Version of 18 Aug 2011

Overview of System Dynamics Modeling

Written for AEM/IARD 6180

Richard G. Dudley

General

The purpose of this course is to help us harness the potential of system dynamics modeling as a tool for enhancing our understanding of interrelationships within complex systems. While system dynamics modeling has long been used as a tool to study and understand such systems, the emphasis in this course will be on its role in enhancing inter-disciplinary work related to understanding, and developing solutions to, problems inherent to food systems, poverty and related spheres, especially in Africa.

In analyzing such problems, we need to have an ability to examine complex relationships within and among different subsystems. We want to do this in a reasonably rigorous scientific manner, but we also want an approach that is transparent and understandable across different sub-disciplines. In fact, we want to use these techniques to enhance discussion across disciplinary barriers to create reasonably unified understanding of complex multifaceted problems. If we have such tools then the likelihood of finding solutions is improved.

While the emphasis, at present, is the application of system dynamics modeling to the study of food systems and poverty in Africa, the course will strive to provide sufficient understanding of system dynamics modeling at a fundamental level to allow its application to a wide variety of problems and disciplines.

The general themes to be covered during the course are discussed below.

Introducing System Dynamics

System dynamics is a method for "looking at" systems, but here "looking at" means describing, understanding, analyzing, with a view toward redesigning policies that affect a system.

However, successful system dynamics models typically ***focus on a problem rather than a system***. So, more accurately, we might say that SD is a good tool for examining, understanding, and finding solutions to complex problems, especially if these problems involve complex interrelationships, feedbacks and delayed effects.

System dynamics modeling emphasizes ***causal relationships***, ***change over time, and feedbacks***. It uses a ***stock and flow*** modeling paradigm, and is ***equation based***, as opposed to computer code based.

Essentially SD models are ***systems of differential equations***, but use numerical integration over time which avoids need for "neat" equations that can be solved analytically. These characteristics permit the development of complex models that are relatively understandable.

Overall SD is a reasonably transparent, fairly standardized, model building approach, compared to other options. There is a well established literature about SD model building, validation, and analysis techniques. Models are built of many simple, understandable relationships. Models can be complex, but each relationship between elements within a model should be understandable… thus enhancing transparency. This makes meaningful discussion, and modification, of a model involving domain experts and stakeholders from several disciplines more likely, but not automatic.

The conceptualization, design and building of an SD model can be thought of as the creation of a ***dynamic hypothesis*** concerning the question: ***Why do things happen the way they do?*** If a good model can be created, a detailed analysis of this causality is possible.

SD models are, or should be, built in such a way that they are fairly easy to modify. If designed well, models are flexible, and never final. Good models, and good modelers, should be ***open to suggestions***. A good modeling approach should strive to accommodate, or at least consider, alternate views… creating less need to defend a model, and a higher likelihood that the model can be used as a collaborative problem solving tool.

Model Conceptualization

Model early and often! A model should be considered a tool for thinking about a problem, a research tool. A good model is less a final problem-solving product, and more a collaborative problem-solving tool. One of the underlying assumptions of the SD paradigm, is that humans are not well equipped to understand complex feedback systems. A good SD model can help overcome that shortcoming. It can enhance our ability to think about complex problems that we wish to solve.

SD models emphasize the importance of feedback loops and delays in systems. In fact, model conceptualization often starts with the development of a ***causal loop diagram*** or a slightly more detailed ***stock and flow diagram***. These ***qualitative*** tools both assist us in thinking about how parts of a system, or components of a problem, fit together. They can also help us in development of a ***quantitative model*** that can lead to better understanding of the ***dynamic interactions*** of the parts of the system under consideration.

Models require information… sometimes quite a bit of information. The model building process guides us as to what information is needed for investigation of the problem of interest. Importantly, the SD philosophy tends to emphasize the idea that even approximate data and information are better than no information. Thus the ***use of soft variables and estimated values*** (when better data is not available) ***is encouraged***. The reasoning for this approach is that omitting important elements from a model would make the implicit assumption that these elements have zero influence on the problem of interest. Inclusion of soft variables and preliminary data can also help refocus research programs toward areas where information is limited and more research is necessary.

Once we abandon the idea that our SD computer models must only use hard numbers and experimental data, the potential sources of useful information expand dramatically. These sources of information, information for thinking about our problem, now include not only 'real' data, but also information contained in written descriptive material, interviews, observations, results from group discussions, and many other sources. This does not mean that we blindly use any information, but only that we do not reject information merely because it isn't written in a log book or research report. In fact, it is quite possible that the most important components of a model result from information that has never been researched or reported.

Basically the series of questions becomes: What is the problem of interest? What information is needed to understand the problem? What information is needed for our model? Is the information available? If not what alternatives are available?

The Importance of "soft" variables cannot be overemphasized. After all, people make decisions based on feelings, beliefs, and their mental models of the world. For example, expectation of payment, belief that chemical fertilizer is bad (or good), level of disappointment, amount of community cohesion, belief that the rains will start next week, can all be valid model components because they really affect what people actually do.

Identifying stocks and flows in a system, and its model, is an important part of model conceptualization because these model components determine model, and system, behavior. In the standard bathtub example, the identification of stocks and flows is fairly obvious: the volume of water in the bathtub is a stock (liters) and the movement of water into or out of the tub are flows (liters per second). But identification of stocks and flows is not always straightforward. In the case of factories or other means of production we will generally model production capacity as a stock which gradually changes as factories are added or shut down. The units for this stock will be units of production per unit time (barrels per month, tons per year, kilos per hour). The

corresponding flows will be in terms of *barrels per month per month* which is the change in production capacity per unit time. Obviously correct assignment of units of measurement is important for creating and understanding a model. It is important to remember that SD models are actually systems of differential equations and the flow elements of the model represent the rate of change of some stock with respect to time (e.g. dX/dt) where X represents the stock.

SD models can be made more understandable to a wider audience than can differential equations, and this relative transparency also frees us to focus more on the problem being modeled. Nevertheless, SD models can easily become very confusing and overly complex. We always need to strive to make models as transparent and as understandable as possible, both to enhance understanding across disciplines and to reserve our energy to focus on solving a problem rather than on modeling details.

In fact, experience has demonstrated the value of making relatively simple, generic, models as opposed to highly detailed, very specific, models. Generic models lead to a more rapid and better understanding of the target problem, and can subsequently be used to develop more detailed model versions if those are necessary. Thus, building a conceptual model of a complex problem is ideally initiated as an exercise in minimalist critical thinking: What are the critical elements of the whole problem? How do they fit together?

SD Model Building Techniques

In actually building SD models, the first step is to know where we are headed: What is the problem of interest? Thus clear problem identification, and a clearly written problem statement are essential starting points. But the very nature of SD modeling implies an element of discovery, making the whole process of model building an iterative one. There are many published versions of this iterative model building process but most involve 1) problem articulation and boundary selection, 2) development of a dynamic hypothesis complete with proto-model diagrams and graphs describing expected behavior, current behavior, and perhaps desired and feared system behavior, 3) model building, 4) model testing, 5) policy testing using the current model version, then 6) review of the problem, the problem statement, the boundaries, 7) reformulation and so on (e.g. see Sterman, 2000 pages 86-87). However, in many ways the development of a good SD model is less like a recipe, where we follow a step by step procedure, and more like the gradual development of a picture, or the solving of a puzzle, where all elements gradually fall into place.

Often the use of ***causal loop diagrams*** (CLDs) is an early step in mapping out the problem and the model of it. The use of CLDs early in the process helps to define and understand feedbacks within the general structure of the system/problem under consideration. On the other hand, some SD professionals prefer to use CLDs only as model summaries, after the model is in a relatively final form. These modelers prefer to start the modeling process by mapping the model as a ***stock and flow diagram***, identifying the stocks early in the process, and filling in the model details fairly quickly thereafter. This approach permits the modeler to (almost) always have some sort of working model available for testing. In either case, a competent modeler needs to have a knowledge of both CLDs and stock and flow diagrams.

During the iterative model building process it is very easy for the modeler to become sidetracked in the details of working out equations for interactions among the various model components. There is always the danger of becoming too focused on these details and losing sight of the overall problem. Thus a good SD modeler will periodically recall the original purpose of the model and, if necessary, will adjust the problem statement to incorporate ideas revealed by the model-building process itself.

Following the initial use of CLDs, or the direct jump into stock-flow diagrams, the detailed aspect of the modeling building process begins. This requires an understanding of the problem domain, the specifics of the

software being used and, importantly, also requires an understanding of the workings of the SD modeling paradigm in general.

For example, because SD models are systems of differential equations, the equations for flows in SD models must always be consistent with the dX/dt concept: the change in something per unit time. Thus the flow of water into, or out of, a tub must be expressed in terms of $\frac{amount\ of\ water}{unit\ time}$, e.g. $\frac{gallons}{hour}$, $\frac{liters}{second}$, or $\frac{barrels}{year}$. Consequently all expressions describing flows in the model must include a time constant, a requirement that is often ignored by novice modelers.

Specifying equations for the relations among model elements appears to take up much of our modeling time, but this process is greatly simplified if we spend sufficient effort ensuring that the structure of the model is clear, and that we have selected the elements of the model carefully.

Nevertheless, even specifying the outflow from a tank of water might be problematic. Take, for example the simple statement that: "After 10 minutes all 100 liters of water will have been drained from this tank." This can mean a number of things. A common interpretation would be: 1) There is a constant flow of 10 liters per minute for ten minutes: $Flow = 10$. But, 2) more reasonably we might expect that the flow is proportional to the amount of water in the tank which would mean, if the tank is mostly empty after ten minutes, that our equation would be something like $Flow = water\ in\ the\ tank * a\ constant\ fraction$. Of course the statement could also mean that all the water is drained from the tank, all at once, at the end of 10 minutes.

Flows are the only way that stocks (also called levels or state variables) in SD models can be changed, so it is important to know how they work. Very often SD models use the proportional type of outflow as a default, and in that case the time constant specifies the ***mean time*** (not the total time) material is held in a stock. Such flows equal a constant fraction of the material present. So in our second example above this turns out to be about $\frac{W\ liters}{2\ minutes}$ or $0.5 * W\ \frac{liters}{minute}$. At any given time the flow is equal to one half of the water in the tank at that instant. Obviously many other types of flows can occur, the constant fraction idea just happens to be a commonly observed phenomenon.

Many types of relationships are used to describe the causal connections between SD model components. These are entered as simple mathematical equations or sometimes as graphical relationships. Although sometimes necessary for turning on and off certain effects, ***use of computer coding such as IF THEN ELSE statements is discouraged***.

Overall, the philosophy of SD modeling encourages modelers to ensure that model elements and their connecting equations reflect the real world as much as possible. This is in keeping with the idea that ***an SD model attempts to describe not only what happens, but why it happens***.

On the other hand SD modeling also relies on known generic patterns of system behavior. These patterns can be helpful both in interpreting observations seen in the real world and in structuring a model to reflect that reality. Some commonly occurring patterns are ***exponential growth***, ***exponential decline***, ***goal seeking behavior***, ***s-shaped growth***, ***overshoot and collapse.*** System patterns can also be a result of commonly observed larger structures composed of a series of stocks. One example is the ***aging chain*** such as employees moving up in a company hierarchy, or animals moving through a series of ages in a herd. Another example is the ***supply chain*** such as agricultural goods moving from harvest facilities, to storage facilities, through various steps of processing, and on to markets and into homes. These patterns have very specific model structures associated with them, are familiar to trained SD modelers, and actually create specific, sometimes confusing, patterns in the real world. Knowledge of these patterns, and the structures creating them, can greatly speed up the model creation process. Importantly, ***system structure causes system behavior***. System dynamics practitioners realize that cycles and similar patterns are often caused by interactions among elements within a system and not by external factors. This knowledge is critical for building good models.

Model Testing and Validation

Often the validity of a model is measured by how accurately it can predict happenings in the real world. While this type of measure can be used for SD models, the philosophy of SD modeling diverges substantially from this approach. Barlas and Carpenter (1990) stress that validity of an SD model is closely tied to the fact that it tries to explain causality – and does not only try to predict outcomes. Because of this each model equation must be, in their words, "defended and justified".

Consequently, true validation of SD models is time consuming and difficult. As with all models of real world complex systems, SD models have many elements creating many sources of variation, and also many possibilities for error. Ideally validation of an SD model involves justifying each model element and equation, and also ensuring that the model outcomes are both "valid" and "useful".

More importantly Forrester (1961 appendix k) pointed out that, even with good models, long term point prediction of complex dynamic systems (using any type of model) is impossible. This is because even small variations in system components can cause large differences in future outcomes. Forrester also pointed out that complex systems have considerable inertia and are relatively slow to respond to policy change. This fact combined with the prediction problem means that trying to change policies to create specific detailed future outcomes is difficult at best. We can't change the system quickly, and we can't predict the effects of change very far into the future! However, with a good model, general patterns of system behavior can be forecast, and more importantly adjustments to the system structure (e.g. different policies) might help to adjust for expected future problems.

This leaves us with the question: What is a "valid" model? We can ensure that each model component is reasonably valid but, if point prediction in complex models is not possible, how can we justify our model outcomes except in a very general way. Barlas (Barlas and Carpenter, 1990; Barlas, 1996) stresses the contrast between empirical truth and model usefulness. Validity is not only related to model structure and outcomes but also to the original model purpose: Does the model address the purpose? Has it created new insights and better understanding? Are the model builders able to make specific recommendations to the client? By the way… who is the client? Why was the model created in the first place? Ultimately the validity of a model is related to its usefulness. *Can* the model be applied in some useful way? *Is it* applied?

In spite of these philosophical questions, there are numerous approaches for validating and testing SD models (e.g. Sterman, 2000, section VI). Typically these consist of tests related to direct inspection of the model to verify that its elements and structure are a reasonable, simplified, view of real things in the real world. Are model units correct? Are model parameters realistic?

Testing then extends to exercising the model, pushing its limits. Under extreme conditions does it still produce reasonable outputs. Also, keeping in mind that real data can be wrong, are model outputs reasonable compared to the real world. Importantly, are ***model output patterns over time realistic?***

At this point the model becomes a more useful tool for investigation. Through sensitivity analysis we learn which model elements create the biggest influence on outcomes. We can investigate which feedback loops are dominant at different times, under different situations. We can investigate why certain outcomes occur. These investigations can help us better understand the problem of interest, and, importantly, can also guide further research.

Of course this is all important for examining proposed policy scenarios. How do policy changes we might propose change specific model outcomes? Or, conversely, if we have particular desired outcomes in mind, how might those outcomes be obtained? The ideal model will help us determine how to accomplish systematic, sustainable, desired change.

Group Model Building Approaches

Models of social systems that represent the real world should include the views of real people who deal with the real problem of interest. Group model building, in the sense of ***building models directly with domain experts and stakeholders*** provides a set of established and tested approaches that can help accomplish this (Vennix, 1996; Andersen, 1997; Andersen *et al.*, 1997; Vennix, 1999; van den Belt, 2004).

Group model building adds considerable extra work to the task of model creation. Most stakeholders are not modelers, and imparting at least a basic understanding of the process is necessary. In tackling controversial problems group participants will have differing views making agreement on even general model structure difficult. For this reason group modeling protocols include approaches for moderating and focusing discussion, and for ensuring that all views are heard. Importantly, the details of the model building process is deliberately somewhat separated from discussion, but discussion outcomes are modeled and then reflected back to the group as modeling progresses. A good model, even a graphical representation of the model (such as a causal loop diagram), can become a tool to guide and focus group discussions.

Group model building provides a number of benefits. Models built in cooperation with groups are more likely to include views of all stakeholders. Ideally, the model, and its outcomes, become a group product, making implementation of recommendations more likely. An additional benefit is that participants tend to gain a bigger picture, systems view, of their problem of interest. Their ability to think in terms of a larger interconnected system is often enhanced.

Additional SD Modeling Tools

There are some additional topics of interest for students intending to go further with SD modeling. There will not be time to cover these topics in the current course, but students may wish to investigate these areas on their own.

It is often desirable to calibrate models to given data sets, and some SD software makes this process reasonably straight forward. During this process, a selection of model constants is adjusted so that the model output more closely fits data provided from the real world. Although this can be attempted manually, automated approaches simplify the process. The difficulties lie in selecting the appropriate model constants for adjustment, in providing the correct weighting of those constants, and in correct interpretation of the results. Models that are calibrated to real world data are usually more acceptable to stakeholders, especially to those who have provided the data set. On the other hand, matches to data do not necessarily mean a model is "correct" because 1) the data could be wrong and, more importantly, 2) the model may still not represent correct causality within the system.

Similarly, techniques for model optimization are also available to help us find the "best" set of "policies" to solve our modeled problem. To use these methods the modeler defines a ***payoff function*** within the model and then uses optimization to maximize or minimize this. However, defining the payoff function can be tricky in complex models, or impossible if there are multiple sets of goals involving multiple stakeholders. Also, the response surface of a payoff function needs to be examined critically to determine if there are multiple peaks (at various parameter settings) and to see the narrowness or breadth of these peaks. That is, are very exact policies required, or are a wide range of policies sufficient to attain the best outcome. A number of other techniques have also been developed to assist in model analysis and improvement and these may be of interest to the advanced modeler.

Other techniques have been developed to simplify and enhance the development of large system dynamics models. Although the SD philosophy tends to emphasize the value of simple models, large models are possible

and sometimes necessary. In these cases standardized modularization techniques can be useful. This is especially true when different teams are assigned to particular aspects of a larger problem.

Lastly, most SD software packages allow the creation of models that can be used as so called ***flight simulators***: models that have a game-like interface from which users run the model. Users have the option of selecting different inputs to discover outcomes from various scenarios. This approach is sometimes useful when training people about the nature of a particular problem.

Possible Problem Areas

As with any course, there are time constraints. How much can be learned in a semester? What has to be left out? Nevertheless, one semester *is* sufficient to become familiar with the system dynamics paradigm, and to become familiar with, and able to use, a basic set of modeling skills. Perhaps the skill most difficult to obtain is that of model conceptualization. There is sufficient time to cover the basic subject matter, but one semester is not enough time to become a fully competent modeler.

System dynamics software is easy to use. On the surface, system dynamics models are easy to create. Learning the software is not the same as learning to build good models any more than learning to use a word processor makes one a novelist. Models are easy to create… but creating good models takes a bit more work.

Modelers can become overly enamored of their newfound abilities. Those who especially enjoy modeling may tend to work on their own to create their special masterpiece. However, these same people may fail to fully consult domain material, experts, and stakeholders. Being a good modeler does not also make one a domain expert. To create good models one must be open to new ideas and alternate views. This is especially important when working collaboratively.

All problems may begin to look like system dynamics models. While many complex, real world problems are good candidates for system dynamics modeling, some are not. We need to keep in mind other modeling paradigms. We should maintain an open view to other useful modeling paradigms, (e.g. agent based modeling, spatial modeling) that may be appropriate in some circumstances. However, we also have to be aware of the tendency for spurts of interest in newer approaches because they are new. Each approach has it best use, and often approaches can be used in a complementary manner. In this course we attempt to develop the use of system dynamics as a unifying cross disciplinary tool, but it is not the only tool.

Extra: Challenges for "Research" Using System Dynamics Modeling

SD modeling is a useful tool for guiding research because it can help to map out a problem and can help select specific area of interest. Can a SD modeling be the actual focus of a research project? SD models may not fit into the traditional testable null hypothesis formulation often desired by scientific research approaches. But other fields, including some within science, don't use that paradigm. Also, while SD modeling may not fit the ideal H_0: vs H_1 "research" scenario, an SD model can be viewed as working hypothesis – a starting point for further investigation.

Can SD modeling be a research end in itself? In a larger sense an SD model is similar to a hypothesis of why a problem exists, and formal SD models can be tested. So, perhaps we should be able to work with an SD model as a testable dynamic hypothesis. Typically, this isn't really our goal, and the model building process is iterative thus confusing the process and the outcome. The research formulation phase (building the model, i.e. the dynamic hypothesis) is long, and always open to adjustment. Creating a model is perhaps more like theory building. A theory of why a problem exists, with suggested approaches toward a solution.

Several fields (e.g. economics, fisheries) have well established quantitative approaches. This creates limited space for new paradigms unless these are applied to problems that cannot be solved conventionally, merge smoothly with existing approaches, are clearly superior to existing approaches, or provide complementary benefits. At present our hope is that SD modeling provides the latter: it provides a tool for merging interests across interdisciplinary boundaries.

What about intractable problems? For many real world problems there are no easy solutions, no technological fixes. SD modeling will not suddenly reveal answers -- nor will any other approach. What can we do with these ***messy / wicked problems.*** These are problems that have no real "solution" -- only a series of sub-optimal solutions. They have many stakeholders with different views and desired outcomes, often conflicting. They include most real world problems. SD is ideal for working on this class of problem.

Literature Cited

Andersen, D.F., 1997. Scripts for Group Model Building. System Dynamics Review 13, 107–129.

Andersen, D.F., Richardson, G.P., Vennix, J.A.M., 1997. Group model building: adding more science to the craft. System Dynamics Review 13, 187-201.

Barlas, Y., 1996. Formal aspects of model validity and validation in system dynamics models. System Dynamics Review 12, 183-210.

Barlas, Y., Carpenter, S., 1990. Philosophical Roots of Model Validation: Two Paradigms. System Dynamics Review 6, 148-166.

Forrester, J.W., 1961. Industrial dynamics. M.I.T. Press, Cambridge, Mass.

Sterman, J.D., 2000. Business dynamics: systems thinking and modeling for a complex world. Irwin/McGraw-Hill, Boston.

van den Belt, M., 2004. Mediated Modeling: A system dynamics approach to environmental consensus building. Island Press, Washington, DC.

Vennix, J.A.M., 1996. Group model building: facilitating team learning using system dynamics. John Wiley and Sons, New York.

Vennix, J.A.M., 1999. Group model-building: tackling messy problems (Jay Wright Forrester Prize Lecture, 1999). System Dynamics Review 15, 379-401.