PN-ABT-844

The Dynamics of Project Implementation *

Karl H. Clauset, Jr.

[redacted]

August, 1987

I. PURPOSE

The purpose of this project has been to build a computer simulation model which embodies the key concepts and relationships on project implementation described in a paper by Donald P. Warwick, "Core Elements in Implementation." Inevitably, the work described here also reflects the author's own ideas about project implementation which stem from his research on the implementation of educational innovations in the US and his teaching on project planning and implementation in the International Educational Development Program at Boston University.

This is a preliminary report on the project. It attempts to describe the key features of the model and the assumptions that underlie its structure. More work is needed to refine the theoretical structure, test model behavior for robustness and consistency, simulate the model under realistic project scenarios, and analyze the results.

A. The Problem

The problem specifically addressed in this work is that projects, when implemented, rarely unfold as envisioned. Many experience significant delays in achieving desired outcomes; others lose momentum and wither away. Yet a number of projects, even when there are problems with demand or fit between the project and its environment, are able to move forward to achieve their goals. Why do these differences occur? Are they pre-ordained, given the particular social, political, or economic context? Are there lessons to be learned from decades of development project experience in a range of sectors that can be synthesized to help project managers increase the likelihood of success in implementation?

*This paper was prepared under the auspices of the BRIDGES Project.

B. The Model

1. [redacted] the life cycle of the implementation of a [redacted] It begins after adoption and is set to track a project for 5 years.

2. The structure of the model assumes, as does Warwick, that all projects face decisions around the same set of core elements.

3. A central assumption of the model is that **one** causal structure can produce a variety of implementation scenarios simply by varying input conditions or the strength of the relationships between variables; therefore, one does not build a separate model for each scenario.

4. Another key assumption is that, in most cases, the nature of the scenario that unfolds is not preordained and is the result primarily of actions taken, or not taken, by the project staff rather than external forces.

5. The model also assumes that it takes effort by project staff to produce outcomes, increase demand for services, maintain support, improve the fit of the project with the environment, and improve staff competency. As a consequence, **how** effort is allocated is a key determinant in which implementation scenario will unfold.

6. Effort allocation implies decisionmaking. The model portrays information that project managers might use to make decisions and the options available to them. Relationships within the model can be modified to simulate different approaches and priorities.

7. As the following section illustrates, there are many more assumptions embedded in the model. The advantage of the modeling process is that it forces one to make the assumptions **explicit**. If there are differences of opinion about assumptions, alternative structures can be formulated and tested for their implications. The purpose of laying them out here is to stimulate discussion about their relevance and

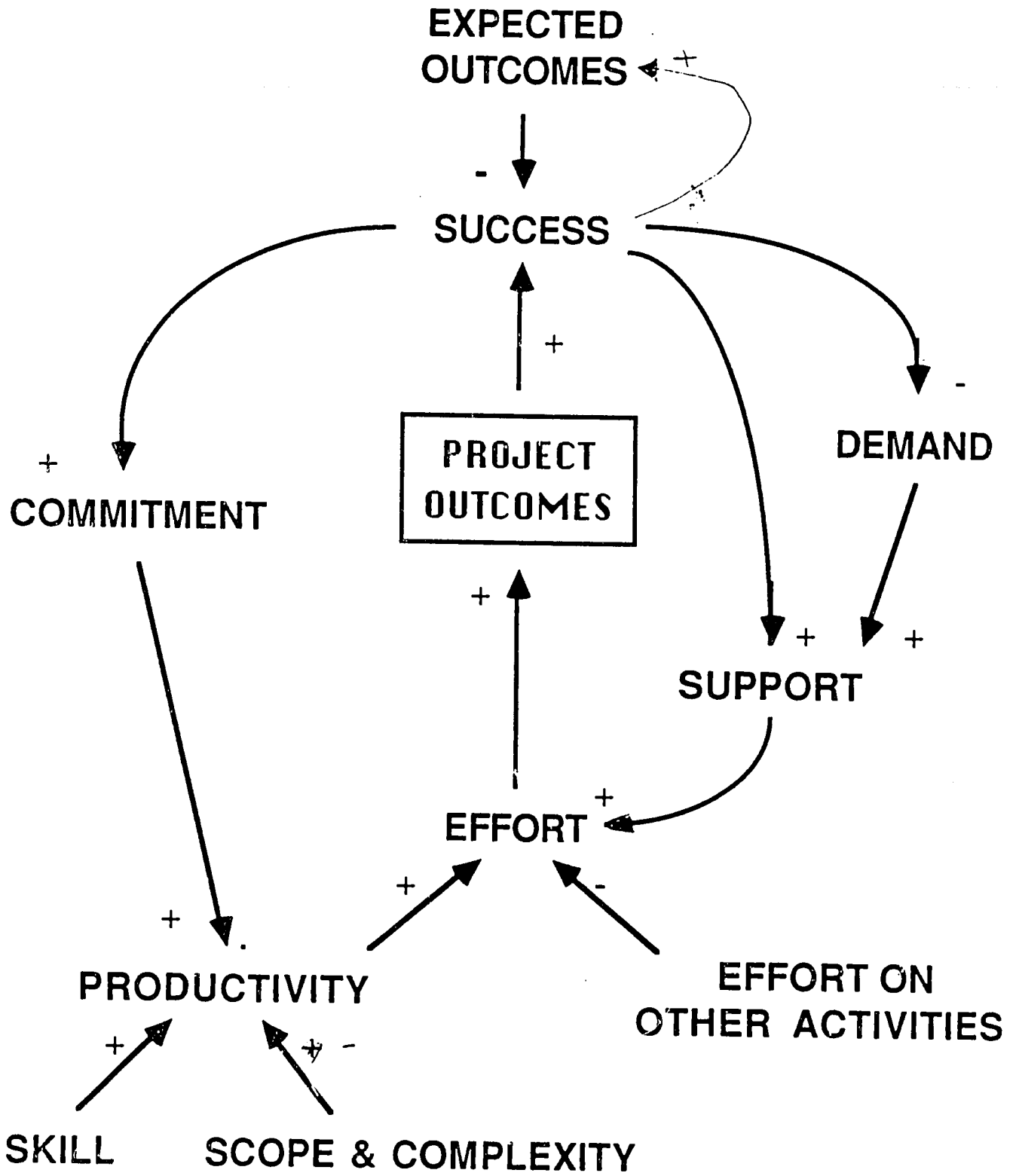


Figure 1: Influences on Project Outcomes

validity.

C. CORE ELEMENTS OF THE MODEL

1. In the model, **Project Outcomes**, is used to signify the outputs to be achieved by the project. These might be such outputs as the number of primary schools built, number of couples served by family planning, and changes in the quality of school curriculum.

   Project outcomes are achieved through effort. The rate at which these outcomes are achieved is influenced by a number of factors - the **scope and complexity** of the project, the **fit** of the project design and technology with its environment, the level of political and client **support** for the project (and the resources that follow from this support), the client **demand** for the proposed outputs, the **commitment and skill** of project staff, and the extent to which staff efforts are directed toward other activites, such as increasing demand or participation, improving fit or staff skills.

   Unlike traditional path models where outcomes are assumed to be the dependent variable, this model assumes that the perceived success or failure of the project staff to produce outcomes on schedule influences directly or indirectly all of the factors listed above. As a consequence, these factors are mutually **interdependent** with project outcomes. In the model, success is measured by comparing, at any given point in time, the actual level of project outcomes with the expected level of outcomes, which are in turn influences by the scope and complexity of the project.

   This relationship is illustrated in Figure 1. The arrows show the direction of the influence. A positive sign at the head of the arrow indicates a direct relationship between the two variables - as "A" increases, "B" increases; as "A" decreases, "B" decreases. A negative sign indicates an inverse relationship - as "A" increases, "B" decreases; as "A" decreases, "B" increases.

Several characteristics of this diagram are noteworthy.

‡ The diagram presents a macro level view of the system. It is not meant to portray all of the interactions or all of the factors that might influence key variables. These are shown in subsequent figures.

‡ Implied in the diagram is a production model of projects where the amount of effort is shaped by the number of staff (a function of support and resources), their productivity (a function of skill, commitment, and the scope and complexity of the project), and the extent to which their efforts are allocated to the direct production of outcomes.

‡ There are three "feedback" loops in the diagram. Loops 1 and 2 are positive feedback loops - they are self-reinforcing. Increased success leads to more support and higher staff commitment, both of which increase effort and lead to more success. This is the growth process, However, they also have a dark side. If productivity or support suddenly decline, a downward failure-reinforcing spiral is triggered unless there are countervailing forces.

‡ The countervailing force is Loop 3. This is a negative feedback loop that serves as the self-correcting thermostat for the system - both for runaway growth and failure. As efforts mount and outcomes are produced, these outcomes begin to satisfy demand. Demand declines and policy makers reduce their support - figuring that it is now time to turn their attention to other problems. As support declines, so does effort and the project comes to a close. Conversely, limited success in producing outcomes does little to satisfy demand and there continues to be pressure and support to produce outcomes.

‡ This countervailing brake on decline does not continue indefinitely. At some point, people get so frustrated with a project that they no longer try to correct the situation, but instead give up. At this point, the self-correcting negative feedback loop ceases to function and the downward spiral accelerates.

‡ Other connections beside those shown in the diagram are possible. One typical response to project success is to expand the scope and

complexity of the project. This changes the quantity and quality of outcomes required and may affect the relative skill and commitment of project staff. It may also redefine the nature of the client population and the relevant political constituencies, thus affecting demand and support. The value of this approach to modeling is that one can begin with simple structures, develop an understanding of them, and gradually expand in complexity as needed.

A key decision point for project managers is how to allocate efforts among competing tasks. In the model, managers set priorities among the following:

- Effort on Outcomes
- Effort to Increase Demand
- Effort to Increase Participation
- Effort to Increase Fit with the Environment

In the model, the Effort on Outcomes is the effort **left over** after effort is directed toward the other activities. The assumption here is that project managers tend to these other areas before devoting energio outcomes. (Other assumptions can easily be modeled.)

The relative priorities for the other categories of effort can either be constant, assuming a particular mindset about management, or variable, assuming a contingency approach to effort allocation. The current version of the model treats them as constants.

2. Critical to the achievement of project outcomes is the continued **Support** of clients and other constituencies that control access and resources.

Support changes over time as the different groups change their commitment to the project. Commitment is a function of several factors: the degree of **environmental stability**, the perceived **fit** of the project's design and technology to the environment, the degree of **participation** that the groups have had in policy making and project design and in project implementation, the **demand** for project

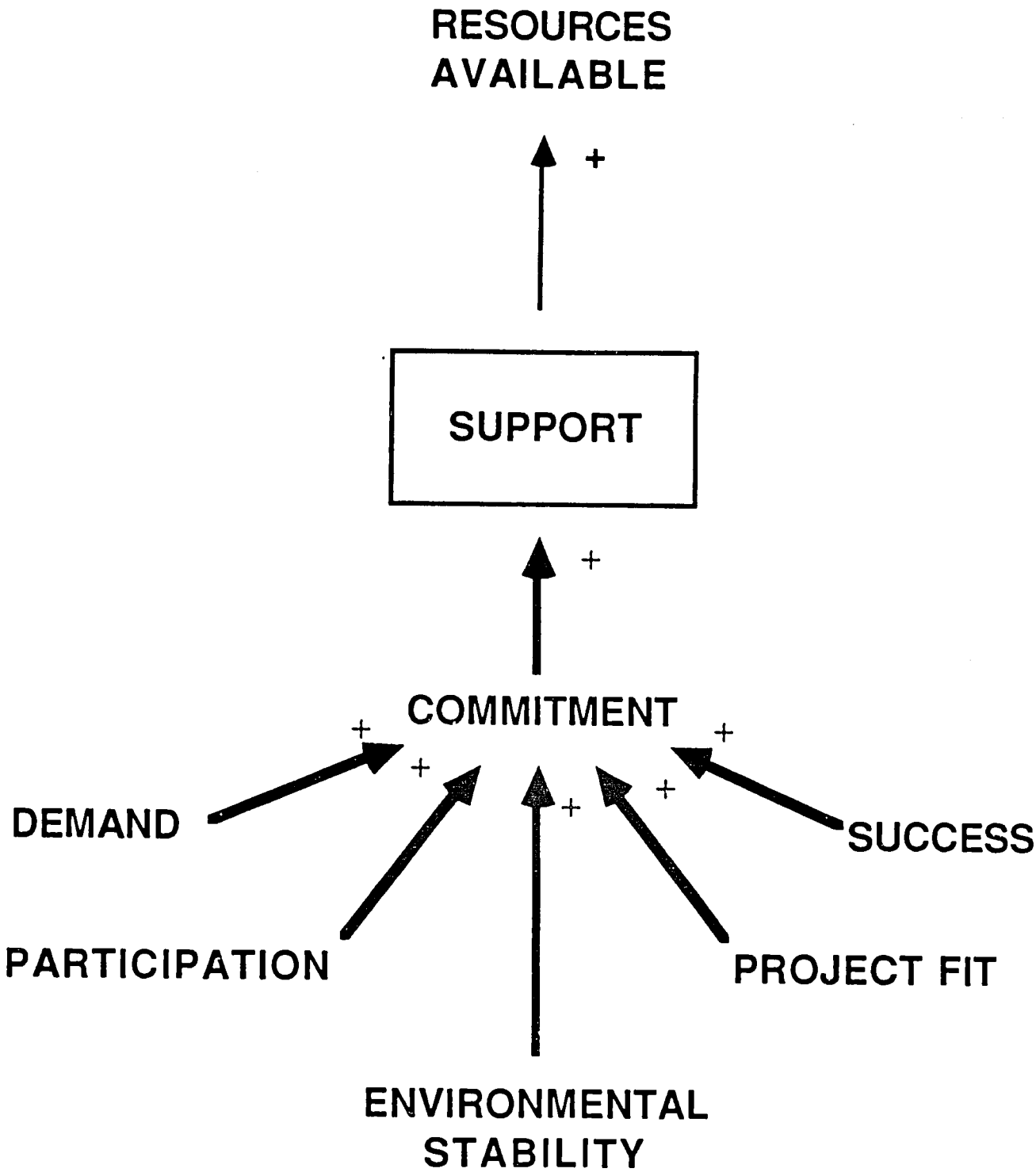


Figure 2: Influences on Support

outcomes, and whether or not the project is being implemented successfully. Figure 2 illustrates these relationships.

The effect of environmental stability may vary. Clearly, great instability and chaos makes it impossible to do any kind of planning and implementation. On the other hand, complete stability might not be a blessing. It seems reasonable to argue that too much stability might undermine support because groups would be totally committed to maintaining the status quo. In the model, the impact of environmental stability on commitment can be portrayed as a nonlinear relationship to capture this changing effect.

Embedded in this part of the model's structure are several assumptions worth noting:

‡ Support does not change instantaneously even though perceptions have changed. One question for empirical field research is to determine what typical response times are and what factors might cause the response time to change.

‡ Project staff do not directly intervene to change the level of support but work indirectly to change participation, demand, fit, and project outcomes.

‡ Differences among constituencies are not modeled explicitly. One might hypothesize that increasing differences would lower environmental stability or lengthen the response time to changes in commitment.

3. Support for a project will be limited if the **Demand** for project services is low. Demand for services changes as people's perceptions of the **fit** of the project with its environment changes and as the project begins to achieve outcomes, which reduces the felt need for services. As described in the preceeding section, demand influences the amount of support the project has.

   Like support, the level of demand does not change instantaneously but gradually adjusts over time to changing perceptions and pressures.

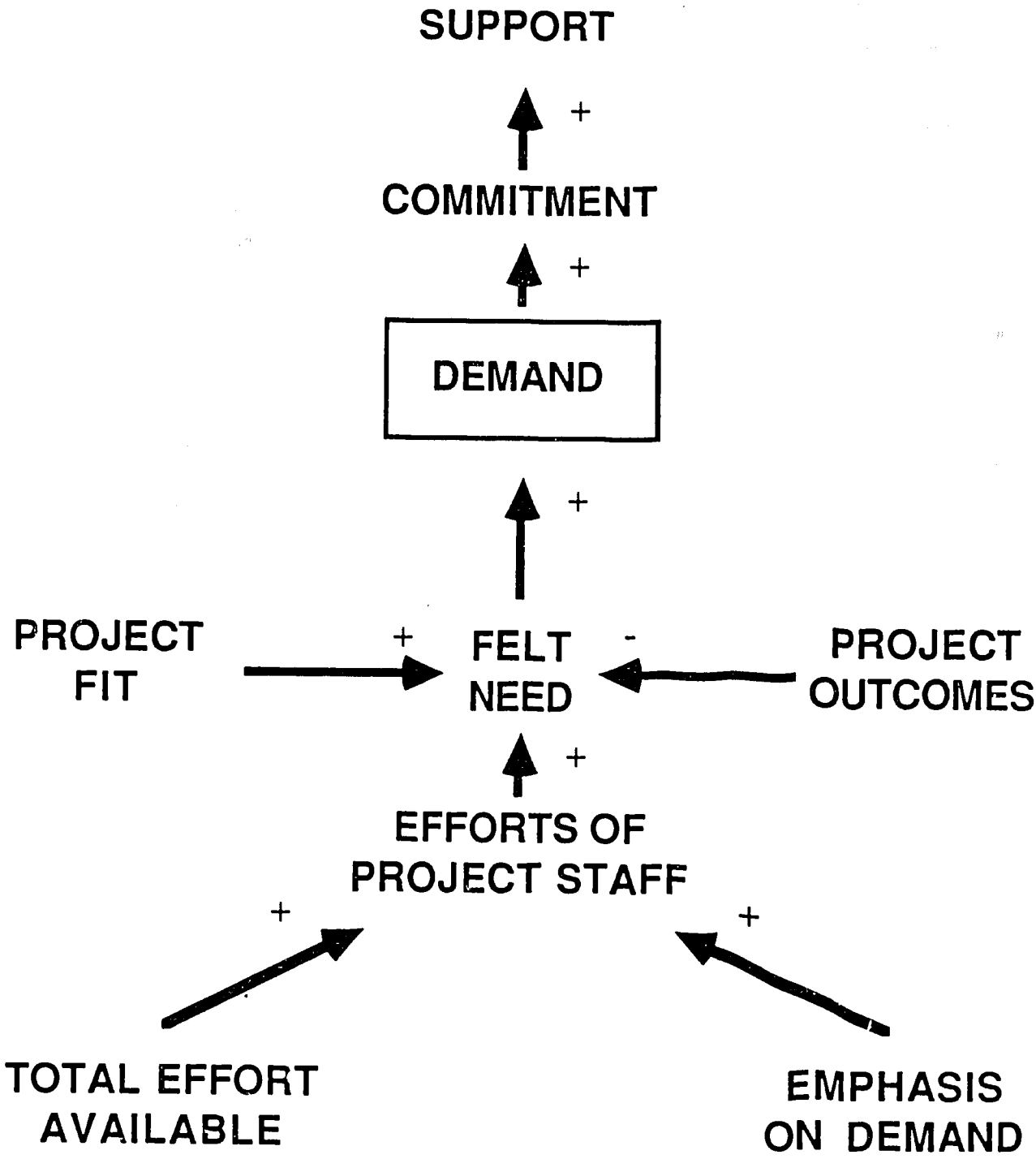


Figure 3: Influences on Demand

Unlike support, project staff can intervene directly to try to increase the felt need for the project - demand mobilization as some would say. How much effort is applied to increase felt need is a function of the size of the discrepancy between the current level and what project staff desire the level to be, the total effort available, and the amount of effort allocated to changing demand (see Figure 3).

This conceptualization assumes two things - both of which are problematic: first, that all effort to raise demand is of equal quality and effectiveness, and second, that whatever target project staff set for felt need is a reasonable one.

‡ Quality of effort surely depends on the skill and commitment of the project staff. Effectiveness of effort probably depends on two factors - the perceived fit of the project with its environment and the level of need already attained. As the level of felt need increases, it becomes harder and harder to continue to raise the level - the law of diminishing returns takes over.

‡ Worldwide, the media are filled with examples of companies and governments trying to stimulate demand artifically. The question is a what point does natural skepticism take over and prevent too great an inflation of need - and how does one model this phenomenon?

4. Support, it is argued above, is enhanced by increased **Participation** of client groups and other stakeholders in policy formulation, planning, and implementation. The degree of participation during project implementation is a direct consequence of the goals project staff set for participation and the amount of effort they put into making it work. As with increasing demand, the effectiveness of their efforts are tempered by their skill and commitment and the law of diminishing returns. However, participation also depends on how much participation the various stakeholder groups desire. The model assumes that increasing demand stimulates participation and that people are more likely to participate if they see tangible outcomes to the project and perceive that the project fits well with the environment.

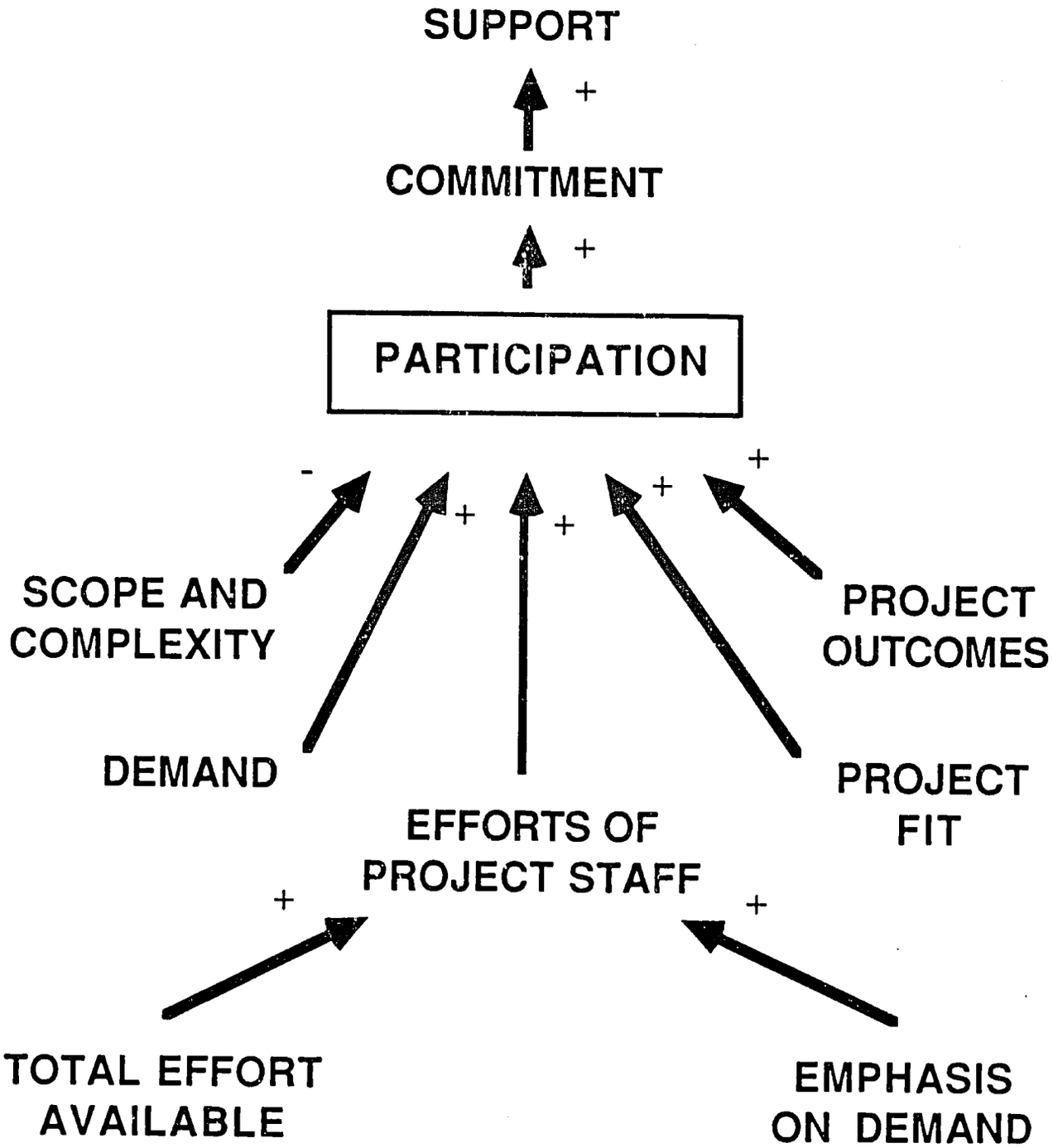


Figure 4: Influences on Participation

The time it takes to change the level of participation increases with increased scope and complexity of the project - often directly related to the number and diversity of clients, decreases with increases in the perceived fit of the project with the environment and with perceived project success. The usual adjustment time probably depends on cultural and political norms on participation.

The underlying assumption here is that participation in social service projects is beneficial and that it means more than token consultation in needs assessment.

5. A common thread running through much of the discussion above is a concern for the **Fit** of the project with its environment - political, organizational, cultural, and economic. Fit relates both to how the project is designed and operated (the tasks) and to the technology used. Fit ultimately affects project outcomes. Poor fit undermines demand for project services, political and financial support, and willingness to participate. It also means that staff will have to direct their efforts to improving fit instead of producing outcomes or services. As with demand and participation, the effectiveness of efforts is influenced by the skill and commitment of staff and the law of diminishing returns. Changing the fit takes longer as the scope and complexity of the project increases. These relationships are illustrated in Figure 5.

   Again, there are opportunities to enrich the model structure. It probably also takes much longer to change the fit if the initial fit, at adoption, is really bad - minor changes are easy, the difficult changes take far more effort. At some point staff may decide it really isn't worth more effort and stop trying to improve fit.

6. A final component of the model structure is the **scope and complexity** of the project. The underlying assumption of the model is that the larger and more complex the project, the more difficult it is to accomplish tasks, regardless of whether they are related to outcomes or to changing demand, fit, and participation.

   Scope and complexity can be treated as fixed for the lifetime of the project or as a variable that changes in response to environmental

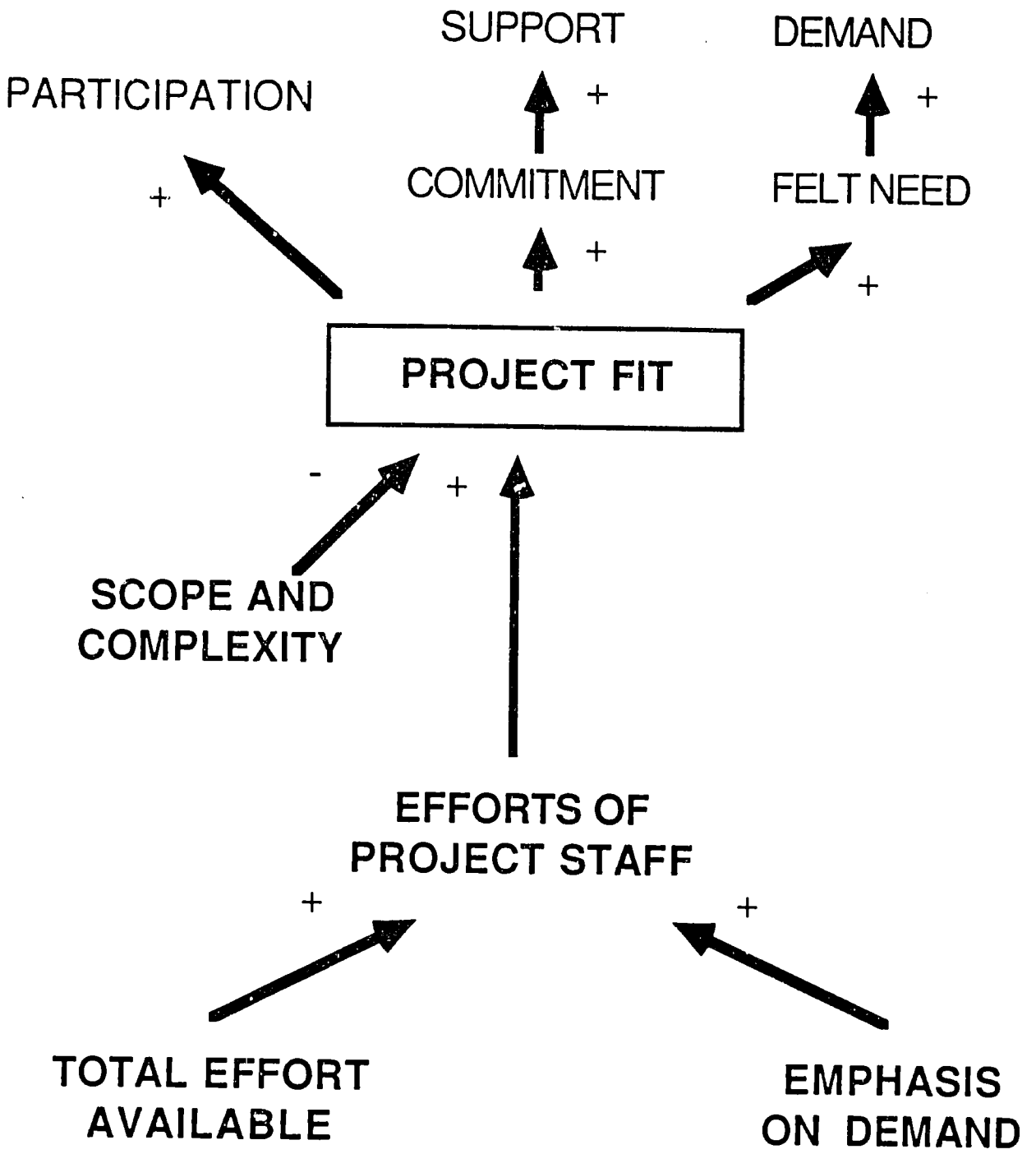


Figure 5: Influences on Project Fit

pressures. In the current formulation of this model, scope and complexity is a fixed constant. In another implementation model I have built to look at educational innovations in the US, I treated it as a variable. Faced with difficulties in maintaining schedules or in generating demand or in improving fit, managers could, as one strategy, scale back the innovation and then try to achieve success with a smaller and less complex project before expanding. Field data suggested that innovations, once downsized, were rarely restored to their original glory.

III. THE COMPUTER SIMULATION MODEL

A. The Model Structure

The ideas embodied in the discussion above were translated into a computer simulation model using STELLA™ on a Macintosh 512KE. The model was constructed in about 1.5 days of work. Testing and debugging the model to produce the simulation runs shown in Figures 11 and 12 has taken another 1.5 days and there still are problems with the model structure.

STELLA™ enables one to build a model with pictoral icons representing different kinds of structural elements (stocks or levels, flows, converters, and connectors) and a series of dialogue boxes to explicate the relationship between variables (see Attachment 1). Unlike other computer simulation modeling software, the software, not the user, writes the model equations. Elements of an "expert system" prevent the user from making structural mistakes. It does not, however, prevent conceptual mistakes!

The model has five stocks, or levels, that correspond to the five core elements described above - **outcomes**, **support**, **demand**, **participation**, and **fit**. These levels increase or decrease in response to inflows or outflows.

For project outcomes, the flow is analogous to an industrial production function - effort and productivity produce outcomes over time (see Figure 6). For the other four stocks, a common stock and flow

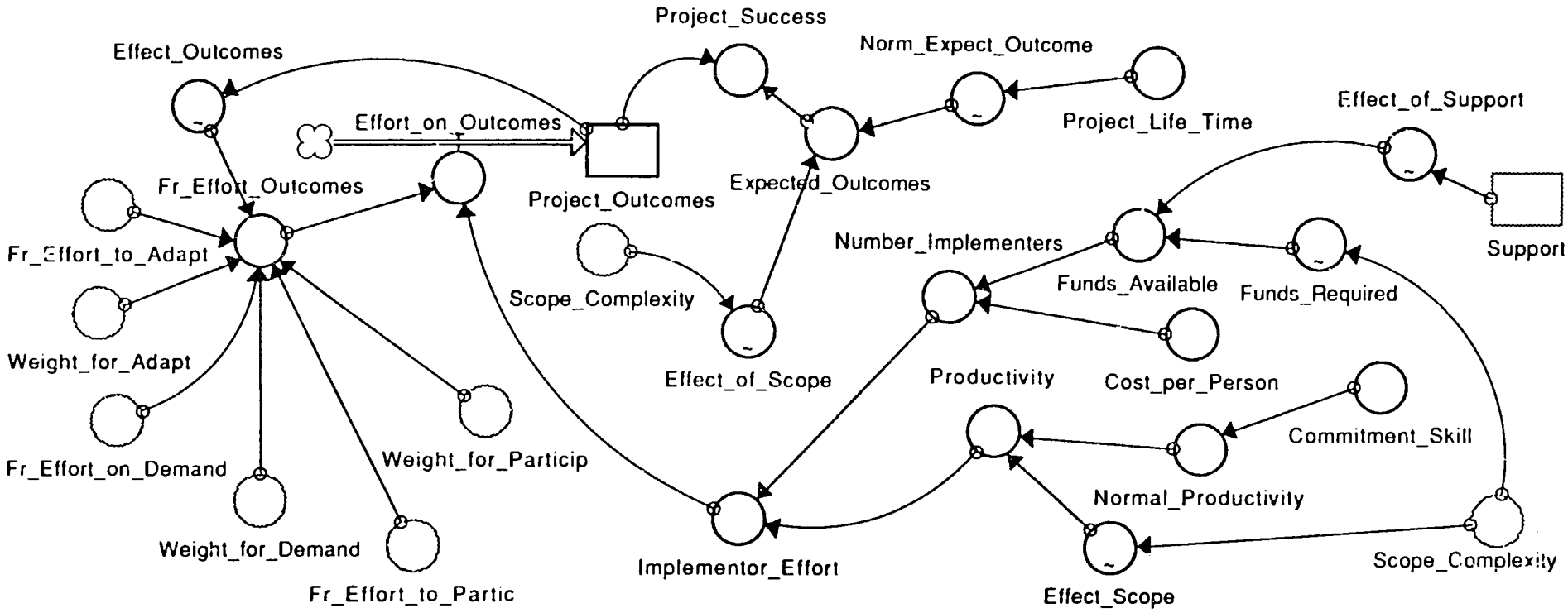


Figure 6: Project Outcomes

structure is used (see Figures 7 - 10). Each stock changes to close a discrepancy between the actual or current value and a desired or perceived value. How fast the discrepancy is closed is a function of the adjustment time, which may be constant or variable, and the size of the discrepancy.

The primary intellectual interest in the model structure lies in the web of connectors and converters that influence the flows for this web is the explication of the concepts and assumptions described above in Section II. The equations that this structure produces are shown in Attachment 2. There are many assumptions embedded in the structural diagram and equations that have not been discussed above.

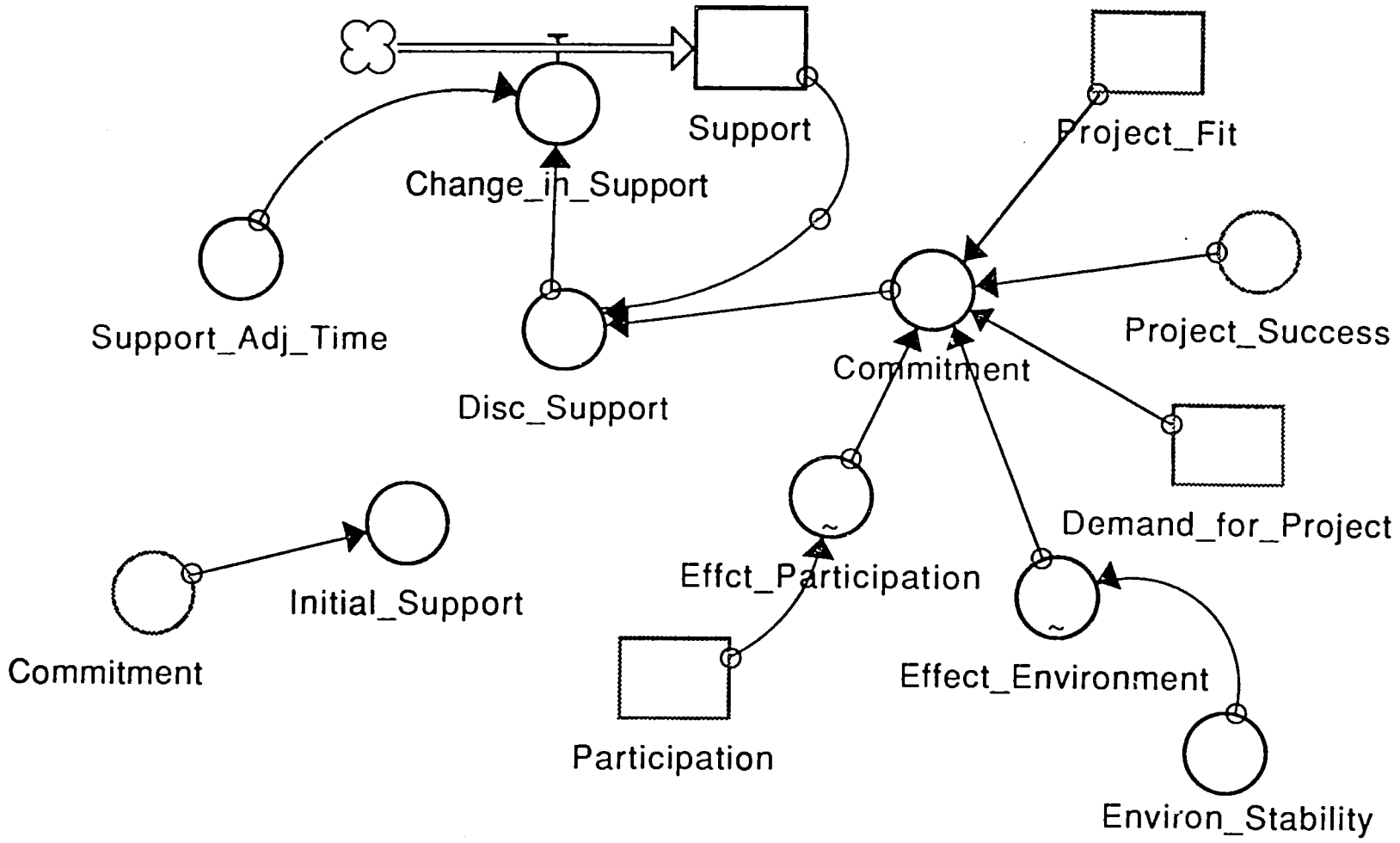
Two points need to be addressed about the quantification of variables in the model:

1. No attempt has been made to model a particular project. Rough estimates have been made on typical response times for changing fit, demand, support, and participation and for other variables - assuming that a typical project unfolds over a five year time span. Parameters can easily be adjusted for different time frames.

2. All five stocks have been formulated as varying between 0 and 100%. In concentrating on the dynamics of interaction among variables, I have purposefully avoided the issue of how to measure in the field such factors as the level of demand or participation. This is **not** to say the issues are not important, they are simply not the issues I addressed. If the model makes sense to others, and helps to illuminate issues around project implementation, then more careful definition of terms might be appropriate.

The next logical step in the development of this model would be the careful analysis by a group of people knowledgeable about project implementation of all the structural linkages within the model and the nature of the nonlinear relationships between variables. This process of model analysis has already, and will continue to, generate questions for field research. In this sense, the modeling process becomes a guide and a teacher.

Figure 7: Project Support



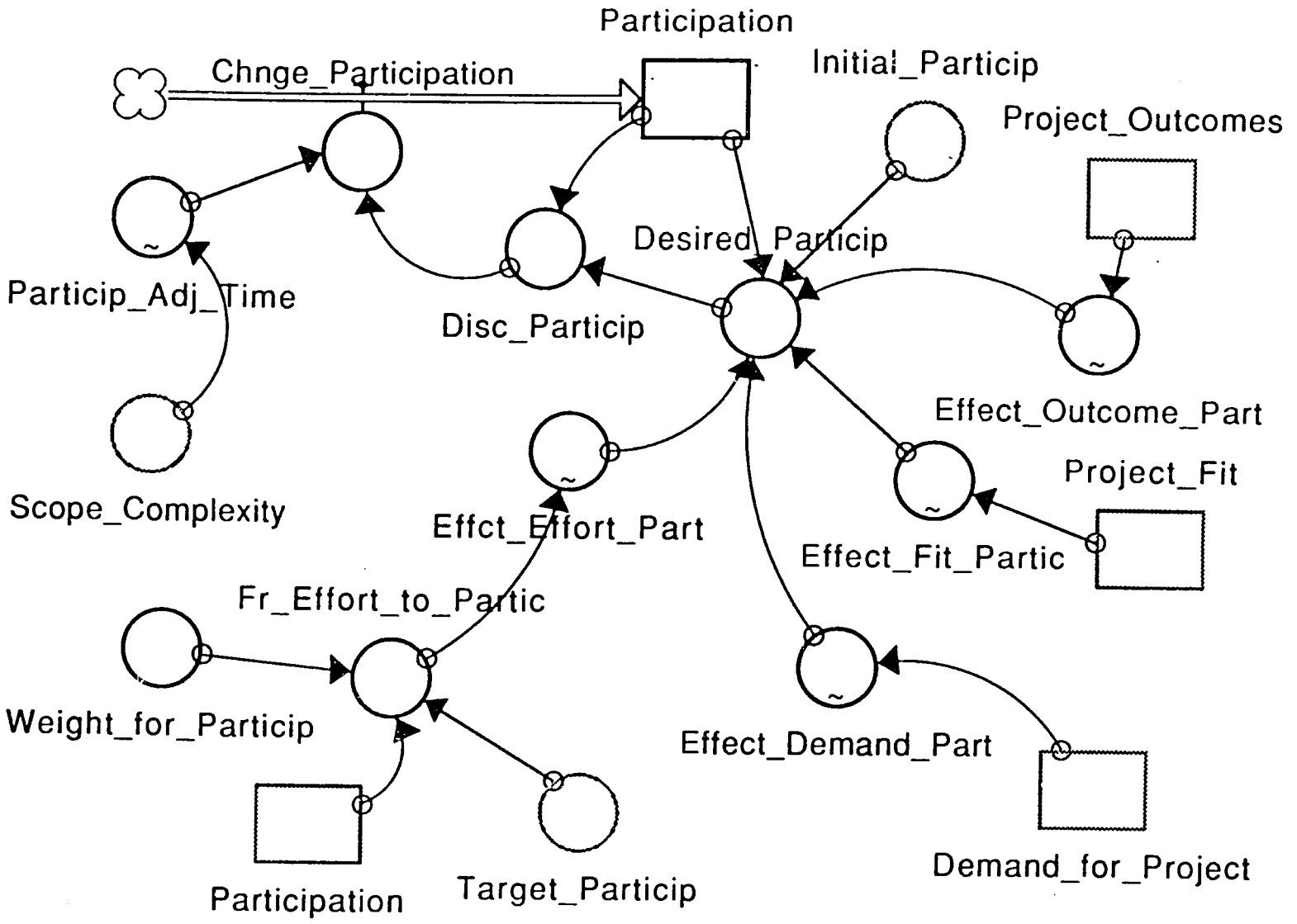


Figure 8: Participation

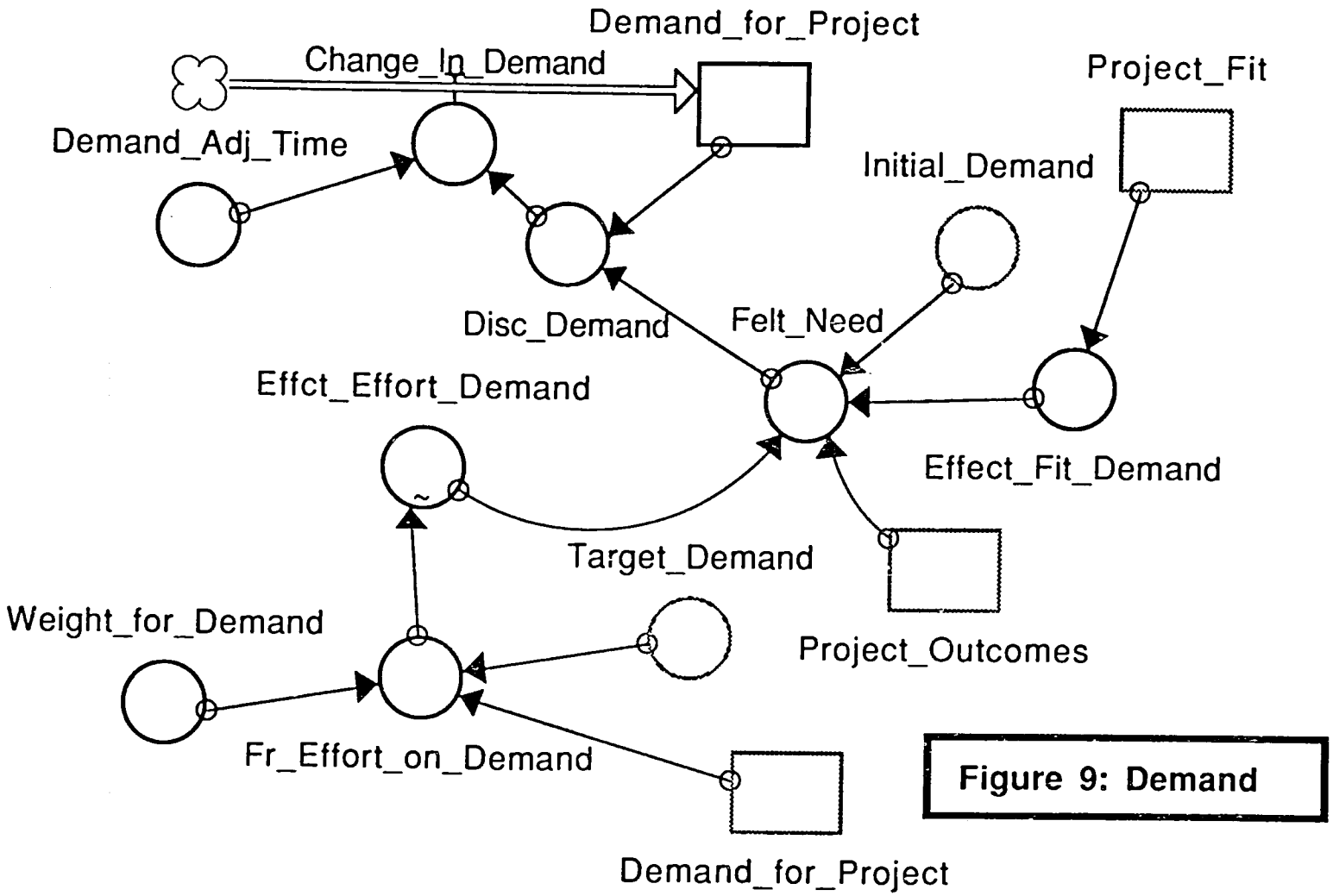
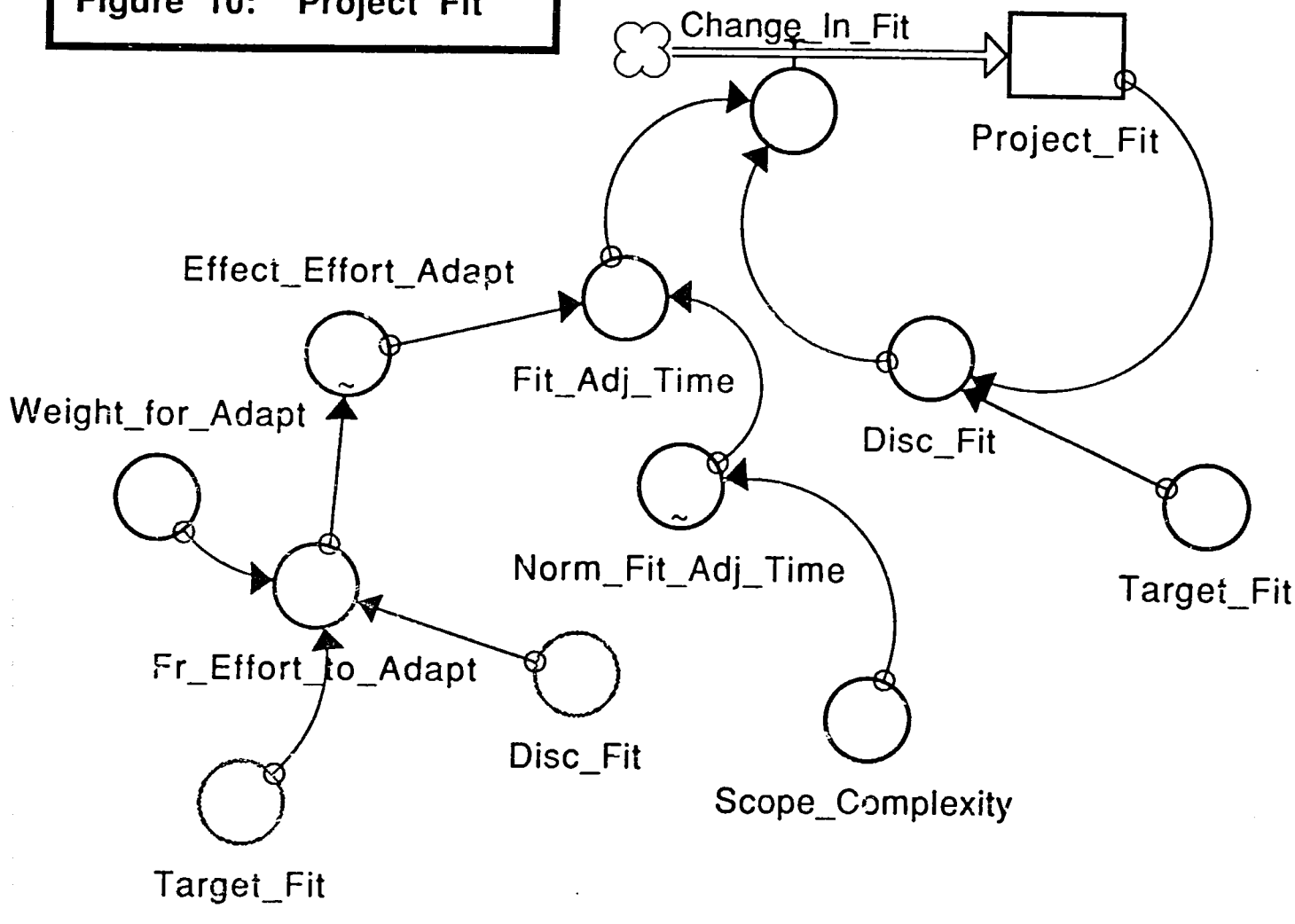


Figure 9: Demand

Figure 10: Project Fit



B. Model Behavior

Given that much work still needs to be done on model development, the behavior of the model shown in Figures 11 and 12 are merely illustrative. Figure 11 shows the behavior of the model under "ideal" conditions, Figure 12 shows behavior under more typical conditions. The differences in parameters are listed below:

	IDEAL	TYPICAL
Initial Fit	100%	50%
Initial Demand	100%	50%
Initial Participation	80%	20%
Commitment & Skill	100%	50%
Environmental Stability	100%	70%

In the ideal scenario, initial support is less than 100% because the project has not produced any outcomes. As outcomes are produced on schedule, the project is perceived as successful and support increases. Over time, demand starts to fall as more outcomes are produced but support remains high because of project success. That outcomes have not quite reached 100% after 5 years is more a function of the parameters chosen for the determination of effort (see Figure 7 and Attachment 2) rather than structural dynamics.

With the typical scenario, the results are markedly different. Initial support is very low - a function of moderate demand, moderate fit, lower environmental stability, and a low rate of participation. Because support is low, less funding is available and project effort on outcomes is reduced - as is effort in all areas. Only project fit improves significantly; however, this change alone is not sufficient to turn the project around.

This typical scenario can now become a laboratory for (1) exploring the reasonableness of the behavior patterns shown (e.g., would real project managers allow project fit so gradually?) and (2) experimenting with different interventions to see if it is possible to rescue this project.

11a

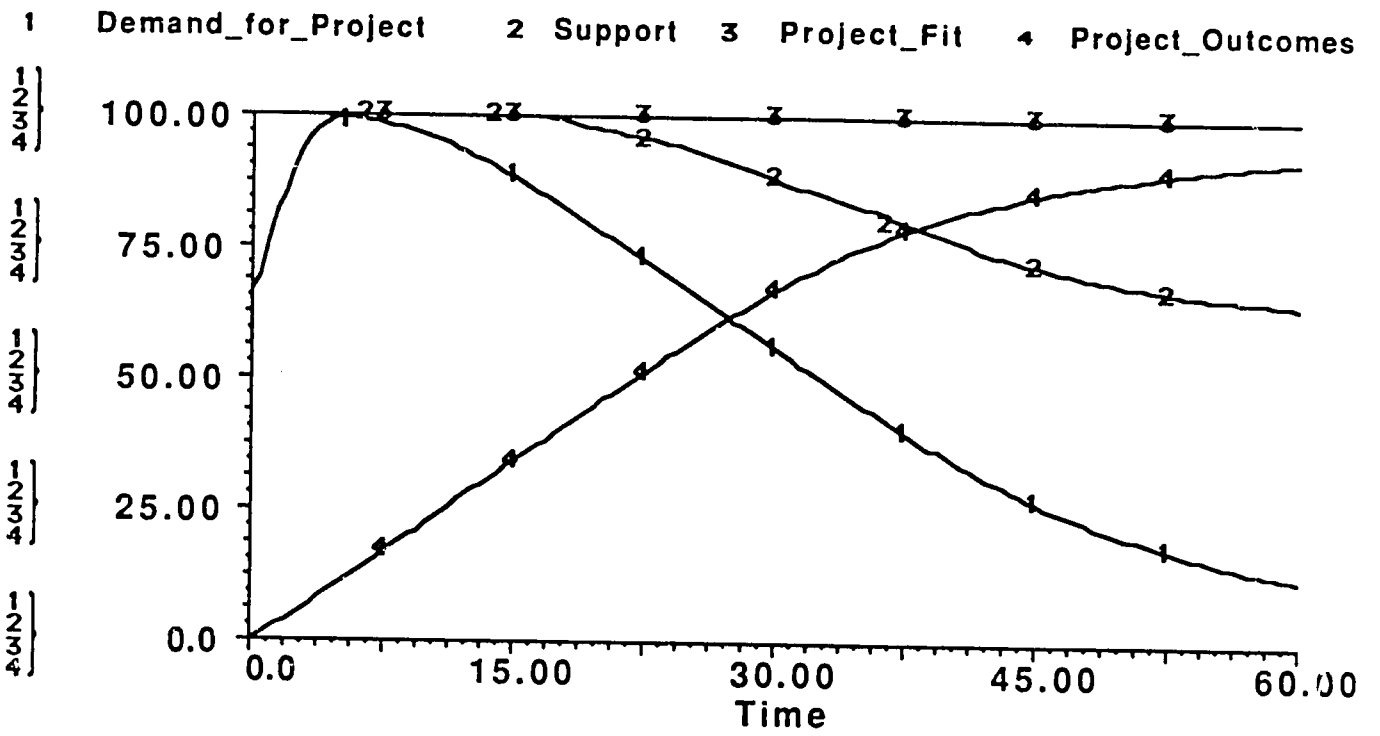


Figure 11: Project Behavior under Ideal Conditions

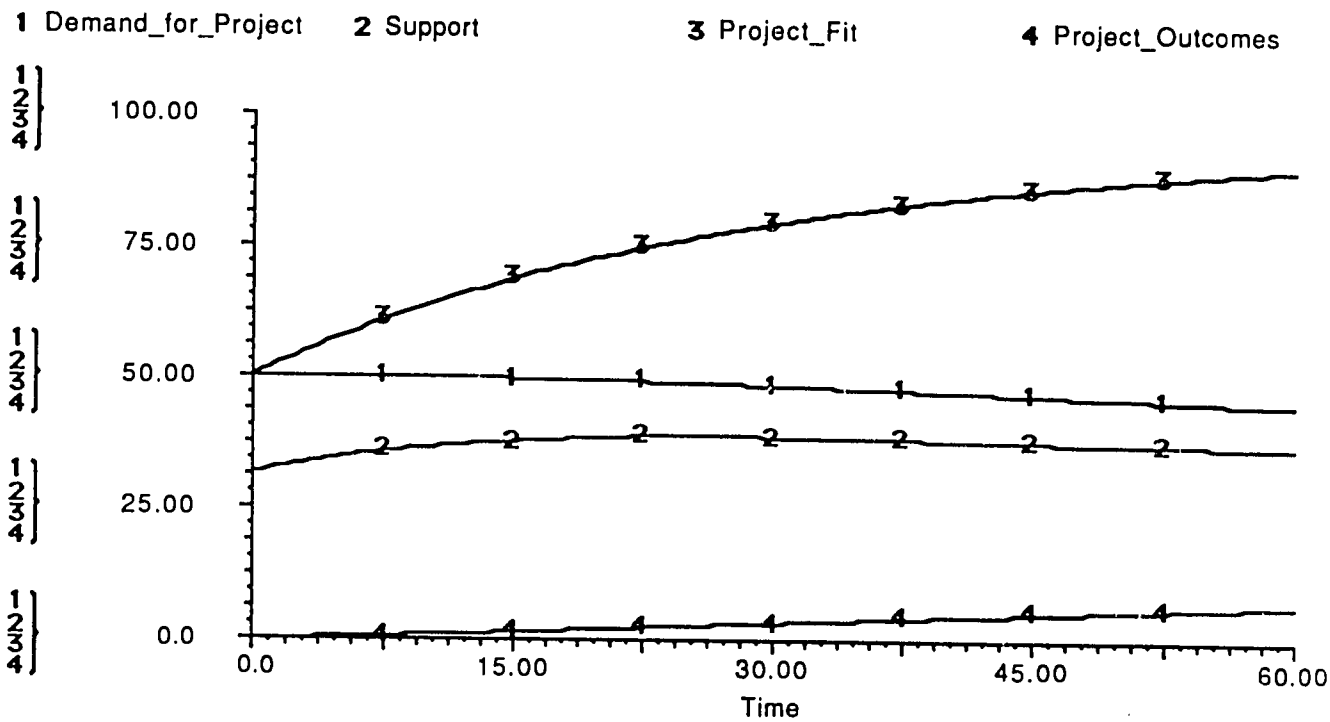


Figure 12: Project Behavior under Typical Conditions

An example of this type of experiment would be to restructure the model to make the decisions about effort allocation responsive to information about the key variables instead of being driven by constants as they now are. This would be one element in the next phase of work on this project.

IV. CONCLUSION

The purpose of this paper has been to sketch, in broad brush strokes, a general theoretical model showing how core elements mutually interact over time to influence, and be influenced by, project outcomes. Much work still needs to be done.

The beauty of the STELLA™ software is that it enables content experts to engage directly in the exploration of model structure and policy options without having to use a computer programmer as an intermediary. Traditionally, computer simulation models have been judged solely on the insights they produce when they are exercised after they are constructed. I would argue that 90% of the learning comes from the process of model building and testing. It has for me.

ATTACHMENT 1

HOW STELLA WORKS

Building the Model

The visual orientation of STELLA is its most unique quality. Macintosh graphic capabilities make it possible to use icons to construct a model. In using STELLA, a structural diagram is constructed by selecting from an icon bar (Figure 1) positioned along the left-hand margin of the screen.

The bar contains four building blocks (the stock, flow and flow regulator, converter and connector) and three tools (a hand for pointing and moving things around, a stick of dynamite to remove elements from the diagram, and a ghost to make copies of elements in the diagram). Each element selected is positioned on the screen and then connected to reflect the structural relationship being considered.

The four building blocks are:

1. The **stock** represents accumulation processes of net inflows minus outflows and is signified by a rectangle, in keeping with longstanding diagramming traditions in engineering and the physical sciences.

2. The **flow and flow regulator** represent the rate of various flows into and out of the stocks. The algebraic direction of flow is automatically established as the arrowhead points the direction of positive flow. It is represented by a pipe, valve and a circle. Signals arrive as inputs to the circle which is the "logic receptacle" portion of the flow regulator. It translates information inputs into outputs that regulate a rate of flow. The valve settings regulate the flows into and out of the stocks.

3. The **converter**, represented by a circle, is a receptacle for logic. It generates outputs by using the logic to manipulate inputs. Within this framework, constants are seen as converters with no inputs.

4. The **connector**, represented as a thin arrow, connects elements to show the flow of information. Built-in rules for constructing

diagrams prevent the user from drawing inappropriate links.

Figure 2 below shows these four building blocks as they relate to each other in a simple structural diagram. In this diagram, the stocks generate signals (pressures) to take actions. Actions regulate the flows. As flows change in volume, they in turn determine new values for the stocks. Stocks then generate new signals. This ongoing, circular control process in which conditions lead to actons which return to influence conditions is known as the feedback loop. The inclusion of feedback loops is a key characteristic of models built with STELLA.

Defining Relationships Among Variables

A model can not be simulated until the logic of each stock, flow and flow regulator, and converter is defined. A building block is defined by double clicking on it to display a Dialogue Box. The Dialogue Box (Figure 3) is used to define and display either the initial values for stocks or the equation logic for the relationship among inputs to a converter or a flow regulator. Each box contains a rectangle that displays the list of allowable inputs which reflect the connector links shown in the diagram. A calculator is supplied for entering numbers and algebraic operators and for calculation. The keyboard can be used for documenting equations with comments. A list of "Built-Ins" (built-in macro functions) is also provided to enable the easy incorporation of mathematical or logical functions and to generate regular, time-varying behavior patterns used in testing the response patterns of a model. A rectangle at the bottom of the box prompts for an equation which can be inserted by typing or by clicking on the list of allowable variables, the calculator, and the built-ins. The example in Figure 3 shows the equation logic for the birth rate in a simple human population model. The two inputs into the birth rate in the structural diagram were the average number of kids per couple and the average number of childbearing couples.

Version 1.3 of the software increases the number of built-in functions from 20 to 33 with six new mathematical functions (arctan, integer, log10, percent, pi, and square root), three financial functions (present value, future value, and periodic payment), and four new special purpose functions (forecast/trend extrapolation, trend of an input, and first and third order

Figure 1: Diagram Window and Icon Bar

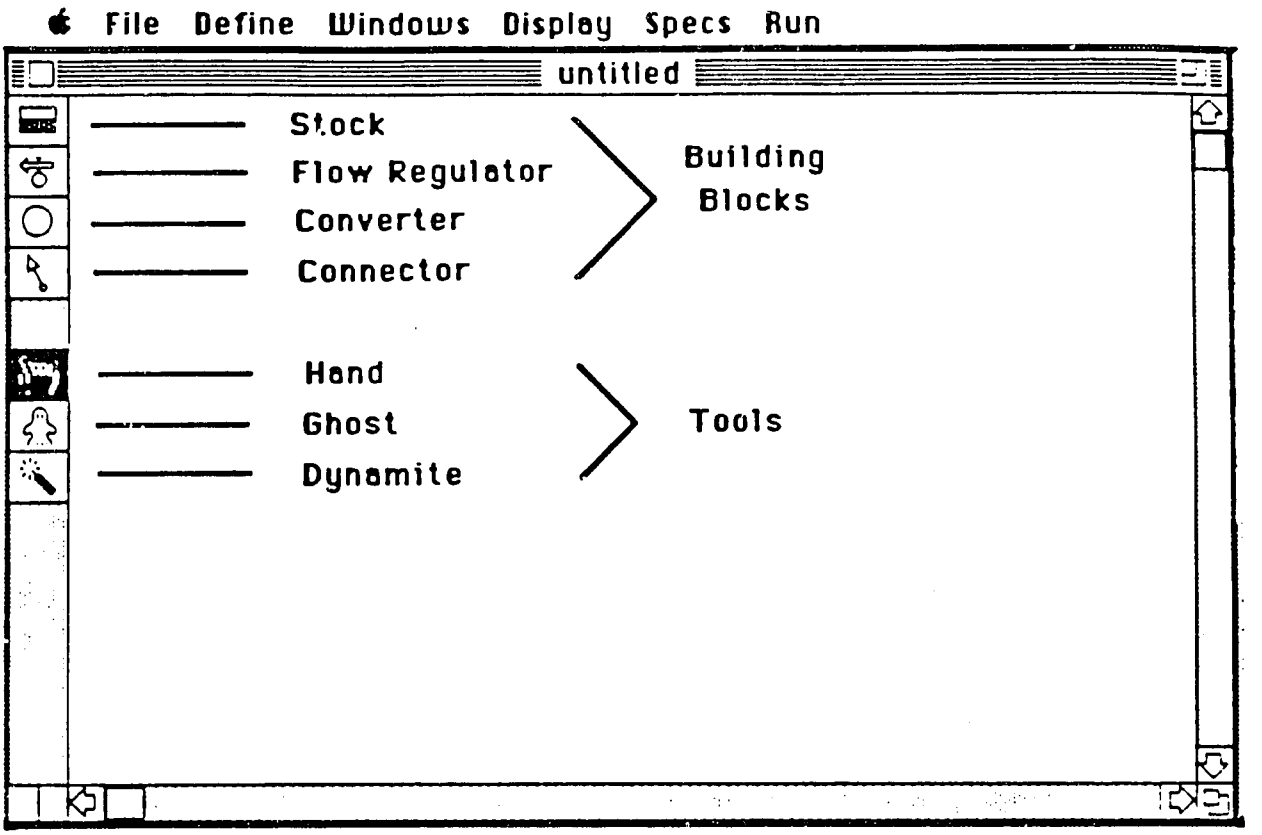
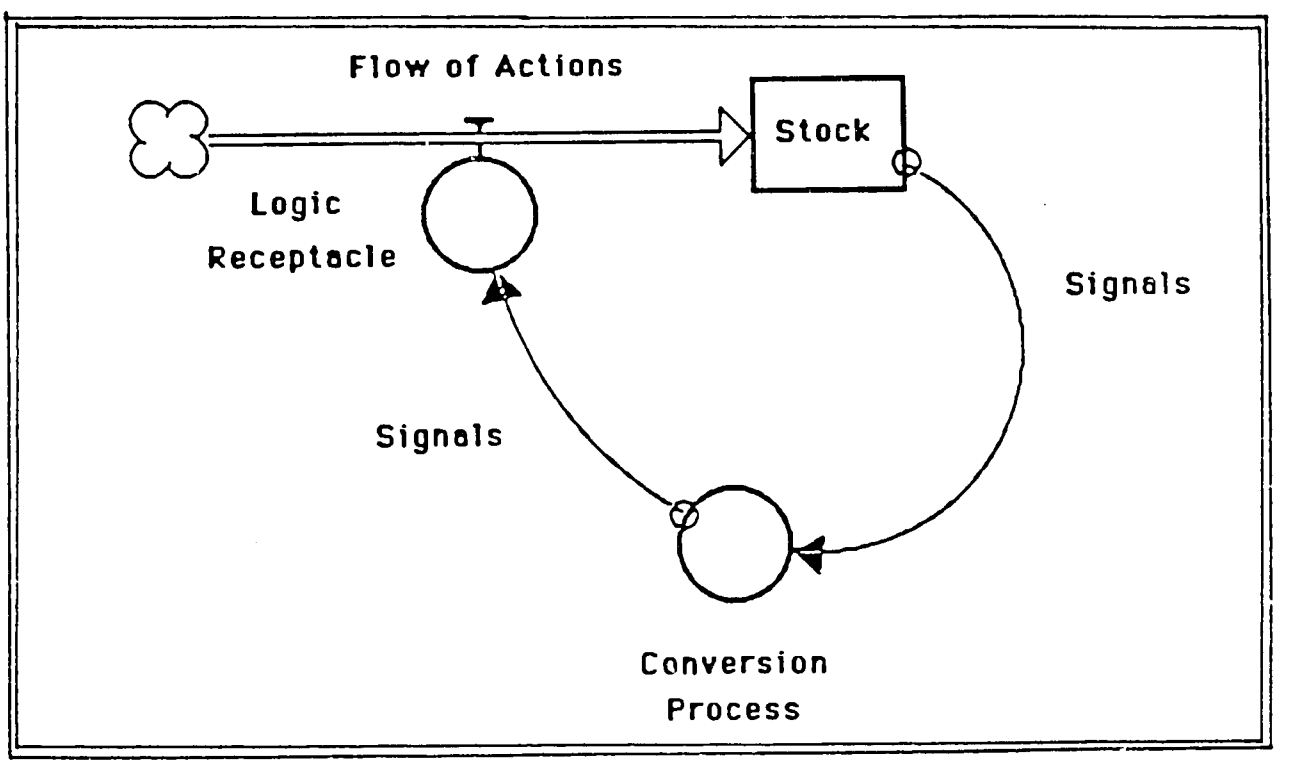


Figure 2: A Structural Diagram for a Simple Feedback Loop



exponential smoothing).

The most significant change in terms of modeling capability of the new software is this increase in the number of built-in functions. These built-in macro functions are used as shortcuts in defining how two variables relate to each other. The additional special purpose functions greatly reduce the time it takes to build complex models because each of the built-ins represents a structural diagram "module" with associated equations that one does not have to recreate each time they are used. This is particularly helpful for the first and third order exponential smooths which are often used to represent the time delays as an opinion, perception, memory or belief - something that is based on experiences - changes slowly relative to changes in the experience itself.

If the relationship between two variables is difficult to express algebraically, STELLA provides a mechanism for graphically representing the relationship. By clicking on the "Become Graph" option in the dialogue box (see Figure 3), a box appears showing a set of coordinate axes and two columns labeled "input" and "output" (Figure 4). One first defines the scales along each axis. Then, to create the graphical function, one can either "draw" the curve on the axis with the mouse, or enter "y" values into the "output" column (the input values are defined automatically when one sets the scale). As the curve is drawn, the numerical values associated with "y" are computed automatically and appear in the column as the sketching is taking place. If numerical values are entered into the column, the corresponding curve will be drawn automatically. Figure 4 illustrates how the death fraction (which produces the death rate when multiplied by the number of people) in the population model example might change as the population increases. This ability to define a graphical relationship easily makes STELLA especially suited to representing nonlinear relationships between variables.

Viewing the Model

STELLA offers two different ways to view the model - as a structural diagram or as a set of equations. The Diagram window (Figure 2), which shows pictorally the structure of the model, can be viewed and printed at different levels of magnification. Regardless of the level of magnification

Figure 3: Defining Relationships: The Dialogue Box

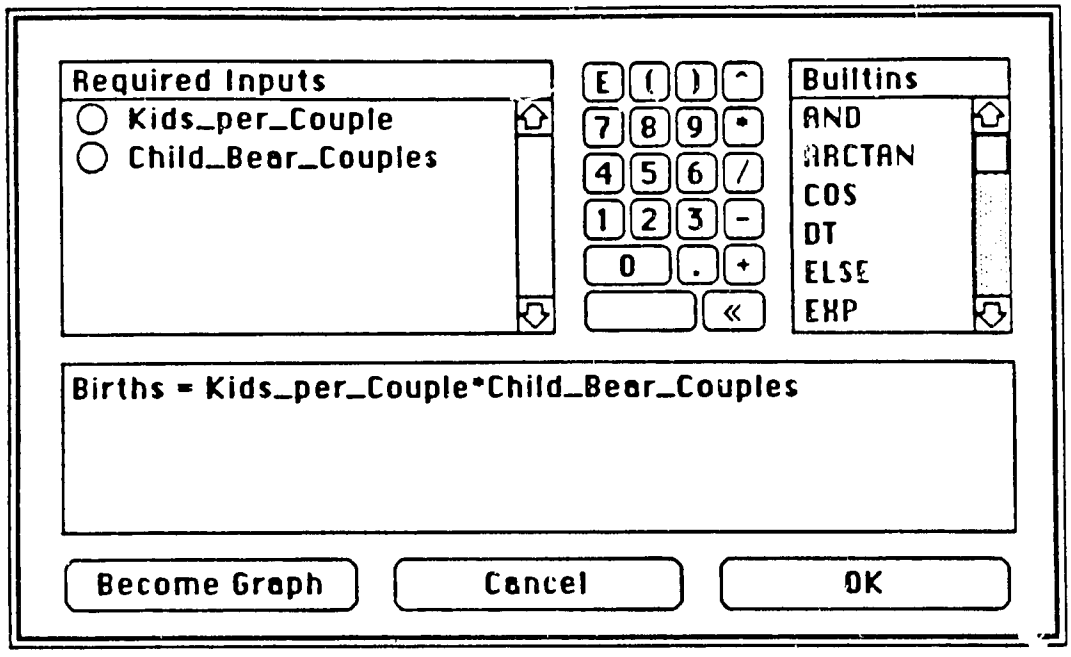
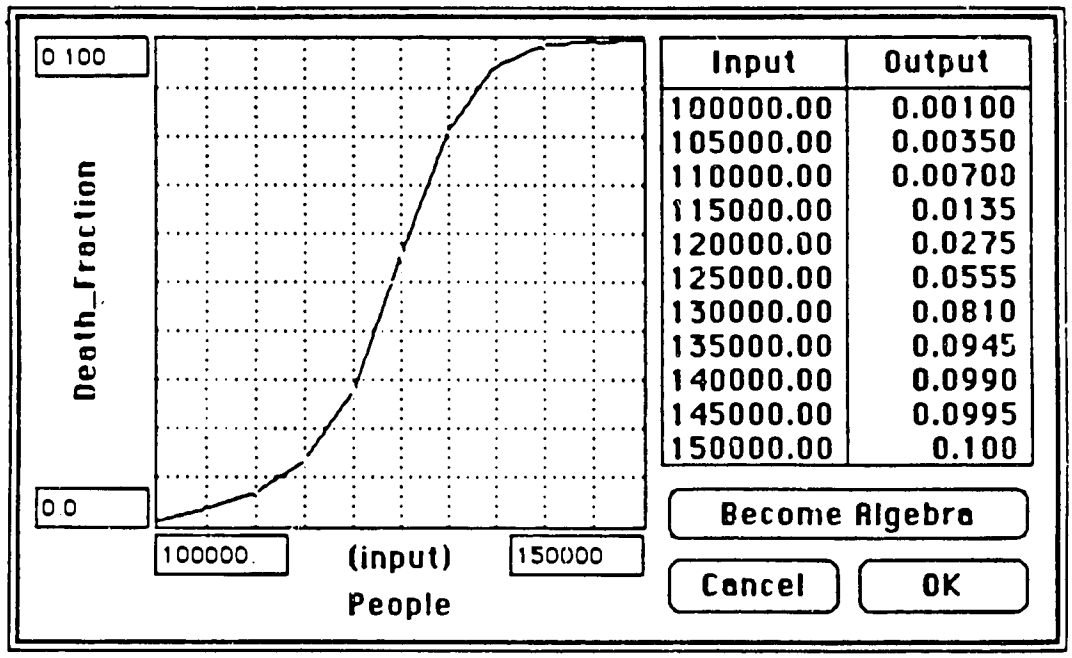


Figure 4: Defining a Non-Algebraic Relationship Between Variables



one can modify the diagram by adding or deleting elements or by moving them around. Bringing the Equations window to the front allows one to scroll through all the equations in the model or to print them out. In the new version of STELLA , diagrams and equations can be saved as disk files to be accessed and manipulated by other MacIntosh applications such as MacPaint™, MacDraw™, and MacWrite™. To facilitate printing, the scale of diagrams can be set and page numbers can be shown on large diagrams to indicate the order of printing. Grid lines indicate the edges of pages.

Running the Model

STELLA greatly simplifies the process of preparing a computer model for simulation. Through its Specs Menu, one selects the computation method, the start and stop times, and the size of the calculation step. One can either specify values or use the default values already programmed in the software. The Run Menu - in addition to the start, pause, stop commands one would expect - contains a scale box which allows one to scale, either manually or automatically, numerical and graphical outputs of the model. The Autoscale feature prompts a trial simulation in which STELLA calculates the maxima and minima for each variable in the model.

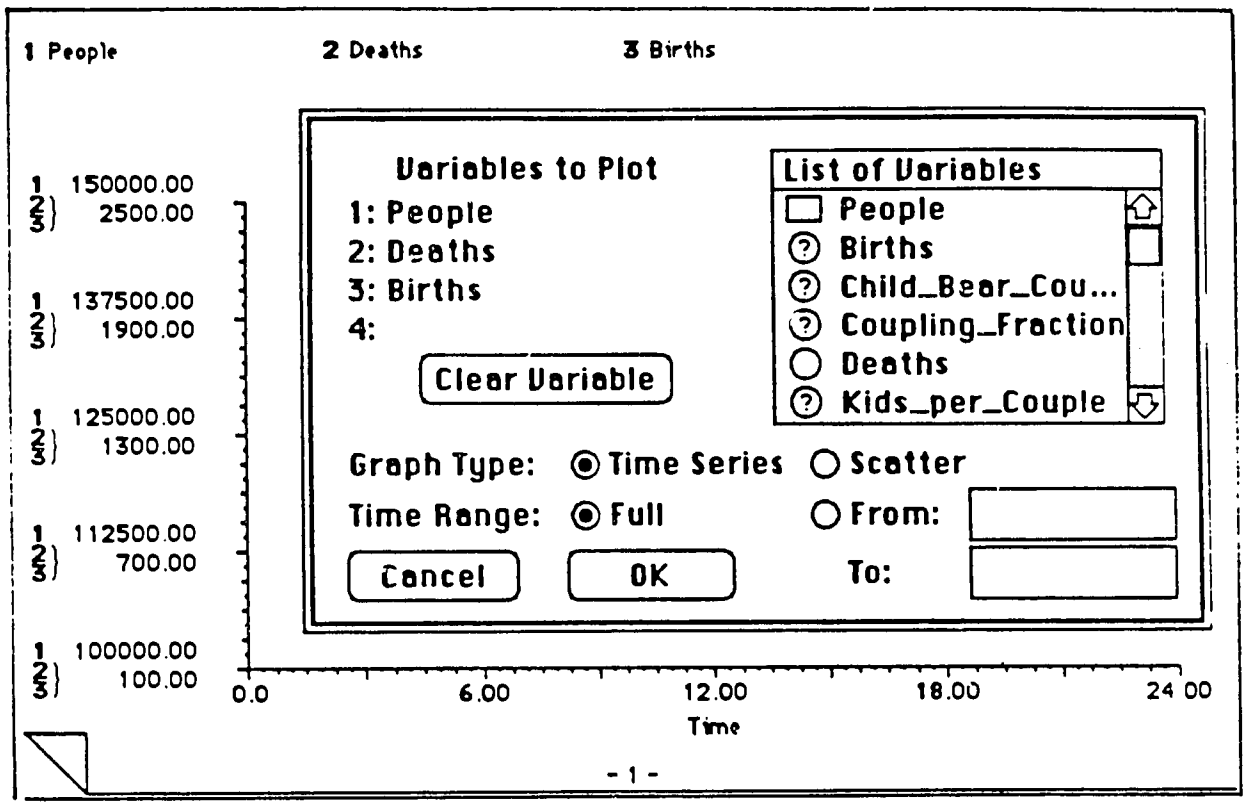
Looking at Model Behavior

There are three different ways to view model behavior during a simulation run - by animation, in a table, or in graphic form. The animation feature allows one to animate stocks, flow regulators, and algebraic or graphical converters. When animated, the stock icon will fill or empty as the level of the stock changes. The regulators and converters display a pointer as on a pressure gauge which moves as the variables value changes.

The traditional table format gives numerical values of any selected variable at specified time intervals. While Version 1.2 limited table output to five variables, the number of variables in table output for Version 1.3 is limited only be available memory. The new version of STELLA also allows one to vary column widths and load tabular data directly into spreadsheet applications such as Excel™.

The Graph window provides a real-time plot of up to four variables

Figure 5: Graphical Output: A Page from the Graph Pad and the Dialogue Box for Defining the Graph



19

over time or a scatter plot of one variable against another. A dialogue box (Figure 5) allows one to specify quickly the variables to be plotted by clicking on the variables in the displayed list. Scales on the axes are set in the scale box of the Run Menu. Since the graph window really contains a graph pad, different "pages" can contain different graphs. Pages of the graph pad can be "locked" to allow one to compare graphical output from one simulation run to the next. Version 1.3 also allows one to save graphs as MacPaint™ or MacDraw™ documents.

ATTACHMENT 2

MODEL EQUATIONS

STOCKS

Demand_for_Project = Demand_for_Project + dt * (Change_In_Demand)
INIT(Demand_for_Project) = Initial_Demand

Participation = Participation + dt * (Chnge_Participation)
INIT(Participation) = Initial_Particip

Project_Fit = Project_Fit + dt * (Change_In_Fit)
INIT(Project_Fit) = Initial_Fit

Project_Outcomes = Project_Outcomes + dt * (Effort_on_Outcomes)
INIT(Project_Outcomes) = 0{percent}

Support = Support + dt * (Change_in_Support)
INIT(Support) = Initial_Support

FLOWS

Change_In_Demand = Disc_Demand/Demand_Adj_Time

Change_In_Fit = Disc_Fit/Fit_Adj_Time

Change_in_Support = Disc_Support/Support_Adj_Time

Chnge_Participation = Disc_Particip/Particip_Adj_Time

ALGEBRAIC CONVERTERS

Commitment = MIN(100,Effct_Participation*Effect_Environment*
(Demand_for_Project+Project_Success+Project_Fit)/3)

Commitment_Skill = 50{percent}

Cost_per_Person = 2E5{dollars}

Demand_Adj_Time = 1{month}

```
Desired_Particip = SMTH3((MIN(100,Participation*(Effct_Effort_Part+
    Effect_Outcome_Part+Effect_Fit_Partic+
    Effect_Demand_Part)/4)),10,Initial_Particip){percent}

Disc_Demand = Felt_Need-Demand_for_Project

Disc_Fit = Target_Fit-Project_Fit

Disc_Particip = Desired_Particip-Participation

Disc_Support = Commitment-Support

Effect_Fit_Demand = Project_Fit/100

Effort_on_Outcomes = Implementor_Effort*Fr_Effort_Outcomes
    {percent/month}

Environ_Stability = .7

Expected_Outcomes = Effect_of_Scope*Norm_Expect_Outcome

Felt_Need = SMTH3(MIN(100,Effct_Effort_Demand*(Initial_Demand-
    (Effect_Fit_Demand*Project_Outcomes))),10){percent}

Fit_Adj_Time = Norm_Fit_Adj_Time*Effect_Effort_Adapt{months}

Fr_Effort_on_Demand = Weight_for_Demand*((Target_Demand-
    Demand_for_Project)/Target_Demand)

Fr_Effort_Outcomes = Effect_Outcomes*(1-(Weight_for_Particip*
    Fr_Effort_to_Partic+Weight_for_Adapt*Fr_Effort_to_Adapt+
    Weight_for_Demand*Fr_Effort_on_Demand))

Fr_Effort_to_Adapt = Weight_for_Adapt*(Disc_Fit/Target_Fit)

Fr_Effort_to_Partic = Weight_for_Particip*((Target_Particip-
    Participation)/Target_Particip)

Funds_Available = Funds_Required*Effect_of_Support
```

Implementor_Effort = Number_Implementers*Productivity

Initial_Demand = 50{percent}

Initial_Fit = 50{percent}

Initial_Particip = 30{percent}

Initial_Support = Commitment{percent}

Normal_Productivity = Commitment_Skill/300{percent/month/person}

Number_Implementers = Funds_Available/Cost_per_Person

Productivity = Normal_Productivity*Effect_Scope

Project_Life_Time = TIME

Project_Success = (Project_Outcomes/Expected_Outcomes)*100

Scope_Complexity = 50{percent}

Support_Adj_Time = 1{month}

Target_Demand = 100{percent}

Target_Fit = 100{percent}

Target_Particip = 100{percent}

Weight_for_Adapt = .3333

Weight_for_Demand = .3333

Weight_for_Particip = .3333

GRAPHICAL CONVERTERS

Effct_Effort_Demand = graph(Fr_Effort_on_Demand)
0.0 -> 1.00
0.100 -> 1.00
0.200 -> 1.00
0.300 -> 1.00
0.400 -> 1.03
0.500 -> 1.13
0.600 -> 1.25
0.700 -> 1.40
0.800 -> 1.61
0.900 -> 1.80
1.00 -> 1.95

Effct_Effort_Part = graph(Fr_Effort_to_Partic)
0.0 -> 0.0
0.100 -> 0.205
0.200 -> 0.475
0.300 -> 0.685
0.400 -> 0.795
0.500 -> 0.865
0.600 -> 0.935
0.700 -> 0.965
0.800 -> 0.985
0.900 -> 1.00
1.00 -> 1.00

Effct_Participation = graph(Participation)
0.0 -> 0.695
10.00 -> 0.780
20.00 -> 0.905
30.00 -> 0.950
40.00 -> 0.960
50.00 -> 0.975
60.00 -> 0.980
70.00 -> 0.995
80.00 -> 1.00

90.00 -> 1.00
100.00 -> 1.00

Effect_Demand_Part = graph(Demand_for_Project)
0.0 -> 0.0
10.00 -> 0.120
20.00 -> 0.225
30.00 -> 0.380
40.00 -> 0.545
50.00 -> 0.700
60.00 -> 0.850
70.00 -> 0.980
80.00 -> 1.00
90.00 -> 1.00
100.00 -> 1.00

Effect_Effort_Adapt = graph(Fr_Effort_to_Adapt)
0.0 -> 10.00
0.100 -> 7.85
0.200 -> 5.85
0.300 -> 4.55
0.400 -> 3.50
0.500 -> 2.70
0.600 -> 2.00
0.700 -> 1.30
0.800 -> 0.650
0.900 -> 0.500
1.00 -> 0.500

Effect_Environment = graph(Environ_Stability)
0.0 -> 0.0
0.100 -> 0.285
0.200 -> 0.570
0.300 -> 0.800
0.400 -> 0.930
0.500 -> 0.970
0.600 -> 1.00

25

0.700 -> 1.00
0.800 -> 1.00
0.900 -> 1.00
1.00 -> 1.00

Effect_Fit_Partic = graph(Project_Fit)
0.0 -> 0.205
10.00 -> 0.215
20.00 -> 0.290
30.00 -> 0.390
40.00 -> 0.490
50.00 -> 0.660
60.00 -> 0.815
70.00 -> 0.940
80.00 -> 1.00
90.00 -> 1.00
100.00 -> 1.00

Effect_of_Scope = graph(Scope_Complexity)
50.00 -> 1.00
55.00 -> 1.00
60.00 -> 1.00
65.00 -> 0.950
70.00 -> 0.860
75.00 -> 0.710
80.00 -> 0.490
85.00 -> 0.370
90.00 -> 0.260
95.00 -> 0.220
100.00 -> 0.200

Effect_of_Support = graph(Support)
0.0 -> 0.00500
10.00 -> 0.0200
20.00 -> 0.0300
30.00 -> 0.0550
40.00 -> 0.135

50.00 -> 0.305
60.00 -> 0.630
70.00 -> 0.825
80.00 -> 0.955
90.00 -> 0.995
100.00 -> 1.00

Effect_Outcomes = graph(Project_Outcomes)
0.0 -> 1.00
10.00 -> 1.00
20.00 -> 1.00
30.00 -> 1.00
40.00 -> 1.00
50.00 -> 1.00
60.00 -> 1.00
70.00 -> 0.830
80.00 -> 0.565
90.00 -> 0.260
100.00 -> 0.0

Effect_Outcome_Part = graph(Project_Outcomes)
0.0 -> 0.800
10.00 -> 0.840
20.00 -> 0.876
30.00 -> 0.912
40.00 -> 0.954
50.00 -> 1.00
60.00 -> 1.06
70.00 -> 1.12
80.00 -> 1.18
90.00 -> 1.20
100.00 -> 1.20

Effect_Scope = graph(Scope_Complexity)
0.0 -> 1.00
10.00 -> 1.00
20.00 -> 1.00

30.00 -> 1.00
40.00 -> 0.940
50.00 -> 0.790
60.00 -> 0.585
70.00 -> 0.455
80.00 -> 0.385
90.00 -> 0.340
100.00 -> 0.310

Funds_Required = graph(Scope_Complexity)
0.0 -> 0.0
10.00 -> 150000.00
20.00 -> 225000.00
30.00 -> 400000.00
40.00 -> 775000.00
50.00 -> 1825000.0
60.00 -> 3275000.0
70.00 -> 4325000.0
80.00 -> 4875000.0
90.00 -> 5000000.0
100.00 -> 5000000.0

Norm_Expect_Outcome = graph(Project_Life_Time)
0.0 -> 1.00
6.00 -> 12.00
12.00 -> 22.50
18.00 -> 35.00
24.00 -> 45.50
30.00 -> 60.00
36.00 -> 71.50
42.00 -> 85.00
48.00 -> 93.50
54.00 -> 98.50
60.00 -> 100.00

Norm_Fit_Adj_Time = graph(Scope_Complexity)
0.0 -> 0.700

78

10.00 -> 0.900
20.00 -> 1.15
30.00 -> 1.65
40.00 -> 2.65
50.00 -> 4.45
60.00 -> 7.35
70.00 -> 8.70
80.00 -> 9.35
90.00 -> 9.85
100.00 -> 10.00

Particip_Adj_Time = graph(Scope_Complexity)
0.0 -> 0.525
10.00 -> 0.600
20.00 -> 0.825
30.00 -> 1.27
40.00 -> 1.98
50.00 -> 3.23
60.00 -> 4.20
70.00 -> 4.72
80.00 -> 4.90
90.00 -> 5.00
100.00 -> 5.00