PERGAMON

Evaluation and Program Planning 24 (2001) 119–128

EVALUATION and PROGRAM PLANNING

www.elsevier.com/locate/evalprogplan

The program logic model as an integrative framework for a multimethod evaluation

Leslie J. Cooksy[a,*], Paige Gill[b], P. Adam Kelly[b]

[a]University of Delaware, Center for Community Development and Family Policy, 297 Graham Hall, Newark, DE 19716-7350, USA
[b]Florida State University, Florida, USA

Accepted 21 September 2000

Abstract

The use of the program logic model as an integrative framework for analysis is illustrated in a multimethod evaluation of Project TEAMS, a middle school curriculum delivery program. The logic model was used to: (a) focus the data collection activities on relevant activities and outcomes, (b) organize the data and (c) interpret the data from multiple methods and sources within an integrative framework. One of the anticipated outcomes, *computer skills*, is examined in detail to illustrate the utility of program logic models as an analysis framework.

Keywords: Logic model; Evaluability assessment; Pattern matching; Triangulation; Data analysis; Multimethod

Logic models depict assumptions about the resources needed to support program activities and produce outputs, and the activities and outputs needed to realize the intended outcomes of a program (United Way of America, 1996; Wholey, 1994). These assumptions are often referred to as program theory (Bickman, 1987, 1990; Weiss, 1997). Recently, program theory was identified as a potential integrative framework for the design and analysis of evaluations using qualitative and quantitative methods (Caracelli & Greene, 1997). By examining data from multiple methods within a single framework, integrated designs are intended to leverage greater insight into a program's operations and effectiveness than component designs in which data collected by different methods and from different sources are not analyzed or interpreted together (Caracelli & Greene, 1997). Despite its potential as an integrative framework, the use of program theory as a framework for mixed-method evaluations is not well-documented (Caracelli & Greene, 1997), and program theory in general 'appears to be having only marginal influence on evaluation practice' (Weiss, 1997, p. 501).

This paper offers an example and discussion of the use of program theory, in the form of a program logic model, as an integrative framework. The example is drawn from an evaluation that used multiple methods, sources, and sites. The first section of the paper provides an overview of program theory and logic models. The second section describes the program from which the illustration is drawn and presents the logic model that guided the evaluation. Then, the use of the logic model as an integrative framework is discussed and illustrated with data on one of the program's desired outcomes and related activities. The paper concludes with a brief summary of the connection between the logic model approach and triangulation and pattern matching.

1. Program theory and program logic models

Ideally, program theory guides an evaluation by identifying key program elements and articulating how these elements are expected to relate to each other. Data collection plans are then made within the framework in order to measure the extent and nature of each element's occurrence. Once collected, the data are analyzed within the framework. First, data that have been collected by different methods or from different sources on the same program element are triangulated (Denzin, 1970; Greene, Caracelli, & Graham, 1989; Mathison, 1988). Second, the pattern of relationships found in the data is compared to the pattern of relationships articulated in the program theory (Marquart, 1990; Scott & Sechrest, 1989; Trochim, 1985, 1989a; Yin, 1994).

Before the term 'program theory' became popular, evaluators were recommending models of evaluation that involved going beyond the simple identification of cause

* Corresponding author. Tel.: +1-302-831-0765; fax: +1-302-831-4225.
E-mail address: ljcooksy@udel.edu (L.J. Cooksy).

0149-7189/01/$ - see front matter
PII: S0149-7189(01)00003-9

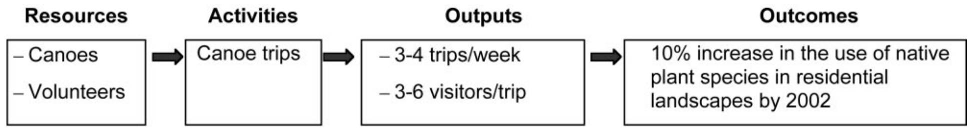

Fig. 1. Logic model for a hypothetical environmental education program.

and effect constructs to the articulation of what we would now call program theory. For example, Stake (1967) presented a model that calls for describing the intended antecedents (whatever needs to be in place before a program is operational), transactions (activities and outputs), and outcomes of a program. Then data on the program in operation are compared to what was intended and to what the standards are for that kind of program. Stufflebeam's (1971) CIPP model is similar to Stake's in its content (CIPP stands for Context–Inputs–Processes–Products) and was designed to encourage a systems approach to evaluation. According to Stufflebeam (1983), the CIPP model does not necessarily lead to the formulation of hypotheses, but it does 'provide a rich array of background data against which to interpret and understand outcomes' (p. 128). Another early proponent of program theory, Weiss (1972) recommended using path diagrams to model the sequence of steps between a program's intervention and the desired outcomes. This kind of causal model helps the evaluator identify the variables to include in the evaluation, discover where in the chain of events the sequence breaks down, and stay attuned to changes in program implementation that may affect the pattern depicted in the model (Weiss, 1972).

Despite this rich tradition of approaches to articulating patterns of relationships, evaluation continued to be dominated by models based on methodological choices instead of on program design (Chen & Rossi, 1980). Because evaluations based on these models tended to provide little evidence of program effectiveness, Chen and Rossi (1980, 1983) began advocating what they called 'theory-driven evaluation'. Chen and Rossi (1980) argued that theory-driven evaluations would be more likely than methods-driven evaluations to discover program effects on the grounds that theory-driven evaluations would identify and examine a larger set of potential program outcomes. The longer list of program outcomes would be theory-based, drawn from existing social science theory and the implicit program models of program stakeholders. Developing the theory-driven approach further, Chen (1990) articulated two major types of theory-driven evaluation. The first, normative evaluation, compares a prescriptive theory of what the program should be to data on the program in operation in order to discover any inconsistencies between the two. In contrast, causative evaluation focuses on the causal relationships underlying a program in order to assess program impact and understand the causal mechanisms associated with program effects.

One of the distinguishing features of theory-driven evaluation is that it explicitly includes a connection to social science theory. However, other writings about program theory have argued that social science theory is generally not relevant to program stakeholders. For example, in Patton's (1997) user-focused approach, the 'evaluator's task is to facilitate intended users, including program personnel, in articulating their operating theory,' also known as the 'espoused theory of action' (p. 221, 223). The espoused theory is then tested by comparison to program reality, the 'theory-in-use'.

A similar approach, evaluability assessment, is based on the theories of program managers about their program (Rutman, 1980; Smith, 1989; Wholey, 1979, 1994). The purpose of evaluability assessment is to discover whether a program is ready to be evaluated so that costly summative evaluations are not conducted prematurely; that is, before there are clear and logical links between a program's resources, activities, and outcomes (Wholey, 1994). An evaluator starts an evaluability assessment by developing a logic model from statements made in program documents and interviews with program managers. The logic model displays these statements in a simple flow chart that outlines the needed resources, intended activities, expected outputs, and desired outcomes.

Once the model is developed, the logic of the linkages is assessed. For example, as can be seen in the logic model of a hypothetical environmental education program displayed in Fig. 1, there are logical connections between the resources, activities, and outputs, but these do not seem to be logically related to the outcome desired by the program. In situations like this, the evaluator can use the program logic model to initiate a dialogue with program stakeholders to clarify the program's processes and goals (Wholey, 1994). Once the assumptions underlying the program seem logical, data on the program in operation are collected in order to learn if the program described in the logic model matches the program in its implementation (Riggin, 1990). If there is a mismatch, the program should not be evaluated for effectiveness until the inconsistencies have been resolved (Wholey, 1994).

Logic models are not rigid in their specifics. While the logic models used in evaluability assessment usually include resources, activities, outputs, and outcomes, logic models can be defined generally as flow charts that display a sequence of logical steps in program implementation and the achievement of desired outcomes. Within this definition, logic models can easily be adapted to display the prescriptive normative theory described by Chen (1990), the antecedents, transactions, and outcomes of Stake's (1967) approach, or Stufflebeam's (1971) context, inputs, processes, and products. As this flexibility suggests, there

is no single correct model for a program. The logic model components can vary, as can the assumptions held by different stakeholders about how a program works (Greene, 1993).

Although logic models have the potential to increase the use of program theory and leverage greater insight from multiple method designs, the approach has its disadvantages. Articulating program theory and developing a logic model takes resources (Bickman, 1989). On the other hand, the cost can be justified by its benefit to program stakeholders above and beyond its use to the evaluator (Patton, 1997). Moreover, in evaluability assessment, logic models are part of a process that is intended to avoid costly evaluations in situations where it is unlikely that program effects will be observed (Wholey, 1994).

Another concern is that a logic model can become a rigid statement of the program's plan and thereby limit the program's responsiveness to new information (Weiss, 1997). Patton's (1997) approach may help limit this unintended side-effect. He describes how developing program theory with program stakeholders can be a useful and educational process for those in the program. Going through this process may help to avoid misuse of the theory by the program operators.

Program operators are not the only ones who might use the logic model as a rigid statement of the program design. Program evaluators may use the logic model inflexibly, assuming that compliance with the model is a measure of the quality of the program and ignoring unintended effects that are not part of the program theory. However, using the evaluability assessment approach to guide the use of logic models can help evaluators use the tool appropriately. In evaluability assessment, the operating theory depicted in the logic model may change based on evidence from the program in operation about available resources or the success of unplanned activities, for example (Wholey, 1994).

Despite its disadvantages, logic models have more potential as an integrative framework than some of the other tools for expressing program theory. The alternatives include path diagrams, program templates, concept maps, and narrative. Path diagrams articulate the sequence of cause–effect relationships that explain how a program intervention yields its effects (Cook & Campbell, 1979; Mark, 1990; Smith, 1990; Weiss, 1972). Program templates are matrices in which the first column identifies key activities of effective programs (identified from research and/or program personnel), the second column lists how a specific program plans to incorporate each activity, and the third column summarizes data on what actually occurred in the program (Loucks-Horsley, 1996; Scheirer, 1996). Concept maps display the items that make up a program variable in a map with distances between items and groups of items indicating their similarity (Trochim, 1989b). Written descriptions of how a program is supposed to operate and the relationship of the program activities to the desired effects are another way of presenting program theory.

Compared to the options, logic models are unique in communicating the relationship of program resources and operations to outcomes in a simple picture. Path diagrams share the simplicity of logic models, but do not include the operational detail that a logic model has. In addition, they usually start with program activities or outputs, rather than with antecedent conditions. Without outlining expected resources and support activities, path diagrams are likely to be less useful than logic models when diagnosing why a program does not have the intended effects. Like logic models and path diagrams, program templates distill detailed descriptions of the assumptions underlying a program into a format that is easy to follow, however they emphasize program activities instead of the connections between resources, activities, and outcomes. Similarly, concept maps tend to be limited to a single step in the sequence of resources, activities, outputs, and outcomes. Finally, textual descriptions can be more complete than charts, diagrams, or matrices, but written presentations of program theory are not consistent in their content and therefore are not useful as a generally recommended framework.

2. The Project TEAMS evaluation and logic model

To illustrate the use of a logic model as an integrative framework, an example is drawn from the evaluation of Project TEAMS. Project TEAMS is a middle school curriculum delivery program intended to integrate active learning strategies, computer access, and interdisciplinary instruction into regular classroom activities (Reiser & Butzin, 1998). TEAMS classrooms are organized into stations where small groups of students work on individual work at computers, collaborative hands-on tasks at exploration stations, or writing tasks at text stations, among other kinds of activities. In a 1–3 week unit on a topic, students rotate through all the stations. Stations are intended to: (a) engage students by presenting material in a variety of ways, (b) encourage teamwork by requiring collaborative projects, and (c) ensure regular and equal access to computers in the classroom. In addition, teams of four teachers covering the subjects of language arts, mathematics, science and social studies are supposed to work together to make interdisciplinary connections to increase the integration and relevance of the content presented to middle school students.

Project TEAMS was developed for sixth grade, the first year of middle school, with plans to expand to the seventh and eighth grades. For 2 years, the TEAMS developers worked with teachers at a pilot site to develop, test, and refine classroom materials and a teacher's manual. After 2 years of formative evaluation (1993–95), program developers decided that TEAMS was ready to be evaluated for its effects on anticipated outcomes ('Project TEAMS Formative Evaluation Report,' n.d.; Atkins, 1995). The primary purpose of the 1995–96 summative evaluation was to communicate progress to the foundation that had funded

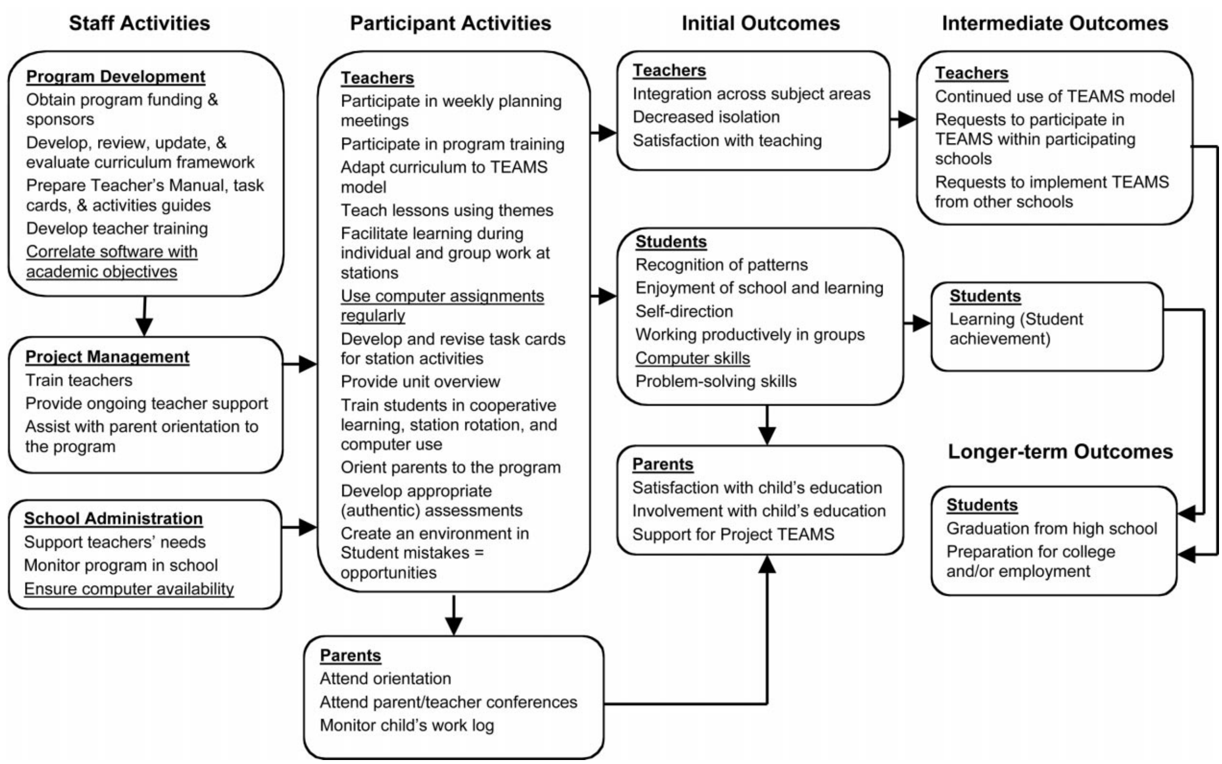

Fig. 2. Project TEAMS logic model. (Underlined items indicate the variables discussed in this paper).

the program's development. During this year, the new evaluation team emphasized the accountability purpose by collecting data primarily on outcomes of interest to the program developers.

At the beginning of the evaluation, the program developers provided us with a variety of documents, including the teacher's manual and previous formative evaluations. We used these documents to identify an initial set of program activities and intended outcomes. Organizing these in a preliminary logic model, we then used the model in interviews with the program developers and coordinators to clarify the intended outcomes of TEAMS and the resources and activities expected to achieve these outcomes. The interviews led to further refinements of the TEAMS logic model. While recognizing that this preliminary model needed further development, we used it to guide the data collection in the first year of the evaluation.

We also continued to refine the logic model during 1995–96. The next major step in the refinement process was a sort-and-rate task completed by the two program developers and the program coordinator. The task used a list of items that we developed from a variety of sources, including program documents, comments on our draft survey instruments, responses to open-ended questions on the 1995–96 surveys, and literature related to potential TEAMS' outcomes (e.g., Grow, 1991, on self-direction; Smith, Young, West, Morgan, & Rhode, 1988, on off-task behavior). The specific instructions given for the sort-and-rate task followed recommendations by Trochim (1989b). Specifically, the program developers and program coordinator were asked to: (a) sort the items into piles 'in a way that makes sense to you', (b) identify the concept or idea underlying the items in the pile and (c) rate each item on a 5-point scale of importance in an evaluation of TEAMS. Using the results of the sort-and-rate task and inductive analyses of the qualitative data from the evaluation, we grouped outcomes into general categories for a final version of the program logic model, shown in Fig. 2.

The theory of how Project TEAMS is supposed to operate, expressed in the form of the logic model found in Fig. 2, was discussed with and approved by the program developers. Where possible, the model used the terms that the program developers used. As a result, instead of a resources or inputs column, there were 'staff activities,' some of which were focused on obtaining resources, such as program funding and computers. Similarly, outputs were not included in the model because the program developers wanted the program to be somewhat flexible within the basic elements of stations, rotations, and teacher teams, and therefore did not identify specific frequencies with which any of the activities would occur. (However, in the construction of the data collection instruments, the program developers stated an expectation that students would have access to computers on a weekly basis.) The 1996–97 evaluation focused on the initial outcomes for students and the intermediate outcome, learning.

At the end of the 1995–96 school year, the evaluation had not found strong evidence of program effects on the initial

Table 1
Features of Project TEAMS and the evaluation at the three schools

Feature	School		
	FMS	SMS	TMS
When was TEAMS first implemented?	1994	1995	1996
Were the teachers organized into teams of four?[a]	No	Yes	Yes
What proportion of 6th graders in the school participated in TEAMS?	60%[b]	100%	50%
What data sources and data collection methods were used?	→ Teacher survey → Student survey → Parent survey → Admin. data → Student case study[c]	→ Teacher survey → Student survey → Parent survey	→ Teacher survey → Student survey → Parent survey → Admin. data

[a] A teacher team consisted of the language arts, math, science and social studies teachers.
[b] Since FMS students were not assigned to teacher teams, this number represents the proportion of students taking at least three of the four core courses with TEAMS trained teachers.
[c] The case study consisted of observations of a student in school and interviews with the student, her mother, and her teachers.

outcomes. Because of concerns about how the program was being implemented, the program developers asked us to look at implementation as well as outcomes in the second year of the evaluation. At this point, the evaluation became more like an evaluability assessment, focusing as much on the feasibility of achieving desired outcomes based on program operations as it did on the outcomes themselves. The Project TEAMS logic model served as the framework for the collection and analysis of data on the implementation and outcomes of TEAMS in the second year of the evaluation.

The data from the second year are used in the illustration below. In contrast to the first year when Project TEAMS was evaluated at only one school, the second year looked at three schools implementing Project TEAMS. These schools are referred to here as First Middle School (FMS), which was the pilot site for TEAMS, Second Middle School (SMS), and Third Middle School (TMS). The differences in how the three schools organized the program in 1996–97 are described in Table 1.

The multiple schools and multiple data sources imply a strong design because patterns of outcomes based on differences in school implementation can be predicted and the consistency of a finding across data sources can be examined (Davis, 1989; Denzin, 1970; Trochim, 1989a). However, the Project TEAMS evaluation had several weaknesses. First, only initial outcomes were examined with multiple data sources and methods. Resources and activities each depended on a single source. Second, the bases for attributing observed outcomes to Project TEAMS were weak and varied across the three schools. (This is discussed further below.) Third, the parent samples were limited to the select groups whose children both delivered the survey to the parents and returned the survey to school (around 36% of the FMS parents and 37% of the SMS parents). (See Cooksy, 1999, for more information on the methodology.) Finally, because so many outcomes were being addressed and the evaluation resources were scarce, none were measured with much sensitivity. The integrating framework of the logic model does not overcome these weaknesses. In fact, as the illustration will demonstrate, it makes the limitations of the data clear by organizing analysis and interpretation around the program elements instead of by data source or by site. As a result, systematic biases are more likely to be revealed and considered in evaluation findings.

3. The use of the logic model as an integrative framework

As an integrative framework for a multimethod evaluation, a logic model offers two ways to look at the data. First, evidence from different data sources and collection methods are organized by program element rather than by source or method, so that the consistency of findings from different data sources and methods can be examined. In this case, the logic model simply facilitates the process of triangulation (Cambell & Fiske, 1959; Denzin, 1970). However, as Greene et al. (1989) discovered in their review of mixed method evaluations, studies that give triangulation as a reason for using different methods do not always follow through by actually examining the results for consistency. Logic models may therefore provide a needed framework for considering the nature and extent of convergence of different kinds of data on a single program element.

The second way that the logic model integrates data collection and analysis is by defining each program element in relation to its antecedents and consequences. As a result, evidence about a program element is interpreted not in isolation, but in light of its expected relationships. These

relationships indicate a pattern in which, for example, more complete implementation would be associated with stronger evidence of initial outcomes, stronger evidence of initial outcomes would be associated with stronger evidence of intermediate outcomes, and so on. In addition to the first two ways that logic models frame the data, in this evaluation, the logic model also enriched the interpretation of the data by elucidating patterns across the three schools.

The following discussion reviews these three roles of the logic model in the context of the evaluation of Project TEAMS, using the initial outcome of *computer skills* as an illustration. As the model in Fig. 2 indicates, *computer skills* are one of the links between program activities, such as regular access to computers, and the intermediate outcome of increased student learning. Using the logic model format, Fig. 3 displays the data for each of the program elements related to computers. The contents of the figure are the foundation for the following discussion.

3.1. Program resources: Computers and software

Although program resources were not an explicit part of the evaluation, observations at FMS and SMS and reports from the program coordinator about TMS provided evidence that each TEAMS classroom had computers. In addition, the case study of an FMS student offered some clues about resource issues related to the outcome of *computer skills*. On the one hand, there were complaints about the relevance of the software. As Sarrao's (1996, pp. 9–10) case report (using pseudonyms) stated: "Several teachers expressed concern with the software that was available. For example, Mr. Brown indicated that he had little software dealing with his subject, which is why his students sometimes work on language arts skills during his class. Additionally, another teacher told me that she had no software for the current unit, so the students were not using the computers for this period. A third teacher expressed concern about the old computers in her room which require floppy discs, making it difficult for her to acquire relevant software".

On the other hand, the case study (Sarrao, 1996, p. 9) also described student computer activities that were appropriate to the subject area: "Students were using a computer database to obtain facts for a report they were writing in geography, and in language arts students were playing a game involving nouns. Additionally, Kelly [the student] told me that she uses the computer to get facts, to read stories and answer questions, and sometimes to play games, if she finishes her work early…. For example, Mrs. Green described a budgeting activity that the students did in mathematics and Mr. Brown described a geography activity that involved students designing, taking, and mapping a trip".

The case report identified the kinds of activities that the students use computers for, but did not provide any information about the frequency with which teachers incorporated or students had access to computers.

3.2. Program activities and outputs: Computer use by teachers and students

Although we learned very little about program resources, the knowledge that computers were available was enough to justify continuing the investigation into TEAMS activities related to *computer skills*. The Project TEAMS logic model (Fig. 2) shows that improved *computer skills* depended first on teachers' incorporation of computers into class assignments and second on students' access to computers. The sources of evidence about these two activities were surveys of the teachers and students. As recorded in Fig. 3, compared to the FMS teachers, teachers at SMS and TMS reported slightly more frequent use of computers in general and of computer activities that specifically supported the lesson objectives. The TMS teachers had the least varied range of responses, indicating that TMS TEAMS students had the opportunity to use computers more consistently across teachers than the students at SMS and FMS. Given both the lower average response and the variability of the reports on regular use, FMS appeared to have a weaker implementation of the computer element of TEAMS relative to SMS and TMS. We therefore expected to find fewer students reporting regular use of computers at FMS than at SMS or TMS.

We measured the output of the teachers' use of computers by asking students if they had used computers on a regular basis in the classes, with response options of 'yes' or 'no'. The results from this item are presented in Fig. 3 in the boxes labeled *Output—Student Access*. The proportion of students stating that they used computers on a weekly basis for class work in 6th grade was small—less than one-third—in all three schools. Moreover, there was no meaningful match between the pattern of results anticipated from the teachers' reports of computer use and the actual access reported by the students. Specifically, although the FMS teachers appeared to have used the computers less often and less consistently than the SMS and TMS teachers, the proportion of FMS students reporting regular use of the computer was similar to those reporting regular use at SMS and TMS.

The other pattern examined was the relationship between regular computer use in TEAMS and computer use in non-TEAMS conditions. Since TEAMS expected to increase computer access, we expected to see larger proportions of TEAMS students reporting that they had used computers than in the results from the non-TEAMS data. Data on non-TEAMS conditions were collected in two ways, depending on the school. A pre-test, post-test approach was used at FMS and SMS, with a survey administered at the beginning of 6th grade to collect information on the 5th grade experiences of the TEAMS students and then repeated at the end of 6th grade with the questions adjusted to 6th grade experiences. As a new implementor of TEAMS, TMS wanted limited involvement in the evaluation so the survey was not administered at the beginning of the school year.

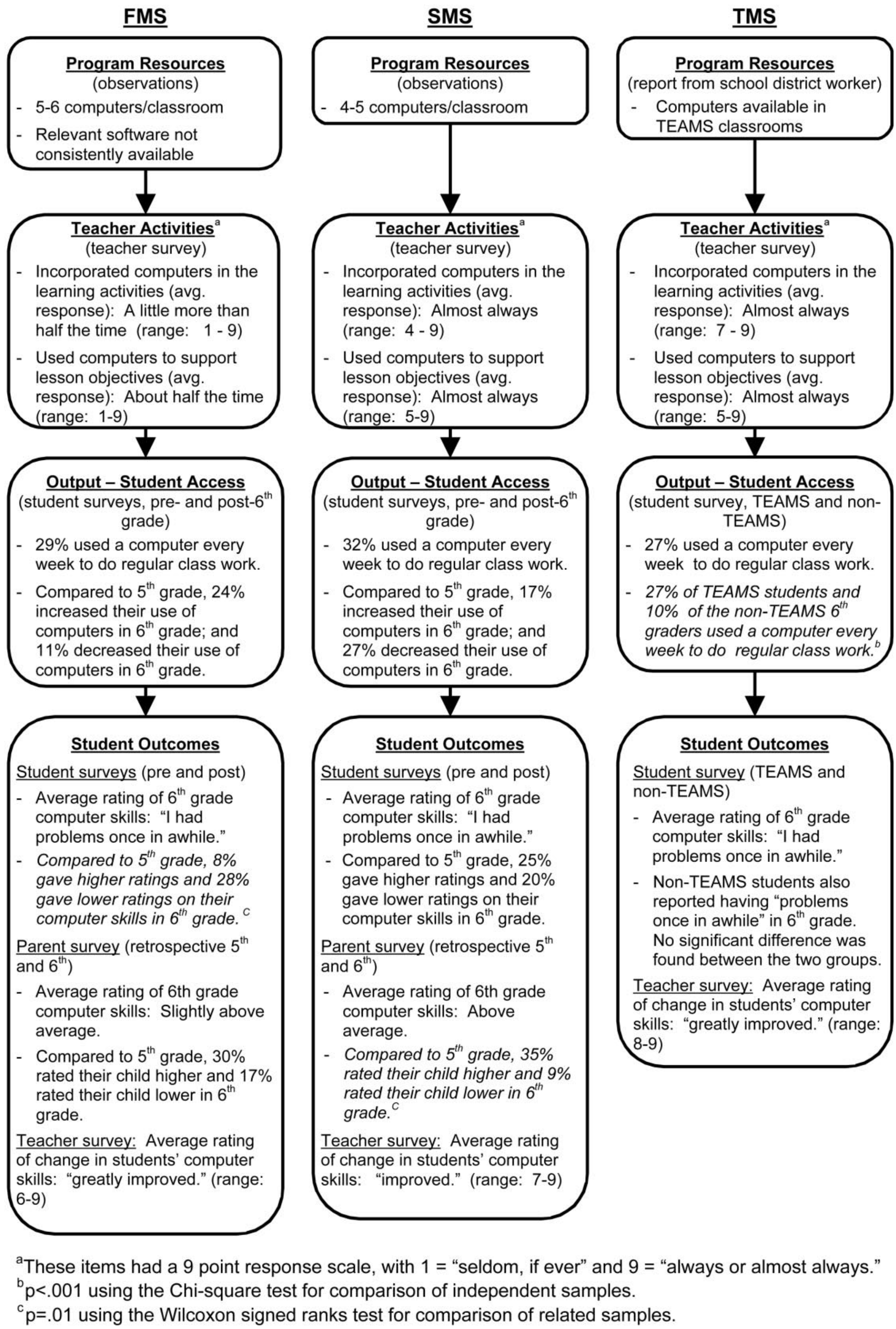

Fig. 3. Logic model display of results related to computer skills (italics indicate statistically significant differences).

Instead, both TEAMS and non-TEAMS students responded to the survey at the end of the 6th grade school year.

McNemar tests on the pre- and post-6th grade data in FMS and SMS, and the χ^2-test on the program and non-program groups at TMS yielded only one statistically significant result. As can be seen in Fig. 3, only TMS showed TEAMS students with a significant and positive difference in access to computers relative to the non-TEAMS students. This result is consistent with the TMS teacher reports of using computers and could be the result of the guidance provided about using computers for station activities in the TEAMS training and teachers manual. An alternative explanation is that TEAMS teachers received a disproportionate amount of the school's computer resources, thereby limiting the opportunities for students in the non-TEAMS classes to use computers in their class assignments.

Although the difference was not statistically significant, the high proportion of SMS students reporting decreased access to computers in 6th grade relative to 5th grade warranted attention since it was the opposite of what would be expected if Project TEAMS had been implemented as planned. We discovered that many of the SMS students had used computers in elementary school through a precursor to Project TEAMS, Project CHILD. While not necessarily indicating a failure of Project TEAMS, this result suggests that the TEAMS implementation at SMS was not as effective at that of Project CHILD in the feeder elementary schools.

Summing up the data to this point, the evidence on resources, activities, and outputs were not strong, yet there was some consistency between the anticipated patterns and the findings. In particular, the TMS teachers had computers, and they consistently reported frequent use of the computers in the learning activities and to support lesson objectives. In line with this evidence, TEAMS students were more likely to report regular use of computers than the non-TEAMS students at TMS were. If this pattern holds out, then we would expect to find stronger evidence of increased skills at TMS than at the other two schools. In contrast, the FMS teachers indicated less frequent and less consistent use of the computers than at the other two schools, and the proportion of students reporting increased access to computers was larger than but not significantly different from the proportion reporting decreased access. Thus, we expected any evidence of a TEAMS effect on computer skills at FMS to be weak. For SMS, the elementary school program that emphasized access to computers confounded the expected link between implementation of the teacher activities and student access to computers, making it difficult to anticipate how the computer skills of the SMS students might change.

3.3. Initial outcome: Computer skills

This discussion summarizes the data on *computer skills* for each school, looking across the different data sources and what was expected based on the information about teacher implementation and student access. At FMS, with little evidence of a strong implementation of the computer component of TEAMS or of improved access to computers by the students, we found inconsistent results across the data sources on computer skills. While there was a slight positive trend in the reports of the parents, a statistically significant proportion of the FMS TEAMS students reported that their computer skills had decreased from 5th to 6th grades. (See Fig. 3 for specifics).

Three explanations of the divergence were considered. First, either the data collection instruments themselves or the data sources were biased to such a degree that we had measured something quite different from what we wanted to measure. Of these two, the different perspectives of the data sources was a more plausible explanation than the data collection instruments since the survey instruments were similar enough in format to make it doubtful that they would yield opposite results. The other possible explanation was that the 24% of students who were getting more access to computers than they had in 5th grade had developed a greater awareness of what they did not know, and thus were less optimistic about their skills at the end of the year. The difficulties of the transition from elementary to middle school could also have contributed to the lower student assessments. In any case, it was inappropriate to draw any conclusion about changes in the students' *computer skills* based on the inconsistent data.

At SMS, the difference in parent and student results was less dramatic than at FMS. A statistically significant proportion of parents gave their child higher ratings on 6th grade computer skills than they had on 5th grade computer skills. The student data were neither positive nor negative, with similar proportions reporting improved and decreased computer skills in 6th grade compared to 5th grade. The teacher responses about changes in computer skills were positive, but not as positive as the FMS and TMS teacher responses. The results from the different sources at SMS did not converge on a single conclusion about changes in the students' computer skills. As with FMS, the lack of convergence could have been the result of the different perspectives of the parents and students. Alternatively, any effect of TEAMS could have been overshadowed by the effects of Project CHILD, the elementary school program that used a similar approach as Project TEAMS, or the transition to middle school.

If there was a TEAMS effect on *computer skills*, it was most likely to be found at TMS. At TMS, a larger proportion of the TEAMS students than of the non-TEAMS students had reported regular access to computers. TMS was also considered a better test of any effects on *computer skills* because, according to the TEAMS coordinator, the students who were assigned to TEAMS had not been selected on any special criteria, so the TEAMS and non-TEAMS groups were likely to be pretty similar. For example, both groups experienced the transition from elementary school to middle school during the year of the evaluation, so that

the transition itself is not a plausible explanation for any differences observed, as it is for the FMS and SMS groups. So, although still a weak design, the TMS comparison was the best we had for the examination of the outcome of computer skills. The result of this comparison was that the difference in the computer skills reported by the two groups was negligible. Both TEAMS and non-TEAMS groups said that, on average, they 'had problems once in awhile' when using the computer. Not surprisingly, no significant difference was found between the group averages. TMS TEAMS teachers were positive about their students' computer skills. Unfortunately, when teacher responses were looked at across the full set of initial outcomes, we found that the teachers were positive about the achievement of all of the initial outcomes, making their responses to any single item suspect. In short, no TEAMS effect on *computer skills* was found at TMS.

3.4. Discussion

It could be argued that the limitations of the evaluation used here were so great that it can not accurately illustrate the value of logic models to integrate the analysis and interpretation of evaluation data. However, when used as an integrative framework, the logic model forces the evaluator to look at patterns in the data across sources and in relation to the intended sequence of events. As a result, inconsistencies across data sources or from one logical link to the next are highlighted, not downplayed as they can be when findings are reported by method. For example, this evaluation could have reported that teachers gave high ratings to the children's computer skills. In isolation, this information casts a positive light on the program. But the logic model exposes the problems with such claims by showing that: (a) the teacher assessments were different from student assessments of their skills, (b) the positive reports were not logical at FMS and SMS where there were questions about implementation, and (c) the teachers were positive about every outcome. In other words, the logic model encourages a complete assessment of the value of the evidence by providing a context within which to interpret the data.

In the TEAMS evaluation, juxtaposing the results from the different sources highlighted the inconsistent reports about *computer skills* from the students, teachers, and parents within and across the schools. Similarly, examining the outcome data in light of what the data on program implementation predicted revealed a mismatch between the expected pattern and the data. In combination with the lack of convergence across the data sources, the inability to match the pattern of outcomes relative to the schools' implementation reinforced our conclusion that a TEAMS effect on computer skills had not been observed. However, the evaluation, in keeping with its evaluability assessment approach, identified concerns in program operations for program improvement. For example, both the relevance of the computer software and students' regular access to computers were likely barriers to improving computer skills. Another concern was that of the three schools, FMS, the school with the longest implementation of TEAMS, appeared to have the weakest implementation. This was not only found in the data on computer activities, but also in the administrative information about the dissolution of the teacher teams which kept the program from being delivered at the same intensity as at the other two schools (Table 1). In short, although the evaluation did not produce definitive information about program outcomes, it served a formative purpose by describing where and how program operations were not in line with the program theory.

4. Conclusion

This paper has argued that program theory can be a useful integrative framework for evaluations using multiple methods. As the example illustrates, its value lies in facilitating triangulation and pattern matching, both ideas with a long history in our profession (Campbell & Fiske, 1959; Cronbach & Meehl, 1955) but continued need for reinforcement (Caracelli & Greene, 1997; Greene et al., 1989; Weiss, 1998). Logic models in particular assist these analyses by identifying program elements for which data collected from different methods or sources can be triangulated and enriching the anticipated pattern of results across the sequence of program operations and outcomes.

Acknowledgements

The authors wish to acknowledge Dr. Robert A. Reiser for providing the opportunity to conduct the evaluation and for his comments on draft reports issuing from the evaluation work. An earlier version of this paper was presented at the 1998 Annual Meeting of the American Educational Research Association, 13–17 April, 1998, San Diego, California. This version is substantially improved in response to the thoughtful comments of anonymous reviewers.

References

Atkins, D. (1995). Project TEAMS final evaluation report. Unpublished manuscript.

Bickman, L. (1987). *Using program theory in evaluation. New directions for program evaluation, 33*, San Francisco: Jossey-Bass.

Bickman, L. (1989). Barriers to the use of program theory. *Evaluation and Program Planning*, *12*, 387–390.

Bickman, L. (1990). *Advances in program theory. New directions for program evaluation, 47*, San Francisco: Jossey-Bass.

Campbell, D. T., & Fiske, D. W. (1959). Convergent and discriminant validation by the multitrait–multimethod matrix. *Psychological Bulletin*, *56*, 81–105.

Caracelli, V. J., & Greene, J. C. (1997). Crafting mixed-method evaluation designs. In J. C. Greene & V. J. Caracelli, *Advances in mixed-method evaluation: The challenges and benefits of integrating diverse paradigms. New directions for program evaluation* (pp. 19–32). Vol. 74. San Francisco: Jossey-Bass.

Chen, H-T. (1990). *Theory-driven evaluations*, Thousand Oaks, CA: Sage.
Chen, H. T., & Rossi, P. H. (1980). The multi-goal, theory-driven approach to evaluation: A model linking basic and applied social science. *Social Forces*, *59*, 106–122.
Chen, H-T., & Rossi, P. W. (1983). Evaluating with sense: The theory-driven approach. *Evaluation Review*, *7*, 283–302.
Cook, T. D., & Campbell, D. T. (1979). *Quasi-experimentation*, Boston: Houghton Mifflin.
Cooksy, L. J. (1999). The meta-evaluand: The evaluation of Project TEAMS. *American Journal of Evaluation*, *20* (1), 123–136.
Cronbach, L. J., & Meehl, P. E. (1955). Construct validity in psychological tests. *Psychological Bulletin*, *52*, 281–302.
Davis, J. E. (1989). Construct validity in measurement: A pattern matching approach. In W. Trochim (Ed.), A special issue of Evaluation and Program Planning, 12(1), 31–36.
Denzin, N. K. (1970). *Sociological methods: A sourcebook*, Chicago: Aldine.
Greene, J. C. (1993). The role of theory in qualitative program evaluation. In D. J. Flinders & G. E. Mills, *Theory and concepts in qualitative research: Perspectives from the field* (pp. 24–45). New York: Teachers College Press.
Greene, J. C., Caracelli, V. J., & Graham, W. F. (1989). Toward a conceptual framework for mixed-method evaluation designs. *Educational Evaluation and Policy Analysis*, *11*, 255–274.
Grow, G. O. (1991). Teaching learners to be self-directed. *Adult Education Quarterly*, *41* (3), 125–149.
Loucks-Horsley, S. (1996). The design of templates as tools for formative evaluation. In M. A. Scheirer, *A user's guide to program templates: A new tool for evaluating program content. New Directions for Evaluation* (pp. 5–24). Vol. 72. San Francisco: Jossey-Bass.
Mark, M. M. (1990). From program theory to tests of program theory. In L. Bickman, *Advances in program theory. New Directions for Program Evaluation* (pp. 37–51). Vol. 47. San Francisco: Jossey-Bass.
Marquart, J. M. (1990). A pattern-matching approach to link program theory and evaluation data. In L. Bickman, *Advances in program theory. New directions for program evaluation* (pp. 93–107). Vol. 47. San Francisco: Jossey-Bass.
Mathison, S. (1988). Why triangulate? *Educational Researcher*, *17* (2), 13–17.
Patton, M. Q. (1997). *Utilization-focused evaluation: The new century text*, Thousand Oaks, CA: Sage.
Project TEAMS formative evaluation report. n.d. Unpublished manuscript.
Reiser, R. A., & Butzin, S. M. (1998). Project TEAMS: Integrating technology into Middle School instruction. *Tech Trends*, *43* (2), 39–44.
Riggin, L. C. (1990). Program theory and social science theory. In L. Bickman, *Advances in program theory. New directions for program evaluation* (pp. 109–120). Vol. 47. San Francisco: Jossey-Bass.
Rutman, L. (1980). *Planning useful evaluations: Evaluability assessment*, Thousand Oaks, CA: Sage.
Sarrao, M. (1996). Project TEAMS case study: One student's experience. Unpublished document.
Scheirer, M. A. (1996). A role for program templates in evaluation: A broader look. In M. A. Scheirer, *A user's guide to program templates: A new tool for evaluating program content. New directions for evaluation* (pp. 89–98). San Francisco: Jossey-Bass.
Scott, A. G., & Sechrest, L. (1989). Strength of theory and theory of strength. *Evaluation and Program Planning*, *12*, 329–336.
Smith, D. J., Young, K. R., West, R. P., Morgan, D. P., & Rhode, G. (1988). Reducing the disruptive behavior of junior high school students: A classroom self-management procedure. *Behavioral Disorders*, *13*, 231–239.
Smith, M. F. (1989). *Evaluability assessment: A practical approach*, Boston, MA: Kluwer-Nijhoff.
Smith, N. L. (1990). Using path analysis to develop and evaluate program theory and impact. In L. Bickman, *Advances in program theory. New directions for program evaluation* (pp. 53–60). Vol. 47. San Francisco: Jossey-Bass.
Stake, R. E. (1967). Countenance of educational evaluation. *Teachers College Record*, *68*, 523–540.
Stufflebeam, D. L. (1971). The relevance of the CIPP evaluation model for educational accountability. *Journal of Research and Development in Education*, *5*, 19–25.
Stufflebeam, D. L. (1983). The CIPP model for program evaluation. In G. F. Madaus, M. S. Scriven & D. L. Stufflebeam, *Evaluation models: Viewpoints on educational and human services evaluation* (pp. 117–141). Hingham, MA: Kluwer.
Trochim, W. M. K. (1985). Pattern matching, validity, and conceptualization in program evaluation. *Evaluation Review*, *9*, 575–604.
Trochim, W. M. K. (1989a). Outcome pattern matching and program theory. *Evaluation and Program Planning*, *12*, 355–366.
Trochim, W. M. K. (1989b). An introduction to concept mapping for planning and evaluation. In W. Trochim, *A special issue of evaluation and program planning* (pp. 1–16). , Vol. 12.
United Way of America. (1996). Measuring program outcomes: A practical approach (Item No. 0989). Author.
Weiss, C. H. (1972). *Evaluation research: Methods of assessing program effectiveness*, Englewood Cliffs, NJ: Prentice-Hall.
Weiss, C. H. (1997). How can theory-based evaluation make greater headway? *Evaluation Review*, *21* (4), 501–524.
Weiss, C. H. (1998). *Evaluation* (2nd ed). Upper Saddle River, NJ: Prentice-Hall.
Wholey, J. S. (1994). Assessing the feasibility and likely usefulness of evaluation. In J. S. Wholey, H. P. Hatry & K. E. Newcomer, *Handbook of practical program evaluation* (pp. 15–39). San Francisco: Jossey-Bass.
Wholey, J. S. (1979). *Evaluation: Promise and performance*, Washington, DC: Urban Institute.
Yin, Robert E. (1994). Case study research, Applied social research methods series (2nd ed), Vol. 5. California: Newbury Park.