LoopX: Visualizing and understanding the origins of dynamic model behavior.
By William Schoenberg (University of Bergen, Norway)

Abstract:
It is a fundamental precept of System Dynamics that structure leads to behavior. Clearly relating the two is one of the roadblocks in the widespread use of feedback models as it normally depends on substantial experimentation or the application of specialized analytic techniques that are not easily approachable by most model builders. LoopX is a tool that builds understanding of structure as it determines behavior by rendering and highlighting structure responsible for behavior as the behavior unfolds. The tool builds on the Loops that Matter (Schoenberg 2019) approach to analyzing loop dominance by presenting the outcome of applying that theory in an easy to use, interactive, web based piece of software. This is a significant step forward in the challenges of automatically visualizing model behavior and linking it to generative structures identified in Sterman (2000). LoopX can be used to machine generate high quality causal loop diagrams from model equations at different levels of detail based on the dynamic importance of links and variables as well as animate them based on their importance from a loop dominance perspective. Several examples are provided that demonstrate the comprehensiveness and ease of use of the tool, important attributes supporting its broad uptake.

Introduction
This paper presents a new and highly usable solution to three important challenges identified in the final chapter of Business Dynamics (Sterman 2000): It addresses “Automated identification of dominant loops and feedback structure”, calculating and displaying the evolution of loop dominance as a model simulates; It improves on “Visualization of model behavior”, using animation to coordinate the display of structural dominance evolution with the behavior over time of model variables; It addresses “Linking behavior to generative structure”, using animation of automatically aggregated diagrams that connect the loop dominance analysis with the model structure via connectors and flows that change size and colors over the course of a model simulation.

Simple systems are usually easy to analyze with intuition and trial-and-error, but with larger systems that are characterized by high feedback loop complexity, the risk of incorrect explanation rises (Oliva, 2016). It is this threat of failure which makes these three challenges posed by Sterman (2000) so relevant. I argue that currently the domain of objective feedback loop dominance analysis is limited to a relatively select few practitioners with a high degree of expertise and training. The lack of tools for parsing and developing insight in large causal models often acts as the limit on the utility of large models to general audiences (Schoenenberger et. al, 2017). The incidence of these problems with presenting models with the intent to develop understanding is not a new occurrence, a cursory literature returns a 1976 paper (republished in 1986) which refers to problems in methods for simplifying the presentation of model structure via casual loop diagrams developed even earlier than that (Richardson, 1986). Cleary, any solution to Sterman’s three challenges must help to reduce the barriers to entry for model understanding and analysis, expanding our depth of understanding.
of the models which are at the heart of our field via improved communication of complexity and its origins.

The foundation of this work is the Loops that Matter (LTM) technique for determining loop dominance (Schoenberg et. al, 2019). Building on the LTM method, the solution to these three challenges employs the use of Causal Loop Diagrams (CLDs) as well as Stock and Flow Diagrams (SFDs) as a vehicle for representing system structure to model consumers.

This paper presents LoopX which is a tool that is capable of reading in and analyzing an XMILE model. The tool allows the model to be simulated, and also analyzed by LTM generating a full complement of link and loop scores describing the origins of model behavior from a loop dominance perspective. LoopX is capable of rendering the model as a stock and flow diagram based on the layout decisions made by the model author. LoopX also machine generates high quality CLDs from the network of model interconnections at user specified levels of complexity presenting a minimum number of variables and links which are deemed necessary (by the LTM analysis) to understand the dynamics of the shifting loop dominance at the requested cognitive complexity level. All diagrams, machine or human generated, are animated portraying dominance information via flows and connectors which change colors and size in real time as the model simulates. All loops are identifiable directly within the context of all of the aforementioned diagrams.

**Problem Statements**

LoopX required the development of solutions to the following three main problems:

1. How can high quality CLDs be machine generated from the network of model interconnections?
2. How can models be aggregated and simplified without losing information important to model understanding while retaining relative simplicity?
3. How can the results of an LTM loop dominance analysis be easily visualized and communicated?

**Literature Review**

This review combines literature from the graph theory and system dynamics fields to provide the reader with the requisite knowledge for understanding the current state of the art as it applies to each of the three problem statements. This helps to place the development of LoopX into context among the existing technologies.

**Techniques for machine generation of network graphs, the basis on which CLDs are formed**

Most important to the automated generation of high quality CLDs is the force directed layout technique. A force directed layout algorithm solves the problem of the placement of nodes in 2D space, such that symmetry is generated, and edge length is approximately equal, by running a physics simulation of weights connected by springs and minimizing the total energy of the system. The first force directed layout technique used steel rings to represent each node and then connected those rings using logarithmic springs (Eades, 1984). In this version of the
algorithm, attractive forces were only calculated between neighbors, and repulsive forces were calculated between all node pairs (Eades, 1984). This process ensured that neighbors were always close by but limited the scope of the N-squared problem.

The next evolution in the force directed approach was to introduce the concept of an ideal distance between every node pair based on the shortest path between each node pair, and to use Hooke’s Law, meaning real world realistic linear springs (Kamada and Kawai, 1989). The Kamada Kawai approach solved partial differential equations based on Hooke’s Law to optimize layout applying all forces between all node pairs in an iterative fashion (Kamada and Kawai, 1989). A gradient descent optimization process used to terminate the simulation when a global minimum in the energy state of all the springs was found (Kamada and Kawai, 1989).

Development of Graphviz, an open source toolkit for solving these graph generation problems took place in parallel to these developments at Bell Labs. Graphviz contains many different automated layout mechanisms, but the mechanism most relevant to CLD generation is called neato, which is based on the work of Eades, Kamada and Kawai among others. The layout algorithm used in neato that we are interested in, is derived from the Kamada Kawai algorithm. It assumes there is a linear spring between every pair of nodes, each with an ideal length (Gansner, 2014). The ideal distance between each node pair is the result of a function computed for each pair; the ideal length function we are interested in uses the shortest path between the two nodes to determine the ideal distance between these nodes, although many other choices are offered. Neato is able to turn a static text file with a description of the graph quickly into a 2D diagram quickly (North, 2004).

Neato, like all force directed graph algorithms, produces, by default, edges that are straight. There are disadvantages though for using straight edges especially as it relates to user understanding of the generated graph diagrams (Xu et al. 2012). Graphs of the type produced by force directed algorithms with curved edges are generally called near-Lombardi or Lombardi-style diagrams, and I would class CLDs as a form of Lombardi-style diagram. In Figure 1, I present examples of straight edge force directed graphs and their Lombardi style complements. When using a curved edge, Lombardi-style diagram, there are significant user performance improvements on graph related tasks such as determining the shortest path, node degree calculations and common neighbor determinations (Xu et. al., 2012). The problem with algorithmic Lombardi-style diagram generation and force directed graphs in general, is the lack of the concept of directed edges and especially directed cycles (links and loops as SD conceives of them).
Techniques for model aggregation and simplification
As covered in the introduction, the problems of simplifying dynamic complexity for wider consumption has been studied since the formative years of the field. Early research discusses how CLDs alone do not give an accurate enough picture of model structure so that behavior modes can be predicted and understood (Richardson, 1986). Richardson (1986) argues for caution when using CLDs to aggregate and simplify model diagrams, and that information is often lost in that process.

The most famous examples of model aggregation techniques are the independent loop set and its refinement, the shortest independent loop set (ILS and SILS) respectively (Kampman, 2012) and (Oliva, 2004). These graph theory techniques for the partitioning of the cycles (feedback loops), implicit in the network of model structure, arose due to the complexity faced when performing and analyzing the results of the eigenvalue elasticity method of loop dominance analysis (EEA) or when trying to find high leverage points for policy intervention. In anything but small models both authors were faced with a relatively large list (compared to the number of variables in the model) of feedback loops which were all tightly interrelated. The SILS concept pairs down the number of feedback loops to the set of geodetic (shortest) loops which are necessary to fully describe the feedback loop complexity of the model. This reduces the

Lombardi-style diagrams tend to produce circular shapes that are not loops. Therefore, I needed to implement a better edge curving algorithm in neato to create an ideal tool for CLD generation (directed cyclic graph generation) based on feedback loops.
number of loops present in a fully accurate CLD of the entire network of model structure, focusing user attention on the loops which are most easily influenced by policy.

Built on the ILS and SILS, (Schoenenberger et. al, 2015 and 2017) present the use of variety filters derived from interpretative model partitioning, structural model partitioning and the ADAS method (algorithmic detection of archetypal structures), to communicate intuition from large models. Their audience are those who would normally be overwhelmed by the size and complexity of the models being studied. This work also builds upon earlier studies of model simplification done by Eberlein (1989), which uses linearization, and on Saysel and Barlas’ (2006) aggregation method. The variety filters technique presents the user with structural clusters of model variables based on state of the art statistical and graph theory techniques as a way of visualizing and understanding nearness and hierarchy. With interpretative clustering, model complexity is filtered via studies of the relationships between pairs of model sectors. Using ADAS which is applied to the above generated clusters, users select a stock of interest as well as an archetypal structure to find, and the algorithm returns the feedback loops which contain the variable in the system archetype specified. This significantly reduces the number of feedback loops to be studied by the end user pairing down the complexity of the SILS.

The Forio Model Explorer feature of Forio Simulate is an example of a simplistic Kamada Kawai style force directed rendering of model structure which was later evolved into supporting a secondary hierarchical layout engine with rudimentary aggregation steps taken to either only show two degrees of distance from a variable of interest or all of the links between two variables of interest with a filter based on path length. The Forio Model Explorer was studied and was compared to traditional hand drawn CLDs in an attempt to measure the effectiveness of the automated diagramming and aggregation techniques (Schoenberg, 2009). All tests were inconclusive, showing no reported differences in learning outcomes, but diagrams generated were of significantly less quality, lacked any of the positive attributes of Lombardi-style diagrams and were not focused on feedback loop behavior. I mention this work because it is, to my knowledge, the best previous attempt at using aggregation and force directed graphs to solve the challenges laid out by Sterman (2000).

**Techniques for the communication and determination of automated loop dominance information**

The final area of requisite knowledge is loop dominance analysis and presentation of related results. Much work has been done on various algorithms and techniques to do objective loop dominance analysis. Significantly less effort up until this point has been focused on the presentation of information because the work of solving the problem of objective loop dominance determination has been so hard.

The field of automated loop dominance analysis can be understood as existing in two major sections, with LTM forming a new section of the research.

LTM produces, for each link and loop in the model, a score that represents the strength and polarity of each link in a feedback loop as well as the percentage contribution and polarity of
each feedback loop to the behavior of the stocks in its coupled shortest independent loop set CSILS (the entire SILS for well-connected models) (Schoenberg et. al, 2019). LTM does this by simulating the model and, at each time step, measuring the contribution of each link in the network to its dependent variable by iteratively holding inputs constant. These link scores are then multiplied to determine loop power, which is then normalized across the CSILS (a subset of the SILS) to determine relative strength and interpreted as the percentage contribution (Schoenberg et. al, 2019).

The first major pre-existing area of loop dominance analysis is EEA. Forrester (1982) was the first to document that eigenvalue elasticities could be used to explain the relative contributions of different loops in models of linear systems. Since then, the formal method of eigenvalue elasticity analysis (EEA) has been further developed and is used to determine how model structure produces the dynamic modes of behavior for the model, specifically those characterizing the state variables, (in SD called stocks) (Saleh, 2002), (Kampmann et al., 2006), (Saleh et al., 2010), (Oliva, 2016). Using EEA, the structure of a model is characterized by the eigenvalues and eigenvectors of that model. It may be demonstrated that the dynamic behavior of a linear systems model may be expressed by a linear combination of behavior modes, each characterized by a specific eigenvalue and weighed by a factor that depends on the eigenvector and the initial state of the model (Saleh et. al., 2010). EEA is applied to examine both link and loop significance with regard to the dynamic behavior of the model.

The second major pre-existing area of loop dominance analysis is PPM. The pathway participation metric approach does not use eigenvalues to describe model structure. Rather, it focuses on the links between variables (Mojtabahzadeh et al, 2004). The starting point in the PPM approach is the behavior of a single variable, typically a stock. The behavior of that single variable is partitioned in time, based on phases where the variable maintains slope and convexity across time with the first and second time derivatives not changing sign (Mojtabahzadeh et al, 2004). This then limits the behavior of the variable at each of these phases to 7 patterns enumerated by Mojtabahzadeh et al, (2004). The PPM approach then determines dominance by tracing along the causal pathways between the stock under study and its ancestor stocks to determine which structure is most influential in explaining the pattern of behavior exhibited by that stock during the selected phase. Mojtabahzadeh et al., (2004) explains that it does this by determining the magnitude of the change in the net flow of the stock under study by making minute changes to that stock. The method then compares these changes in the net flow to determine the change with the largest magnitude in the same direction as the stock under study thereby identifying the most important (dominant) pathway governing the behavior of that stock during that phase.

All three methods explained above produce data of similar classes and have, so far, visualized this information in similar ways via overtime (time-series) and sometimes instantaneous plots of loop dominance or link strength or tables of important loops etc. All make efforts to mix the time-series data with behavior via graph overlay, attributing loop dominance information to phases in the behavior over time graph of each specific variable of interest. When measured against Sterman’s challenges, this clearly falls short because it lacks any of the dynamic aspects
necessary to communicate loop dominance information to a wide audience, especially the general public.

Results

1. How can high quality CLDs be machine generated from the network of model interconnections?

As indicated in the literature review, CLDs are machine generated using neato. But the problems of generating Lombardi-style diagrams that do not emphasize non-loops needed to be solved. The solution is depicted in Figures 2 and 3 which were generated in the LoopX tool.

![Diagram of Bass Diffusion model](image)

*Figure 2: Autogenerated Full CLD of Bass Diffusion model. Red links are negative, green links are positive.*

To solve the edge curving problem, I developed a new algorithm which has been accepted into the public version of neato to curve edges (merge:1305). This algorithm follows a simple heuristic derived from the observation of CLD diagram drawing by hand. The heuristic specifies that the center of the cycle which forms the arc that the edge will follow, must be the average center of the nodes which form the shortest feedback loop with length greater than 2 that the edge is a member of. The two-node exception is handled separately within the neato codebase, and produces paired directed edges that do not over-emphasize the cycle, producing elongated ellipse structures that cover an area relative to the number of nodes. This edge curving heuristic relies upon the attributes of force directed graphs which place nodes that are related closest together. This heuristic produces loops that look circular except for in...
degenerate cases where the force directed layout fails to produce good local clusters and the shortest feedback loops are relatively far flung in the 2D space.

There are two key techniques for minimizing the incidence of degenerate diagrams due to path dependency issues in neato. The first one is to set the initial position of each node to the position of the variable it represents in the stock and flow diagram. This is a fairly accurate way to make sure that local clusters are being preserved. The second technique is for when stock and flow information is not available. In this technique, the order of the nodes in the input (dot) file are dictated by the order of the nodes reached when iterating through the variables in all of the loops that will be rendered. The loops must be in descending order by number of variables. This process ensures that items nearest to each other in the graph space end up initially nearest to each other in the physical space, allowing neato to do a better job of optimizing placement.

Other key inputs to neato include the usage of the Prism algorithm (a proximity graph-based algorithm) to prevent the overlap of nodes and the KK mode which uses a variant of the gradient descent process, originally proposed by Kamada and Kawai, for solving the optimization problem during the node placement phase. Generated CLDs have the highest quality when neato uses the ‘shortpath’ model (default option) for computing its distance matrix, i.e. using the shortest path between two nodes as the ideal length of the spring between them. Finally, to invoke my edge curving algorithm, the option for splines needs to be set to curved, - otherwise a straight-edged diagram will be produced.

2. How can models be aggregated and simplified without losing information important to model understanding while retaining relative simplicity?

Building upon the work of Kampman (1996) and Oliva (2004) with the SILS I had to develop additional techniques to further partition the loop set of models. Work on this was started with the development of LTM and the description of the CSILS which can sometimes further partition the SILS into a subset of variables where each stock in the subset affects itself and any other stock in that same subset (Schoenberg et. al, 2019). Figures 2 and 3 depict the CSILS in full without additional partitioning. These diagrams depict the structures responsible for generating behavior across the stocks resident in the CSILS.

Use of the CSILS though, does not provide the necessary granularity adjustments to reshape a subset of the SILS on demand for the user. Based on the LTM loop dominance analysis, I created two new global parameters for a generic model of diagram generation. The parameters are used to filter the CSILS. The parameters are the Link Threshold and the Loop Threshold which are used to create “Simplified CLDs” or aggregated CLDs.

The link threshold is used, mainly to filter the number of auxiliary variables that appear in the rendered graph. The link threshold has a range of [0-1], representing the largest change in the strength portion of the normalized (by dependent variable) link score from LTM. Only variables which are pointed to by links with a change in normalized link strength greater than or equal to
this parameter, are included in the rendered graph. This allows the user to specify from a loop dominance perspective which (mainly auxiliary) variables to remove from the rendered graph since typically flow-to-stock links have high threshold scores (aka large changes in normalized link strength as the model approaches an equilibrium states). Links with a high variability in their normalized link strength are those which change their power the most over the course of the simulation run. They therefore, point to the sources of non-linearity in models and are, typically, the variables most important for understanding model behavior. Links which do not change at all over the course of the simulation run (regardless of explanatory power over their dependent variable) have a link threshold value of 0 and point to variables that are likely the least important and are good candidates for elimination. These variables tend to exist for the modeler to simplify equations.

The loop threshold is mainly used to filter the number of stocks and associated flow variables that appear in the rendered graph. The loop threshold has a range of [0-1], representing the strength portion of the loop score from LTM. Only loops that have a loop score strength greater than or equal to this parameter have their stocks and flows automatically included in the rendered graph regardless of variability in normalized link strength. This allows the user to specify from a loop dominance perspective which stock and flows (as well as their associated direct feedback loops) do not need to be in the rendered graph, facilitating the creation of a smaller graph. The loop strength is measured in terms of the cumulative contribution of the loop to the behavior of the stocks in the CSILS over the entire course of the simulation run. This number describes the percentage of behavior of the stocks being graphed for which the loop is responsible. Loops with a low threshold of, for instance, 0.01 would only describe 1% of the total behavior of the model over the simulated time horizon and probably do not need to be included in a simplified diagram in order to understand the origins of the generated model behavior from a structural perspective.

A default loop score threshold value of 10% and 20% for the link score threshold tends to produce the simplest diagrams that explain the large majority of model behavior while producing the smallest diagrams in terms of cognitive complexity for interpretation. For the market growth model, example diagrams of varying levels of simplicity are presented in Figures 4 and 5 demonstrating how the loop and link thresholds may be applied to simplify model structure.
Figure 3: Autogenerated full CLD of Forrester’s 1968 market growth model (link threshold 0%, loop threshold 0%). Red links are negative, green links are positive.
The link and loop thresholds are used to generate a filtered list of nodes from the CSILS. From that, a corresponding list of edges needs to be generated, - one that matches the reality of the true feedback loop connections, but does not show all of the individual steps along the way. In
other words, the filtering process solves the aggregation problem from a variable perspective and now the links need to be generated such that they make sense at both this new level of aggregation and yet still represent the connections of a totally disaggregated model (AKA a one to many connection from aggregated level to disaggregated level). This technique of loop and link filtering is a new method in the succession of methods for automated model aggregation and can be used independently of CLD drawing.

The process used to generate the links, is termed the skip link process which performs a depth first search (DFS) for each possible link in the new limited directed adjacency matrix, generated by the filtering. The DFS traverses the disaggregated model, testing if we can go backwards from the destination of the candidate aggregated link to the source of that candidate aggregated link without passing through a variable already in the limited set specified by the parameter filters (or a variable already visited in this search). If it passes this test, then we know we have found an aggregated link which is valid and has the property of being accurate to the disaggregated level while representing one step at the aggregated level. We can therefore include this link in our rendered graph and properly preserve the information from the disaggregated model at the new aggregated representation of the model.

3. How can the results of an LTM analysis be easily visualized and communicated?

The final puzzle to solve is the most creative, and least objective. I needed to invent techniques and processes for animating loop dominance behavior across any of the rendered graphs of model structure (SFD, Full CLD, Simplified CLD).

I used colors to represent the polarity of any link and thickness to represent the link strength in any rendered diagram. Users can change the diagrammatic representation of the model at any point in time via a simple dropdown box. Users are able to obtain plots of behavior overtime for any variable in any diagram, as well as link scores and loop scores for any rendered link or loop by just clicking on the variable or connector or loop identifier in any diagram. All generated data is also offered for download in CSV form for external analysis and plotting.

On all diagrams, a table of loop scores is plotted showing the instantaneous contribution of each loop to the behavior of the stocks shown as well as a cumulative total contribution over the entire model run. This table is sorted by cumulative contribution to make the most important loops rise to the top.

The loop identifier for any loop may be pressed, and, while held, highlights all variables and links in any of the diagrams that the loop represents, - removing all dominance information while doing so. This allows users to quickly identify what the meaning of all of the identified loops are and track them through any rendering of model structure as their placement does change since each diagram generation is a totally independent process as of the current writing of this paper.
An animation timeline is provided so that users may scrub through the visualization of loop dominance and pin it at any point in time to examine the state of the model (structurally or behaviorally) at that specific $dt$. Users are also free to adjust the Link and Loop Thresholds at any point during the simulation as results are animating, or while they’re scrubbing through results to explore the various levels of complexity in the explanations of model behavior. Figures 6 and 7 depict this animation in both stock and flow and CLD diagrams of the bass diffusion model.

One interesting difficulty in visualizing link strength and polarity in SFDs, is the case of flows. Flows are often times connected to two stocks and therefore have an in-polarity and strength and an out-polarity and strength. In these cases (bass diffusion model), it would be nonsensical to render the whole flow with a single polarity and strength because it is nearly always guaranteed that the flow has opposite polarities and often times it has different strengths. To solve this problem, I split the flow in half, rendering the pipe sections before and after the flow valve with the polarity and strength associated with the connecting stock. This produces flows which make it very clear to users of the SFD of the hidden information links they contain. This is especially true re. the outflow from stocks where there is no arrowhead from the flow to the stock. This can be observed in Figure 6.

![Figure 6: Screenshot of LoopX showing a stock and flow diagram of the Bass Diffusion model at the final time period. Notice the coloring of the split flow ‘adopting’. Red links are negative, green links are positive.](image-url)
Figure 7: Screenshot of LoopX showing a simplified CLD of the Bass Diffusion model at the final time period. Notice how Potential Adopters is driving potentials contacts with adopters signifying the strength of $B_1$. Red links are negative, green links are positive.

Discussion

CLD generation with neato and the improved edge curving algorithm is successful because the generated CLDs appear to be naturally drawn, yet they tend to follow best practices as laid out by Richardson (1986). Also, the generated CLDs tend to minimize the instances of non-feedback looking like feedback while keeping short loops close in 2D space. This enables users of these diagrams to enjoy not only all of the benefits of curved edged diagrams as measured by Xu et al, 2012, but in addition, feedback becomes much easier to identify at a glance.

The techniques used to produce machine generated CLDs described in this paper can be applied independently from LTM and any other techniques discussed in this paper to generate high quality CLDs from network data. It is my hope that the authors I mentioned in the system dynamics portion of the literature review, use these techniques to better visualize and deliver the structure they determine as important to users, and that those in the graph theory and visualization field use the edge curving techniques presented to improve their diagrams.

The skip link algorithm for model aggregation and simplification works well because it maintains consistency of information regardless of aggregation level. Because each aggregated link is composed of a specific and known list of disaggregated links, animation and visualization of the link score is not affected, because the link score is designed to be multiplied. Because the aggregated diagrams are fully accurate representations of the relationships between the variables selected, we can be confident that information loss is minimal and is primarily controlled by the choice of value of the loop threshold parameter. The loop threshold
parameter will allow for information loss when it is set such that the stocks of specified unimportant feedback loops are removed from the diagram. This means that the representation of those feedback loops in the aggregated diagram can be lost if those stocks are not also resident in feedback loops of more cumulative importance. The link parameter threshold rarely leads to the loss of feedback loops in the aggregated diagrams generated because of the tendency of systems to generate large fluctuations in link score when approaching and leaving those equilibrium states. Therefore, the link threshold exceedingly rarely tends to be the source of removal for stocks and their associated feedback loops from the aggregated diagram. The power of the link threshold is to very quickly identify any relationships in the model that serve to expand the number of variables for reasons of equation simplification, - as opposed to for reasons of dynamic complexity. It acts as a surgical scalpel for cutting away all of the variables in the model that do not serve as the interface between feedback loops, - allowing users to be presented with diagrams that contain a minimum number of variables representing a maximal amount of dynamic complexity.

The skip link algorithm has many uses beside this research. One obvious area of utility for this technique is in the ADAS algorithm referenced by Schoenenberger et. al, (2017) to reduce model structure complexity before attempting to pattern match system archetypes. The ADAS algorithm ought to do a better job of finding system archetypes in their reduced forms. I believe this is possible because it will not need to consider so many different possible mutations of structure. Also if the network searched, was limited by the loop and link threshold parameters then the results would also be limited to the structures which are provably most relevant to the CSILS under study. This would ensure that the algorithm presents the most comprehensive list of matches that are the most relevant to system behavior.

Choice of the link score normalized at each time step for each set of independent-to-dependent variables requires further explanation. A normalized value is required for the animation of connectors and flows because a maximum thickness needs to be set. Otherwise thickness would have no bar to measure it against. Without normalization, using the link score would create links whose thickness explodes towards infinity just as the link score does when models pass through or reach equilibrium states.

Another potential other choice for the source of data for the rendered link thickness could be an approach which would apply loop scores, sizing all of the links in a loop equally to better emphasize loop dominance. The problem with animating the loop score arises often in models of any significance where important loops are derivatives of each other, sharing many links in common. The trouble in these cases boils down to how to represent and display the information that a link is resident in multiple loops, - each with their own level of significance. Theorized techniques include drawing multiple links, one for each loop, or flashing through representations over time (per each time step) in proportion to loop strength. I believe, however, that all techniques we may choose to use to represent loop score over links resident in multiple loops, will not scale with model complexity. This problem is, moreover, compounded by the aggregation techniques presented. Because aggregate links, by their very nature, tend to be resident simultaneously in even more loops.
Other possible options for animating link thickness includes normalization of the link score to all link score values in the entire model at that specific time step, or normalization of the link score to all link score values in the entire model across all time steps. The potential benefit of normalizing link score across all links at all times, is to offer an impression of the power level of the model as a whole overtime. This would make very clear when the model is reaching equilibrium, - as links would then tend to get thicker during these periods. The problem, though, is that link scores even when plotted on a logarithmic scale appear to be exponential in shape during equilibrium events. This means that dynamics would be all but impossible to observe at any time except for during equilibrium events because connectors and flows would constantly remain tiny. Normalization across all links at a specific time period suffers from the same general issue, except its applied not to the diagram over time, but to parts of the diagram that reach equilibrium marginally slower or faster than other parts. That would give a very misleading perception of the model, exhibiting drastic shifts in dominance from one time step to the next, which is not borne out by the data.

The flaws in using normalized link strength as the source of the animation of connector thickness is that loops are not made any more easily identifiable and that the loop power is completely unobservable because values are normalized at each time step. Ultimately, though, these downsides are mitigated via the loop legend that allows for quick and easy access to loop scores and, in the future, to loop power information over time, as well as for a quick and easy way to identify any loop of interest.

**Conclusions**

In the final chapter of Business Dynamics (2000), Sterman issued many challenges for the future of system dynamics, three of which have been answered by the creation of LoopX. The first challenge; “Automated identification of dominant loops and feedback structure”, has been answered previously by other techniques including EEA, PPM. But, for this first time, one of these automated loop dominance analysis techniques has been automated and packaged in such a way that the outcomes are easily accessible to a wide swath of practitioners in the field. The second and third challenges; “Visualization of model behavior” and “Linking behavior to generative structure”, also have a long past set of accomplishments. I believe LoopX represents a major success because it integrates loop dominance analysis techniques with model aggregation and visualization. LoopX produces high quality, easy-to-decipher, animated SFDs and high-quality machine generated animated CLDs of the origins of model behavior via the integration of the results of an automated loop dominance analysis done by LTM.

At the current date, LoopX represents only a start to what ultimately may be possible. Efforts must be undertaken to measure the effectiveness of these techniques for teaching purposes, practitioner purposes and, potentially, after future revisions for use by the general public, before any definitive statements can be made about achieving Sterman’s goals. Problems still need to be addressed include the scalability across giant models of a size such as T-21 or its brethren, which must include a significant re-engineering effort focused on deriving efficient
solutions to the skip link, and CSILS detection algorithms. The ultimate viability of these techniques will be proven via their adoption in mainstream tooling.

References:
1. Chernobelskiy, R., Cunningham, K. I., Goodrich, M. T., Kobourov, S. G., & Trott, L. (2011, September). Force-directed Lombardi-style graph drawing. In International Symposium on Graph Drawing (pp. 320-331). Springer, Berlin, Heidelberg.
2. Eades, P. (1984). A heuristic for graph drawing. Congressus numerantium, 42, 149-160.
3. Eberlein, R. L. (1989). Simplification and understanding of models. System Dynamics Review, 5(1), 51-68.
4. Forrester N. 1982. A Dynamic Synthesis of Basic Macroeconomic Theory: Implications for Stabilization Policy Analysis. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.
5. Gansner, E. R. (2014). Using Graphviz as a Library (cgraph version). published online August 21.
6. Kamada, T., & Kawai, S. (1989). An algorithm for drawing general undirected graphs. Information processing letters, 31(1), 7-15.
7. Kampmann CE, Oliva R. 2006: Loop eigenvalue elasticity analysis: Three case studies. System Dynamics Review. 22(2): 141-162.
8. Kampmann CE. 2012. Feedback loop gains and system behaviour (1996). System Dynamics Review 28(4): 370–395.
9. Mojtabahzadeh M, Andersen D, and Richardson GP. 2004. Using Digest to implement the pathway participation method for detecting influential system structure. System Dynamics Review 20(1): 1-20.
10. North, S. C. (2004). NEATO user's guide. Murray Hill, NJ: AT&T Bell Laboratories.
11. Oliva, R. (2004). Model structure analysis through graph theory: partition heuristics and feedback structure decomposition. System Dynamics Review, 20(4): 313-336.
12. Oliva R. 2016. Structural dominance analysis of large and stochastic models. System Dynamics Review 32(1): 26-51
13. Richardson, G. P. (1986). Problems with causal-loop diagrams. System dynamics review, 2(2), 158-170.
14. Saleh MM. 2002. The Characterization of Model Behavior and its Causal Foundation. PhD Thesis, University of Bergen, Bergen, Norway.
15. Saleh M, Oliva R, Kampmann CE, Davidsen P. 2010. A comprehensive analytical approach for policy analysis of system dynamics models. European Journal of Operational Research 203(3): 673-683.
16. Saysel, A. K., & Barlas, Y. (2006). Model simplification and validation with indirect structure validity tests. System Dynamics Review, 22(3), 241-262.
17. Schoenberg, W. (2009). The Effectiveness of Force Directed Graphs vs. Causal Loop Diagrams: An experimental study. In The 27th International Conference of the System Dynamics Society.
18. Schoenberg, William, Davidsen, Pål, Eberlein, Robert, (2019). Understanding model behavior using loops that matter. arXiv preprint arXiv: submit/2818896.
20. Schoenenberger, Lukas, Alexander Schmid, and Markus Schwaninger. "Towards the algorithmic detection of archetypal structures in system dynamics." *System Dynamics Review* 31.1-2 (2015): 66-85.

21. Schoenenberger, L., Schmid, A., Ansah, J., & Schwaninger, M. (2017). The challenge of model complexity: improving the interpretation of large causal models through variety filters. *System Dynamics Review*, 33(2), 112-137.

22. Sterman JD. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin/McGraw-Hill, Boston.

23. Xu, K., Rooney, C., Passmore, P., Ham, D. H., & Nguyen, P. H. (2012). A user study on curved edges in graph visualization. *IEEE Transactions on Visualization and Computer Graphics, 18*(12), 2449-2456.