Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

FAST AND EFFECTIVE LIVING BUSINESS MODELS WITH SYSTEM DYNAMICS: A TUTORIAL ON BUSINESS CASES

Kim Warren

Strategy Dynamics Ltd
Princes Risborough, HP27 0JS, U.K.

ABSTRACT

This tutorial is for analysts, consultants, and teachers of business modeling. No prior experience is needed. Simulating business challenges and plans has until recently been difficult and time-consuming, but things have changed! We will take you through an "agile" process that makes quantified, working simulations practical for non-experts to build, quickly and reliably. And, we need these tools! Spreadsheet-based methods just cannot handle the interdependencies, feedback, thresholds, and intangible factors that pervade all but the simplest cases. The resulting Living Business Models display with total transparency the factors that management recognizes, and the causality driving the performance outcomes they are interested in. Since everyone sees the same rigorous picture, they get a "joined-up" view that fully explains how everything has been changing. This allows them to explore likely future outcomes under alternative assumptions, decisions, and strategies.

1 INTRODUCTION

Simulation modeling has contributed greatly to many fields, both for research and for real-world planning and problem-solving. Its use for operational purposes in business is also widespread. However, simulation of organizational strategy and management challenges has progressed rather slowly. The tutorial to which this paper refers will explore why this has been the case for one of these methods – system dynamics (SD) – and demonstrate a process that might break through these limitations. It will demonstrate that SD can be deployed faster, more easily, and more reliably than common spreadsheet-based alternatives, and is well within the capabilities of non-specialists.

SD is one of three simulation methods most appropriate for modeling organizational performance issues, the others being agent-based modeling (ABM) and discrete-event simulation (DES) – see Maidstone (2012) for a short comparison of the methods. DES has a substantial heritage and is very widely used for modeling discrete events concerning individual entities moving through a process, such as queuing cases and production or supply chain systems. ABM is ideal for modeling the actions and interactions of autonomous agents, in order to assess their effects on a wider system. It is especially powerful for cases where geo-spatial phenomena are important.

SD, in contrast, is a continuous simulation method that addresses changes to, and the interactions between, the quantities of related *populations* of people, things, or materials. These populations are modeled as “asset stocks” that accumulate and deplete over time. SD lacks the entity-specific benefits of ABM or DES methods, but has the countervailing advantage that its models can be quick to build and compact, while capturing interactions between several diverse classes of asset stock. This makes SD ideal for strategic planning and management of longer-term challenges concerning different parts of an organization or issue.

There are also benefits from combining these methods in hybrid models (Mustafee et al. 2015). This tutorial will focus on the process for developing and using SD models, although that process may also have useful implications for ABM and DES modelers.

978-1-5386-6572-5/18/$31.00 ©2018 IEEE

2 A COMMON SYSTEM DYNAMICS MODELING PROCEDURE

Concern has grown over recent years about the slow adoption of SD among those with authority over significant policies in many domains – environment, health, economics, business, security, and so on (Forrester 2007; Homer 2013) – and the resulting lack of beneficial real-world impact. Other, non-simulation methods have overcome this challenge and gained rapid, widespread adoption, such as balanced score-card, value-based management, systems engineering, 6 Sigma, and business process engineering.

One might reasonably expect a method to be adopted if it meets three criteria – delivering clear and demonstrable benefits, with reasonable effort and cost, and doing so reliably, where "reliable" implies that the method can be deployed with confidence in similar cases. The low recognition, slow adoption, and limited impact of SD thus raise doubts about the method's performance on all three of these criteria. Mature professional disciplines typically feature some best-practice by which their methods are implemented – standard procedures to ensure reliable delivery of valuable benefits, at reasonable cost. Professional societies publish such procedures, for example the APICS Supply Chain Reference Model (SCOR), the Balanced Scorecard Institute's Nine Steps to Success, and INCOSE's Systems Engineering Handbook.

Although no similarly procedure has been specified for the SD field, some approaches have been codified. A group model-building process (Vennix 1996; 1999) aimed to make SD more accessible, and Coyle (1996) offered a practical approach based on similar steps. Guidelines in other sources are generally similar to that provided in these books (Lane 1994; Sterman 2000; Maani and Cavana 2000; Morecroft 2007; Pruyt 2013; Wolstenholme 1994). Published SD articles suggest that these guidelines are widely followed, although professional practitioners often adapt previously proven structures, rather than starting models from scratch.

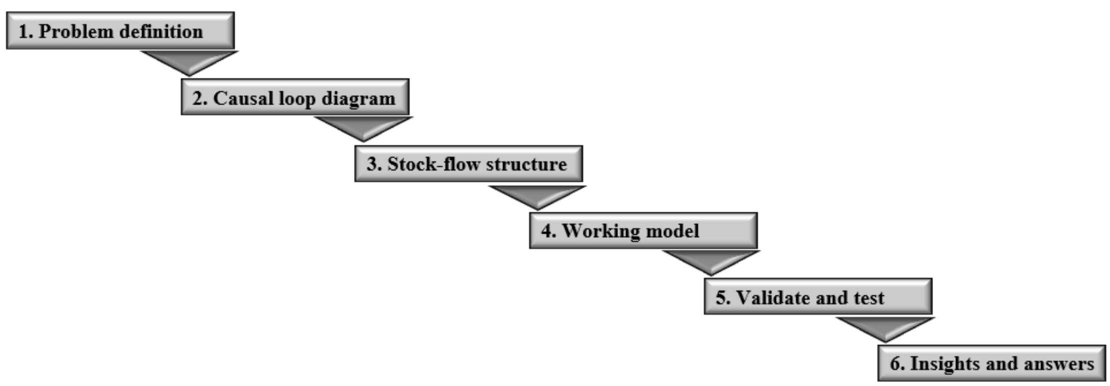

Figure 1: A common process for developing system dynamics models.

Step 1: The common method (summarized in Figure 1) first identifies the scope of the problem and defines the outcome of concern – plotting a time-chart of how that outcome has changed up to the present time and how it might change in future. This is a quantified task, both for the outcome indicator itself and the time-scale over which change occurs and leads to the "reference mode" time-chart against which the resulting model is assessed, scenarios are explored, and policy options are tested. There may be more than one such chart – the reduction in some harmful outcome *and* the cost of doing so, for example.

Step 2: People involved in the situation are then asked for their views of what other factors may be causing that outcome, and for the causal relationships they believe exist between those factors. This step focuses on identifying feedback loops in the causal structure. Some collective process is used to negotiate and combine these views into a single causal-loop diagram (CLD), taken to capture the stakeholders' so-called "shared mental model" of the system structure driving the outcome of concern.

Step 3: Any accumulating stocks in the diagram are identified, and a modified diagram produced. An accumulating stock is a collection, group, population, mass, or volume of things or material that accumulates or depletes over time. Common categories include people or other populations, cash, physical assets, and so on. In some cases, steps 2 and 3 may be combined, with facilitators attempting to develop causal diagrams that include stocks and flows from the start. This brings some benefits, but also some costs (Lane 2008; Fisher 2010).

Step 4: The feedback diagram with stocks and flows provides the basis for actual modeling, where data are gathered and arithmetical relationships specified to produce a working, quantified model. Although simply stated, this step is a large and technically demanding part of the model development process.

Step 5: The model is then validated and tested in various ways, including confirmation that it does indeed explain the reference-mode behavior of the outcome indicator(s) so that it can be used in Step 6 to test alternative scenarios and policy options.

Although Figure 1 suggests that model-development is a one-pass, linear process, Sterman (2000) emphasizes that modeling should be an iterative process, with lessons from one phase of the process leading to revisions to the outcomes from earlier steps. However, it is not evident from published work whether this iteration is commonly carried out.

3 PROBLEMS WITH THE COMMON MODELING PROCESS

Although the process summarized in Figure 1 conforms with the field's accepted principles, a careful assessment suggests it may not be easy to perform nor reliable. This is due to its neglectance of important issues that are well known in the field, its attempt to "boil the ocean" by mapping the entire problem space, its initial reliance on purely qualitative methods, and the linear process it implies, even if later iteration occurs.

Step 1 is usually straightforward. However, attention is rarely paid to whether the item of concern is a stock (fish population, for example) or is instead an indicator whose value *depends on* one or more stocks (the fishing catch-rate). Skilled SD facilitators know exactly which is the case, but newcomers will not. The perceived difficulty of explaining this distinction between stocks and other entities often leads to participants being guided through a causal-mapping process that omits the distinction entirely.

Step 2 is more problematic. First, having quantified the performance indicator's time-path, the search for causal relationships proceeds in an entirely qualitative manner. Stakeholders proffer their view that factor B depends on factor A, whether or not any evidence exists to support that assertion. Nevertheless, their status as actors in the system is taken to justify including that relationship. Yet it is axiomatic in the SD field that human cognition is incapable of understanding how feedback systems work (Sterman 1989 and 1994; Paich and Sterman 1993; Moxnes 2000). Furthermore, human perception is notoriously subject to selection and bias (Kahneman et al 1982 ; Haselton et al 2005). It is, therefore, implausible that the sum of individual mental models will describe in any reliable manner how any system actually functions.

Secondly, qualitative consultation raises the risk that social and power relationships between participants influence the content and structure of the CLD and resulting model. Important factors or links may be missed, simply because those with that knowledge are not consulted, because they hesitate to offer their views, or because those views are ignored.

Thirdly, step 2 typically avoids identifying stocks and flows, even though the accumulating behaviour of stocks is fundamental to any system's behavior. Even if stocks and flows are identified and included in the qualitative diagram, it is again axiomatic that people cannot reliably estimate the behavior of stock-flow structures (Cronin et al 2009; Cronin and Gonzalez 2007; Sterman 2010) and since at least one accumulating stock *must* exist in any feedback loop, every such loop must include at least one causal mechanism that people do not understand (Morecroft 1982; Richardson 1986). Omitting stocks from the CLD is especially dangerous where those segments of the model are made up of aging chains (which rarely appear in CLDs in any case), since these exacerbate the stock-item's influence on long-term system behavior.

Fourthly, step 2 depends on the reliability of relationships between feedback loops and their behavior – reinforcing feedback is assumed to produce accelerating growth or decline, and balancing feedback produces goal-seeking behavior. However, it is entirely possible that feedback loops exist *without* generating such behavior, due to the existence of other causal mechanisms. It is also possible for the behavior ascribed to feedback loops to arise from other mechanisms entirely. Accelerating growth may arise when a factor growing in a linear manner overcomes a constant loss-rate, for example when a product's performance wins potential customers increasingly fast. Participants can, therefore, neither estimate outcome behavior from the feedback structure, nor intuitively estimate the feedback structure from the outcome behavior, problems exacerbated by the absence of data from step 2.

Fifthly, step 2 also relies on the belief that the "mental database" of participants far exceeds the written or numerical information available. Yet the sheer quantity of numerical information in any situation has long exceeded that which people can carry in their memory. In practice, any actor can have reliable information only on that part of the system in which they personally are active. We should not care, for example, how anyone *thinks* a company's customers behave; the only reliable information will come from customers themselves (and even that may not be reliable!) or from data about their observed behavior.

Finally, an over-arching problem is that step 2 seeks to identify and encompass the whole of the problem space of concern. Since the entire step is performed without data, it is not clear how we can know whether step 2 has indeed captured the whole problem space or alternatively strayed beyond its boundaries. A focus on participants' views on feedback – or "closing the loop" – also risks neglecting important exogenous factors.

Having agreed the shared mental model of the situation, step 3 should not be problematic. Nevertheless, novices generally struggle with the task, incorrectly identifying items as stocks, and missing items that *are* stocks. Failure is especially likely where aging chains of stocks are involved (even if they were identified in step 2). The extensive set of difficulties and risks inherent in the common procedure for developing CLDs means that only highly experienced SD practitioners can reliably lead steps 2 and 3.

Step 4 – building a quantified working model of a problem – follows established model-building procedures, but must also be done by the same experts who are able to facilitate steps 1 – 3. Typically, it is only at this stage that significant numerical information is looked for, and since the wide audience previously consulted already signed up to the feedback diagram, that structure determines the data that are sought. There is, thus, a risk that data about important factors are not looked for, because they were not identified in the CLD, and that data about insignificant factors are forced into the model, because they *were* included in the CLD.

Step 5 – validation and testing – also follows accepted procedures (Forrester and Senge 1980; Graham 1980; Barlas 1989; Coyle and Exelby 2000; Peterson and Eberlein 1994; Sterman 2000, Chapter 21; Schwaninger and Groesser 2009; Yücel and Barlas 2011). However, much effort will have been expended, much time will have elapsed, and great commitment to the model structure will have been built before this vital task is carried out. Fundamental problems easily survive from the very start of the process, making it necessary to go back to the beginning and revise all of the prior work!

Step 6 is where the simulation model is actually used to answer the questions identified in step 1. For a model to affect any organization's performance, someone must do something additional to or different from what would otherwise have been done – and such action must have some scale and occur at some points in time. Since these are quantitative features, they must be informed by quantitative findings. Thus, whilst qualitative insight might arise as early as step 2 in the process, participants cannot confidently act on anything until steps 3 to 5 are completed.

4 ADVERSE CONSEQUENCES FOR MODELING PROJECTS

Much insight may be gained from the commonly accepted process above, and successful simulation models have certainly been developed with that process. However, the issues outlined above may cause serious adverse consequences, both for particular projects, and for the wider adoption and impact of the SD method.

Solving the problem; Since a shared mental model is regarded as critical to developing a good SD model, the concept has received substantial attention. Evaluation of individual, group, and method outcomes has fallen into three categories: participant satisfaction and acceptance (McCartt and Rohrbaugh 1995; Vennix et al 1993; Huz et al 1997), changes in participants' and group thinking (Franco and Rouwette 2011), and improvements in participants' capability (Scott et al. 2014). However, since the ultimate purpose is to make a positive *impact* on the real world, these issues are not of primary concern. Two key questions are absent from the criteria for assessing the value of building shared mental models. First, is the shared mental model actually valid, or does it contain significant objective omissions or errors? Secondly, what did anyone *do* as a result of the process (spend more or less money, commit more or less effort, and so on), and with what significant improvement in outcomes? If SD models *had* reliably enabled better policies and resulted in improved outcomes, then much stronger adoption of SD might reasonably be expected.

Reliability; The risk of serious errors becoming embedded in any model arising from the standard process was already highlighted. This, too, would likely discourage repeated adoption of SD modeling.

Cost and time; It takes much time to bring together, several times, the wide audience whose shared mental model is sought. A separate effort must then follow to specify and build the working model, including lengthy efforts to find required data. A long time, therefore, elapses between starting the effort and obtaining actionable answers. The absence of data or validity-checking make it likely that a CLD and the resulting model will omit critical items (especially stocks) and causal relationships (especially multi-stock-flow structures), or will include irrelevant or mis-stated items.

Repeatability; Since the content and structure of the developed CLD depends heavily on the views of the participants and the facilitation of the process leaders, models cannot be transferred to other situations, no matter how similar they may seem. If we believe that the CLD built for any case must reflect the shared mental models of those involved, then no CLD developed by one team can be an acceptable starting point for any subsequent project on the same or similar topic with any other group. This idiosyncrasy of related models stands in stark contrast to other professional disciplines, which typically feature repeatable solutions – a proven method is applied to one example of a common class of problem; that procedure is then tested, refined, and documented for other test cases, from which point it can be deployed, with suitable adjustments, to many similar cases. The SD field, too, needs such repeatable models – as Forrester (2013, p. 30) remarks "Seldom, if ever, should a person model the specific situation of interest but, instead, should model the family of systems to which the specific one belongs", or as Homer (2013, p. 125) puts it, our method should "… be able to project a comprehensible 'sameness', as other modeling disciplines have managed to do".

Together, the problems with the commonly accepted process seem to explain why SD projects can be costly, time-consuming, and of uncertain value. And, if that is the case, then persuading any potential client to commission a first project or to repeat the effort for other challenges will be difficult. We, thus, have a plausible explanation for the slow adoption and limited impact of SD.

Those problems also cause difficulties in developing professional capacity. The complexity and uncertainty in the method mean that there is little chance of any novice developing from first principles a good working model of even a simple system. Guarding against the risks in the process relies entirely on the expertise and stature of the facilitators, making it hard for inexperienced practitioners to run successful projects. And, since the real value only emerges at the end of a long and costly process, it is also difficult for young professionals to win projects in the first place or to demonstrate value as the project progresses.

5 AN "AGILE" ALTERNATIVE PROCESS FOR SYSTEM DYNAMICS MODELING

The process suggested in Figure 1 is very similar to the "waterfall" approach to software development that dominated the information systems (IS) field until the 1990s. That process, too, starts with identifying the full scope of the desired solution, then tries to define the entire architecture of the application, before the whole solution is coded. The software is then tested and debugged, before being installed for users to employ. Like many SD projects, software solutions produced with the waterfall development process were widely felt to take too long, cost too much, and deliver uncertain value (Fowler 2001).

The waterfall approach has since been superseded in most IS-development cases by incremental and adaptive "agile" methods. Agile IS development processes feature many details and variants (Abrahamsson et al. 2002) that are beyond the scope of this paper. However, certain elements of those processes are transferable to SD methodology. First, user engagement is paramount, not just in the initial scoping of the application (which the common SD process does in steps 1 and 2, above), but throughout the development of the working software application itself. Applying the same principle to SD projects would imply that the user should be closely involved in building the working simulation from the start.

Secondly, agile IS development aims to achieve working software very early in a project, and to deliver incremental features continually. For SD, this implies getting to a working, if limited, simulation right from the start, then repeatedly extending that model. This further implies, most importantly, that the model be continually *validated* – it should at all times demonstrably match the real-world system's behavior, not just for the performance outcomes of concern, but also for all other elements.

Other agile principles that SD modeling can usefully adopt include close collaboration with knowledgeable individuals, rather than large-group reviews, welcoming changing requirements driven by learning as the software (or model) evolves, and simplicity (maximizing the amount of work *not* done).

Such an agile SD modeling process would still require technical rigor, but that can be ensured by observing the rigorous science of the SD method, consisting of four core principles:

1. We seek to explain, anticipate, and improve how outcomes of concern change over time.
2. The behavior over time of such outcomes depends directly on the behavior over time of the stocks on which they depend (plus changes due to exogenous factors or actors in the system).
3. The quantity of every accumulating stock at each point in time is derived from the cumulative total of all prior values for the stock's associated flow rates.
4. Those flow rates depend, in each period, on current stock quantities, decisions of human actors, and exogenous factors.

Items 3 and 4 together create interdependence, including feedback, but that feedback is a *consequence* of the underlying principles – it is not in itself one of those principles.

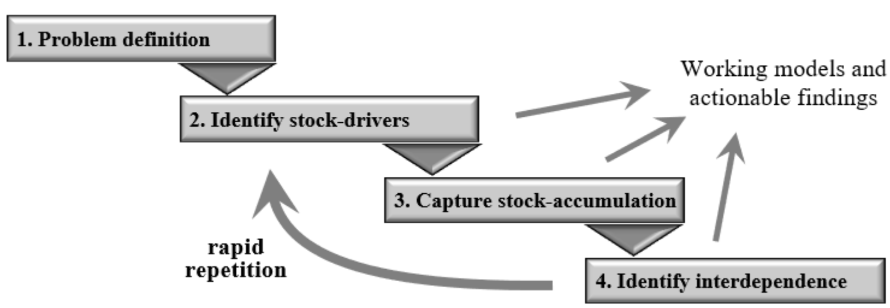

Figure 2: An agile model development process.

These principles can be translated into actual steps in a modeling process (Figure 2). In the first iteration, the whole cycle is completed with concerned stakeholders in a matter of hours. A diagram is built out from the factor(s) of concern in which every item and every causal relationship between those items is supported by quantified time-charts, even if many values must be estimated. Subsequent iterations may happen with people having particular knowledge of the parts of the model under scrutiny. Every iteration extends the working model, which confirms that the emerging causal analysis is valid and realistic.

6 CASE EXAMPLE OF AGILE SD MODELING

This section demonstrates the agile SD procedure with the diagnosis and resolution of a problem facing a mid-size company. The company provides IT support to mid-scale businesses, such as small retail groups, accounting and law firms, and construction companies. It supports clients' needs for computer hardware, software, and communications, and employs young, skilled technical staff to do so. After some successful years, the company embarked on a growth effort, which initially succeeded, but two years later was facing high rates of service problems and the loss of clients. The simple case is chosen to most clearly demonstrate the principles. The following sections show how questioning the company's CEO and reflecting his replies in quantified diagrams proceeded alongside a working model that he helped develop from step to step.

6.1 Step 1: How Performance is Changing Over Time

Step 1 is exactly the same as in the standard process – draw a time-chart(s) of the issue(s) of concern, including relevant history and desired future. The time-chart *must* have a scale on it and the chart's line must reflect at least an estimate of how the performance indicator (call it 'A') has actually changed. Clarifying if this chart is for a stock or something that *depends on* a stock must be done at this point, because it determines whether to do step 2 or jump straight to step 3.

<u>Question: You say customer problems are increasing and client numbers are falling. Can you sketch a chart of each item?</u> "Two years ago we had about 90 clients, and were doing well, so took on a sales executive to win more new clients. We grew to about 135 clients, but we then had reports of rising client problems, peaking at nearly 3 per month for each client. Staff must fix those problems fast, rather than get on with normal client support. We have since lost some long-standing clients and are now down to 115. We fear it will take time to reduce the rate of problems, and clients will likely continue to be lost."

Figure 3 shows the whiteboard sketch that resulted from this dialogue.

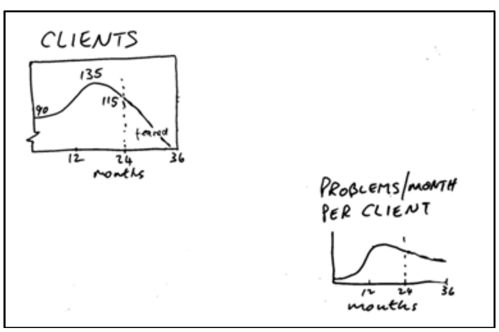

Figure 3: The time-charts of concern to the IT-support company.

6.2 Step 2: How Stocks Drive Performance

In step 2, the direct explanation for the outcome's values is identified. This *must* include one or more stocks (unless the outcome of concern is itself a stock, in which case this step is unnecessary). This is often a simple calculation from two or more other items (call those B and C). It should be possible to calculate or estimate outcome A from B and C, so time-charts are also added to those causal factors, to prove that the causality is valid. This enquiry is repeated for each of B and C, again with values or time-charts for every item identified, until the causal relationships reach one or more of the following – stocks, decisions, or exogenous factors.

Question: What caused the increasing rate of problems for your clients? “It looks like it was the pressure of work on our staff. We log the work they do – installing equipment and software, upgrading systems, training users and so on. That workload increased much more than our capacity to do the work. People were working into the night and at weekends. We estimate the workload may have been at its peak 60% more than people could do in a regular work week.” Has anything changed the amount of work each customer needs, like new software releases or equipment needs? “No more than usual.” Did you have problems with unskilled or unproductive staff? “No.” So what happened to staff numbers? “We managed a slow increase for the first half of the period, then numbers fell to about 37 today.”

Figure 4 shows the sketched time-series and the causality between the items from this dialogue. Items driving performance outcomes are usually identified within just 2 – 4 causal steps. Any uncertainty is resolved by seeking numerical evidence to support the causal relationships involved, by unit-checking and by developing in parallel a working model (available at sdl.re/m203) that captures the system’s behavior as it is built. This model is shown in Figure 5 (with corrected whiteboard estimates). Red chart lines are actual data for the clients’ problem rate and the CEO’s feared future. Data to validate the causal pathways from stocks to outcomes may not be easily available, but the process can proceed with plausible estimates. It is also self-validating – if A depends on B and C, then data on A and B imply values for C.

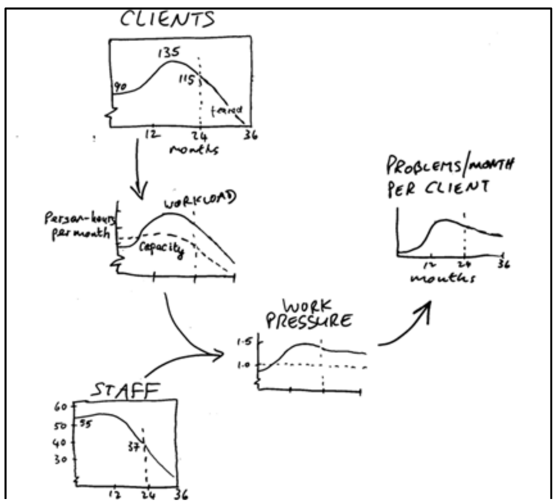

Figure 4: Stocks driving performance at the IT-support company.

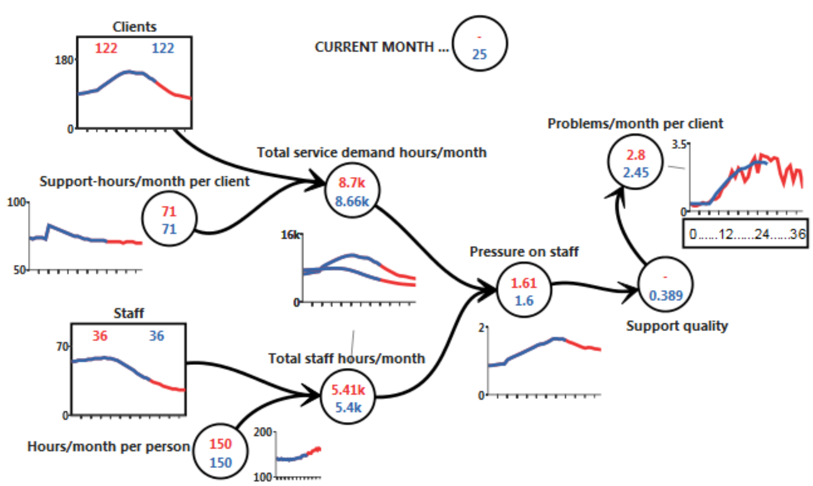

Figure 5: The model of stocks driving performance at the IT-support firm.

Critical to this process is that the diagrammatic picture and the working model develop in parallel. Modern SD software is sufficiently user-friendly that there is no reason to develop the diagram first, then follow that with a further, lengthy procedure to put that image to a working model. A first-pass model built alongside the diagram may need to be checked and improved, but stakeholders should be able to see their working model develop in parallel with their discussion of that model on whiteboards or other media.

6.3 Step 3. How Flows Change the Quantity of Stocks

Step 3 explains the behavior over time of the stock items. That behavior must reflect, and *only* reflect, how the stocks' in- and out-flows have changed over time, so those, too, are populated with time-charts. If the issue identified in step 1 is itself a stock, the process starts here and misses out step 2. There can be no arithmetical ambiguity whatever in Step 3 – the value of each accumulating stock is identical to the cumulative sum of all in- and out-flows to date. There may, however, be uncertainty when multiple flows exist, again resolved by reference to numerical evidence.

Question: How fast have you been adding and losing clients and staff? "The sales guy was successful at first, he upped the new client rate from 2 per month to about 7, but now we are hardly winning any clients at all. Previously, we rarely lost any clients, but starting about a year ago, that increased to a peak of 8 a few months back. We are losing fewer now. We kept hiring about 2 people per month as before. We previously lost people occasionally, but in recent months many more have left."

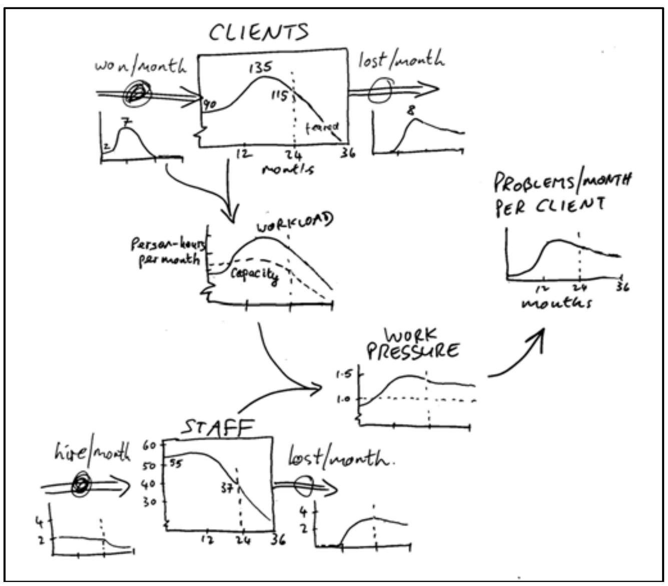

Figure 6: Flow rates changing numbers of clients and staff at the IT-support company.

Figure 6 shows how the diagram developed to reflect these answers. The model was again developed in parallel with the diagram to confirm that the behavior patterns were consistent (see sdl.re/m302). By this point, the outcomes of concern are entirely and reliably explained by changes to stock quantities (no decisions or exogenous factors directly affect performance in this case, although that is often found in other cases), and changes to the stocks are explained by the varying flow rates.

6.4 Step 4: How Stocks, Decisions, and External Factors Cause the Flow Rates

The only remaining question is what has caused the flow rates to change over time. This is answered in step 4 by following the same causal logic of step 2, and once again, every causal chain must originate at one or more stocks, decisions, or exogenous factors, and every such chain is validated by values and time-charts for all the items along that chain.

Question: So the growth in clients was simply a result of taking on the sales guy? "Correct – we didn't pick any up from competitors, or so far as we know from recommendations of existing clients." And why did clients leave at the increasing rate they did? CEO: "Well I guess it was inevitable. We were getting calls about problems that we created for them, or delays in fixing things that went wrong. It is a nuisance and takes a while to find another provider, but if you have had a lot of problems over many months, you will probably take the jump. Modeler: What about staff hiring and losses? CEO: "By the time we realized we had a problem, it was too late to bring in more staff (which takes a while in any case). The faster turnover was inevitable, too. People cope with over-work at first, but after a while they have had enough and decide to leave, and it's easy for skilled people to find other good jobs."

Since the causal explanations for each flow rate may involve any of the existing stock levels, including those of the stock whose flows are themselves being assessed, step 4 identifies interdependencies and feedback. In contrast to the standard process, however, the existence and relevance of such loops arise from the evidence, rather than from participants' speculation.

6.5 Exploiting the Model

The framework and model are now at a point where decisions to improve performance or solve the problem can be taken. The solution and action plan for the IT-support firm was to drop clients who were causing disproportionate difficulties, by helping them find alternative providers. This quickly killed the work overload, allowing support quality to recover. After a period of additional hiring, the firm was able to start growing once again, although now more carefully. This "preferred" future can be added to the diagram and tested in the model (Figure 7, which plays out this solution scenario; see sdl.re/m4a03).

In more complex cases, step 4 may identify additional stocks, not found in step 2 (although that is not the case in this example). Those may not be involved directly in explaining outcome behavior but *are* involved in explaining the flow rates of those resources that drive behavior. A company's product range, for example, may not directly explain its profits, but is part of an explanation for its customer win-rate. These additional stocks may include prior stocks and subsequent stocks in aging chains – explanations for changes to a stock of adult fish, for example, must include the stock of juveniles, or a company's graduate hiring rate will reflect the stock of available candidates. Such additional stocks are readily identified by posing the question "From where does this flow originate, and is its source important?" or "To where does this flow go, and is the destination-stock important?" The only such additional stock in this case is the small number of trainee staff, who take a few months to gain experience and become fully productive (see the model at sdl.re/m602).

By this point in the agile process, the core interdependencies between the stocks in the system are clear and supported by numerical evidence – what might be termed the "physics" of the system, including the *physical* feedback it implies. The final step is to complete the *policy* feedback, identifying what information is used, and in what way, to inform what decisions. Those decisions in the IT-support firm were the initial plan to add more clients and the hiring rate. The decision to hire the sales person and go for growth was driven by the firm's previous success, both in terms of good quality client support and profitability. Profitability is not shown in Figure 7, but is easily calculated from the fees paid by clients minus the costs of staff and other items. The resulting model – including policy feedback – is at sdl.re/m912.

7 BENEFITS OF THE AGILE PROCESS

The agile procedure deals with most of the challenges found in the currently common process. Since participants constantly see the need for numerical evidence, they identify entities whose definitions and values are well understood. Since they also see quantitative changes in those entities being caused by quantitative changes in the factors on which they depend, they can immediately check that the causality is in fact correct. Differences of opinion or understanding do not survive this process, so problem owners and stakeholders not only arrive at a shared mental model, but one that is demonstrably valid and usable.

Actionable insights arise early and continue throughout this process. The IT-support case above is too simple to demonstrate the importance of this finding, because the work was completed in a matter of hours. For more complex cases, however, the process may take longer and require periods of data discovery. In such cases, it is useful to learn, for example, that a sales decline is due to a fall in customer numbers rather than lower average purchase rates, or that staff shortages arise, not because of inadequate hiring, but because turnover has increased.

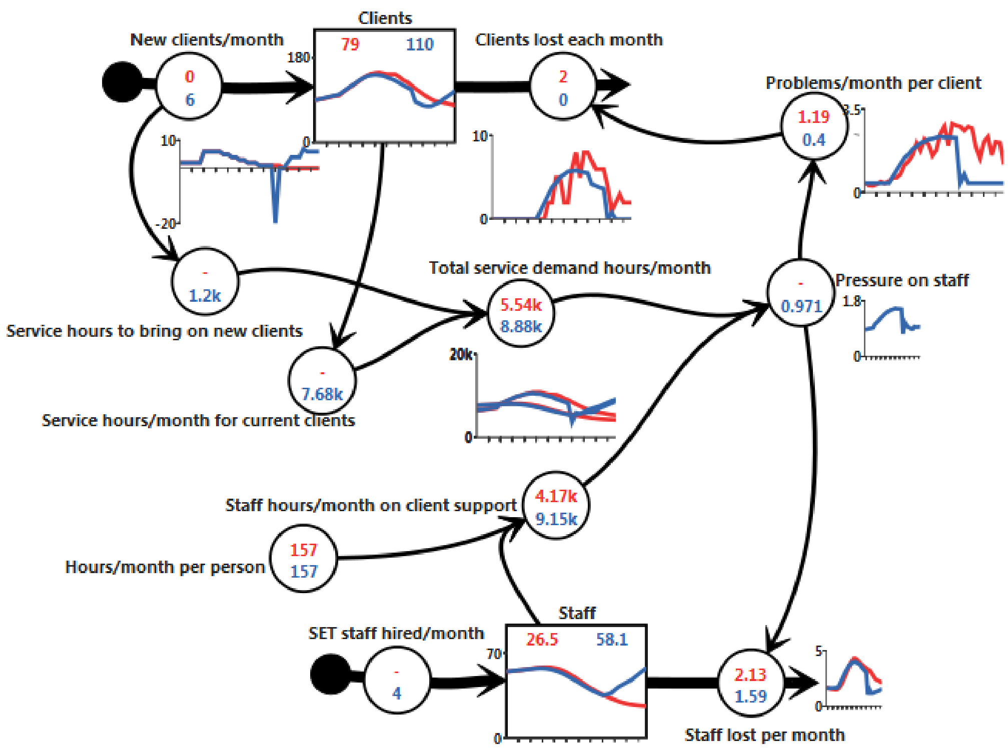

Figure 7: An example solution-test in the IT-support company model.

Fewer large-group meetings are needed, cutting both the time-demands on busy staff and the elapsed time for the project. Initial diagnosis may be carried out with the team who owns the problem, rather than engaging all who may have wide system knowledge. However, those others may become involved as the building of the model runs into topics on which they have knowledge – such consultation starts by explaining the model's progress to date. At each stage, however, the simple, logical, and data-supported steps of the agile process take less time than the open-ended consultation needed to develop CLDs.

Since problem owners readily understand what is required and why, they can trust the modeler to work with those who have specific knowledge on particular points and need only reconvene to see the results of adding and quantifying those points. Engaging with the full range of stakeholders may be useful at the end, so that the entire audience shares the understanding to which they contributed.

To be clear – qualitative causal loop or influence diagrams do not feature *at any stage* in this process. The procedure *discovers* feedback mechanisms that are evidenced by the data, and as long as the behavior of concern is well-explained by the analysis, there is no reason to expect that additional feedback mechanisms are significant. Nevertheless, it may be legitimate to enquire as to whether additional feedback *could* arise or *could* be designed into the structure to improve its management. Such enquiry, though, must follow the same logic as the foregoing analysis – asking which stocks in the system need to change their time-path of growth or decline, and what changes to their flow rates must happen for that to arise; and what interdependencies not previously identified might bring about that change? This enquiry may be especially useful when seeking to improve policies or to redesign the system, such as by adding or removing stocks.

The agile process also identifies the relevant boundaries of the problem-space, which are discovered when no unanswered questions remain, rather than by trying to define those boundaries at the outset. Model boundaries may, therefore, be more limited than otherwise, but could include an *additional* scope arising from important mechanisms that the more common process missed through lack of attention to data.

The logical, self-validating agile process allows inexperienced professionals to achieve impactful work early in their SD career. For more complex cases, they can also start work from the numerous standard structures that have already been discovered from extensive experience with the agile process or from standard templates developed by experienced colleagues. The stature and experience required to facilitate qualitative, multi-stakeholder debate is no longer critical. Those proven cases can also make clear to consumers of SD work what they are going to get from it, and give them confidence that the result will work and be helpful, because it has been so for other users.

Two important insights have arisen from repeated and varied use of this process to tackle real-world business challenges. First, performance improvements are entirely focused on the flow rates in the system – if performance of the system depends entirely on the quantity of its stocks, and exogenous factors remain constant, then zero flow rates mean no change in performance. Secondly, decisions must change performance by altering the flow rates. Indeed, many decisions are about what those flow rates should be – client acquisition and hiring, in the IT-support case. Others, such as price changes, marketing expenditure, or training, alter the flow rates less directly.

It was noted earlier that successful SD professionals might not follow the common procedure described in the field's key sources and summarized in Figure 1. Two pieces of evidence support this supposition. First, leading academics and successful consulting groups in SD tend to specialize in certain fields of application – Ford (2009) in water and power resources; Moxnes (2004) in renewable resources; Paich et al. (2009; 2011) in pharmaceuticals; Lyneis and Ford (2007) in project management, and so on. This is not to imply that these individuals have not carried out other types of work; merely that they have a cumulative body of work in those domains. Such professionals do not start each new enquiry with a blank sheet and seek to build qualitative, shared mental models from each fresh set of stakeholders whilst bringing no known structures or phenomena from previous experiences.

Whatever the method by which the very first model in a domain was developed, subsequent projects identified and developed standard structures that are found to be widespread. This principle should be extensible to many more domains than have been documented. Not only does every fishing region have fish, vessels, and fishermen, but every law firm has clients, lawyers, and cases; every health-care system has patients, doctors, nurses, treatments, and hospitals; every city has criminals, victims of crime, and police. Many of the causal dependencies are also shared between similar cases – vessels catch fish and doctors treat patients – so, although the numerical values and strengths of relationships will differ markedly between different examples, the underlying physics will be shared.

8 CONCLUSIONS

Concern within the SD field regarding the method's slow adoption and limited impact, compared with the considerable relevance and value it could potentially offer, has now reached a level that calls into question the very means by which it is practiced and communicated. Such questioning suggests that the process of modeling should be fundamentally rebuilt, starting from the method's basic principles. Following those principles has led to a process that bears striking similarity to the "agile" method now used in many software development projects. Solutions are built hand-in-hand with users, and working models and actionable insights are delivered continuously. Experience over many years with the agile process described in this paper suggests that it is fast, effective in delivering actionable insights, and reliable in generating similar structures for similar cases. Although that experience has focused mostly on business-related challenges, the principles and process should be equally practical and reliable in all other application domains.

Together, the agile process for modeling new problems and the use of proven structures for well-known situations may deliver valuable SD models, faster and more reliably, and thus contribute to greater adoption and impact of the method in the real world. The two procedures could have the incidental benefit of accelerating young professionals' mastery of the science and practice of SD, adding further to its adoption.

A practice-guide based on this tutorial is at sdl.re/AgileSD.

REFERENCES

Abrahamsson, P., O. Salo, J. Ronkainen, and J. Warsta. 2002. *Agile Development Methods: Review and Analysis*. Helsinki: VTT Publications.

Barlas, Y. 1989. "Multiple Tests for Validation of System Dynamics Type of Simulation-Models". *European Journal of Operational Research* 42(1):59–87.

Coyle, R. G. 1996. *System Dynamics Modelling: A Practical Approach*. London: CRC Press.

Coyle, R. G. and D. R. Exelby. 2000. "The Validation of Commercial System Dynamics Models". *System Dynamics Review* 16(1):27–41.

Cronin, M., C. Gonzalez, and J. D. Sterman. 2009. "Why Don't Well-educated Adults Understand Accumulation? A Challenge to Researchers, Educators, and Citizens". *Organizational Behavior and Human Decision Processes* 108:116–130.

Cronin, M., and C. Gonzalez. 2007. "Understanding the Building Blocks of System Dynamics". *System Dynamics Review* 23(1):1–17.

Fisher, D. 2010. *Modelling Dynamic Systems*. Lebanon, New Hampshire: ISEE Systems.

Ford, A. 2009. *Modelling the Environment: An Introduction to System Dynamics Models of Environmental Systems*. Washington DC: Island Press.

Forrester, J. W. 2007. "System Dynamics – the Next Fifty Years". *System Dynamics Review* 23(2–3):359–370.

Forrester, J. 2013. "Economic Theory for the New Millennium". *System Dynamics Review* 29(1):29-41.

Forrester, J. W., and P. M. Senge. 1980. "Tests for Building Confidence in System Dynamics Models". In *System Dynamics: TIMS Studies in the Management Sciences*, edited by A. A. Legasto et al., 209–228. Amsterdam: North-Holland.

Fowler, M. 2001. "Is Design Dead?". In *Extreme Programming Explained*, edited by G. Succi and M. Marchesi, 3–18. Boston, Massachusetts: Addison-Wesley.

Franco, L. A., and E. A. J. A. Rouwette. 2011. "Decision Development in Facilitated Modelling Workshops". *European Journal of Operational Research* 212:164–178.

Graham, A. K. 1980. "Parameter Estimation in System Dynamics Modelling". In *Elements of the System Dynamics Method*, edited by J. Randers, 143–161. Cambridge, Massachusetts: Productivity Press.

Haselton, M. G., D. Nettle, and P. W. Andrews. 2005. "The Evolution of Cognitive Bias". In *The Handbook of Evolutionary Psychology*, edited by D. M. Buss, 724–746. Hoboken, New Jersey: Wiley.

Homer, J. 2013. "The Aimless Plateau Revisited: Why the Field of System Dynamics Needs to Establish a More Coherent Identity". *System Dynamics Review* 29(2):124-127.

Huz, S., D. F. Andersen, G. P. Richardson, and R. Boothroyd. 1997. "A Framework for Evaluating Systems Thinking Interventions: An Experimental Approach to Mental Health System Change". *System Dynamics Review* 13(2):149–169.

Kahneman, D., P. Slovic, and A. Tversky. 1982. *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge, Massachusetts: Cambridge University Press.

Lane, D. C. 1994. “Modelling as Learning: A Consultancy Methodology for Enhancing Learning in Management Teams”. In *Modelling for Learning Organisations*, edited by J. D. W. Morecroft and J. D. Sterman, 64-84. Portland, Oregon: Productivity Press.

Lane, D. C. 2008. “The Emergence and Use of Diagramming in System Dynamics: A Critical Account”. *Systems Research and Behavioral Science* 25(1):3–23.

Lyneis, J. M., and D. N. Ford. 2007. “System Dynamics Applied to Project Management: A Survey, Assessment and Directions for Future Research”. *System Dynamics Review* 23(2):157-189.

Maani, K. E., and R. Y. Cavana. 2000. *Systems Thinking and Modelling*. Auckland, New Zealand: Pearson Education.

Maidstone, R. 2012. *Discrete Event Simulation, System Dynamics and Agent Based Simulation: Discussion and Comparison*. White Paper, https://www.researchgate.net/publication/306157680_Discrete_Event_Simulation_System_Dynamics_and_Agent_Based_Simulation_Discussion_and_Comparison, accessed August 18th, 2018.

McCartt, A. T., and J. Rohrbaugh. 1995. “Managerial Openness to Change and the Introduction of GDSS”. *Organizational Science* 6:569–584.

Morecroft, J. D. W. 1982. “A Critical Review of Diagramming Tools for Conceptualizing System Dynamics Models”. *Dynamica* 8(1):20-29.

Morecroft, J. D. W. 2007. *Strategic Modelling and Business Dynamics*. Chichester: Wiley.

Moxnes, E. 2000. “Not Only the Tragedy of the Commons: Misperceptions of Feedback and Policies for Sustainable Development”. *System Dynamics Review* 16(4):325-348.

Moxnes, E. 2004. “Misperceptions of Basic Dynamics: The Case of Renewable Resource Management”. *System Dynamics Review* 20(2):139-162.

Mustafee, N., J. Powell, S. C. Brailsford, S. Diallo, J. Padilla, and A. Tolk. 2015. “Hybrid Simulation Studies and Hybrid Simulation Systems: Definitions, Challenges, and Benefits”. In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz et al., 1678-1692. Piscataway, New Jersey: IEEE.

Paich, M., C. Peck, and J. Valant. 2009. *Pharmaceutical Product Branding Strategies: Simulating Patient Flow and Portfolio Dynamics.* New York: Informa Healthcare.

Paich, M., C. Peck, and J. Valant. 2011. “Pharmaceutical Market Dynamics and Strategic Planning: A System Dynamics Perspective”. *System Dynamics Review* 27(1):47-63.

Paich, M., and J. D. Sterman. 1993. “Boom, Bust, and Failures to Learn in Experimental Markets”. *Management Science* 39(12):1439-1458.

Peterson, D. W., and R. L. Eberlein. 1994. “Reality Checks: A Bridge Between Systems Thinking and System Dynamics”. *System Dynamics Review* 10(2–3):159–174.

Pruyt, E. 2013. *Small System Dynamics Models for Big Issues*. Delft, Netherlands: TU Delft.

Richardson, G. P. 1986. “Problems With Causal-Loop Diagrams”. *System Dynamics Review* 2(2):158–170.

Schwaninger, M., and S. N. Groesser. 2009. “System Dynamics Modelling: Validation for Quality Assurance”. In *Encyclopedia of Complexity and System Science*, edited by R. A. Meyers, 9000–9014. Berlin: Springer.

Scott, R. J., R. Y. Cavana, and D. Cameron. 2014. “Evaluating Immediate and Long-Term Impacts of Qualitative Group Model Building Workshops on Participants’ Mental Models”. *System Dynamics Review* 29(4):216-236.

Sterman, J. D. 1989. “Modelling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment”. *Management Science* 35(3):321-339.

Sterman, J. D. 1994. “Learning In and About Complex Systems”. *System Dynamics Review* 10(2):291-330.

Sterman, J. D. 2000. *Business Dynamics*. Boston, Massachusetts: McGraw-Hill.

Sterman, J. D. 2010. “Does Formal System Dynamics Training Improve People’s Understanding of Accumulation?”. *System Dynamics Review* 26(4):316–334.

The Balanced Scorecard Institute. 2013. *Building & Implementing a Balanced Scorecard: Nine Steps to Success*. https://www.balancedscorecard.org/BSC-Basics/The-Nine-Steps-to-Success, accessed August 18th, 2018.

The International Council on Systems Engineering. 2011. *Systems Engineering Handbook v3.2.2*. Los Angeles, California: INCOSE.

The Supply Chain Council. 2013. *The SCOR Framework.* Chicago, Illinois: APICS.

Vennix, J. A. M. 1996. *Group Model Building.* Chichester: Wiley.

Vennix, J. A. M. 1999. "Group Model-Building: Tackling Messy Problems". *System Dynamics Review* 15(4):379-401.

Vennix, J. A. M., W. Scheper, and R. Willems. 1993. "Group Model Building: What Does the Client Think Of It?". In *Proceedings of the 1993 International System Dynamics Conference,* July 23rd-26th, Bergen, Norway, 208-209.

Wolstenholme, E. F. 1994. "A Systematic Approach to Model Creation". In *Modelling for Learning Organisations*, edited by J. D. W. Morecroft and J. D. Sterman. Portland, Oregon: Productivity Press.

Yücel, G., and Y. Barlas. 2011. "Automated Parameter Specification in Dynamic Feedback Models Based on Behavior Pattern Features". *System Dynamics Review* 27(2):195–215.

AUTHOR BIOGRAPHIES

KIM WARREN is an experienced strategy professional, teacher, and publisher of online courses and teaching resources on business modeling. After senior strategy roles in the petrochemicals, brewing, hotels, and leisure industries, Kim joined London Business School, to teach on MBA and Executive programs. Realizing serious limitations with the strategy methods available, he developed the powerful strategy dynamics frameworks for designing and managing enterprise strategy. He went on to develop a practical and effective process for converting those frameworks into working, quantified models of any enterprise, function, or challenge. Once a specialist skill, building these simulations is now easier, faster, and more reliable than spreadsheet modeling. His LinkedIn profile is at https://www.linkedin.com/in/kimwarren/, his website is www.strategydynamics.com, and his email address is kim@strategydynamics.com.