can be transferred from a system dynamics model to a client in an easy-to-comprehend manner. To date, a number of archetypal structures have been presented by Richmond, Senge, and Wolstenholme. This paper discusses two systems archetypes proposed by Senge: "shifting the burden" and "fixes that fail." By developing sets of precise code and simulating the models, the authors document the written descriptions of these two archetypal structures and explore the extent to which the structures behave as expected. The authors demonstrate that the development of formal models for systems archetypes is not an easy task.

Introduction

In recent years an important component of the research agenda in the field of System Dynamics has focused on the definition and use of archetypal structures. Although the primary objective of such research is to develop an intrinsic set of system structures that can be used to categorize insights in dynamic systems, the ultimate goal is to provide an effective mechanism by which information can be transferred from a system dynamics model to a client in an easy-to-comprehend manner. Research efforts in this area have focused on the identification and development of models of archetypal structures. In 1988, Richmond et al presented a series of activity and infrastructure arcyhetypes for use, while in 1990 Senge identified and described nine systems archetypes. These efforts were followed in 1993 by Wolstenholme and Corben who proposed condensing archetypal structures to a minimum set of four, each representing one of four possible ways of ordering a pair of feedback loops.

The activity and infrastructure archetypes developed by Richmond et al in the late 1980s are widely recognized and used by system dynamicists today as tools for learning about the fundamentals of complex systems. The acceptance of these archetypal structures, compounding, draining, external resource production, stock-adjustment, and co-flow, can be attributed to a number of factors, including the simple stock and flow diagrams developed for each that make them readily understandable, the graphs accompanying them that show the possible behavior patterns generated by the structures, and the guidelines provided for their use in specific circumstances. An additional factor contributing to their acceptance is that they resemble the rate-level structures used by Richardson and Pugh (1981) to formulate common rate equations.

The nine systems archetypes presented by Senge (1990) are: balancing process with delay, limits to growth, shifting the burden, eroding goals, escalation, success to the successful, tragedy of the commons, fixes that fail, and growth and underinvestment. Wolstenholme and Corben (1993) propose that Senge's systems archetypes can be reduced to a set of four: 1) growth intended-stagnation/decline achieved, 2) control intended-unwanted growth achieved, 3) control intended-compromise achieved, and 4) growth intended-at expense to others. Although Senge's and Wolstenholme and Corben's archetypes are designed to provide insights about dynamics that are easy to understand by lay persons, they have not yet received universal acceptance by system dynamicists. One possible explanation for this may be that sets of precise code have not been available for them in the literature, making it problematic for researchers to easily understand the descriptions of the structure and behavior.

While causal loop diagrams are a simple tool to use in communicating system structure, they are inherently weak because they do not distinguish between conserved flows and information flows. As such, they obfuscate direct causal relationships between rates and levels. This is an important factor since most of the structures presented by Senge and by Wolstenholme and Corben represent conserved flows. Further, as would be argued by Richardson (1986), it is impossible to determine behavior simply from loop polarity because loop polarity does not create behavior. Rather it is the rate-level structure that determines behavior. The fact that causal loop diagrams do not reflect important factors such as hidden loops, net rates, and parameters further limits their ability to provide a clear understanding of structure and behavior.

To advance the process by which the definition and use of archetypal structures is refined, the authors have developed sets of precise code for each of the systems archetypes presented by Senge. The authors transform Senge's causal loop diagrams and written descriptions into formal quantitative models. The primary purpose for undertaking this task is to "capture" these systems archetypes in greater detail, generating an expanded framework within which system dynamicists can continue their dialogue on generic structures. In carrying out this task, the authors hypothesized that the models might be more complicated than the causal loop diagrams and written descriptions suggest. By developing sets of precise code and simulating the models, the authors document the written descriptions of the proposed systems archetypes and explore the extent to which these structures behave as expected. While the formal models developed may not necessarly be the optimal models for these systems archetypes, the authors argue that in order for a set of archetypes to be accepted by the system dynamics community, formal code for the causal loop diagrams based on appropriate modeling techniques needs to be written and made available for discussion through the literature.

This paper focuses on two of Senge's systems archetypes: "shifting the burden" and "fixes that fail." These particular two systems archetypes were chosen for inclusion in this paper because they produced simple but thought-provoking observations. A discussion of the sets of code developed for all of Senge's archetypal structures is presented in a separate paper available directly from the authors, entitled "Formal Models of Systems Archetypes."

Fixes That Fail

Senge (1990) describes "fixes that fail" as a situation in which the solution or fix to a problem is effective in the short-term, but has unforeseen long-term consequences that usually get addressed by reapplying the same fix, but with an even greater vengeance. Thus, after initial better conditions, the unintended consequences exacerbate the problem such that it not only reappears but becomes steadily worse. Senge's causal loop diagram for "fixes that fail" is shown in Figure 1A.

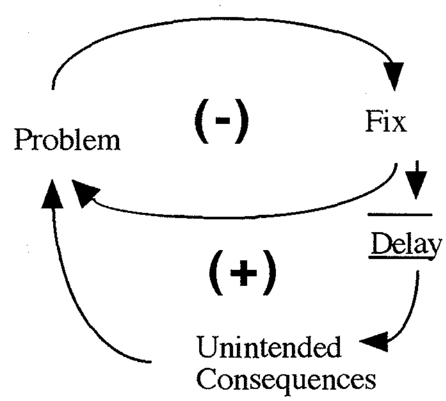

Figure 1A
Causal Loop Diagram of "Fixes That Fail"

The task of developing a STELLA model for "fixes that fail" highlights the difficulties that can be encountered in building models of systems archetypes. For this particular archetype, the authors have developed two different, albeit plausible, models. While both models reflect the behavior described by Senge, they differ with regard to their structure. As described below, the structure for one model is based on the concept that the Problem is a level with the Fix being its outflow, while the structuce for the second model is based on the premise that the Problem and the Fix are both levels.

As shown in Figure 1B, the first model developed for "fixes that fail" is a three level model with Problem and Unintended Consequences modeled as levels. There is a single inflow into the problem, Increase in problem, and a single outflow, Fix. This structure is based on the premise that as the Problem increases due to Unintended Consequences, the Fix increases in a proportional manner. Thus, the Problem elimination normal variable is exogenous to the structure and is arbirtrarily set at 10 percent.

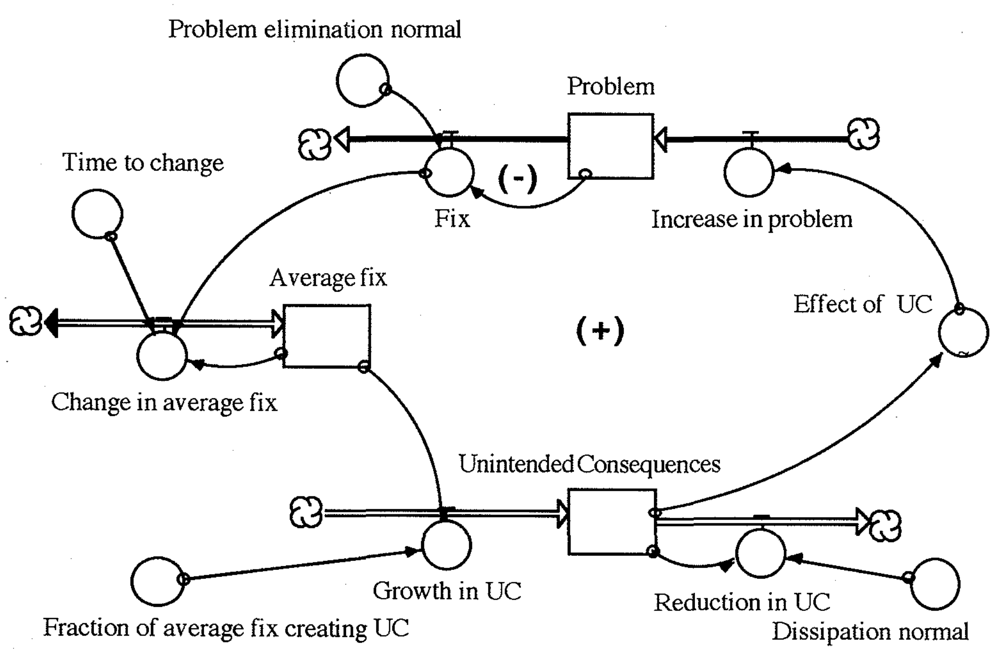

Figure 1B
STELLA Model 1 of "Fixes That Fail"

In Senge's causal loop diagram, Unintended Consequences is directly determined by the Fix after a time delay. As shown in Figure 1B, a smooth is used to capture Senge's notion of a time delay. The set of precise code for this model is presented in Table 1 of the Appendix.

The second model developed for "fixes that fail", shown in Figure 1C, is a four level model with Problem, Fix, and Unintended Consequences each modeled as levels. Modeling the Fix as a level was based on the premise that in specific circumstances the Fix could be a variable that accumulates and changes moderately over time. Again, as in the first model, the time delay is modeled as a smooth. The set of precise code for this second model is presented in Table 2 of the Appendix.

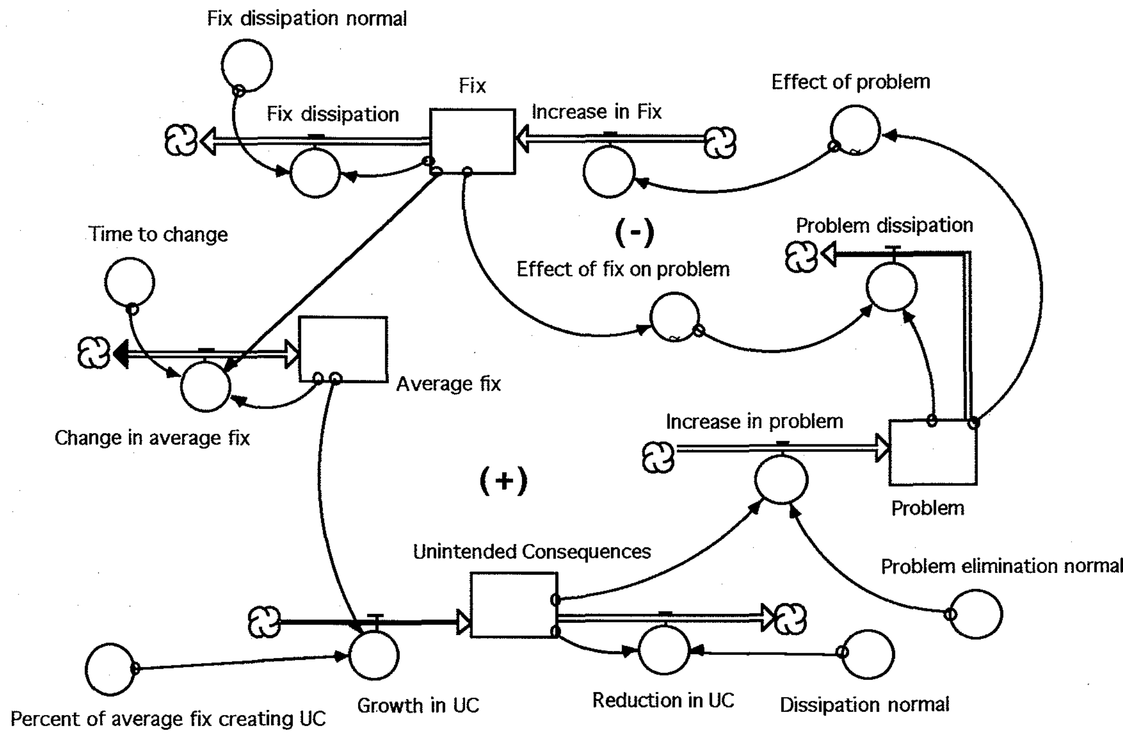

Figure 1C
STELLA Model 2 of "Fixes That Fail"

A simulation of each model, reflecting the behavior of the structures represented in Figures 1B and 1C and Tables 1 and 2, is shown in Figures 1D and 1E, respectively. As indicated in Figures 1D and 1E, the Fix does have an initial positive effect on the Problem. These figures further show that over time the Fix can no longer control the Problem, that the Problem reappears and begins to increase in severity. At this point, the Fix no longer has a positive effect on the problem, but rather serves to steadily worsen the situation. This occurs because the Fix in both models is part of both the negative and positive loops. When loop dominance shifts from the negative loop to the positive loop, the Fix contributes to the problem. The behaviors, as shown in Figures 1D and 1E, reflect the behavioral description provided by Senge.

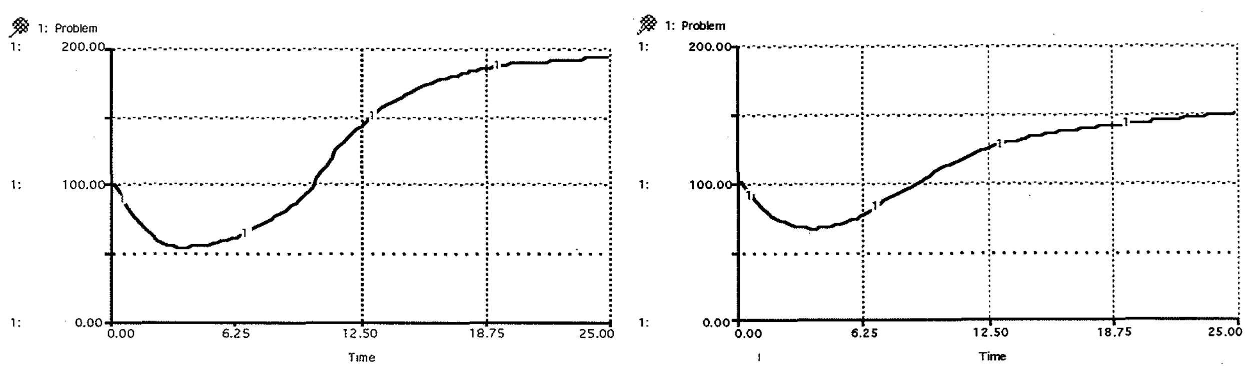

Figure 1D

Figure 1E

Figure 1D-E: Simulation of "Fixes That Fail"

A good solution to the Problem would involve increasing the Fix so that the Problem goes to zero before the Unintended Consequences grow to the point where they cause the inflow into the problem to exceed the outflow. Neither model, as structured, can do this since the inflow into the Unintended Consequences is formulated as the product of the Average fix and the Fraction of average fix creating Unintended Consequences. A substantial increase in the Problem elimination normal can reduce the Problem quickly, but cannot eliminate the Problem completely. In effect, the negative loop which reduces the Problem quickly is strengthened, keeping the Problem from coming back as quickly. However, even a "stronger" Fix fails in the long run.

The models presented in Figures 1B and 1C show that the structure causes the UC level to have a negative effect on the Problem. Thus, no matter how large the Fix, as long as the Problem exists, the Fix will always contribute to the growth of Unintended Consequences, with the strength of the positive feedback loop growing until it dominates the structure. The reader will observe that the structure generates the nonlinearity that shifts loop dominance. This occurs because Unintended Consequences, which is part of a positive loop, is a conserved flow. The authors argue that while the effect of the Average fix can be delayed by increasing the Time to change, in the long run the Fix cannot stop the Problem from increasing.

In essence, each of the formal models presented above captures the concept that the Fix to the Problem creates Unintended Consequences which in turn exacerbates the Problem. A basic difference between the two formal models and Senge's representation is that the models explicitly state the relationships between information and conserved flows in the appropriate manner. However, the fact that both models produce similar behavior raises a fundamental dilemma: whether one model is more correct or better than the other, if in fact one is, or whether there is still some other formal model that needs to be examined.

Shifting The Burden

Senge (1990) defines the "shifting the burden" systems archetype as a short term solution which has the immediate effect of reducing some problem. He defines this short term solution as a "Symptomatic Solution." In addition to its immediate reduction in the Problem symptom, the Symptomatic solution also has a side effect which causes the Fundamental solution to atrophy. The Fundamental solution is capable of reducing the Problem sympton to a greater degree than that of the Symptomatic solution, but there is a delay in the time it takes for the Fundamental solution to take effect. Senge's causal loop diagram for "shifting the burden" is shown in Figure 2A.

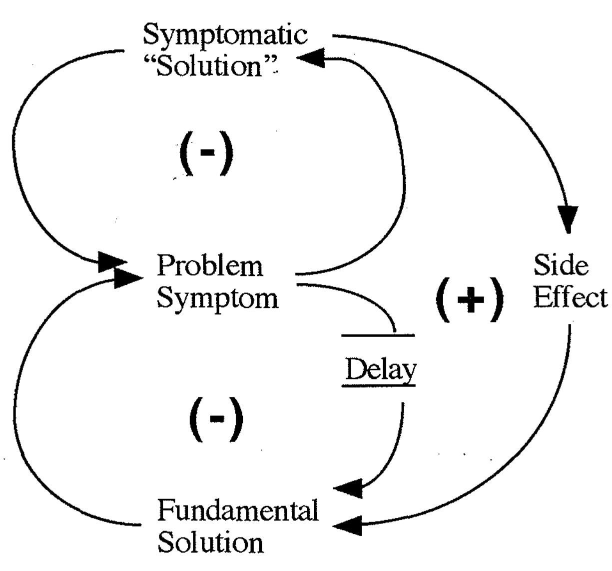

Figure 2A
Causal Loop Diagram of "Shifting the Burden"

A three level STELLA model was developed to document the structure and behavior of "shifting the burden". As shown in Figure 2B, the model is structured so that Problem symptom, Fundamental solution, and Side effect are all levels. The Problem symptom has a Problem inflow which is determined by multiplying the Problem symptom level by a constant term, the Problem inflow normal. This structure adds a second positive feedback loop to the model that is not in Senge's causal loop diagram. The authors believe that this structure is necessary to capture the concept that the Problem symptom is dynamic - it can either increase or decrease over time. This positive feedback loop dominates the structure when policies favoring the Fundamental solution are instituted. The Problem symptom is reduced by the Problem outflow, formulated by multiplying the Problem solution by the sum of the Symptomatic solution and the Fundamental solution.

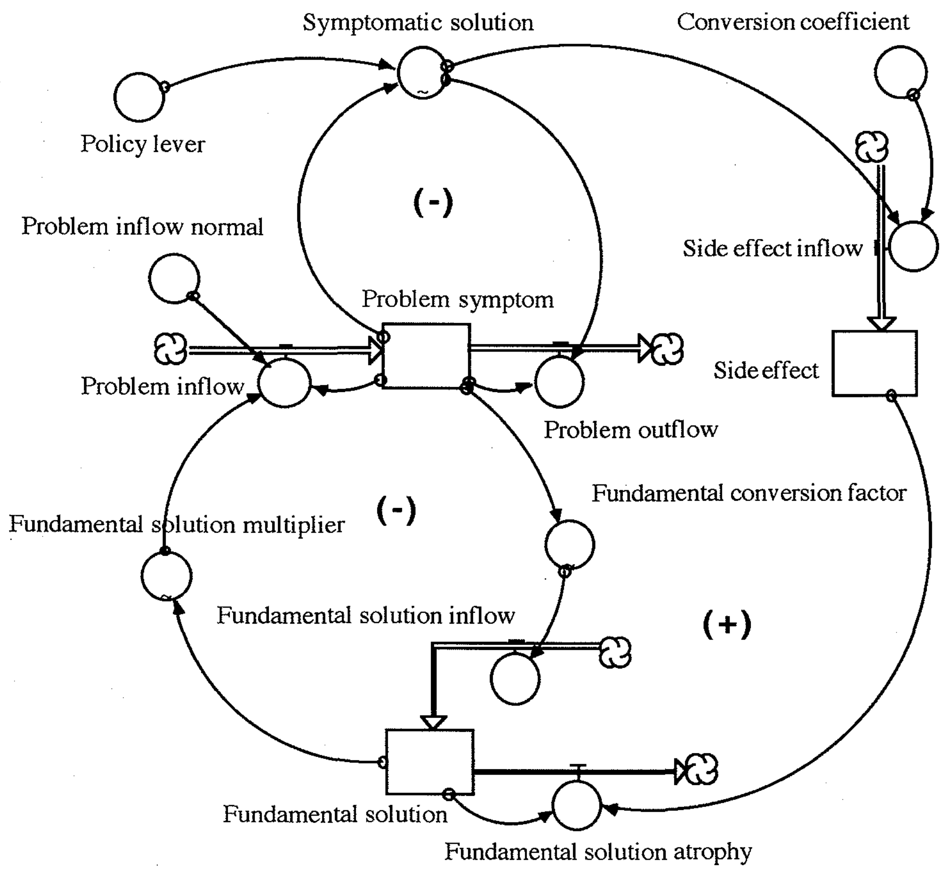

Figure 2B
STELLA Model of "Shifting the Burden"

The Symptomatic solution is formulated as a table function of the Problem symptom and the Policy Lever, with the Policy lever being used as a "switch" to turn the Symptomatic solution on and off. The Symptomatic solution table function reflects the concept that as the Problem symptom increases, the Symptomatic solution increases. As indicated in Table 3 (see Appendix), the Problem symptom level is initialized at 100, while the value for the Symptomatic solution at this point is greater than the Problem inflow normal. Therefore, as the Problem symptom is reduced, the Symptomatic solution is also reduced. This represents the negative feedback loop shown at the top of Figure 2B. Equilibrium is achieved when the Symptomatic solution plus the Fundamental solution are equal to the Problem inflow normal.

The Symptomatic solution also has a Side effect which is determined by the Side effect inflow, formulated by multiplying the Symptomatic solution by the Conversion coefficient. The Conversion coefficient captures the concept that the Symptomatic solution creates an additional conserved flow. The Side effect level is responsible for determining the Fundamental solution atrophy rate. This rate drains the Fundamental solution level which, in turn, reduces the Problem outflow rate and increases the Problem symptom. The increase in the Problem symptom, in turn, increases the Symptomatic solution, forming the positive feedback loop shown on the right side of Figure 2B.

The negative loop at the bottom of Figure 2B begins with the Problem symptom and runs through the Fundamental conversion factor. The larger the Problem symptom the larger the Fundamental conversion factor. This effect can be modeled as either a table function or as a constant. The authors believe that a table function is a more appropriate modeling technique in this situation because it captures the concept that pressure for a solution would increase as the Problem symptom grows.

Two simulations of the model are shown in Figures 2C-E. One simulation reflects the behavior of the structure when the Policy lever switch is turned on, while the second reflects the behavior when the switch is turned off. When the Policy lever switch is turned on, the Symptomatic solution begins to decrease, reducing the Problem symptom by approximately 40 percent before reaching equilibrium. At the same time, the Fundamental solution initially increases and then begins to atrophy as the Side effect increases. These behaviors are reflected by Problem

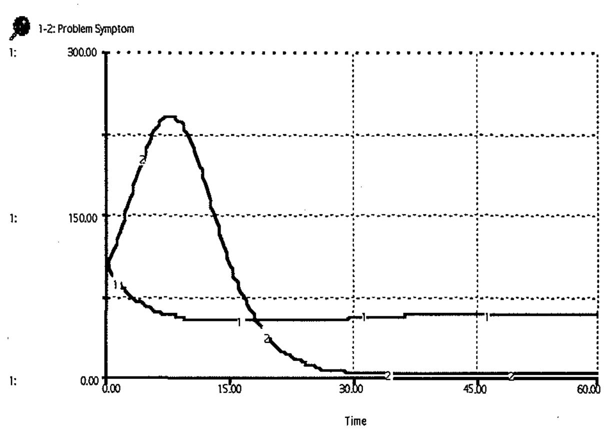

Figure 2C

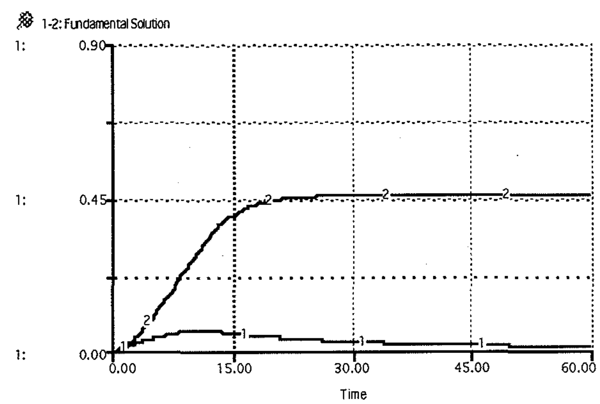

Figure 2D

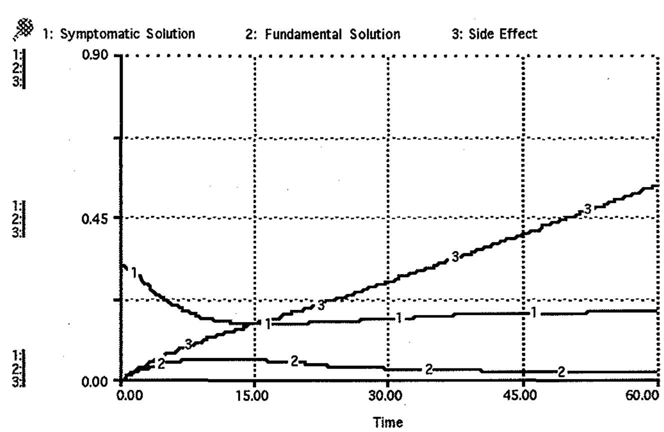

Figure 2E

Figures 2C-E: Simulation of "Shifting the Burden" symptom 1 in Figure 2C and Fundamental solution 1 in Figure 2D. To facilitate behavioral comparisons, the Symptomatic solution, the Fundamental solution, and the Side effect can be observed in Figure 2E.

When the Policy lever switch is turned off, the Problem symptom initially increases, then declines and finally approaches zero. This behavior, reflected by Problem symptom 2 in Figure 2C, is a typical case of "worse before better" behavior. Although the initial behavior of the Problem symptom is much worse than when the Symptomatic solution is turned on, in the long run its behavior is much more desirable because the Fundamental solution is not allowed to atrophy. The behavior of the Fundamental solution when the Policy lever is turned off is indicated by Fundamental solution 2 in Figure 2D.

The behavior of this model reflects the behavioral description provided by Senge. The reader will note subtle differences, however, when Senge's causal loop diagram is compared to the model's structure. For instance, the authors chose to model the Symptomatic solution as a table function, whereas the diagram appears to suggest a level.

Summary

The models presented above highlight differences that can exist between causal loop diagrams and stock and flow diagrams that are important factors in understanding the structure and behavior of systems. The causal loop diagrams for these particular archetypal structures do not reflect the causal relationships between the rate and level variables. These models, especially those presented for "fixes that fail", also demonstrate that the development of formal models for systems archetypes is not an easy, clear-cut task. Specifically, in "Fixes That Fail" the question of whether to model the Fix as a stock or flow presented an unresolved dilemma for the authors. Likewise, the decision to omit a stock in the formal model that was represented in Senge's causal loop diagram of "Shifting the Burden" - Symptomatic solution - was not a trivial one.

The authors recognize that the code they have developed for "shifting the burden" and "fixes that fail" is simply the first step in producing a definitive set of codes for these archetypes. Regardless of whether future research determines that these sets of code are "good" or "bad", the authors will consider their efforts a success if they challenge and stimulate discussion among the research community working with archetypes.

REFERENCES

Graham, Alan K. 1988. "Generic Models as a Basis for Computer-Based Case Studies." In Proceedings of the 1988 International System Dynamics Conference (San Diego, CA).

Graham, Alan K. and D. Neal Scogin. 1985. "Generic Models as Educational Tools: Teaching About Managing Technology Conversions." In Proceedings of the 1985 International System Dynamics Conference (Keystone, Colorado).

Kim, Daniel H. and Gary Burchill. 1992. "Systems Archetypes as a Diagnostic Tool: A Field-Based Study of TQM Implementations." Unpublished.

Paich, Mark. 1995. "Generic Structures." System Dynamics Review 1, no.1 (Summer): 126-132.

Richardson, George P. and Alexander L. Pugh III. 1981. Introduction to system Dynamics With DYNAMO. Cambridge: MIT Press.

Richardson, George P. 1986. "Problems With Causal-Loop Diagrams." System Dynamics Review 2, no.2 (Summer): 158-170.

Richmond, Barry M., S. Peterson, and P. Vescuso. 1988. An Academic User's Guide to STELLA. Hanover, NH: High Performance Systems, Inc.

Senge, Peter M. 1990. The Fifth Discipline: The Art and Practice of the Learning Organization. New York: Doubleday.

Wolstenholme, E.F. and D.A. Corben. 1993. "Towards a Core Set of Archetypal Structures in System Dynamics." In The Role of Strategic Modelling in International Competitiveness, ed. by Enrique Zepeda and Jose A. E. Macuca. Cancun, Mexico: System Dynamics Society.

APPENDIX

Table 1
Equations List For Proposed Model 1 of "Fixes That Fail"

Average_fix(t) = Average_fix(t - dt) + (Change_in_average_fix) * dt
INIT Average_fix = Fix
INFLOWS:
Change_in_average_fix = (Fix-Average_fix)/Time_to_change
Problem(t) = Problem(t - dt) + (Increase_in_problem - Fix) * dt
INIT Problem = 100
INFLOWS:
Increase_in_problem = Effect_of__UC
OUTFLOWS:
Fix = Problem*Problem_elimination_normal
Unintended_Consequences(t) = Unintended_Consequences(t - dt) + (Growth_in_UC - Reduction_in_UC) * dt
INIT Unintended_Consequences = 0
INFLOWS:
Growth_in_UC = Average_fix*Fraction_of_average_fix_creating_UC
OUTFLOWS:
Reduction_in_UC = Unintended_Consequences*Dissipation_Normal
Dissipation_Normal = .1
Fraction_of_average_fix_creating_UC = .88
Problem_elimination_normal = .25
Time_to_change = 1
Effect_of__UC = GRAPH(Unintended_Consequences)
(0.00, 0.00), (10.0, 0.00), (20.0, 0.00), (30.0, 1.00), (40.0, 6.50), (50.0, 12.5), (60.0, 16.0), (70.0, 21.0), (80.0, 26.0), (90.0, 37.0), (100, 49.0)

Table 2
Equation List For Proposed Model 2 of "Fixes That Fail"

Average_Fix(t) = Average_Fix(t - dt) + (Change_in_average_fix) * dt
INIT Average_Fix = 0
INFLOWS:
Change_in_average_fix = (Fix-Average_Fix)/Time_to_change
Fix(t) = Fix(t - dt) + (Increase_in_Fix - Fix_dissipation) * dt
INIT Fix = 100
INFLOWS:
Increase_in_Fix = Effect_of_problem
OUTFLOWS:
Fix_dissipation = Fix*Fix_Dissipation_Normal
Problem(t) = Problem(t - dt) + (Increase_in_Problem - Problem_dissipation) * dt
INIT Problem = 100
INFLOWS:
Increase_in_Problem = Unintended_Consequences+Problem_normal
OUTFLOWS:
Problem_dissipation = Problem*Effect_of_fix_on_problem
Unintended_Consequences(t) = Unintended_Consequences(t - dt) + (Growth_in_UC - Reduction_in_UC) * dt
INIT Unintended_Consequences = 0
INFLOWS:
Growth_in_UC = Average_Fix*Fraction_of_Average_Fix_creating_UC
OUTFLOWS:

Reduction_in_UC = Unintended_Consequences*Dissipation_normal
Dissipation_normal = .1
Fix_Dissipation_Normal = .1
Fraction_of_Average_Fix_creating_UC = .05
Problem_normal = 10
Time_to_change = 2
Effect_of_fix_on_problem = GRAPH(Fix)
(0.00, 0.00), (20.0, 0.02), (40.0, 0.107), (60.0, 0.158), (80.0, 0.212), (100, 0.293), (120, 0.354), (140, 0.4), (160, 0.5), (180, 0.6), (200, 0.7)
Effect_of_problem = GRAPH(Problem)
(0.00, 0.00), (20.0, 2.00), (40.0, 4.00), (60.0, 6.00), (80.0, 8.00), (100, 10.0), (120, 12.0), (140, 14.0), (160, 16.0), (180, 18.0), (200, 20.0)

Table 3
Equation List For "Shifting The Burden"

Fundamental_solution(t) = Fundamental_solution(t - dt) + (Fundamental_solution_inflow - Fundamental_solution_atrophy) * dt
INIT Fundamental_solution = 0
INFLOWS:
Fundamental_solution_inflow = Fundamental_conversion_factor
OUTFLOWS:
Fundamental_solution_atrophy = Fundamental_solution*Side_effect
Problem_symptom(t) = Problem_symptom(t - dt) + (Problem_inflow - Problem_outflow) * dt
INIT Problem_symptom = 100
INFLOWS:
Problem_inflow = Problem_symptom*(Problem_inflow_normal-Fundamental_solution_multiplier)
OUTFLOWS:
Problem_outflow = (Problem_symptom*Symptomatic_solution)+(Problem_symptom*.1)
Side_effect(t) = Side_effect(t - dt) + (Side_effect_inflow) * dt
INIT Side_effect = 0
INFLOWS:
Side_effect_inflow = Conversion_coefficient*Symptomatic_solution
Conversion_coefficient = .05
Policy lever= 1
Problem_inflow_normal = .2
Fundamental_conversion_factor = GRAPH(SMTH1(Problem_symptom,1))
(0.00, 0.00), (20.0, 0.01), (40.0, 0.02), (60.0, 0.03), (80.0, 0.04), (100, 0.05), (120, 0.062), (140, 0.077), (160, 0.098), (180, 0.117), (200, 0.133)
Fundamental_solution_multiplier = GRAPH(Fundamental_solution)
(0.00, 0.00), (0.1, 0.00), (0.2, 0.001), (0.3, 0.0163), (0.4, 0.035), (0.5, 0.0663), (0.6, 0.0963), (0.7, 0.126), (0.8, 0.152), (0.9, 0.182), (1, 0.2)
Symptomatic_solution = GRAPH(Problem_symptom*Policy)
(0.00, 0.00), (20.0, 0.0675), (40.0, 0.155), (60.0, 0.188), (80.0, 0.228), (100, 0.268), (120, 0.295), (140, 0.34), (160, 0.405), (180, 0.46), (200, 0.492)