Creating System Dynamics Models Hierarchically Using SADT

M. B. Hommel & C. McGowan
GTE Laboratories Incorporated
Waltham MA, USA

Abstract

SADT™(A trademark of SofTech, Inc., Waltham, MA), a hierarchical system description notation, was used to create System Dynamics models. This paper discusses the two SADT model types, data and activity, and their correspondence with System Dynamics patterns. Rules for transforming an SADT data model to a System Dynamics model, semi-automatically, are proposed. This information is then used in a step by step translation from a SADT data model to a System Dynamics simulation model. An example is given showing how the SADT hierarchy enhances the understanding of the simulation model.

Introduction

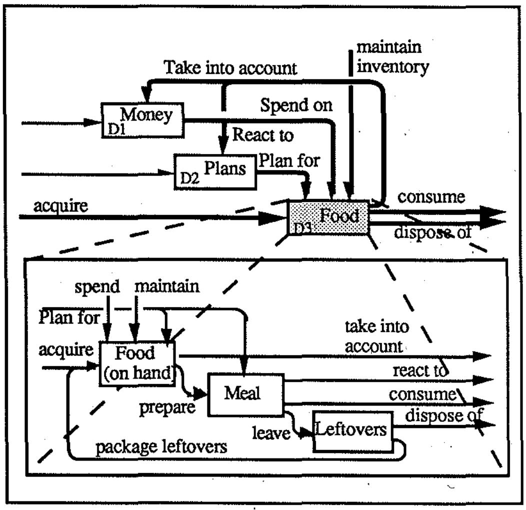
SADT (Structured Analysis and Design Technique) is widely used to create functional, object and state-transition models of systems.(1) Similar to System Dynamics, SADT emphasizes modeling to answer questions, including the relevant system context The large number of existing SADT models has evoked interest in a method to convert these models into simulation models. Because the hierarchical nature of SADT makes it easily understood by non-modelers, we were also interested in using SADT to design System Dynamics models.

This paper describes a method for translating an SADT model into a System Dynamics model. The first part of the paper contains a general description of SADT activity models and data models. The second, contains a list of System Dynamic generic patterns with corresponding SADT models for each. The last part describes the relation between System Dynamics models and SADT data models. This section also discusses a method for converting an SADT data model to a simulation model, semi-automatically. An example is shown of the translation to a System Dynamics model from an SADT data model.

Part I SADT Models

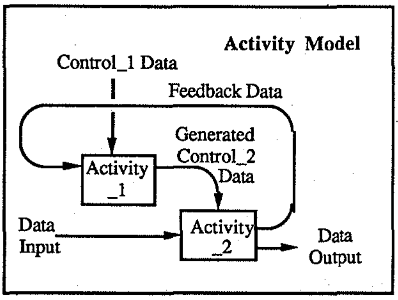
SADT is a highly structured system description notation. A subset, IDEF0 (the Integrated Computer-Aided Manufacturing Definition Method) sponsored by the U.S. Air Force, has been widely used to model manufacturing systems. SADT has a formal syntax, and a methodology that features configuration control and quality/context review to converge on a common group understanding. It features a graphical representation which is recognizable world-wide. It emphasizes modeling to answer questions, including the relevant system context (i.e., how the modeled system is embedded as a component of a larger system), and identifying feedback loops.

SADT models are focused on one subject; the boundaries of the system are precisely defined. Thus, the model includes a description of the contents of the system. Each model also has a specific viewpoint. This characteristic helps identify important, and unimportant system features. A particular model is either an activity or a data model, and each model has a defined purpose, viewpoint, and context. Moreover, the "semantics" (or meaning) of boxes and lines are, in effect, model specific as determined by the model's author. This flexibility makes SADT useful for designing simulation models.



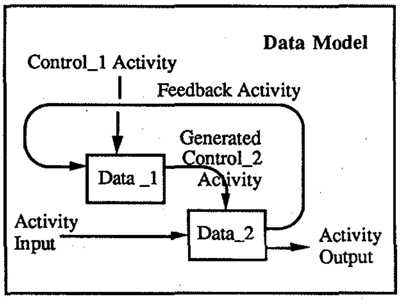
Models in SADT contain a coordinated set of diagrams, each with the same graphical form. Diagrams consist of text plus only two types of objects: boxes and directed lines. A model consists of a set of diagrams, and lower level diagrams are an expansion of a single box on an upper level diagram. The diagram on the right shows two levels of the SADT model used in Part III, and illustrates this principle.

There are two types of SADT models. One, focuses on functions or activities. In this type, activities are successively decomposed into more and more detail and sub activities. This first type is called an activity (or functional) SADT model.



The second type of SADT model has system objects as its focus. Here, the hierarchy is of more and more detail about the things in a system. This type is called a data model.

In a functional model, boxes represent activities, and the lines are data which are input to, output from, or control of the activities. In an object or data model, boxes are things, and the lines are activities which act on the things. All inputs enter a box on the left. All outputs leave a box on the right. Data which enters the top of a box controls the data or activity in the box. The diagrams on the right show these concepts.



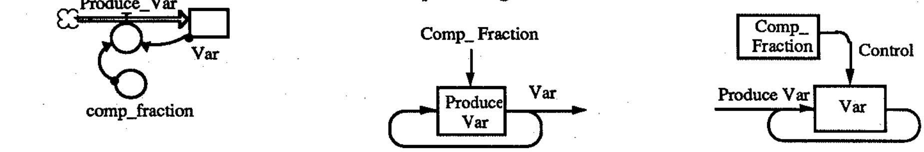
The activity model contains two activity boxes and several types of data lines. There can be feedback in the model, either as shown for input data, or as feedback control data. In the data model, the reverse is true. It contains two data boxes and several types of activity lines. In this model, the control lines are also activities, not data as in the activity diagram.

Part II Generic Patterns

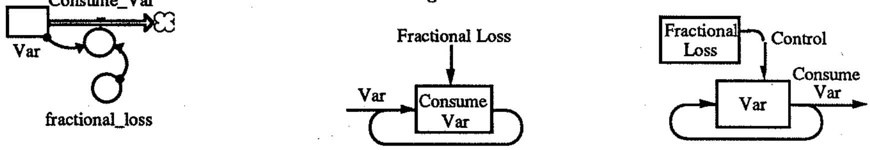
Richmond and Peterson have classified System Dynamics behavior patterns into generic processes and sub-systems.(2) This section lists these processes, with the System Dynamics diagram, the corresponding SADT activity diagram, and the SADT data diagram. The five examples following are simple flows.

System Dynamics Model	SADT Activity Model	SADT Data Model

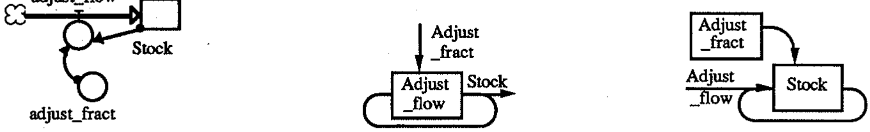
Compounding Process



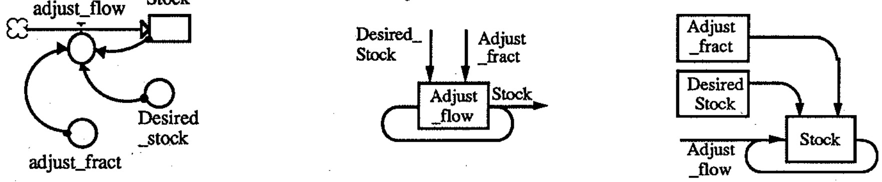
Draining Process



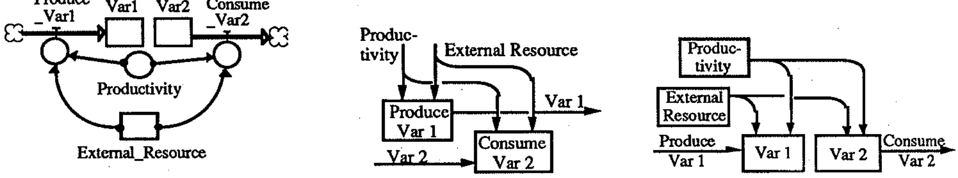
Implicit Goal-Seeking Process



Stock Adjustment Structure



External Resource Production



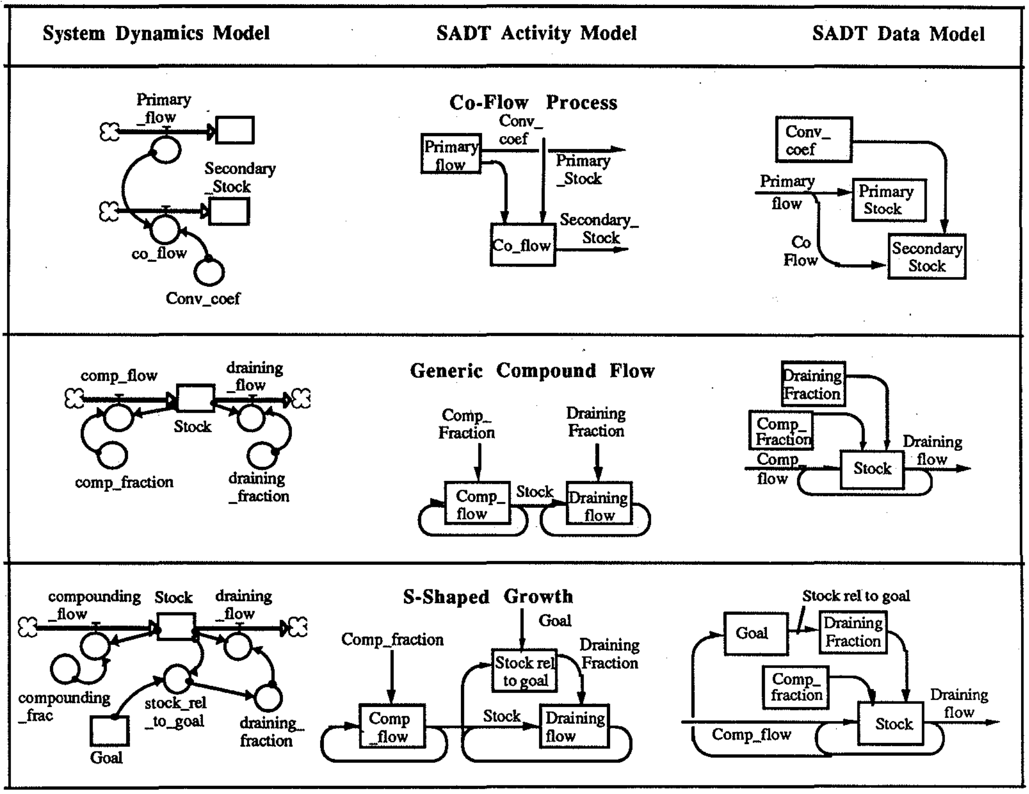
These examples lead us to conclude:

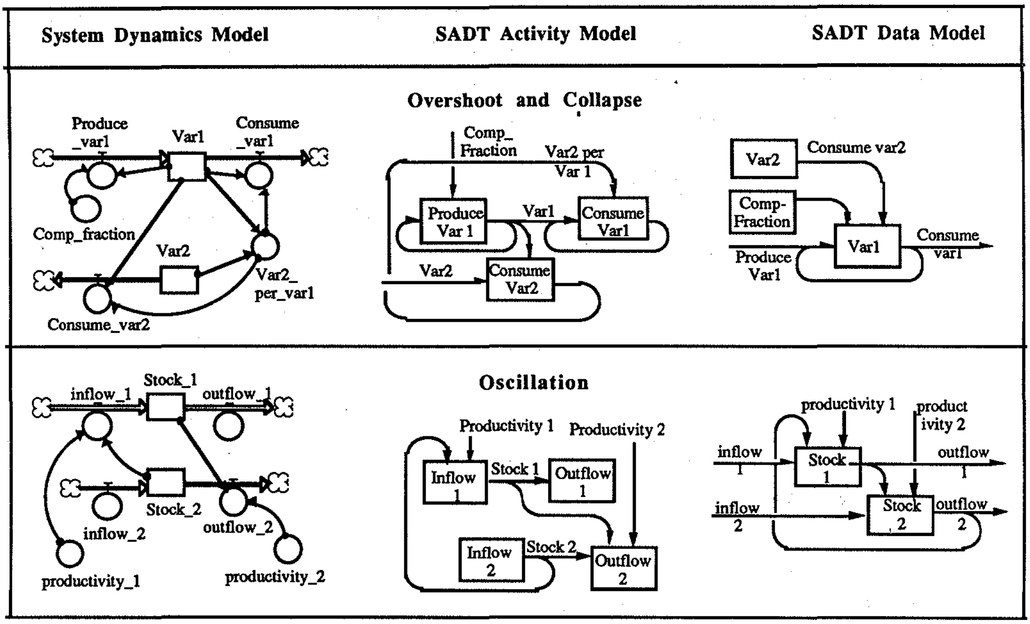
- In general, System Dynamics stocks translate to lines in the activity diagrams, and to boxes in the data diagrams. Flows, on the other hand, translate to boxes in the activity diagrams, and to lines in the data diagrams.

- Converters become, or generate, control inputs to the activity or data boxes. For example, the Implicit Goal Seeking Process is a subset of the Stock Adjustment Structure, where the latter has two control lines, instead of one.

- Feedback in the SADT models occurs in a manner similar to feedback in the System Dynamics model. For example, the External Resource Production shows a produce and a consume component. The produce part differs from the Stock Adjustment Structure only because there is no feedback from the value of the stock to the flow producing it. Thus, feedback in the SADT model is absent, also.

The next five examples are compound flows.





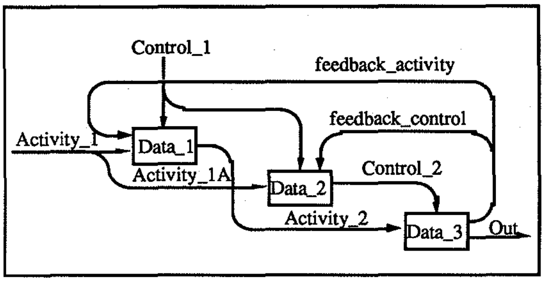
These examples show that SADT diagrams for compound flows combine in the same manner as the System Dynamics models. Thus, the Generic Compound Flow joins a Compounding and a Draining Process.

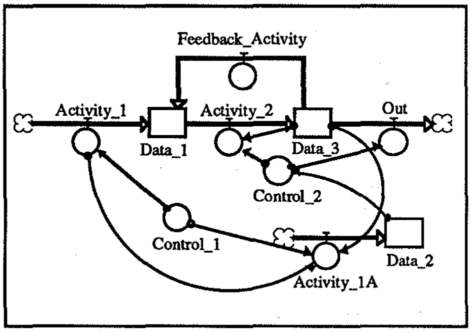
Part III Translation of SADT Data Models

We have studied transforming both activity and data models into System Dynamics models. Although it is easier to understand hierarchy in an activity model than in a data model, and activity models are more widely used, it is more difficult to do the translation. We found that additional user input was required. Our rules for the semi-automatic translation of a hierarchical activity model to a System Dynamics model have been defined and applied successfully to several business system models.(3)

For two reasons, we found it easier to translate an SADT data model to a System Dynamics model. The first related to system context. The boundaries of a System Dynamics model are flows and therefore correspond to activities. Since the boundaries of an SADT data model are also activities, the contexts can be matched. In contrast, the boundary of an SADT activity model is a set of data. The second reason was the direct relationship between System Dynamics stocks and SADT data boxes. There is an easy one-to-one correspondence when comparing stocks connected by flows with data boxes connected by activity lines. Boxes stay as boxes, and the arrow showing the direction of a flow remains as the directed line entering or leaving the data box.

This is illustrated in an abstract SADT data model of a system consisting of three data groupings and their associated input, output, and control activities. The model, and the System Dynamics translation, are given next. Note: Activity_1A is shown as a function of Activity_1. This depends upon the actual quantities involved. Note also: Control_2 is shown controlling the input and the output activities for Data_3. This also depends on the actual model.



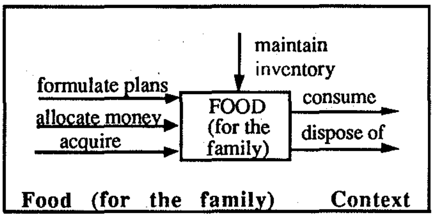


The general rules for creating a System Dynamics model from an SADT data diagram are as follows:

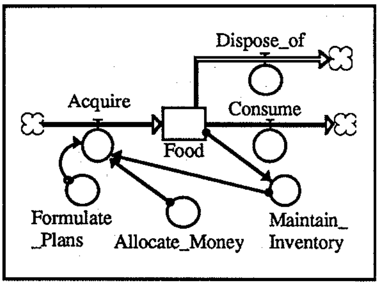
1. SADT data boxes become stocks.
2. All stocks of the same variable are connected. If the output of one data box is input to another, these should be different states of the same stock, and should connected by a flow. If there is more than one output, the stock flows in more than one way.
   a. Inputs to boxes are flow rates into the stock.
   b. Outputs that are inputs to the next box are flow rates to the next stock.
   c. An output of the diagram becomes a flow rate to a cloud, where a cloud is considered a sink, outside the system.
   d. An input to the diagram becomes a flow rate from a cloud, and this cloud is a source from outside the system.
3. Control activities translate to converters. Controls contain information to regulate the flow rates.
   a. Converters are input to flow rates.
   b. The input to the converters comes from the stock that corresponds to the source of the control.
   c. A control that comes from outside the diagram originates a converter. (i.e., that converter has no input)
4. There will be a different set of interconnected stocks and flows for each variable.
5. The sets of stocks are connected through converters

Example - Feed Family

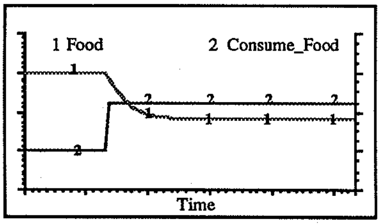
A completely worked out activity model for Feed Family appears with comments in the book SADT, Structured Analysis and Design Technique.(4) We created a dual data model by considering the birth to death life history of "the food." The diagram shows the top level, or context, of the SADT data model.



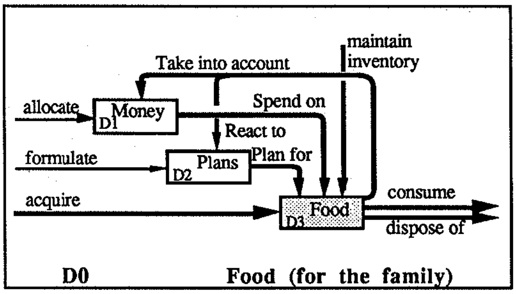
This data model translates to a top level simulation model in a straight-forward way. The decisions made in this translation process only involved separating the activities that directly add food to the stock, and those that affect the rate at which the food is added. Thus, "allocate money," and "formulate plans" become modifiers of "acquire." The activities, "dispose" and "consume" flow directly out of the stock.

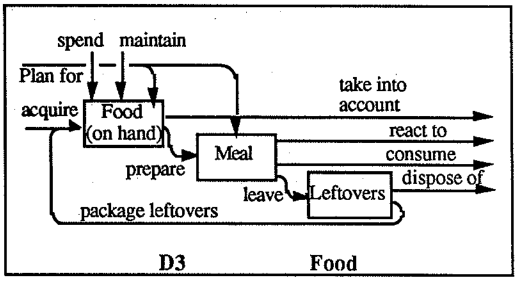


Provided that the "Acquire" activity equaled the sum of the "Dispose of" and the "Consume" activities, the model would be in steady state with the supply of food being constant. If "Consume" is increased, and then held constant, the stock of food would stabilize at a lower level, as expected. Thus, this simulation model does not include many things that would make the model "interesting."



The FOOD (for the family) box was decomposed two levels as shown in the following data diagrams. The highlighted lines, the inputs and outputs of D3, are the inputs and outputs to the expanded diagram.

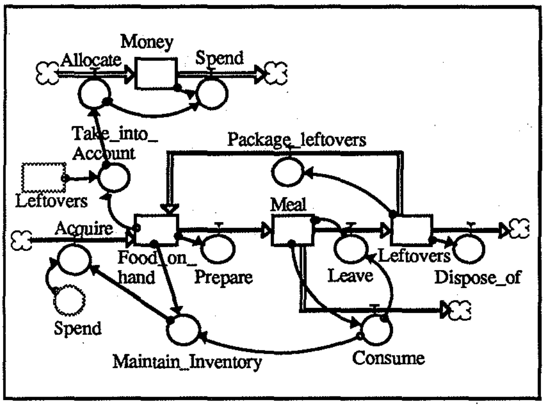


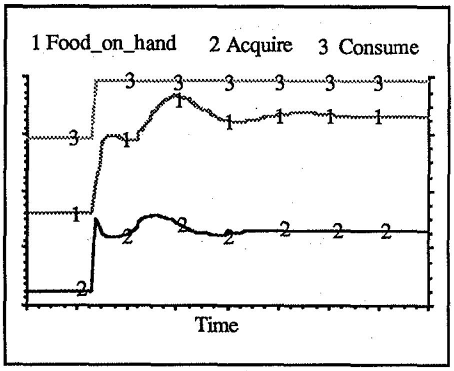


Quantify the D3 variables, and adding part of its D0 context, leads to the simulation model and behavior shown in the diagrams on the next page. When there was a step increase in the amount of food consumed, Food_on_Hand recovered to a higher level after a delayed overshoot.

Conclusions

This paper shows that it is possible to relate SADT models and System Dynamics models. A hierarchical SADT model is much more easily understood than a large simulation model, since it can be examined in separate pieces. As demonstrated in the first section, the predefined interfaces also make the connection of the whole model more evident. Therefore, hierarchical SADT data models can be created and then converted into System Dynamics simulation models.





We have developed procedures for developing System Dynamics models from both SADT data and functional models. Functional SADT models are more common, but require more user input to make the translation.

The translation of a data model is nearly automatic, because the model boundaries are the same. Numeric information and some understanding of the dynamics of the system provide the necessary information to complete the model. The dynamic behavior of the data can then be studied and analyzed.

It is very significant that the translation from SADT to System Dynamics is possible. Simulation enhances the value of SADT system models, giving insight into dynamic behavior. SADT enhances a System Dynamics model, making it easier to understand by the non-modeler, and allowing a hierarchical analysis of the modeled system. The semi-automatic translation from SADT to System Dynamics allows the creation of a simulation model more quickly.

(1) Ross, Douglas T., "Applications and Extensions of SADT", IEEE Computer, April, 1985.

(2) Richmond, Barry, Peterson, Steve, and Peter Vescuso, An Academic User's Guide to *Stella*™, High Performance Systems, Inc. Lyme, NH

(3) Hommel, M. B. & C. McGowan, "Creating System Dynamics Models Using SADT", Proceedings of the 1989 Eastern Multiconference, March, 1989.

(4) Marca, David and Clement McGowan, SADT Structured Analysis and Design Technique, McGraw-Hill, Inc., 1988.