"Do I have to?": Convincing Program Implementers that Program Logic Models are Valuable[1]

Victoria S. Engel, NYSERDA, Albany, NY
Scott Albert, GDS Associates, Inc., Manchester, NH
Lynn Hoefgen, Nexus Market Research, Inc., Cambridge, MA
Lawrence J. Pakenas, NYSERDA, Albany, NY

ABSTRACT

Logic modeling documents program inputs and activities and how they are expected to lead to outputs and outcomes. Reaction to and acceptance of logic modeling[2] can be mixed among program implementers. This paper explores reasons for this mixed reception and identifies ways to overcome misgivings.

Several barriers cause apprehension among program implementation staff regarding the logic modeling process, including: real imposition on their time; perceived questioning of their expertise; concern regarding content accuracy; objections to logic model formats (aesthetics); and lack of understanding or appreciation of the timing and associated benefits to be gained from their efforts. Related to all of these barriers is the common perception among some that they "already know" everything about a given program that could emerge from a logic model. Nonetheless, some implementers embrace logic models and the logic modeling process—including some who have initially been resistant, but through experience have come to value and depend on them.

This paper assesses these barriers and how they can be overcome. It includes examples from direct experience with development of nearly 40 program logic models for a variety of ongoing and new energy efficiency, renewable resource, and research and development programs, as well as at the sector and portfolio levels. It also discusses how logic modeling has helped shape evaluation and provides a framework for tracking program progress. The paper includes a number of lessons learned and ends with suggestions to help others use logic models more effectively in their organizations.

Background

A program logic model, as defined by the New York State Energy Research and Development Authority (NYSERDA), is a diagram and text document that discusses the logical relationships among key program elements. The diagrams are constructed with boxes, circles, and other geometric figures to pictorially 1) map the relations among inputs, activities, outputs and outcomes embedded within the programs; 2) identify hypotheses and key indicators; and 3) document potential external influences. In addition, the logic model document includes text, in report format, that identifies the context of the program compared to broader sector and portfolio-level efforts, key targeted market actors, program activities, external influences and potential testable hypotheses that can suggest additional sectors, actors or technologies for the program to consider. Figure 2 (located at the end of this paper) shows the logic model diagram for NYSERDA's EmPower New York Program and is included to illustrate NYSERDA's logic model format (GDS Associates, Inc.).

[1] The views expressed in this paper are those of the authors and do not necessarily reflect the views of the New York State Energy Research and Development Authority.

[2] Although program-level logic models are highlighted in this paper, the same process and barriers can be faced when developing sector-level logic models.

At the beginning of the program logic modeling process, NYSERDA's program logic team searches and reviews applicable secondary data sources such as competitive program solicitations and program-specific websites to gain a general understanding of each program. In addition, the team reviews materials for similar programs implemented outside of New York. A draft logic model report is then prepared and shared with program implementers and a feedback meeting is held with staff to discuss comments, make corrections and clarify detail found within these secondary sources. The feedback session also gives program implementers an opportunity to provide additional context-related to the program and provides important insights regarding the rationale for the program's targeted activities and goals.

Given this reliance on secondary data sources and the knowledge of program implementers, the value of the program logic modeling process may not be evident to them in the beginning. Some implementers view the program logic model as a program synopsis, or as mere documentation of the program. Compared to evaluations that produce more tangible results (*e.g.*, impact evaluations), program logic modeling can be seen as more of an academic exercise. With proper explanation of the process, however, program implementers can learn and understand that, for example, indicators generated from the logic model can be incorporated into program-specific databases and data collection activities. Further, these logic model-identified measurement indicators and researchable issues can form the basis for targeted in-depth interview and telephone survey instruments used to track awareness, knowledge or market share of a technology as well as overall program progress toward key goals. Researchable issues identified in the logic model provide potential areas for further investigation.

Logic models also provide a succinct way to communicate with the outside world what a program is intended to achieve and how it is designed to achieve it. Given the dependence of energy efficiency and renewable energy programs on various entities for support, this communication of core program concepts can be essential. Finally, "forcing" (*i.e.*, gently persuading) program implementers to articulate how they expect program activities to lead to given outcomes can reveal gaps or even faults in their logic. Thus, it is very important for program implementers to understand the value of logic models as a guide to informed program design, delivery, and improved communication.

Common Barriers to Embracing Program Logic Models

Common barriers preventing program implementers from embracing logic models include: demands on their time; perceived questioning of their expertise; concern regarding accuracy of content; objections to the format (aka aesthetics) of logic models; and a lack of understanding or appreciation of the timing and associated benefits to be gained from these efforts. A description of each of these barriers follows.

Demand (Real Imposition) on Staff Time

"I'm busy, so let's make this meeting quick."[3]

By its nature, the program logic modeling process can be a fairly time-consuming undertaking. As described above, program implementers' participation is critical to this process as only these staff members can convey the fine points and the intricacies about the program. In addition, program logic model development often includes a few reviews of the draft logic model documents. Therefore, the time spent discussing and reviewing a program logic model report can escalate and require program

[3] These are example quotations that could be expressed by program implementers during the logic modeling process.

implementers to balance this need with their own competing program implementation and oversight priorities, which can be a challenge.

Perceived Questioning of Staff Expertise

"What's the point in doing a logic model now?"
"I already know everything there is to know about this program – why do we have to do this?"

Ideally, a logic model is developed during the initial program planning stage; however, this is not always possible. Implementing the logic model process on a program that is well-established in the field but has not yet been documented can be a daunting task. Staff may consider this request as an implication that something is wrong with the program (*e.g.*, this program has been in the field for years. Why is there sudden interest in looking at the logic of this program?). In addition, there could be a perception that the evaluators are questioning the program implementer's expertise and decision-making regarding the program. Further, the program logic modeling process may be considered a waste of time among some of these staff members due to their in-depth knowledge of the program (*i.e.*, program implementers believe they already know everything there is to know about the program; thus, what is the value of the logic modeling process?). Forcing program implementation staff to make explicit their internal opinions and assumptions regarding key market actors, program goals and barriers that they are working to overcome can be viewed as a direct challenge of their judgment and management authority.

Concern Regarding Accuracy

"They just don't get it!"

Unlike the details required to scope out a process or more direct field data collection-oriented evaluation, logic models are typically developed at a relatively high level, with focus on general activity categories, associated outputs and short, intermediate and long-term outcomes anticipated to result from these activities. There is often concern, therefore, among program implementers that evaluators will be unable to accurately describe and communicate the details of the program in the logic model report. Thus, evaluators' desire to commence the logic modeling process in the first place may be met with resistance from program implementers.

In addition, because of their in-depth knowledge of their programs, there are program implementers who spend the majority of the logic modeling process ensuring that all the specific details of the program (*e.g.,* specific program activities) are described accurately. This rapt attention to detail can inhibit the progress of the logic modeling process by swaying the focus away from identifying the overarching goals of the program; thus, the value of the process can quickly dissipate.

Logic Model Aesthetics

"I can't even read these things!"
"Where are my reading glasses?!?"

For the uninitiated, the format and content of a logic model diagram can be a bit overwhelming. Faced with reviewing a detailed model such as this, program implementers new to the logic modeling process are often quite reluctant and may tend to procrastinate during the review process.

Lack of Understanding or Appreciation of Timing and Benefits

"These things are so far over my head."
"I'm lost."
"I'm so confused."
"I just don't get the point of this."

Ideally, program logic models should be completed during a program's design phase to help identify key program activities and to make explicit the anticipated outcomes from the program. Since this process does not always occur with every program, program implementers may consider the program logic modeling exercise a missed opportunity, unable to be taken advantage of once the program is deployed. Thus, program implementers may be reluctant to conduct such an analysis after the fact. The fact that some program implementers have never gone through the logic modeling process with their programs before might also be the ones most likely to express reluctance to this type of evaluation.

Overcoming These Common Barriers

When faced with the barriers described above, it is important to convey the many benefits of the logic modeling process, some of which include identifying critical program activities within the context of targeted markets; identifying high priority measurement indicators and researchable issues for tracking performance, market changes, and assessing causality; and providing program implementers with insight and feedback on program effectiveness as well as recommendations for modifications to better align activities with desired goals (Albert). What follows are some suggestions for overcoming these common barriers, while at the same time, identifying the value of logic modeling to program implementers.

Identify the Value at the Beginning

Given their depth of knowledge regarding market needs, actors, barriers and other in-the-field realities, obtaining commitment and direct input and review support from program implementers is critical to the success of the logic modeling process. Expecting program implementers to assist in the development of a logic model without providing them with an adequate background of the process, expected deliverables and benefits must be avoided. Program implementers should have a thorough understanding of the process in order to provide useful, effective information and to ensure their buy-in to the process. Further, an explanation of how the logic model work can be used by other evaluators (*e.g.*, how measurement indicators can help define specific program data collection needs and influence market surveys) may also bolster support for this process. NYSERDA has found that program implementers' attention to the logic modeling process is directly proportional to how well the benefits are conveyed at the beginning of the logic modeling effort.

Be Prepared and Hold Efficient Meetings with Staff

Nothing can jeopardize the credibility and the future of the program logic model process more than scheduling a meeting with program implementers without conducting a proper review of the program first. A review of all applicable program materials is necessary prior to meeting with staff to ensure a basic understanding of the program's implementation and its goals. Meetings with staff should

supplement this knowledge and should center on clarifying questions and obtaining additional insights from program implementers. As such, program implementers' time should not be spent identifying the basic purpose of the program. Instead, these meetings should be viewed as valuable to all parties involved.

All participants should get something useful out of every meeting: be it a clearer picture and more articulate communication of the anticipated outcomes from program activities that will help a program manager more easily explain key elements of the program (an "elevator story"), or an evaluator gaining improved insights regarding specific linkages between program activities, outcomes and potential researchable issues that might need to be addressed to help test key program theories or assess causality (the "miracle in the middle"). It will be difficult in the future to gain the favorable opinion of program implementers regarding the logic modeling process if they feel they are merely reciting the basic tenets of the program during feedback meetings.

In addition, meetings with program implementers should be organized well and run smoothly. These meetings offer the evaluators an opportunity to have program implementers' undivided attention as these staff are removed from their day-to-day management activities. These meetings should be focused with clear agendas so everyone involved gains knowledge.

Balance the Level of Detail

Program logic models can easily become unwieldy. Since each logic model is dependent on the detail provided by program implementers, it is important to balance the need for simplification with a level of detail sufficient to reasonably describe key program elements, expectations and influences. For example, instead of highlighting each individual marketing activity of the program in a logic model, consider streamlining the activity into marketing generally.

Program logic models are designed to provide specificity regarding a program, including an understanding of what the program does, why the program does what it does, the expected outcomes of those activities, and what influences external to the program might be helping or hindering achievement of its expected outcomes. However, the question of how much detail to provide can turn an informative logic model into an overwhelming maze of boxes and arrows if the material is not displayed in a concise manner. A good rule of thumb has been to limit the number of program activities documented in a logic diagram to a range of no more than five to seven with enough separation to distinguish between tasks (*e.g.*, marketing activities vs. partner recruiting/training; technical assistance vs. quality assurance; product testing/demonstration vs. knowledge dissemination/education).

Lessons Learned

As NYSERDA began developing its initial program logic modeling activities in 2003, the framework for these diagrams and reports was very much a "work-in-process". As NYSERDA had never before embarked on formal program logic modeling work, NYSERDA's logic team took that opportunity to experiment with some different approaches in the hope of identifying the most efficient and effective method. Drawing on the expertise of its logic modeling team and evaluation assistance contractor regarding best practices in the industry, NYSERDA developed a customized logic modeling approach that has been evolving ever since. The following describes some of the key lessons NYSERDA has learned during the past four years as NYSERDA's program logic modeling process progressed.

Provide Drafts before the Meeting

Early in NYSERDA's logic modeling experience, the team often reviewed the programs to the extent secondary data was available (*e.g.,* reports, websites), but would meet with program implementers to go over specific details. In fact, early on, it can be described that when embarking on a specific program logic model, the NYSERDA team would meet with staff with a "blank slate". In addition to going over the basic process of the programs, the NYSERDA logic team would develop the program logic model diagram with staff on a real-time basis, using colored note pads to identify the inputs, activities, outputs and outcomes, and yarn to show the logical links (threads) between these activities, outputs and outcomes. Generic workshops on the concepts of logic modeling were also conducted with NYSERDA's program implementers. While these were good ways to introduce program implementers to the logic modeling process, and to demonstrate the value of program logic models early on, they were also very resource intensive.

In the years that have followed, NYSERDA's logic team has taken a different approach by becoming more proactive in their review process and by providing program implementers with drafts prior to the beginning of each program-specific logic model meeting. This way, the team has been able to provide NYSERDA's busy program implementers with a document to react to, and in case staff does not have time to review the document prior to the meeting, the NYSERDA logic team can still walk staff through an actual hard-copy report.

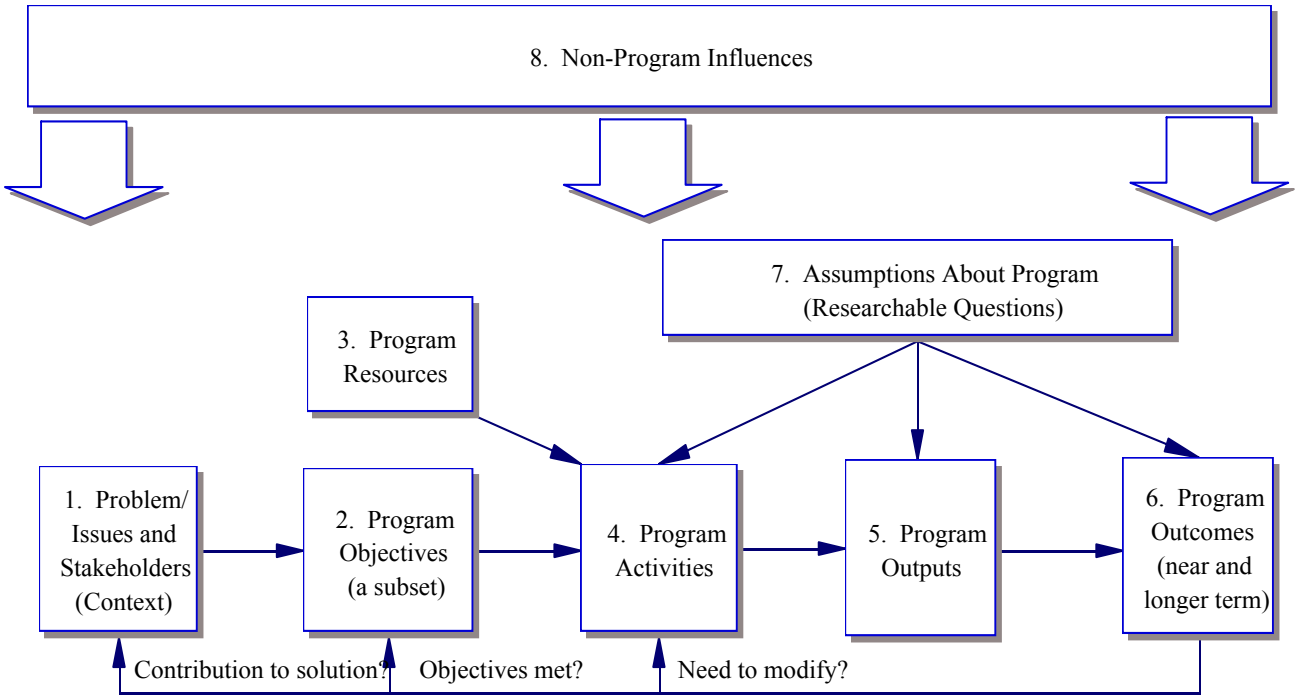
The sequence of logic model documents now typically starts with a draft key elements document and then moves to a draft of the logic diagram and a more developed logic document. The process culminates (after potential additional reviews and revisions) with a final document and diagram for publication and distribution to key program management, implementation, evaluation, and policy stakeholders. The final logic model document includes the following sections:

1. Problem/Issues and Stakeholders (Context): Describes the problem(s) the program is attempting to solve, or issues it will address and the regulatory and stakeholder environments (context) within which the program is working.
2. Program Objectives: Describes, at a high level, the program's ultimate purpose and targets.
3. Program Resources: Identifies the dollar, staffing and partnership, etc. resources the program is providing.
4. Program Activities: Describes the various research, product development, demonstration and commercialization progress support activities and strategies being delivered through the program.
5. Outputs: Describes the anticipated immediate results associated with program activities.[4]
6. Outcomes and Logic Diagram: Describes, in text, what is expected to be achieved in the near, intermediate and longer term, and shows, in .diagram form, the linkages between key activities, outputs, outcomes and external (non-program) influences.[5]
7. Assumptions: Describes assumptions about how program activities and outputs will lead to the desired near, intermediate and longer-term outcomes.[6]
8. Non-Program Influences: Describes factors outside the program that may drive or constrain the achievement of outcomes.

[4] Typically not developed until NYSERDA program implementers' feedback on the activities text and logic diagram have been received.
[5] Ibid.
[6] Ibid.

The relationship among these eight items is presented in Figure 1. As will be described later, this template has been tested on NYSERDA's R&D programs with much success.

Figure 1. Program Design Template



Hold Meetings or Teleconferences to Discuss Models

The NYSERDA program logic modeling team has long agreed that committing time to speak with staff, whether on the phone or in person, is an important and time-efficient way to ask questions and expand on detail. Rather than provide program implementers with draft reports to review and comment, the program logic team has had success in coupling these draft reports with teleconference or in-person review meetings. In these venues, both program implementers and the evaluators have been successful in centering on specific issues and discussing aspects of the program in greater detail than can be expressed in written comments. While in-person meetings are not always the most cost-effective choice when reviewing draft documents, teleconferences have provided an excellent second choice. These phone meetings have been most effective with program implementers that already have developed a solid working rapport.

Remain Flexible!

At the beginning of NYSERDA's program logic modeling process, there was a general desire to develop program logic models that followed the same format. While this generally worked well for deployment programs (*i.e.*, typical of market transformation and resource acquisition programs providing outreach and services directly to end-use customers), the same was not true for NYSERDA's Research and Development (R&D) programs. During the fourth year of the program logic modeling process, the NYSERDA logic modeling team decided to take a new approach with R&D programs and analyzed the utility of the old deployment program-based logic modeling template. A new template was developed which presents key logic model information in a much more user-friendly fashion (see Figure

1 above). The information presented in text is easier to understand in this template and the interrelationships among activities, outputs and outcomes are much more obvious. While consistency can be useful in some applications, logic model report text and diagrams should remain flexible to adapt to different types of programs.

Conclusions

While the logic modeling process can be characterized at times as a painful one by both program implementers and the logic team, general consensus among NYSERDA's logic modeling team suggests that the time and effort spent to address and overcome program implementers' reluctance to this process have proven to be worthwhile. As time has gone on, support for logic modeling has increased.

For example, as part of an overarching study conducted two years ago by reviewing NYSERDA's entire evaluation process, logic modeling was identified as one of the least valuable aspects of the evaluation process (Research Into Action, Inc., 2006). Reasons for these views, and similar to the barriers described earlier in this paper, included the timing of the logic models; a belief that the logic models were restating the program's purpose without providing insight or guidance for future program development; the time needed to complete the logic models; and skepticism regarding the logic model's usefulness for program management.

More recently, however, these opinions have started to change. An update to this evaluation review found that logic models were still listed among the most and least useful components contributed by NYSERDA's evaluation contractors; while some program implementers still did not recognize the value in this process, in one instance, the development of the logic model and the rigor in identifying processes and outcomes was deemed very useful (Research Into Action, Inc., 2007).

In addition, staff from one NYSERDA program took the initiative to develop a sector-level logic model for an important policy decision maker because NYSERDA staff felt that the graphic nature of the logic model was the best way to describe the interactions that needed to occur in order for the intervention to be successful. This example also shows how NYSERDA's logic models have proven to be useful as an effective communication tool for stakeholders.

Lastly, another NYSERDA program implementer has explained that the logic modeling process has provided additional benefits for designing and implementing programs, such as

- Helping to ensure all aspects of the project, that might not have been evident otherwise, are considered, including which aspects should not be dealt with and which aspects should be noted and considered in the big picture;
- Forcing project staff to identify external influences, synergies and challenges that might not have been obvious otherwise; and
- Helping to identify market research needs

Thus, in addition to improving program designs and implementation strategies, program logic models are improving the effectiveness of program delivery and are increasing the likelihood of goal achievement (*i.e.,* encouraging program implementers to see the "big picture"). It appears these efforts have been able to illustrate to program implementers the value of this process, and, in some cases, this value has been embraced by them.

The program logic modeling process has helped to identify important program database and field data collection needs (that can be collected directly through program implementation activities) to track key program performance indicators, assess progress toward goals, and show causality. Further, the process has helped guide more effective program evaluation efforts (targeting need-to-know market

research, indicator tracking, etc.). Given limited evaluation budgets, by reviewing and prioritizing logic model-driven research able issues and measurement indicators, the most important items can be identified and focused on to ensure biggest bang for each evaluation buck

NYSERDA plans to continue its logic modeling activities; as this process has evolved over the past four years, it is likely that this evolution will continue and the process will become more effective as its value increasingly becomes apparent to program implementers.

References

Albert, Scott, Victoria S. Engel, Gretchen Jordan, Lori Megdal, Jane Peters. 2004. "Using Program Theory and Logic to Improve Design and Likelihood of Real Market Change: Experience with a State Public Benefits Program." *Proceedings of the 2004 ACEEE Summer Study on Energy Efficiency in Buildings*. Pacific Grove, CA, p. 6-1 to 6-12.

GDS Associates, Inc. 2006. *EmPower New York℠ Logic Model Report*. Prepared for NYSERDA.

NYSERDA. 2006. ***New York Energy $mart℠*** *Program Evaluation and Status Final Report*.

Research Into Action, Inc. 2006 and updated 2007. *Review of the Evaluation Process*. Prepared for NYSERDA.

Figure 2. EmPower New York Logic Model