METHOD GUIDE

How We Model Matters — Visualizing Program Theories

Sebastian Lemire, Allan Porowski, and Kaity Mumma

March 2023

Abt Global

Abt Global is a global consulting and research firm that combines data and bold thinking to improve the quality of people's lives. We partner with clients and communities to advance equity and innovation—from creating scalable digital solutions and combatting infectious disease, to mitigating climate change and evaluating programs for measurable social impact.

Acknowledgments

We gratefully acknowledge the contributions of many Abt colleagues in developing and producing this method guide. We especially appreciate the thought leadership of Laura Peck who offered thoughtful feedback and guidance throughout the writing process to improve every aspect of the guide.

We are also fortunate to have received generous and constructive feedback from our expert reviewers: Ulla Keech-Marx, Fiona MacTaggart, and Betsy Ness-Edelstein. Their thoughtful comments on multiple revisions of the guide were invaluable.

Finally, the report would not have been possible without the contributions of many other staff at Abt Global. We thank Eric Tolbert and Puneet Kaur for their stunning visuals and Bry Pollack for her thorough editing. Any remaining errors and omissions belong to the authors.

Preferred Citation

Although permission to use and reprint this publication is not necessary, it should be cited as:

> Lemire, S., Porowski, A., & Mumma, K. (2023). *How We Model Matters: Visualizing Program Theories*. Rockville, MD: Abt Global.

Biographies

Sebastian Lemire, PhD, is a Senior Associate at Abt Global and an award-winning author of more than 40 peer-reviewed publications on theory-based evaluation, evaluation capacity building, and evidence reviews. He currently serves as a Board Member of the American Evaluation Association and on the Editorial Advisory Board of Evaluation.

Allan Porowski is a Principal Associate at Abt Global. He has over 25 years of experience in working with clients to create program theories, develop measurement plans based on program theories, and evaluate programs to validate program theories. He has worked with a range of federal clients on program theory development, including the Office of National Drug Control Policy, U.S. Department of Education, U.S. Department of Labor, U.S. Department of Health and Human Services, and AmeriCorps, as well as nonprofit organizations such as Communities In Schools, the Boys and Girls Clubs of America, and the AARP Foundation.

Kaity Mumma, PhD, is a Senior Analyst at Abt Global. She has seven years of experience contributing to research design, data analysis and reporting, and program evaluation, primarily in the field of early childhood education. She is passionate about improving the U.S. education system through evidence-based practices and empirical research.

TABLE OF CONTENTS

Introduction 1

1. Logic Model 4

2. Logical Framework 8

3. Theory of Change 11

4. Context-Mechanism-Outcome Configuration 16

5. Causal Loop Diagram 20

6. Stock and Flow Diagram 25

7. Concept Map 30

8. Network Map 34

9. Path Model 38

10. Nested and Hybrid Models 41

Practical Principles for Designing Program Theories 50

Appendix 60

Glossary 61

References 64

Endnotes 68

EXHIBITS

Exhibit 1. Different Types of Program Theories 3
Exhibit 2. Logic Model—Teaching Assistant Program 5
Exhibit 3. Logic Model with Explicit Assumptions and External Factors 6
Exhibit 4. Logic Model with Clearly Articulated Problem Statement and Program Goal 7
Exhibit 5. Logframe Matrix—Teaching Assistant Program 8
Exhibit 6. Logframe Using Numbering to Make Connections among Components 10
Exhibit 7. Theory of Change—Teaching Assistant Program 12
Exhibit 8. Theory of Change with Color-Coded Causal Links 14
Exhibit 9. Theory of Change with Color-Coded Direct Outcomes and Targets 15
Exhibit 10. Context-Mechanism-Outcome Configuration—Teaching Assistant Program 16
Exhibit 11. CMO Configuration with Positive and Negative Context Factors 18
Exhibit 12. CMO Configuration with Outcome Ratings 19
Exhibit 13. Causal Loop Diagram—Teaching Assistant Program 21
Exhibit 14. Causal Loop Diagram with Thin and Thick Line Arrows for Short-Term and Medium-Term Outcomes 22
Exhibit 15. Causal Loop Diagram with Face Icons for Facilitating and Inhibiting Feedback Loops 23
Exhibit 16. Causal Loop Diagram with Leverage Points for Program Improvement 24
Exhibit 17. Stock and Flow Diagram—Teaching Assistant Program 26
Exhibit 18. Stock and Flow Diagram with Markers for Positive and Negative Processes 27
Exhibit 19. Stock and Flow Diagram with Both Unidirectional and Bi-Flow Arrows 28
Exhibit 20. Stock and Flow Diagram With External In and Out Flow 29
Exhibit 21. Concept Map—Teaching Assistant Program 30
Exhibit 22. Concept Map with Participant Statements about Program Success 32
Exhibit 23. Concept Map Tracking Participants' Thinking on Obesity Over Time 33
Exhibit 24. Network Map—Teaching Assistant Program 35
Exhibit 25. Network Map with Color-Coded Subthemes 36
Exhibit 26. Network Map with Color-Coded Lines for Degree of Collaboration 37
Exhibit 27. Path Model—Teaching Assistant Program 39
Exhibit 28. Path Model with Color-Coded Lines for Statistically Significant Connections 39
Exhibit 29. Path Model with Color-Coded Pathways 40
Exhibit 30. Nested Model—Teaching Assistant Program 41
Exhibit 31. Hybrid Causal Loop/Stock and Flow Diagram—Teaching Assistant Program 42
Exhibit 32. Nested Theories of Change 44
Exhibit 33. Hybrid Causal Loop + Stock and Flow Diagram 45
Exhibit 34. Hybrid CMO + Causal Loop Diagram 46

Exhibit 35. Hybrid Concept Map + Network Map 47
Exhibit 36. Hybrid Theory of Change + Causal Loops 48
Exhibit 37. Hybrid Context-Mechanism-Outcome Configuration + Path Model 49
Exhibit 38. Decision Tree for Selecting a Program Theory Type 51
Exhibit 39. Simple, Complicated, and Complex Programs 53
Exhibit 40. Dimensions of Complexity 54
Exhibit 41. Outcome Trajectories 55
Exhibit 42. Types of Assumptions 57
Exhibit A1. Software for Developing Program Theory 60

INTRODUCTION

Program theories are widely used in monitoring and evaluation. In basic terms, a *program theory* is a description of the way in which a program is intended to bring about a desired set of outcomes—a "plausible and sensible model of how a program is supposed to work" (Bickman, 1987, p. 5). Program theories can be useful for all phases of the program cycle—from initial design and planning to continuous monitoring, learning, adaptation, and evaluation. Across all these phases, program theories facilitate a shared understanding of how a program is intended to work. This shared understanding potentially benefits evaluators and commissioners of evaluation, program funders and staff, as well as the communities and partners with whom the program is involved.

Over the years, evaluators have developed a wide range of techniques and methods for developing and articulating program theories, substantively expanding our program theory toolbox and practice. In theory-based evaluation especially, innovative analytical approaches and techniques continue to emerge and gain prominence (Lemire, Whynot, & Montague, 2019). A broad—and still broadening—range of types of program theories have emerged. For novice—and perhaps even experienced—evaluators, the broadening array of techniques and methods, not to mention the dizzying array of corresponding terminology and visual models, can invoke a mixed sense of methodological promise and peril, opportunity and apprehension. What is a "causal loop diagram"? And how is it different from a "stock and flow diagram"? And what do evaluators mean by a "nested theory of change"?

The motivation for this guide is to promote awareness and a better understanding of different types of program theories. Towards this aim, the guide describes and visually illustrates ten different types of program theories, ranging from more common *logic models* and *theories of change* to less common *causal loop diagrams* and *hybrid models*. The focus of the guide is intentionally on the visual models and strategies illustrated by these different types of program theories. As such, the guide does not describe how to develop or test program theories as part of evaluations, a topic we will cover in a separate forthcoming method guide. Our modest hope is that the present guide will serve to inspire evaluators and other developers and users of program theories to pursue (or at least consider) a broader range of program theories in their future work.

Terminology

As mentioned above, a broad range of terms and labels for what we are terming a *program theory* have emerged in evaluation circles, including "logic model" (Cooksy, Gill, & Kelly, 2001), "program logic" (Funnel, 1997), "implementation theory" (Scheirer, 1987), "theory of change" (Mayne, 2015), and "context-mechanism-outcome configuration" (Pawson & Tilley, 1997), among others. Though important distinctions exist among these labels, their specific uses and interpretation have not been consistent in practice, often reflecting personal preferences, training, source references, or even just plain old habits.

For the purposes of this guide, we use *program* as a generic term to describe any type of program, policy, project, practice, product, or process that is a target of evaluative inquiry. By extension, we use *program theory* as an umbrella term for any type of visual model of the underlying logic of a program.

Audience

The primary audience for this guide is program funders, designers, managers, evaluators, and other practitioners currently engaging with program theories as part of their work. The guide will be of relevance to both new and more experienced practitioners with an interest in broadening their repertoire of program theories. In the presentation of the different types of program theories, we have endeavored to make any complex and technical aspects of the modeling techniques accessible to a broad range of practitioners.

Structure

The guide is structured around brief descriptions of ten different types of program theories:

1. Logic model
2. Logical framework
3. Theory of change
4. Context-mechanism-outcome configuration
5. Causal loop diagram
6. Stock and flow diagram
7. Concept map
8. Network map
9. Path model
10. Nested/Hybrid model

Exhibit 1 offers a summary table of these types of program theories, highlighting comparative benefits and limitations of each. In the individual chapters, we provide a more detailed description of each type of program theory in its scope and purpose, as well as its main benefits and limitations. To facilitate comparison, we illustrate each program theory using the same worked example of a teaching assistant program. By using a common case example, we hope that the similarities and distinctions among the different types of program theories stand out with more clarity. Finally, we include illustrative real-world case applications of each type of program theory and highlight useful resources when relevant.

Teaching Assistant Program

The worked example that we use to illustrate the various types of program theories is a fictitious teaching assistant program designed to support teachers in elementary schools.

The purpose of the teaching assistant program is to reduce classroom disruptions and improve classroom relations. This in turn is expected to improve the classroom climate and the ability of teachers to focus on teaching, resulting in improved student learning.

There is of course no one-size-fits-all recipe for good (or even decent!) program theories in evaluation. Good program theories—and all the types of models and other visual techniques presented in this guide—require equal parts know-how, technical skills, and perhaps more importantly, reflective practice. To facilitate reflective practice, in our last chapter we provide a set of *practical principles for designing program theories*. We highly encourage readers to continue improving the different types of program theories presented in this guide.

Promoting an Equity Perspective in Program Theories

We strongly believe that program theories can be a useful tool towards promoting equity and transformative change in an evaluation (Lemire, Whynot, & Montague, 2019). If designed and implemented well, program theories can address inequities, shift power dynamics; reduce disparities, exclusion, and discrimination; and increase the autonomy and voice of people who have been marginalized or excluded based on race, ethnicity, gender, ability, sexual orientation, and other dimensions. Towards this end, and throughout this guide, we have included suggestions for how to promote an equity perspective in program theories.

Exhibit 1. Different Types of Program Theories

	SCOPE	BENEFITS	LIMITATIONS
1. Logic Model	Lists program inputs, activities, outputs, and outcomes.	Provides a simple and accessible program overview—great for communication purposes.	Limited specification of how specific program inputs, outputs, and outcomes are assumed to be connected.
2. Logical framework (logframe)	Lists program activities, outputs, outcomes, and impacts, as well as underlying risks and measurable, timebound indicators for these.	Provides an accessible, simple program overview that supports program planning, management, and monitoring.	
3. Theory of change	Describes how specific program inputs, activities, outputs, and outcomes are assumed to be connected. Embedded theories of change also describe influencing actors and alternative explanations.	Provides more detailed description than do logic models and logframes of how the program is intended to bring about the desired outcomes. Also supports testable causal hypotheses about how the program works.	Limited ability to address certain types of complexity, such as non-linear and dynamic outcome patterns.
4. Context-mechanism-outcome (CMO) configuration	Describes how specific mechanisms are assumed to generate outcomes in a particular context.	Provides a fine-grained description of the underlying processes explaining why and in what context the program generates desired outcomes.	Can be difficult to distinguish between mechanisms and outcomes, as well as mechanisms and program components.
5. Causal loop diagram	Describes how program activities and outcomes are assumed to be connected, including positive and negative connections, feedback loops, and emergent outcomes.	Provides more detailed and nuanced description of the causal logic underlying programs.	Complex and time-consuming to develop; can be less accessible to stakeholders unfamiliar with the modeling technique.
6. Stock and flow diagram	Describes program dynamics in terms of stocks (program outputs and outcomes) and flows (program processes, activities).	Provides insights into how a program behaves under various conditions.	
7. Concept map	Describes how concepts and ideas are related.	Facilitates conceptual clarity on program components and outcomes comprising program theories, as well as how these develop over time.	Requires specialized software and technical capabilities to develop. Developing maps and models can require substantial time and resources.
8. Network map	Describes the extent to which and how individuals, organizations, or programs are connected.	Provides quantified measures of the extent to which and how individuals, organizations, or programs are connected.	
9. Path model	Describes how program activities are statistically associated with specific outcomes.	Provides quantified measures of the relative strength of influence of program activities on outcomes.	
10. Nested/Hybrid model	Applies two or more modeling techniques in integrated (hybrid) or separate (nested) program theories.	Facilitates differentiating the degree of specificity and focus across different aspects of the program theory.	Complex and time-consuming to develop; can be less accessible to stakeholders unfamiliar with the modeling techniques.

1. LOGIC MODEL

Description

The primary purpose of a logic model is to provide an accessible overview of the main components of a program. Logic models are useful for outlining desired program outcomes as well as the inputs, activities, and outputs that will help the program attain those outcomes. Some logic models also quantify specific benchmarks for outputs and outcomes.

Logic models are often presented in a tabular format structured around program inputs, activities, outputs, and outcomes.

Logic Model versus Theory of Change

The terms *logic model* and *theory of change* often are used interchangeably.

One key distinction between these is that a *logic model* simply lists program inputs, activities, outputs, and outcomes without specifying how each of these is connected.

A *theory of change* uses many of the same building blocks—activities, outputs, and outcomes—but specifies how each of these is connected (see section on Theories of Change).

- **Inputs** refer to program resources, funding, materials, equipment, technology, staff, or any other support services and material resources available to the program.
- **Activities** refer to the concrete actions, events, and strategies implemented by the program.
- **Outputs** refer to the concrete products delivered by the program (emerging directly from the program activities).
- **Outcomes** refer to knowledge, skills, attitudinal, behavioral, and other changes experienced by the program participants (and other stakeholders). Outcomes often are broken down to short-term, medium-term, and long-term outcomes (by some referred to as *impact*)[1] and can be depicted in separate columns or text boxes.

Some logic models also include contextual factors that could influence the program. These are aspects of the program setting that could influence the program's ability to deliver the intended outcomes. In their visual presentation, logic models can be oriented from left to right or from top to bottom. We provide an example of a generic logic model of the teaching assistant program in **Exhibit 2**.

Exhibit 2. Logic Model—Teaching Assistant Program

INPUT	ACTIVITIES	OUTPUT	SHORT-TERM OUTCOMES	MEDIUM-TERM OUTCOMES	LONG-TERM OUTCOMES
Funding	Practical / logistical support	Number of in-class sessions	Teacher can focus on teaching	Better learning environment	Increased learning
Teaching assistants	Pedagogical support	Number of coaching sessions	Fewer disruptions	Better classroom climate	
Coaches			Fewer student conflicts		

Benefits and Limitations

Logic models provide an accessible overview of the main components comprising a program. This can be beneficial when communicating what the program is to funders, stakeholders, and other audiences unfamiliar with the program. By focusing on the most salient program components, logic models also can serve to focus and provide a structure for monitoring and evaluating program implementation and performance. Logic models frequently are featured in reports and publications, and evaluation commissioners and funders often request and expect logic models as part of a proposal.

In terms of limitations, the linear and tabular presentation of logic models provides for an overly simplified description of a program. It can be difficult to adequately capture all pertinent aspects of a program with only the categories of inputs, activities, outputs, and outcomes. Though the tabular format indicates the general logic of the program, a logic model does not describe the ways in which specific program components are connected to one another and with the outcomes of interest. As such, logic models are unable to

Promoting Equity

Centering the purpose of our program theories on equity is a necessary first step towards promoting equity in program theories. Common practice in program theories is to focus on the immediate, surface-level outcomes of the program.

If we are to promote equity in our program theories, we will need to reach beyond traditional outcomes to focus on outcomes that speak directly to the root causes of inequities. Root causes are the underlying factors that create social issues and that make those issues likely to persist even though a program might be in place to alleviate more surface-level needs of the affected individuals and communities.

As just one example, we need to shift our focus on increased student academic achievement to focus instead on gaps in student academic achievement and addressing underlying causes for gaps in student academic achievement. This shift entails being explicit about the population expected to benefit from the program and giving priority to disadvantaged and marginalized groups of students.

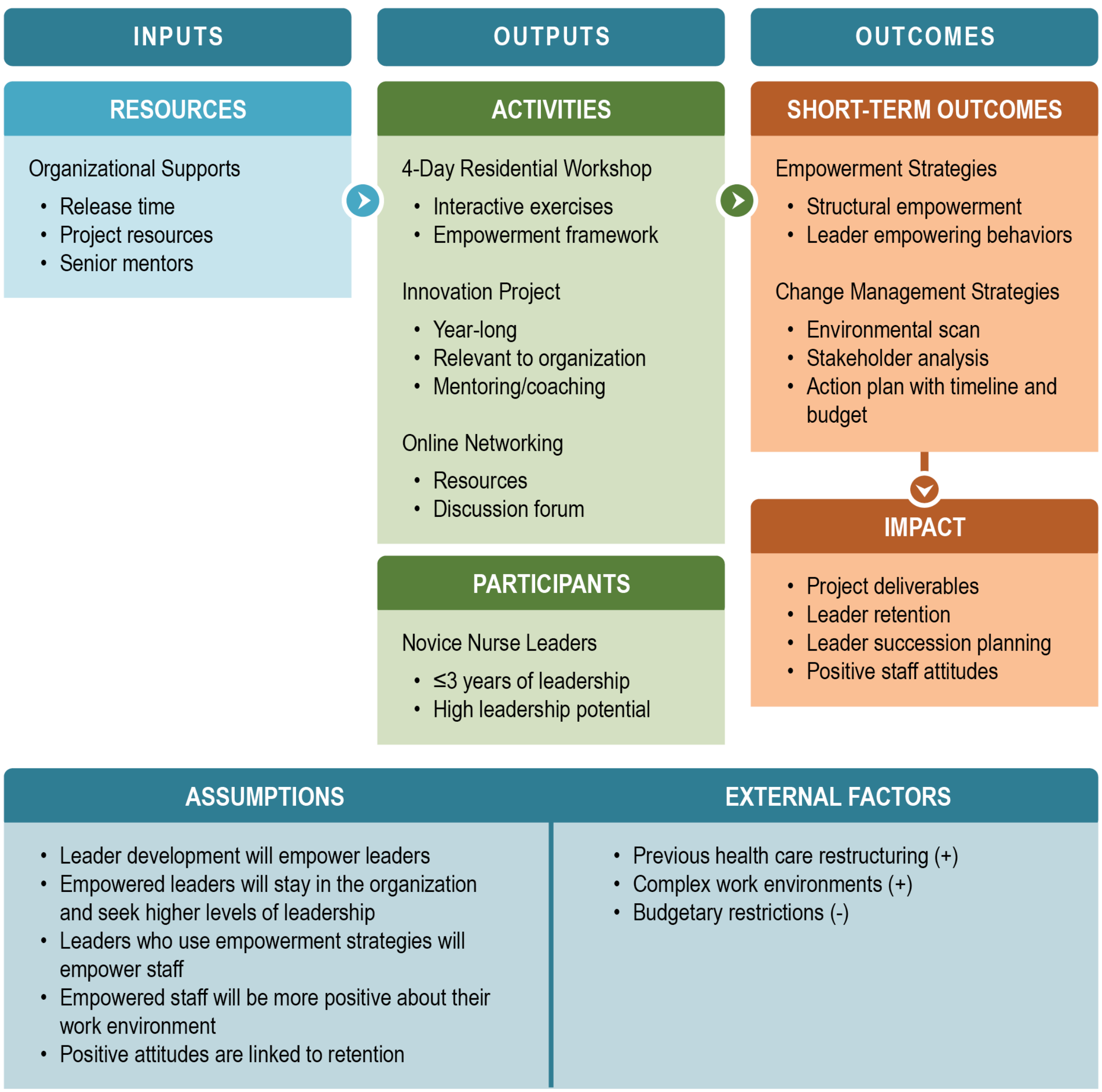
capture the more complex relationships between components and outcomes that can be useful to examine as part of an evaluation.

Real-World Applications

Exhibit 3 illustrates a logic model depicting the main components of an organizational change intervention. The authors grouped activities (*4-Day Residential Workshop, Innovation Project, Online Networking*) and short-term outcomes (*Empowerment Strategies* and *Change Management Strategies*) within the main boxes.

The logic model also highlights assumptions and external factors in separate boxes. The external factors are marked with plus and minus signs to indicate positive and negative program influence, respectively. Whereas the authors listed program participants under *Outputs*, some might prefer to list the participants under *Inputs*.

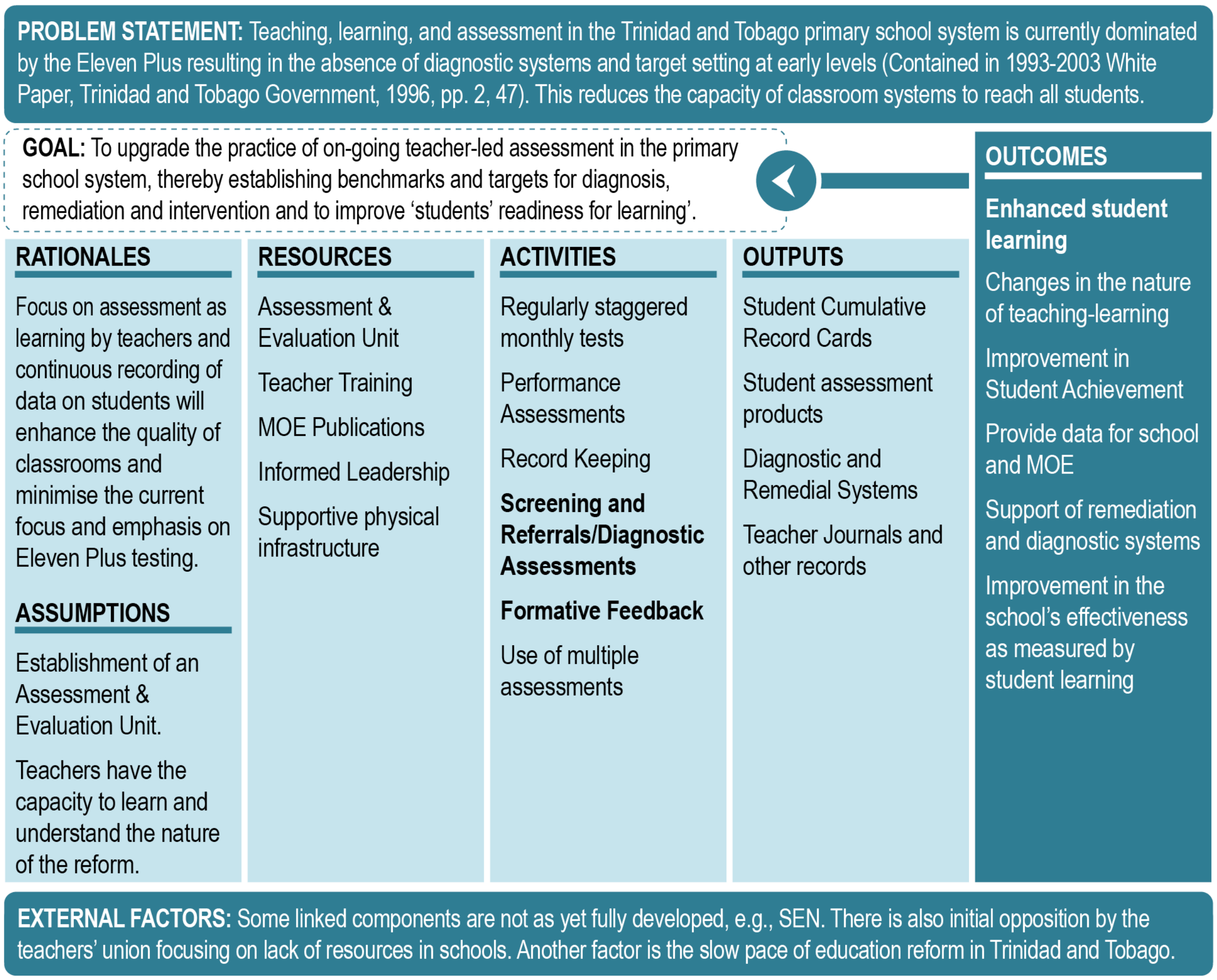
Exhibit 3. Logic Model with Explicit Assumptions and External Factors



Source: Adapted from Havaei and Macphee (2015)

Exhibit 4 presents a logic model of a program promoting the use of diagnostic assessment in primary schools. The logic model is framed by a problem statement (the absence of diagnostic assessment) and oriented towards a corresponding goal (to upgrade teacher-led assessments). The author used bold font to indicate key activities (e.g., screening and referrals) and the key student outcome (enhanced student learning). The model also specifies the underlying rationale and most salient assumptions for the program. External factors are described in a separate text box.

Exhibit 4. Logic Model with Clearly Articulated Problem Statement and Program Goal



Source: Adapted from De Lisle (2015)

Resources

The Education Logic Model (ELM) application is a free, downloadable software that supports development of logic models and theories of change. The ELM was developed by the Regional Educational Laboratory (REL) Pacific. See https://ies.ed.gov/ncee/edlabs/regions/pacific/elm.asp.

2. LOGICAL FRAMEWORK

Description

The logical framework (logframe) matrix is widely used in international development evaluation.[2] Originally intended as a program planning and implementation tool, logframes primarily depict how program activities are intended to lead to a specific set of measurable outcomes and impact. In its practical application, a logframe is developed ideally during the program design and planning stage, with revisions made throughout the implementation of the program.

The logframe matrix is typically in the form of a table, with rows for program components (activities, outputs, outcomes, impact) and columns for measurement information (program summary, indicators, means of verification, risks/assumptions). However, developers can easily adapt this structure to preferences within different government and donor organizations.

A logframe matrix for the teaching assistant program is presented in **Exhibit 5**, with generic structure and category labels.

Exhibit 5. Logframe Matrix—Teaching Assistant Program

	SUMMARY	INDICATORS	MEANS OF VERIFICATION	RISKS/ ASSUMPTIONS
GOALS	-Improved learning	-Average math/ ELA test scores	-State-administered tests	-Access to test scores
OBJECTIVES	-Improved learning environment -Teacher can focus on teaching -Improved classroom climate	-Average number of classroom disruptions -Average climate scores	-Teacher survey and observation	-Teachers allow classroom observation
OUTPUTS	-Number of in-class TA sessions -Number of coaching sessions	-80 TA sessions per school/ per month -20 coaching sessions per school/per month	-Program records	-Teachers are interested in TA support and coaching
ACTIVITIES	-Teaching Assistants and coaches assigned to classrooms	-Number of teaching assistants/coaches assigned	-Program records	-School leadership supports program

Benefits and Limitations

There are many potential benefits of using logframes. If designed and implemented well, the logframe matrix can support program planning, management, and monitoring. As a management approach, the logframe matrix also can serve as the program overview and work plan for the program, guiding program implementation and management (Imas & Rist, 2009). As a monitoring tool, the logframe provides structure and focus by establishing key activities, outputs, and outcomes to be monitored; by connecting these with measurable indicators; and by identifying plausible risks and assumptions (Imas & Rist, 2009). It also serves as an accountability tool to the client by monitoring progress and achievement of targeted outcomes and impacts. The Logframer software allows for logframes to be directly linked with a budget, calendar, and a monitoring interface.

One limitation of the logframe is the linear, tabular visualization—resembling the logic model—which limits its ability to capture more complex causal relationships underlying programs. The common focus on quantitative indicators in logframes also limits the role of qualitative means of verification. This limitation potentially results in oversimplification and omission of key aspects of a program (referred to by some as the "lack-frame" issue) and potentially limits or even blocks program adaptation in response to new opportunities and challenges (referred to by some as the "lock-frame" issue).

In general, the logframe should not be considered a stand-alone tool, but rather designed and implemented in tandem with other types of program theories (Freer & Lemire, 2019; Teskey, 2021).

Promoting Equity

There are several ways of promoting equity in the context of logframes:

- Include equity-focused output/outcome statements wherever possible.
- Include disaggregated data by sex, geography, income, and race in all people-focused indicators; and actively consider how these interact to amplify disadvantage (intersectionality).
- Include equity factors in the risks/assumptions section; for example, informed by a Gender Equality, Disability, and Social Inclusion (GEDSI) analysis.
- Actively consider and include potential unintended consequences in risks, such as gender-based violence, in line with a *do no harm* approach.

Real-World Applications

Exhibit 6 illustrates a logframe for a redesign of an integrated water and pollution management program in Zimbabwe. The author used numbering (1-5) to connect specific activities and outputs described in the narrative summary with specific indicators and means of verification.

Resources

Logframer is a freeware tool that facilitates development of a logframe. Logframer has an integrated project planning and monitoring interface with calendar and budget. See: https://www.logframer.eu/content/step-step-guide-through-logframe.

Exhibit 6. Logframe Using Numbering to Make Connections among Components

	NARRATIVE SUMMARY	OBJECTILVELY VERIFIABLE INDICATORS	MEANS OF VERIFICATION	CRITICAL ASSUMPTIONS
GOAL	Clean and safe water resources in the Chivero Basin	Well-being and health of population (human, flora, fauna) is acceptable according to world standards	Statistics from: • Department of Health and Child Welfare • Natural Resources Board	Water Act of 1998 is implemented and enforced
PURPOSE	Sound water and pollution management in the Chivero Basin	Feasible options are available that will lead to a measurable decrease of: • Per capita/hectare water use • Water losses • Sediment loads • Pollution and nutrient loads • Eutrophication of Lake Chivero	Statistics (and baseline) from: • DWD/ZINWA (Hydrology branch, Water Pollution Control Unit) • City of Harare • Department of Environment • Manyame Catchment Council	• Statistical surveys are reliable and continue to be collected on a regular basis • Institutions and stakeholders are willing to adopt (and adapt) design options
OUTPUTS	1a) Method to identify N and P leachate from diffuse sources 1b) Data on groundwater pollution from selected waste dumps 2a) Alternative operational rules for dams 2b) Alternative land husbandry methods 3) Model for environmental in stream requirement	1a) N and P contribution from two commercial farming areas is quantified 1b) Groundwater pollution from two solid waste dump sites is quantified 2a) Two scenarios for the operation of Lake Chivero dam are developed 2b) A method reducing erosion and increasing rain infiltration is identified 3) Environmental in stream flow requirements for Basin established	1) Working paper research theme 1 (and conference papers etc.) 2) Working paper research theme 2 (and conference papers etc.) 3) Working paper research theme 3 (and conference papers etc.)	Stakeholder groups cooperate with the research, actively participate in workshops, and critically draft design options
ACTIVITIES	1) Identify critical sources of pollution 2) Develop operational rules for dams 3) The environment	Input Personnel 1) Principal investigator (part-time) 3) Senior researchers (part-time) 2) PhD researchers (part-time)	Input Budget (US$) Equipment 3000 Operating costs 2500 Travel and subsistence 2300 Total 7800	Academic, government and non-government organisations make available existing data

Source: Adapted from Wright (2003)

3. THEORY OF CHANGE

Description

The purpose of a theory of change (ToC) is to make explicit how and in what way specific program activities are expected to lead to specific outputs, which in turn are expected to lead to specific outcomes.[3] ToCs visualize the hypothesized connections among activities, outputs, and outcomes—the underlying assumptions of how the program works. More developed ToCs also include contextual conditions—within which the program is embedded—that can influence the ability of the program to generate the desired outcomes. Some ToCs include alternative explanations for the program outcomes, such as “rival” programs influencing the outcomes (Lemire et al., 2012).

Theory of Change versus Theory of Action

A distinction is sometimes made between *theory of change* and *theory of action.*

Theory of change refers to the processes through which change is expected to come about to achieve the program outcomes. It describes how change happens in the context in which we are working and for the people with whom we are working.

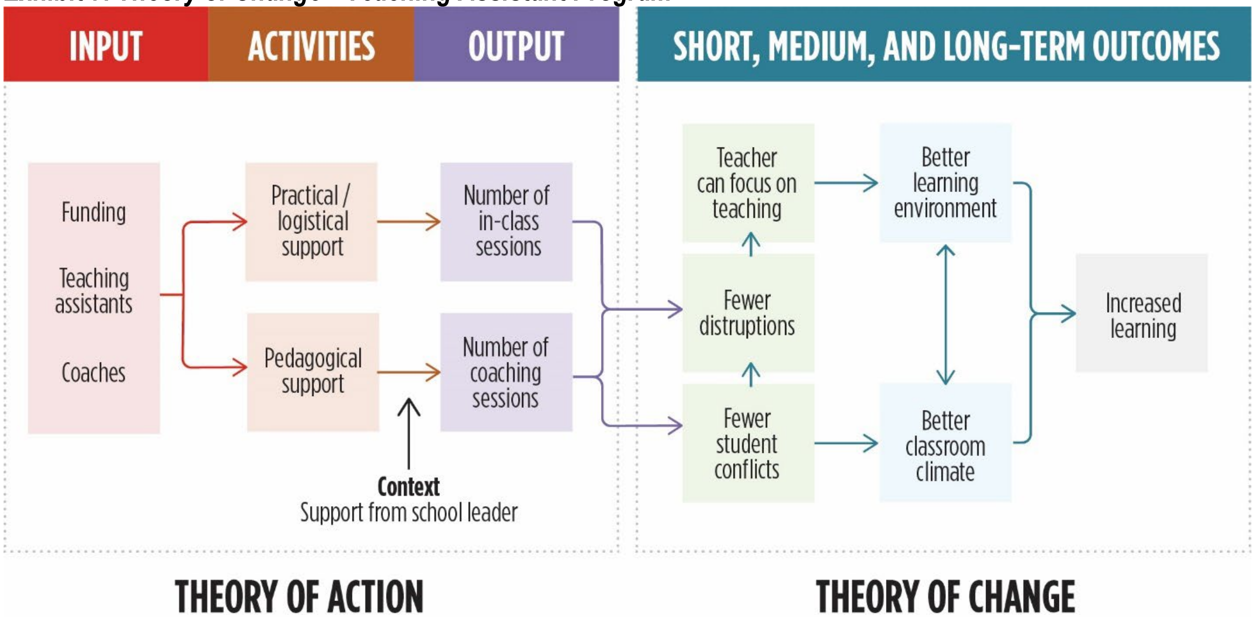
Theory of action refers to the program inputs, outputs, and activities, as well as how these are to be implemented. It describes the mix of things we will do to affect and contribute to these changes.

In practice, many theories of change include both a theory of change and a theory of action.

ToCs usually consist of the same basic building blocks as a logic model: inputs, activities, outputs, and outcomes. These are visually presented in a diagram and accompanied by a narrative description. The narrative description follows an “if/then” logic, with a strong focus on why change is expected to happen in a particular way. The ToC diagram is structured around boxes for activities, outputs, and outcomes, with line arrows (assumptions) indicating how these are connected.

Though there is no standard format for a ToC, a generic version for the Teaching Assistant Program is provided in **Exhibit 7**. As illustrated, the ToC depicts connections between specific activities and outcomes by using arrows. The ToC also indicates the most salient contextual conditions (*Influencing Factors*). The outcome boxes describe the direction of change, for example, *Better Classroom Climate* and *Learning Environment* are intended to lead to *Increased Learning*.

Exhibit 7. Theory of Change—Teaching Assistant Program



Benefits and Limitations

In marked contrast to the tabular logic models and logframes presented in the preceding sections, a theory of change centers on the inner workings of a program. If designed and implemented well, the use of a ToC can be a rich source of information regarding a program's progress, intended and unintended outcomes, and underlying causal structure.

By detailing how specific activities and outputs lead to specific outcomes, and how these are influenced by contextual factors, the ToC can support development of fine-grained hypotheses about how the program is intended to generate the desired outcomes. The ToC approach can also facilitate learning and adaptation, through regular revisiting and updating of the initial ToC as implementation progresses and data is collected and used to refine key aspects of the ToC.

For evaluation and monitoring, well-developed ToCs can structure and focus the data collection and analysis by identifying the most salient program components, contextual conditions, and outcomes. By explicating the underlying logic

Promoting Equity and Do No Harm

Program theories tend to focus on a select set of intended and often positive outcomes; that is, outcomes that represent a beneficial change to program participants and other stakeholders.

However, programs cannot always be assumed to only do good. Many programs have unintended consequences that might or might not be positive. Increasing the risk of gender-based violence or increasing the work burden for women are examples of negative unintended consequences.

In developing/refining a program theory, make sure to consider and (when relevant) include:

- Side effects (adverse spillover effect of the program)
- Paradoxical or counterproductive effects (opposite of the intended program effect)
- Inequitable effects (unfair differences across program participants)

of a program, ToCs also allow us to examine and better understand *how* and *why* programs work (or fail to work), whereby important program learning can take place.

Distinguishing Between *How* and *Why* a Program Works

A useful distinction can and should be made between explaining *how* and *why* a program works.

Explaining *how* a program works entails determining and describing the program activities and outputs that either individually or collectively generate a desired outcome. By identifying and describing these program activities and products, our program theories can support important insights into how the program works.

Explaining *why* a program works entails determining and describing the underlying psychological processes generating a specific outcome. These underlying processes are referred to as *mechanisms*. By specifying the underlying mechanisms, program theories can support important insights into why the program works and why the program makes a difference for the participants.

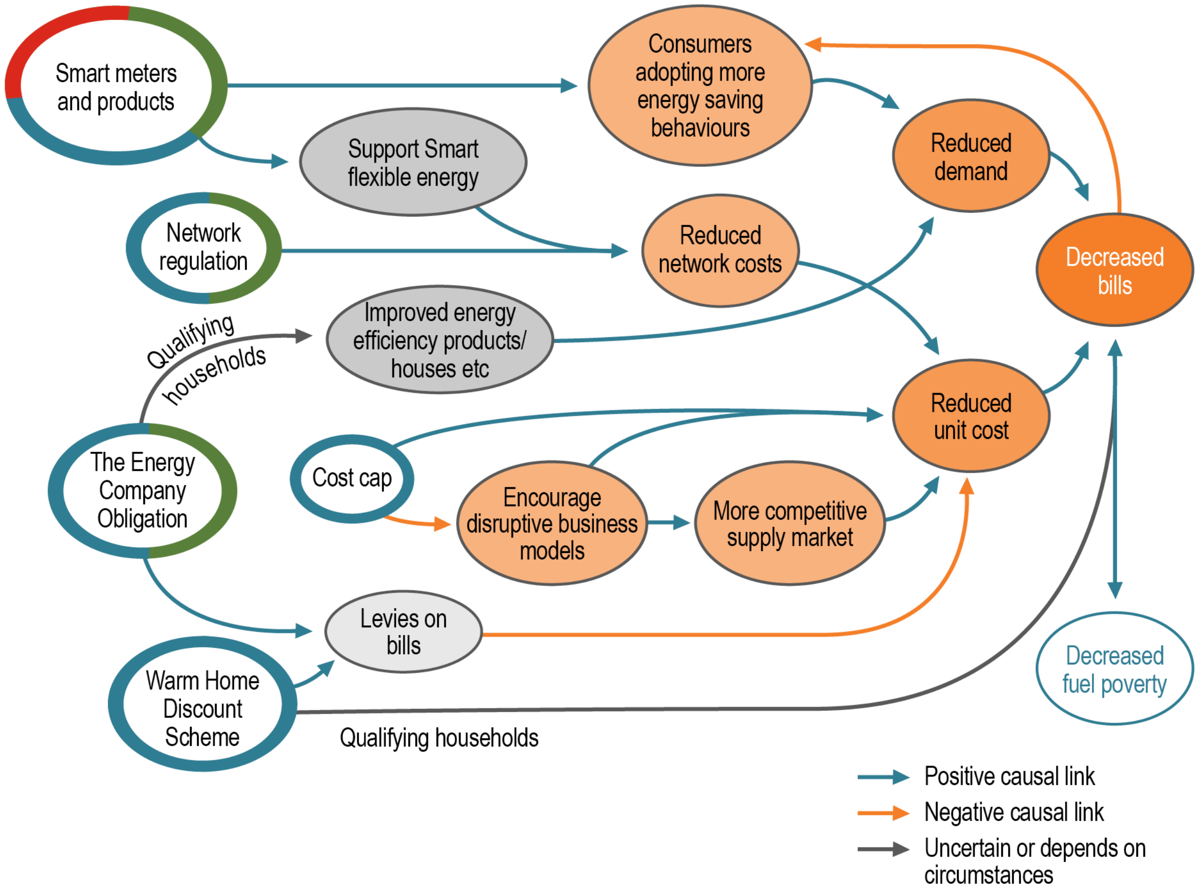
Theories of change support more versatile and complex program modeling, as compared with tabular logic models and logframes. However, ToCs still are somewhat limited in their ability to address certain types of complexity, such as non-linear and dynamic outcome patterns. Though these types of complexity to some extent can be described in the narrative section of the ToC, the more complexity and systems-oriented program theory approaches, such as causal loop diagrams and stock and flow diagrams described later in this guide, are better suited to visualize this type of complexity.

Real-World Examples

The theory of change in **Exhibit 8** is from an evaluation of multiple energy policies. The theory of change describes the interaction between the policies and the sustainability and emissions, affordability, prices, and security of the supply of energy.

The different types of policies (e.g., *Smart Meters and Products*, *Network Regulation, Cost Cap*) are presented on the left side, each of which is color-coded according to its primary focus (e.g., blue indicates reduction of bills; green indicates carbon reduction). The authors also color-coded the connections (causal links) to show whether a positive change in a policy activity, output, or outcome reflects a positive (blue), negative (orange), or uncertain change (grey) in a connected output or outcome. Finally, the authors used different shades of orange to indicate intermediate (e.g., *More Competitive Supply Market* and *Reduced Unit Cost*) to final outcomes (*Decreased Bills*).

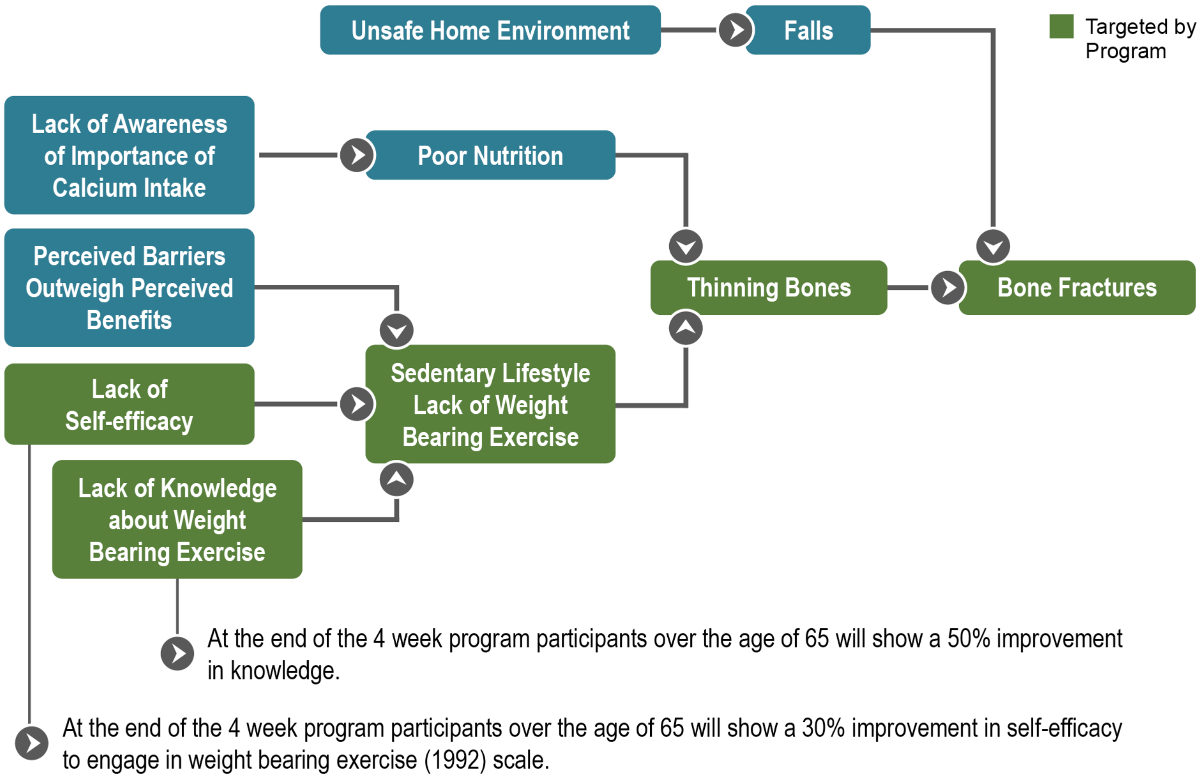
Exhibit 8. Theory of Change with Color-Coded Causal Links



Source: Adapted from Wilkinson et al. (2021)

The theory of change in **Exhibit 9** is from an evaluation of a nutrition and physical exercise program that aims to reduce bone fractures in people who are elderly. The green boxes identify the outcomes directly affected by the program. The authors also specified targets of program success, as well as how and when these targets would be measured, a useful strategy for establishing a clear connection between the theory of change and the data collection. The selected targets for program success were the outcomes closest to and most directly affected by the program.

Exhibit 9. Theory of Change with Color-Coded Direct Outcomes and Targets



Source: Adapted from Renger and Titcomb (2002)

Resources

Theory of Change Online (TOCO) is a web-based software designed to develop and edit theories of change, visualizing outcomes, indicators, rationales, and assumptions. The software requires a subscription. See: https://www.theoryofchange.org/toco-software/.

4. CONTEXT-MECHANISM-OUTCOME CONFIGURATION

Description

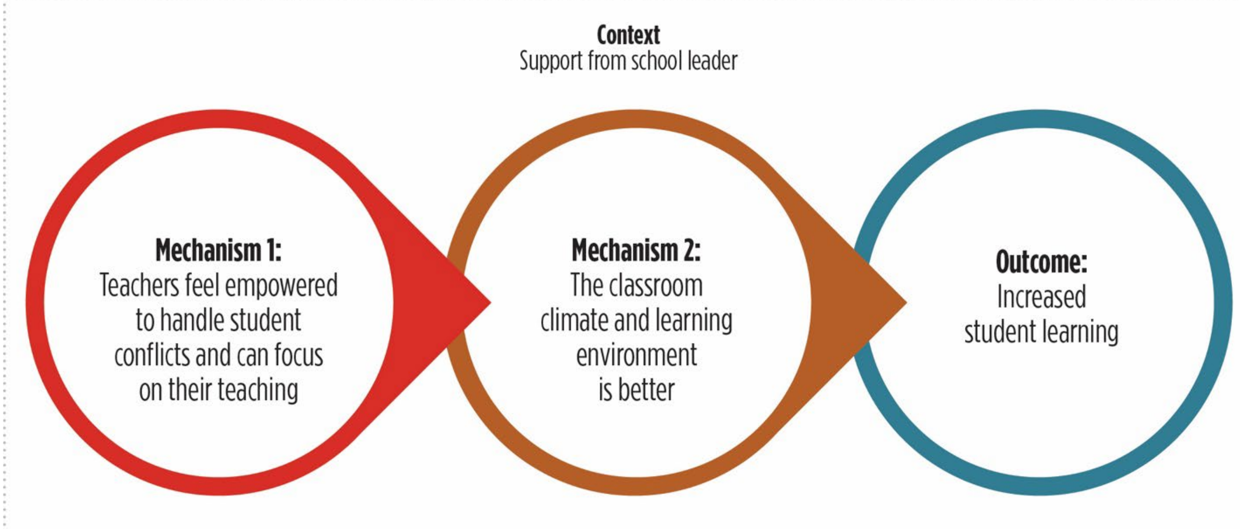
Pawson and Tilley (1997) introduced the context-mechanism-outcome (CMO) configuration as part of their realist evaluation approach. The purpose of CMOs is to describe the generative processes (mechanisms) promoting behavioral changes (outcomes) in a given setting (context).

> **What Is a Mechanism?**
>
> *Mechanism* refers to the underlying psychological processes that bring about the outcome of interest. Mechanisms can be in the form of participant reactions and responses to program activities, such as increased sense of efficacy, empowerment, or motivation, as well as behavioral changes.

CMO configurations most often are presented in the form of a table or in simple box diagrams complemented by a more detailed narrative description. The CMO configuration can involve multiple contextual conditions, mechanisms, and outcomes.

Exhibit 10 presents an illustrative CMO configuration for the teaching assistant program. As compared with other program theories, the focus of the CMO is on the mechanisms (the teaching assistant feels confident and empowered to support the teacher) through which the intended outcome is generated (improved classroom environment).

Exhibit 10. Context-Mechanism-Outcome Configuration—Teaching Assistant Program



Benefits and Limitations

One benefit of CMO configurations is that they bring attention to why and under what circumstance outcomes are generated. The emphasis on making relevant mechanisms explicit provides for a more fine-grained and nuanced understanding of participant reactions and responses to program activities, and in extension a better understanding of why the program made a difference.

In practical applications, one common limitation of the CMO configuration relates to the conceptual confusion surrounding the term *mechanism* (Lemire et al., 2020). Mechanisms have been defined and operationalized as program components, participant psychological reactions to program components, or participant behavioral reactions to program components, as well as combinations of these. Evaluators have experienced difficulties distinguishing between context factors and mechanisms, as well as between mechanisms and outcomes (Nielsen, Lemire, & Tangsig, 2021).

Real-World Applications

The CMO configuration in **Exhibit 11** depicts the interaction among context, mechanisms, and outcomes for a program aimed at reducing long-term sickness absence. The CMO configuration focuses on the interplay among employees, line managers, and senior management in an organization.

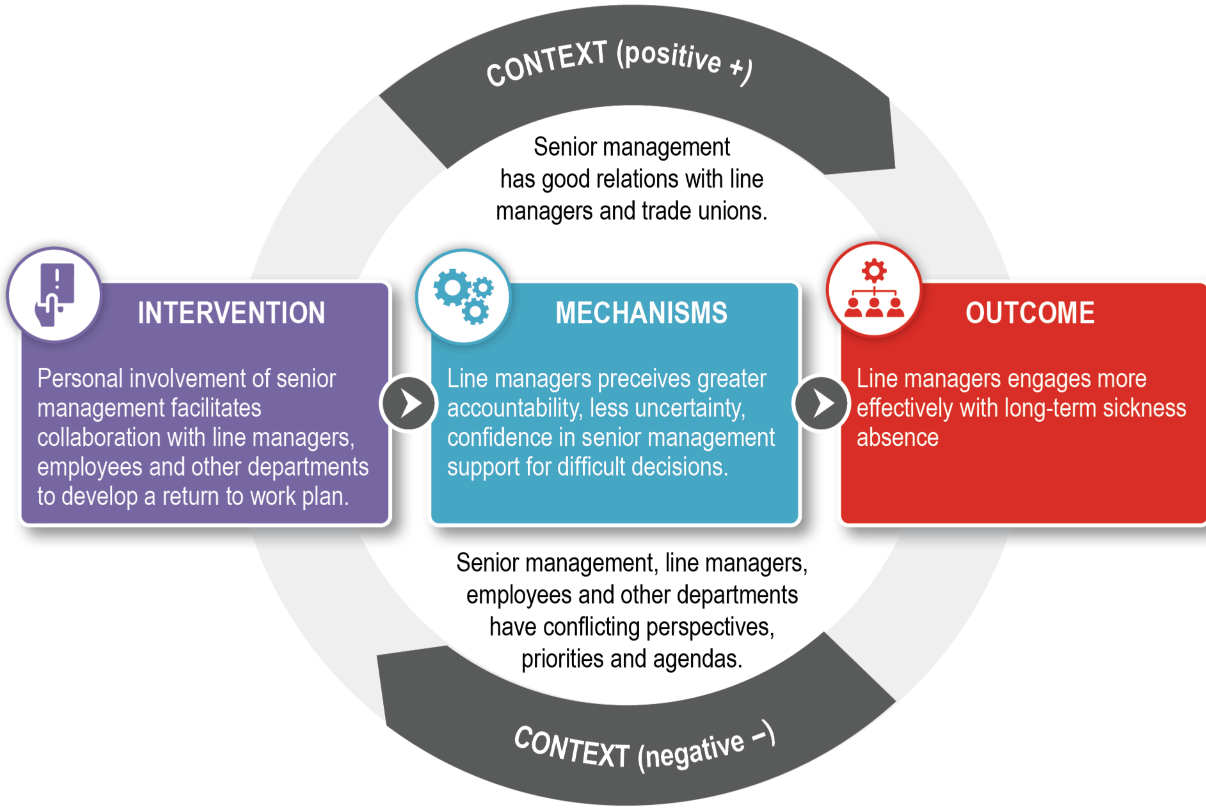
The authors adapted the traditional CMO template by including the program that is intended to trigger the mechanism. The authors also used plus and minus signs to specify contextual conditions that positively or negatively influence the mechanisms of interest. For example, conflicting perspectives and agendas within an organization have a negative influence on the mechanism.

Promoting Equity

Program theories reflect a shared set of assumptions about how and why a program is intended to bring about change. This is potentially problematic because our assumptions are heavily influenced by social norms, biases, and privilege, all of which are internalized at the individual, organizational, and systemic level.

If we are to promote equity in our program theories, we need to critically examine and question the assumptions—and mechanisms—comprising our program theories. Active assumption testing and self-reflection on our own biases is a critical step towards promoting equity and avoiding negative unintended consequences.

Exhibit 11. CMO Configuration with Positive and Negative Context Factors



Source: Adapted from Higgins, Halloran, and Porter (2015)

The CMO configuration in **Exhibit 12** was developed as part of a realist evaluation of an integrated nutritional and smoking cessation program in Scotland. The authors used a tabular format depicting four parallel CMOs for different types of program participants (e.g., participants in the early stages of behavioral change). For each of these subgroups, the authors provided a narrative description of the mechanisms. The outcome column summarizes the extent to which the participants in the program reduced weight gain after smoking cessation. By detailing the CMO configurations for different participant groups, the authors were better able to explain the different ways in which the program made (or failed to make) a difference.

Exhibit 12. CMO Configuration with Outcome Ratings

CONTEXT	MECHANISM	OUTCOME
Participants in the early stages of behavioural change	Encouraged by group support, participants may succeed in making small changes to their diet and physical activity, which in turn may cause them to progress to the 'action' stage in which they actively seek to improve their diet and physical activity	✓
Participants with low motivation and self-efficacy	May be reluctant to attend and participate in classes and this may prevent them from developing a rapport with other group members and the advisor, which may further decrease their motivation	✗
Group with motivated participants that have a good rapport with each other	May share ideas, experiences and support one another, which may encourage them to keep making changes, limit relapse and motivate them to attend further sessions	✓
Participants from disadvantaged areas	May face a variety of stressful situations and may feel unable to commit to quitting smoking and eating halthily simultaneously	✗

Source: Adapted from Mackenzie et al. (2009)

Resources

There is no specific software for developing CMO configurations. Most evaluators use PowerPoint or MS Word.

5. CAUSAL LOOP DIAGRAM

Description

Causal loop diagrams emerged from systems thinking and complexity science. The purpose of causal loop diagrams is to depict more complex outcome patterns, accounting for positive and negative feedback loops as well as emerging, delayed, and unintended outcome trajectories.

Determining Reinforcing and Balancing Feedback Loops

Whether a loop is *reinforcing* or *balancing* depends on the polarity (positive or negative) and the number of the arrows comprising the loop.

A causal loop with an *even* number of causal arrows with a *negative* polarity (or where none are present) is reinforcing.

A causal loop with an *odd* number of causal arrows with a *negative* polarity is balancing.

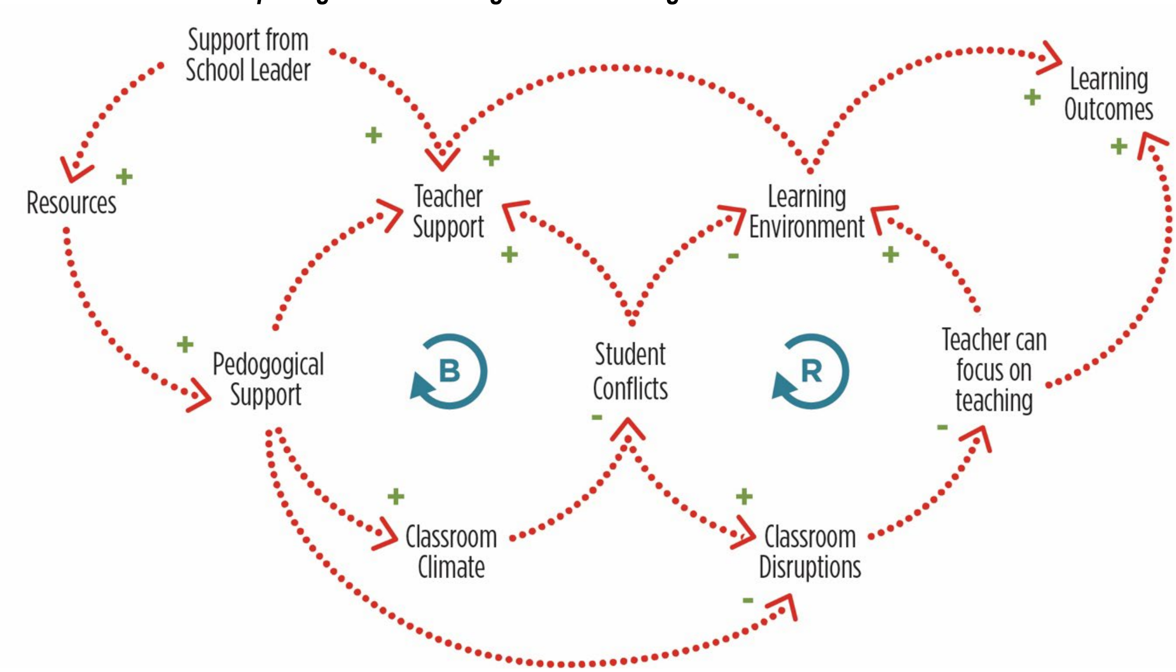
The visual display of causal loop diagrams usually consists of program components and outcomes presented in text boxes and causally linked to one another using line arrows. Each line arrow has a direction and polarity. Arrows with a plus sign (+) indicate that a change in the first variable in a certain direction causes a change in the second variable in the same direction. Arrows with a minus sign (−) indicate that the change in the first variable in a certain direction causes a change in the second variable in the opposite direction.

Emerging outcomes can be presented with dotted line arrows. Delayed outcomes can be represented by double bars (||) on the line arrows.

Causal loop diagrams can include feedback loops that are either reinforcing or balancing. Reinforcing causal loops are indicated with an “R” and balancing causal loops are indicated with a “B.” Reinforcing loops refer to causal loops where a change in one direction is compounded by more change in the same direction. Balancing loops refer to causal loops where a change in one direction results in a change in the opposite direction.

The causal loop diagram for the teaching assistant program in **Exhibit 13** represents causal pathways, with directions of influence depicted by arrows. The long-term learning outcome is highlighted in bold.

Exhibit 13. Causal Loop Diagram—Teaching Assistant Program



Benefits and Limitations

The main strength of causal loop diagrams is that they allow for more complex causal patterns to be visualized, as compared with the more linear logic models, logframes, and to some extent theories of change. The ability to visualize feedback loops is particularly useful when examining the nature of the interactions between the program activities and outcomes, as well as the wider context within which the program is embedded. This type of information supports a better understanding of how and why programs work.

The necessary tradeoff of the added complexity is that causal loop diagrams can be less accessible to stakeholders unfamiliar with the modeling technique and associated terminology. For communication purposes, causal loop diagrams can be overwhelming. The level of complexity also entails that causal loop diagrams are more difficult and time consuming to develop. For these reasons, engaging stakeholders in the process of constructing causal loop diagrams can be difficult.

Promoting Equity

Traditional program theories often assume that change is linear, emerging through a uniform trajectory of short-, medium-, and long-term outcomes across all program participants.

This assumption is likely flawed. Different subgroups of program participants may experience and respond to the program in different ways, resulting in different outcome trajectories.

For this reason, careful thought should be given the different types of outcome trajectories that could be relevant to depict for different subgroups in the program theory.

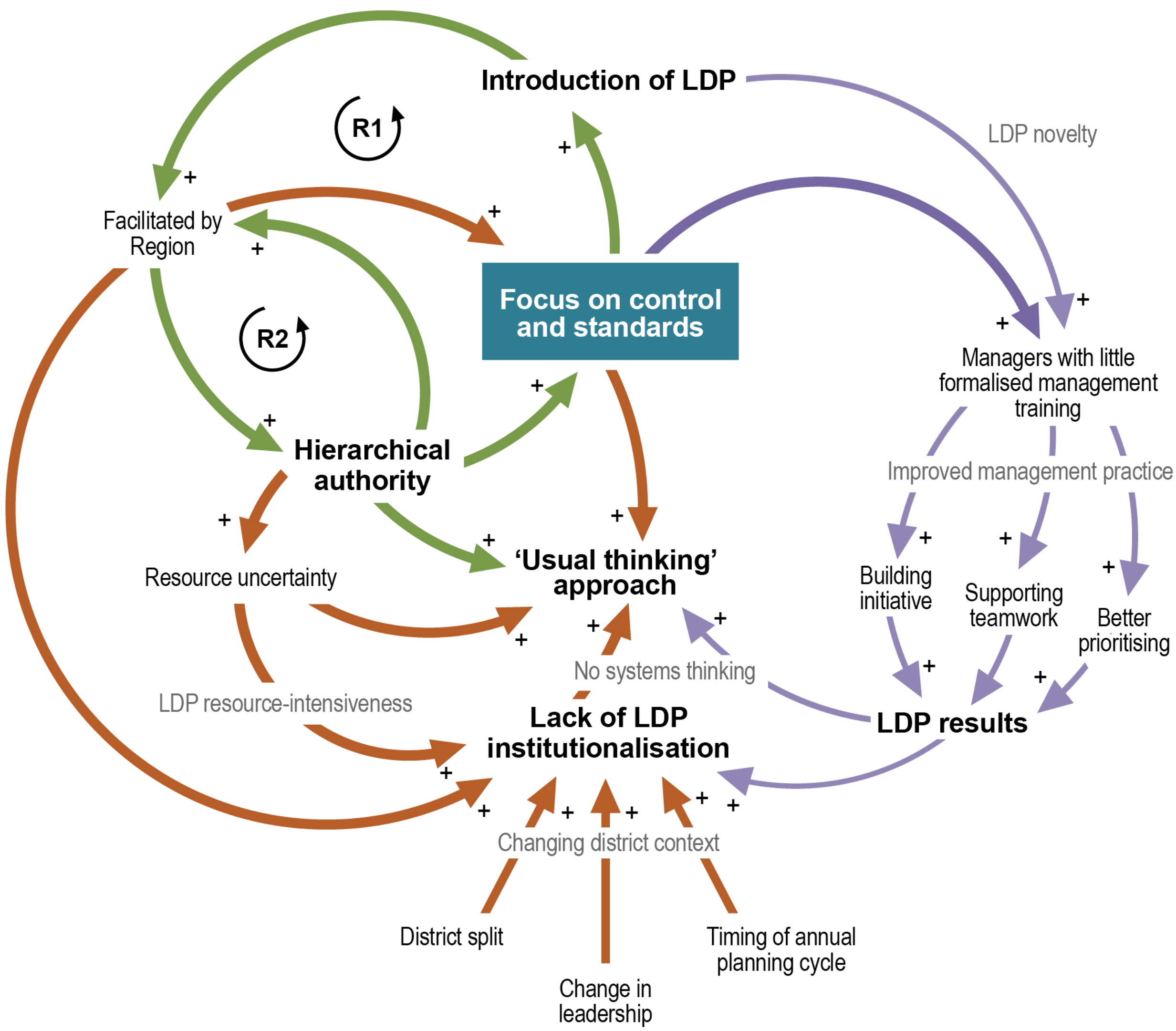
As illustrated in the real-world examples in this guide, there are simple visual techniques to indicate emerging outcomes (dotted lines), delayed outcomes (||), or disrupted outcomes (┤). Different stages of the outcome chain also can be identified by color-coding the lines and symbols.

Of course, many other symbols, icons, and visualization techniques can be relevant potentially when depicting various outcome trajectories.

Real-World Applications

The causal loop diagram in **Exhibit 14** is from an evaluation of a leadership development program ("LDP") for district managers in Ghana. The program aims to advance the application of systems thinking in health care. The directions of influence are depicted by line arrows, with influence in the same direction represented by positive arrows. Two reinforcing feedback loops ("R1" and "R2") are numbered to link with a narrative description of the loops. The authors also use thin lines for the short-term outcomes and thick lines for the medium-term outcomes, visualizing their temporal dimension.

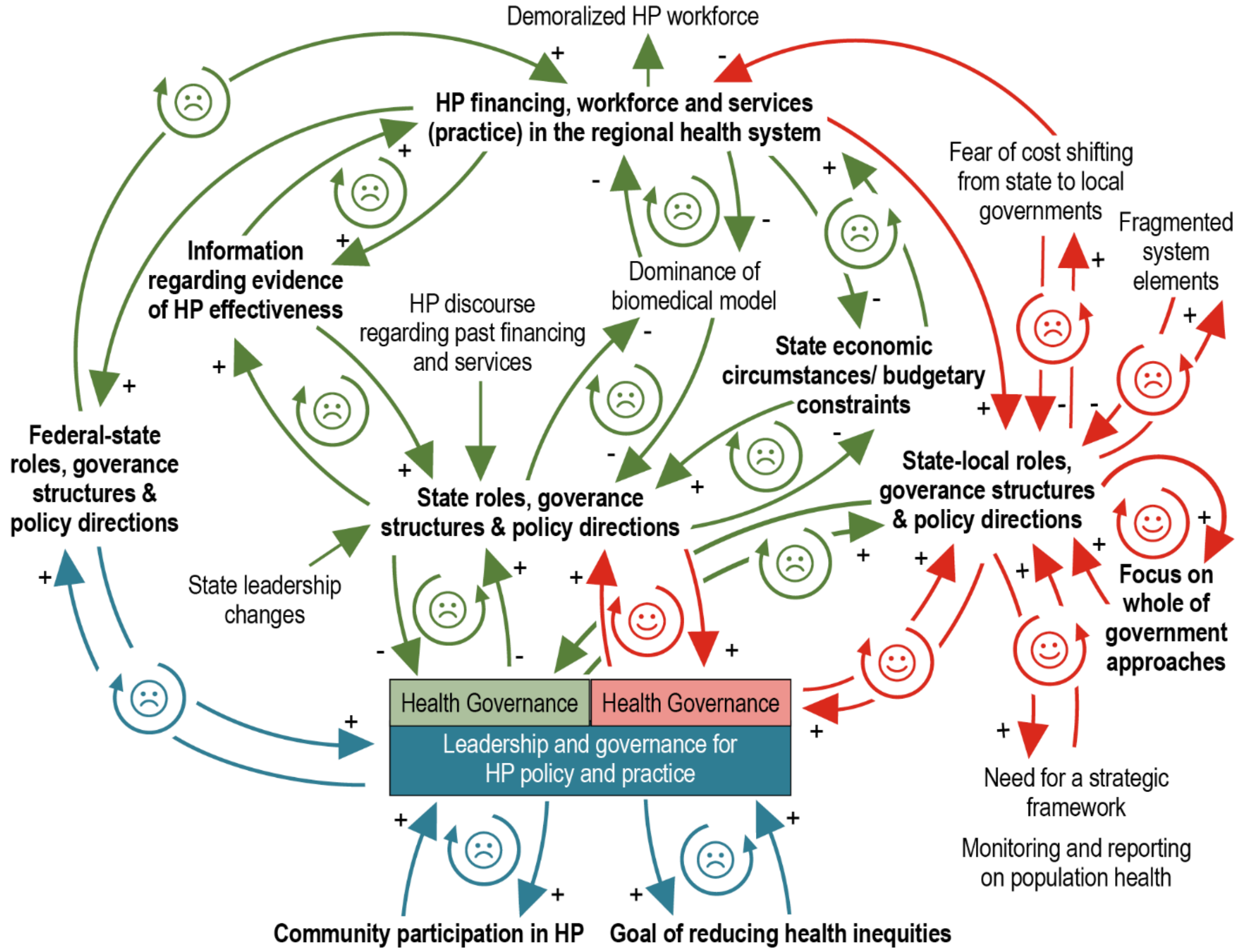
Exhibit 14. Causal Loop Diagram with Thin and Thick Line Arrows for Short-Term and Medium-Term Outcomes



Source: Adapted from Kwamie, van Dijk, and Agyepong (2014)

Exhibit 15 illustrates a causal loop diagram for a health promotion policy in a multisectoral health system in Australia. The dominant factors are highlighted in bold font. The authors used color-coding (blue, red, and green) to distinguish—yet capture the interplay of—different types of leadership and governance related to the health care policy (*Health Governance* versus *Governance for Health*). To provide a more reader-friendly diagram, the authors used icons to represent facilitating (happy face) or inhibiting (sad face) feedback loops on the health promotion policy.

Exhibit 15. Causal Loop Diagram with Face Icons for Facilitating and Inhibiting Feedback Loops



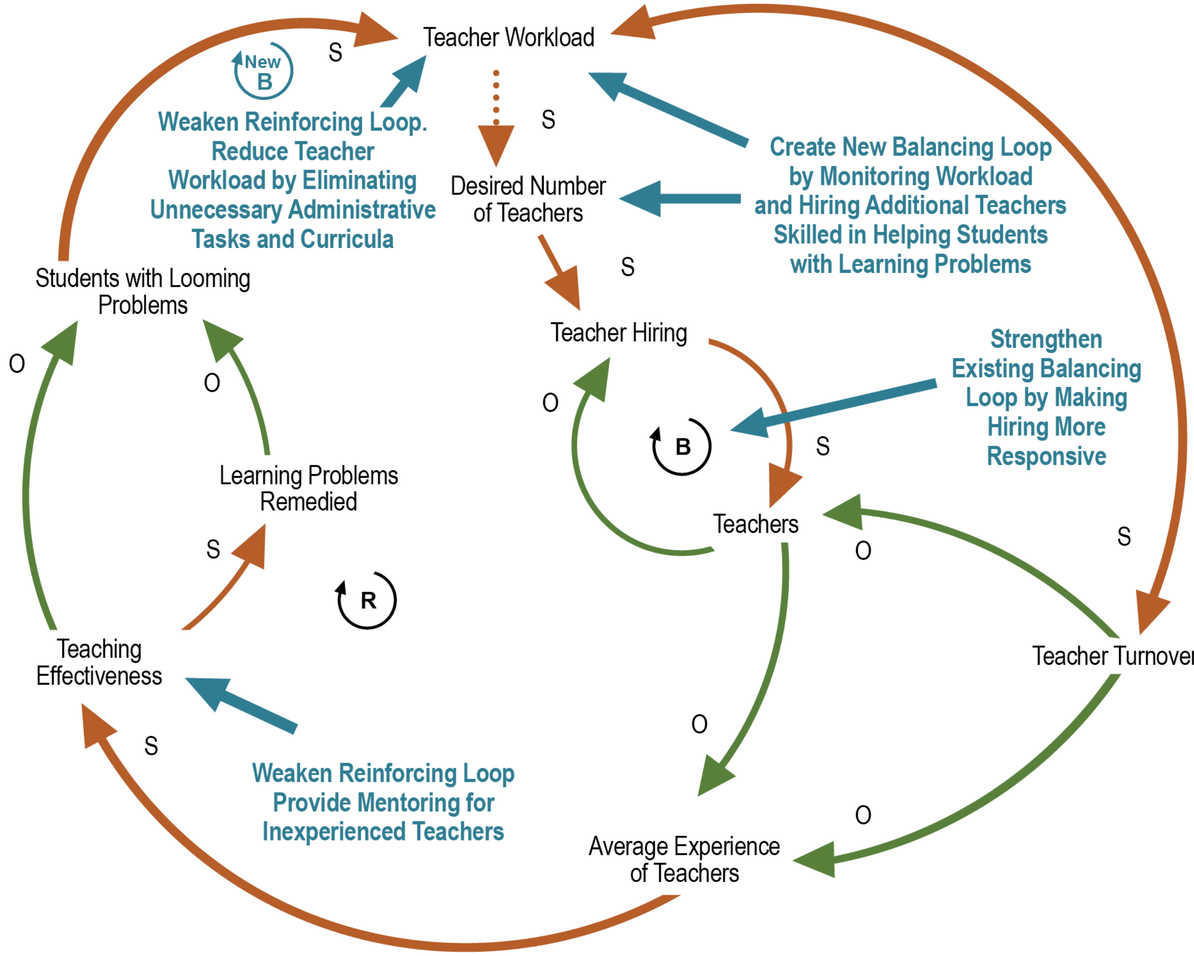
Source: Adapted from Littlejohns et al. (2018)

In **Exhibit 16**, the authors developed a causal loop diagram for a program aiming to improve teaching and student performance. Instead of using a plus or minus sign to indicate the direction of the flow, the authors used "S" for *same* and "O" for *opposite* polarity, respectively.

The arrows color-coded in blue and highlighted in bold represent possible leverage points—or activities—that might positively change the direction of the feedback loops to improve the program. For instance, one might attempt to cope with an increase in teacher workload by monitoring workload and hiring more teachers (loop "I") to reduce workload.

This illustrative example demonstrates how causal loop diagrams can be used to support program planning and development.

Exhibit 16. Causal Loop Diagram with Leverage Points for Program Improvement



Source: Adapted from Hirsch, Levine, and Miller (2007)

Resources

The System Dynamics Loop Generator is a free online tool for developing causal loop diagrams. See: http://sysdyn.simantics.org/.

6. STOCK AND FLOW DIAGRAM

Description

Stock and flow diagrams—by some considered a variant of causal loop diagrams—is a systems-oriented modeling technique. The purpose of stock and flow diagrams is to describe how programs work in terms of variations in *stocks* (key program outputs and outcomes) and *flows* (program processes). Stock and flow diagrams can also be used to identify leverage points for changing the system, which is particularly useful for programmatic adaptations.

Leverage Points

Leverage point is a place in a system where (a) a relatively small local change can produce major effects throughout the system, and (b) communities are likely to be willing, and able, to make the required change (Proust et al., 2012).

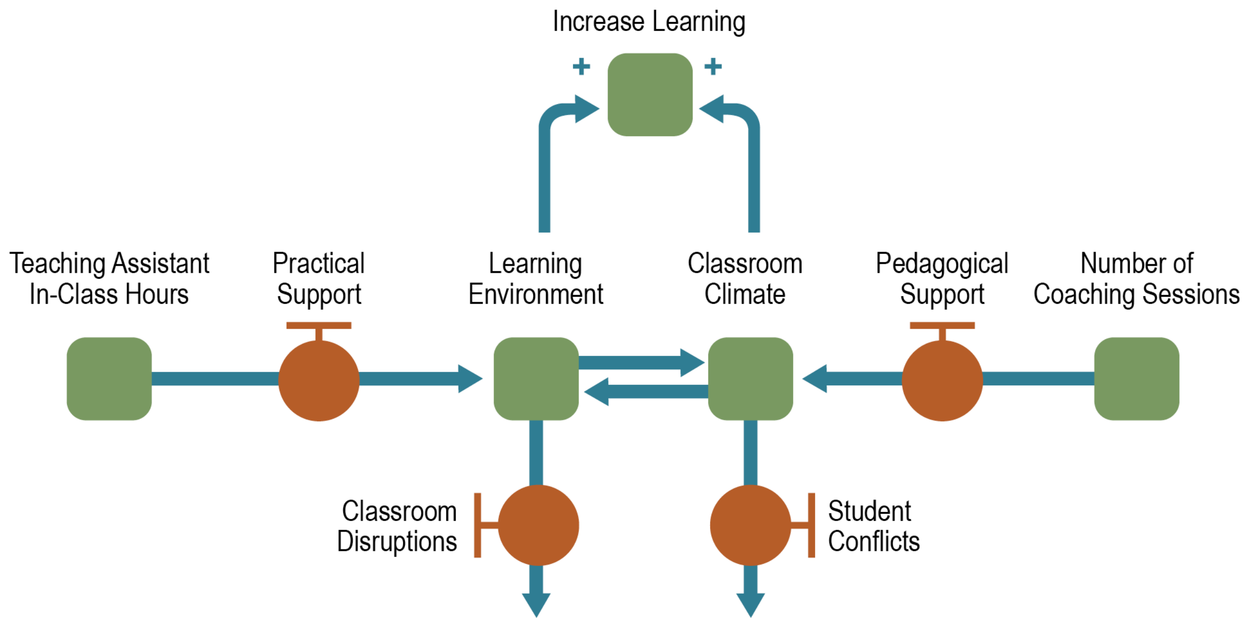
In stock and flow diagrams, stocks are often presented as boxes with text labels. Stocks refer to accumulations of interest to the program, such as number of teachers, high-performing students, or positive classroom climate.

Flows refer to processes, activities, or decisions that increase or decrease the stocks. These are typically represented with thick or double arrows. In line with causal loop diagrams, each flow has a direction and polarity. A *positive* flow means that a change in the first variable in a certain direction causes a change in the second variable in the same direction. Conversely, a *negative* flow means that a change in the first variable in one direction results in a change in the second variable in the opposite direction.

Stock and flow diagrams also can include causal loops. A change in the level of a stock can feed back, around a causal loop, to either amplify or oppose the original change (see section on Causal Loop Diagrams).

In the stock and flow diagram in **Exhibit 17**, the boxes represent stocks (accumulations), such as *Learning Environment* and *Classroom Climate.* The arrows represent inflows and outflows (processes that change the amounts accumulated), such as *Pedagogical Support* (inflow) and *Classroom Disruptions* (outflow) for the *Learning Environment* (stock).

Exhibit 17. Stock and Flow Diagram—Teaching Assistant Program



Benefits and Limitations

One benefit of stock and flow diagrams is that they allow for a better understanding of how a program behaves; that is, how variations in program processes influence variations in program outputs and outcomes. In this way, stock and flow diagrams can help identify both beneficial and problematic program dynamics, such as processes that enhance or reduce program outcomes. Understanding the program dynamics can also motivate and inform further program development.

Stock and flow diagrams also can provide for a better understanding of how participants "flow" through the program. They even can support more advanced simulation modeling of future program developments.

Similar to causal loop diagrams, the complexity of the stock and flow diagrams can pose a barrier to stakeholders who are not familiar with the modeling technique and associated terminology. Developing stock and flow diagrams also requires sufficient technical knowledge and skills to apply relevant

Promoting Equity

Systemic factors are important to include in program theories. If we are to promote equity in program theories, we need to award attention to the influence on structural racism, sexism, and the many other systemic factors that produce and sustain inequities.

Structural racism is a very complex, dynamic system with interlinked social, political, and economic components. For this reason, program theories need to capture how interconnections between the program and existing policies, social and institutional practices, and cultural representations and narratives reinforce inequities.

By capturing and closely examining these in our program theories, we will be better positioned see how and in what ways race, gender, and other privileges and disadvantages fundamentally affect our program. It is important to actively consider intersectionality; that is, multiple forms of inequity and disadvantage. For example, a Black woman with a disability might have a very different experience than might a White woman who had no disability.

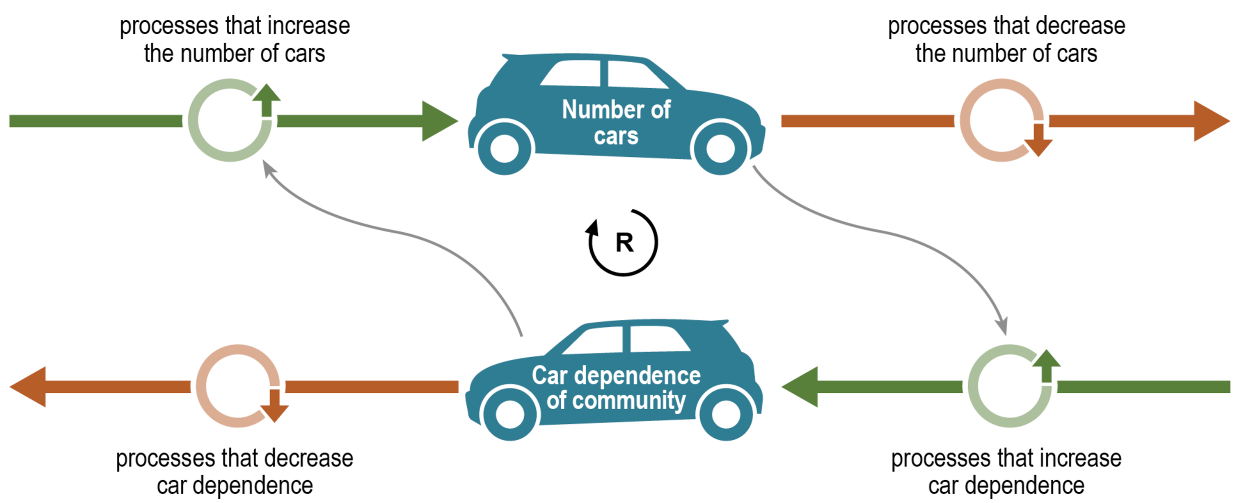
The Aspen Institute (2009) published a helpful guide on how to incorporate structural racism analysis in program theory development.

software—especially if the goal is to simulate future program developments.

Real-World Applications

Exhibit 18 provides a simple example of a stock and flow diagram. The diagram contains two stocks: *number of cars* and *car dependence of community.* The thick arrows (with "tap" symbols) represent processes that can drive the levels of their affected stocks up (inflow) or down (outflow). For example, the processes that increase the number of cars is an inflow to number of cars. The thin-lined arrows represent how stocks can influence the processes. For example, car dependence of community influences processes that increase the number of cars. The encircled R in the diagram indicates that this is a reinforcing (positive) feedback loop—as the number of cars increases, the car dependence of the community increases, which in turn leads to an increase in cars.

Exhibit 18. Stock and Flow Diagram with Markers for Positive and Negative Processes



Source: Adapted from Proust et al. (2012)

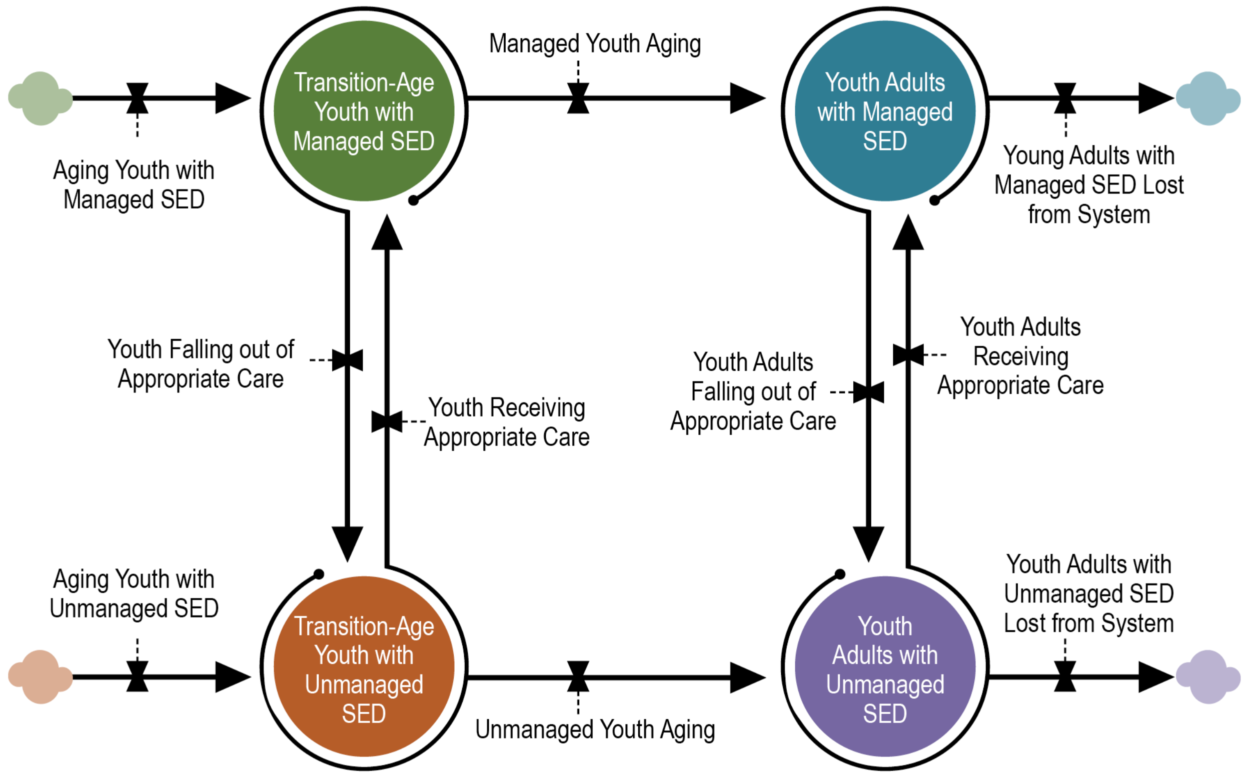
The stock and flow diagram in **Exhibit 19** was used as part of an evaluation of a program for transition-age youth with emotional and behavioral challenges in the southeastern United States. The stocks are depicted as four boxes, one in each corner: *Transition-Age Youth With Unmanaged [Severe Emotional Disturbance], Transition-Age Youth With Managed SED, Young Adults With Unmanaged SED,* and *Young Adults With Managed SED*.

The double-lined arrows represent flows, depicting how transition-age youth might flow from one stock to another over time. The rate of flow, depicted with a valve across each arrow, can be used to quantify the number of units (youth) flowing over a fixed unit of time (e.g., week, month, year) based on data.

There are also external flows in and out of the program. For example, new youth age into the transition-age youth stocks, emerging from a cloud on the left side of the diagram. Similarly, young adults leave the program over time, as indicated by the clouds on the right side of the diagram.

In a separate program theory, the authors enhanced the stock and flow diagram with causal loops (see **Exhibit 33** in section on Nested and Hybrid Models).

Exhibit 19. Stock and Flow Diagram with Both Unidirectional and Bi-Flow Arrows

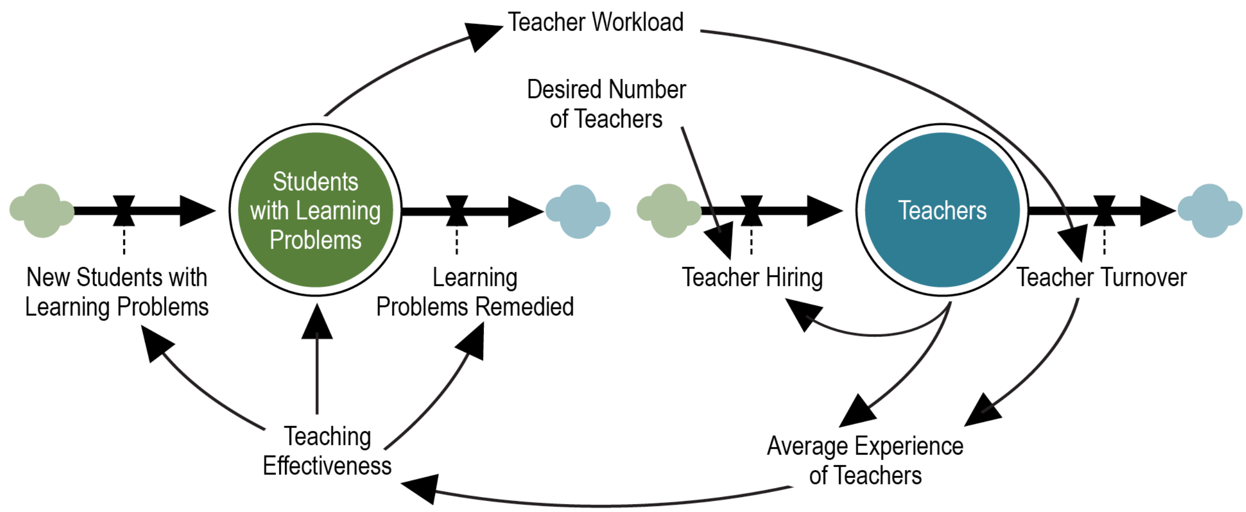


Source: Adapted from Lich et al. (2017)

The stock and flow diagram in **Exhibit 20** was developed for a program aiming to improve teaching and student performance (a corresponding causal loop diagram for the program is presented in **Exhibit 16**). The stock and flow diagram is structured around two stocks: *Students with Learning Problems* and *Teachers*. The external flows in and out of these stocks are depicted with thick arrows (and clouds), such as *New Students with Learning Problems* and *Learning Problems Remedied*. The other processes connecting and affecting these two stocks are depicted with thin-line arrows (e.g., *Teacher Workload*).

The authors used Vensim software to develop and convert the stock and flow diagram into a mathematical simulation model that was used to estimate how different changes to the diagram are likely to affect the stocks and flows.

Exhibit 20. Stock and Flow Diagram With External In and Out Flow



Source: Adapted from Hirsch, Levine, and Miller (2007)

Resources

A brief guide by Daniel Aronson and Daniel Angelakis (1999) offers procedural guidance and practical lessons learned on how to convert causal loop diagrams into stock and flow diagrams. See: https://thesystemsthinker.com/wp-content/uploads/pdfs/100602pk.pdf.

Vensim is a license-based software for developing and analyzing stock and flow diagrams, causal loop diagrams, and other types of models. The software also has several features for model simulation and optimization. See https://vensim.com/.

7. CONCEPT MAP

Description

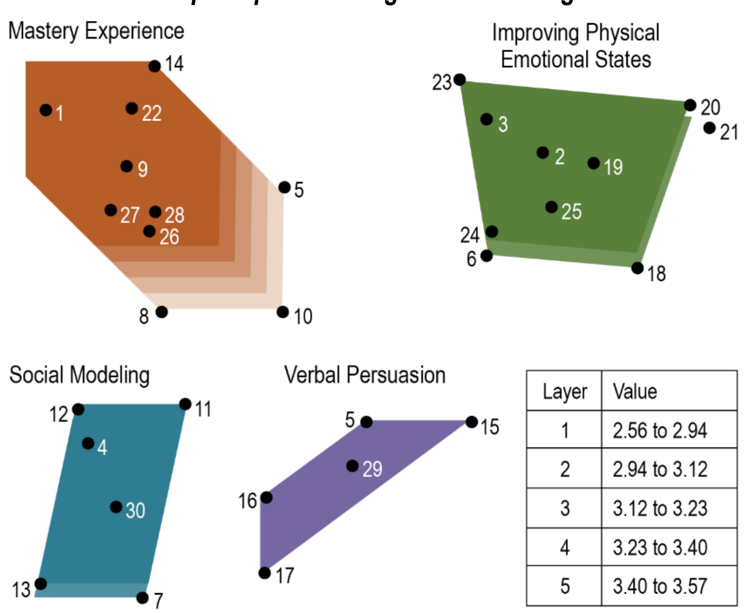
Concept maps—referred to by some as *mind maps*—are diagrams that can be used to clarify and explicate key concepts in a program theory. The purpose of concept mapping is to facilitate a shared conceptual understanding of one or more key concepts or ideas, as well as the relationships between these. Concept maps are particularly useful in unpacking mechanisms and outcomes and in documenting changes in these before and after a program is implemented.

> **Two Types of Concept Mapping**
>
> A distinction is sometimes made between *mind* maps and *concept* maps.
>
> *Mind maps* refer to visualizations of one or more concepts, ideas, or themes. The organization and structure of mind maps are not defined by statistical analyses.
>
> *Concept maps* refer to visualizations that are defined by statistical analyses, such as factor or network analyses. These maps quantify the relationship between different concepts, ideas, and themes.

Concept maps are often—but not always—developed on the basis of statistical analyses, such as factor or network analysis. There is no common template for concept maps. Most maps are visually structured around circles or boxes for key concepts or themes, and relationships between these are indicated by connecting lines.

A generic concept map of the teaching assistant program is provided in **Exhibit 21**. The map depicts four types of strategies used by teaching assistants to reduce classroom conflict (e.g., *Social Modeling* and *Verbal Persuasion*). The numbered squares indicate survey items related to each strategy. The number of layers represents the relative importance (average ratings) attributed to each strategy by the teaching assistants.

Exhibit 21. Concept Map—Teaching Assistant Program



Layer	Value
1	2.56 to 2.94
2	2.94 to 3.12
3	3.12 to 3.23
4	3.23 to 3.40
5	3.40 to 3.57

Benefits and Limitations

Concept maps help to refine and share understanding of key concepts of a program theory, which is particularly relevant in relation to key mechanisms or outcomes. As illustrated in the real-world applications below, concept maps also can be used to visualize changes in thinking or attitudes before, during, and after program participation (see Exhibit 23). Some evaluators use concept maps as a preliminary step towards developing more advanced hybrid models (see **Exhibit 33** in [Nested and Hybrid Models] for an example).

One practical limitation of concept maps is that they require qualitative or quantitative data collection on the concepts of interest. Another limitation of concept maps stems from the technical capabilities needed to develop the maps. A broad array of software has been developed over the years. Still, developing a concept map requires more time and resources and technical know-how than do logic models, logframes, or theories of change.

Promoting Equity

Program theories most often are developed by evaluators based on program documents. If developed in a participatory manner, program theories also can be based on input from program staff and funders (stakeholders). If community members or other partners are included, it is often those in positions of power.

This is problematic because a program theory they develop will not reflect the values and experiences of the people and communities most affected by the program (recipients, participants, frontline staff).

If we are to promote equity in our program theories, we must shift power to those who are most affected by the programs.

Importantly, *shifting* is not just about including a broader range of stakeholders. Simply including people most affected by the program along with funders and program staff likely will not change existing power dynamics or influence the program theory.

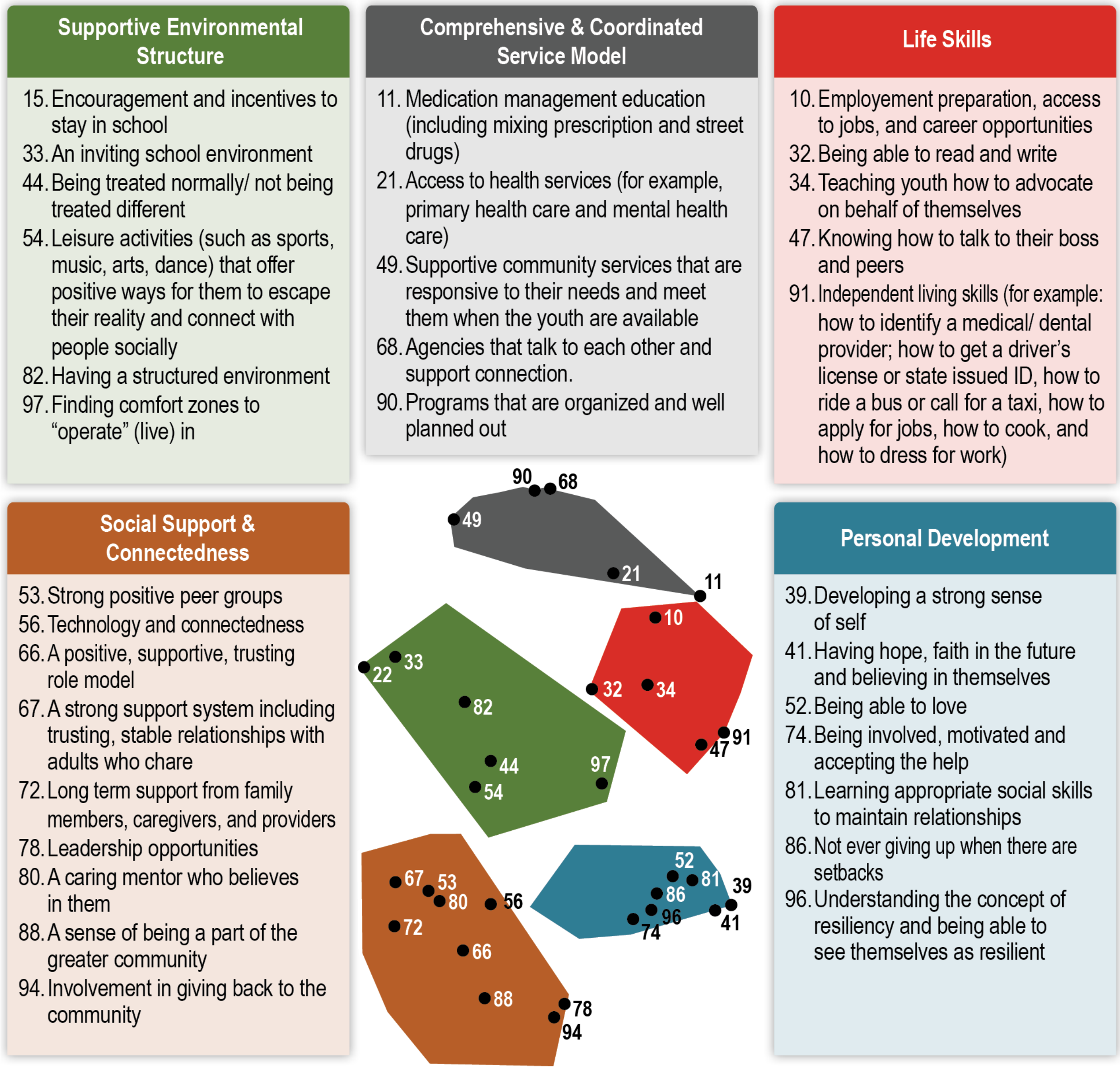
Carefully consider how and with what purpose people are included in the development process. Do so to ensure the different groups feel comfortable engaging with the process and the diversity of their perspectives is genuinely reflected in the program theory that results.

Real-World Applications

In **Exhibit 22**, the authors developed a concept map as part of evaluating a program aimed at strengthening the cross-sector system of care for youth with severe emotional disturbances. The authors applied a *group* concept mapping approach to understand the experience of participants in the program; that is, how participating affected youth's successful transition to adulthood.

Across all brainstorming sessions, program participants generated a total of 830 statements. These statements were synthesized using the Keywords in Context software and structured into a five-cluster solution. The numbers inside each cluster represent a statement about factors affecting program success. Statement numbers in bold were rated above average in importance by the program participants. The map provides an overview of the five clusters of program factors affecting youth transition and, within these, the specific factors that are particularly relevant. The individual statements are spelled out in the text boxes—one text box for each cluster of statements. For example, within the grey cluster *Comprehensive & Coordinated Service Model*, statements on *medication management, access to health services, supportive community services that are responsive, agencies that talk to each other, and programs that are organized and well planned out* were identified as particularly important components for successful transition to adulthood. Expanding on their concept map, the authors also developed a stock and flow diagram (see **Exhibit 19** in Stock and Flow Diagrams) and a hybrid causal loop/stock and flow diagram (see **Exhibit 33** in Nested and Hybrid Models) for the program.

Exhibit 22. Concept Map with Participant Statements about Program Success



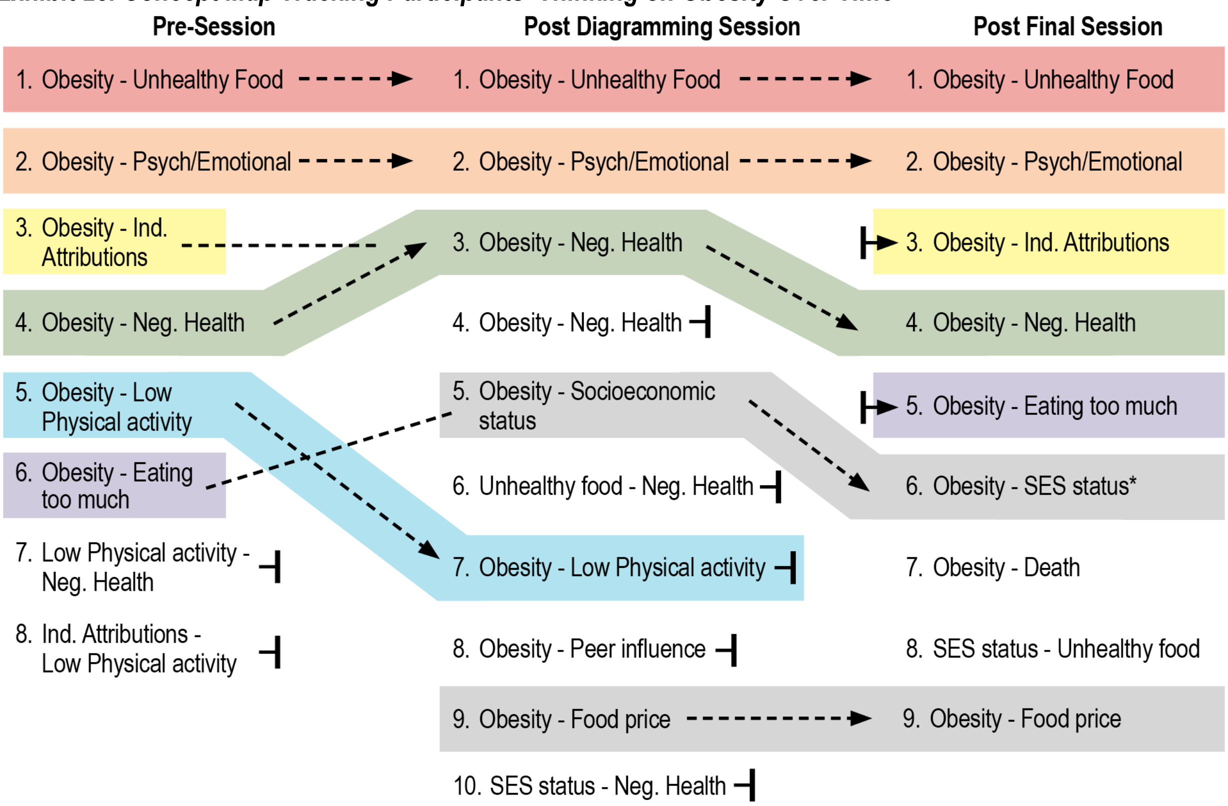
Source: Adapted from Lich et al. (2017)

The concept map in **Exhibit 23** was developed as part of an evaluation of adolescents' conceptual understanding of obesity before and after participating in a series of workshops. As part of the workshops, youth engaged in a mind mapping activity, where each youth developed an individual mind map of their own thoughts on obesity. The authors recorded and organized these individual mind maps in Excel spreadsheets and imported these into R statistical software, using the packages *igraph* and *tnet* to visualize the concept map.

The resulting concept map summarizes how the participants' thinking on obesity changed over the course of the program workshops. The three columns in the concept map represent the three discussion sessions: before, during, and after program completion. The rows in the map figure present the themes that most participants connected across the mind mapping sessions (before, during, and after the final workshops). They list themes from most to least common from top to bottom. Colors highlight the changes in the ranking of the pairs across sessions. Themes without a color appeared only once.

The dotted arrows indicate connected themes that continued to occur across sessions. The "⊣" symbol indicates themes that ceased to occur in subsequent sessions. To illustrate, the theme of *Obesity* related to *Low [Physical Activity]* was ranked fifth during the pre-session, ranked seventh during the post-diagramming session, and ceased to be a relevant theme after the final session.

Exhibit 23. Concept Map Tracking Participants' Thinking on Obesity Over Time



Source: Adapted from Frerichs et al. (2018)

Resources

FreeMind is freeware software for concept mapping that includes graphics to enter and arrange relationships between and among ideas and related concepts. See https://freemind.en.softonic.com/.

The Visual Understanding Environment (VUE) is freeware software that can organize and visualize content and concepts as well as connections between these. VUE also allows for connectivity matrices to be imported to statistical packages. See: https://vue.tufts.edu.

8. NETWORK MAP

Description

Network maps display relationships between people, organizations, themes, or other entities in the form of spiderweb-like diagrams. In their most basic form, network maps show whether a relationship exists, typically through the presence or absence of connecting lines (called *paths*) between two people, organizations, or themes (called *nodes*). More detailed network maps also can show the quality, strength, and directionality of a relationship, the distance between two entities, or in what type of context relationships occur.

Key Network Map Terms

Node refers to the people/places/things that comprise the network, often taking the visual form of circles.

Density refers to the number of connections divided by the total possible connections in the network.

Connectedness refers to the frequency of which one node is related to other node(s).

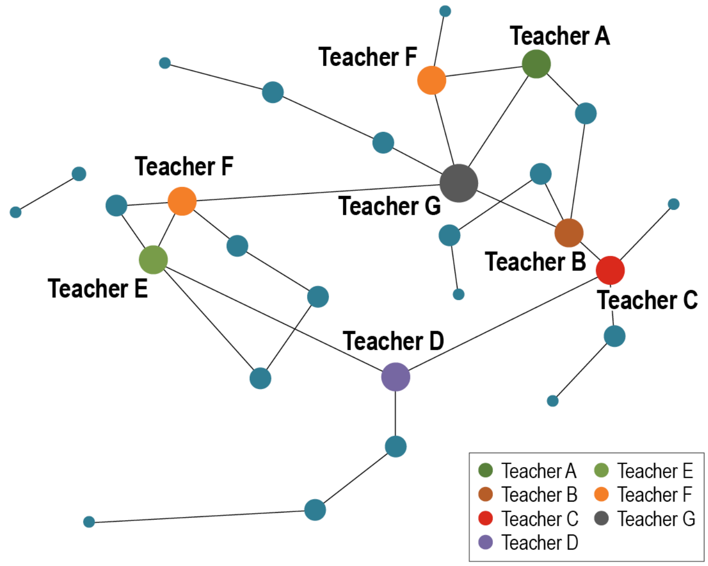
For program theories, network maps can be used to depict program outcomes related to social networks, partnerships, collaborations, social capital, organizational cohesion, and interconnectedness.

There is no common visual template for network maps. They typically display the following:

- **Type/category of node**: using multiple node shapes or icons (e.g., circles to indicate students, squares to indicate teachers; apples to indicate the use of a food assistance program)
- **Direction**: using arrows for paths instead of simple lines
- **Strength**: dashed lines for weaker relationships, thick lines for stronger relationships; background colors (e.g., placing stronger relationships on a more saturated background color, weaker relationships on a less saturated background color)
- **Context**: overlaying shapes that contain parts of the network (e.g., all students within one school exist within a square, all students within another school exist within a circle); background colors
- **Positive/negative relationship**: color-coding each type of relationship

A generic network map for the teaching assistant program is illustrated in **Exhibit 24**. The teachers are represented by circles. The size of the circles denotes the number of professional relationships for that teacher within the program. The eight teachers with the highest number of connections are labeled (“A” through “G”).

Exhibit 24. Network Map—Teaching Assistant Program



Benefits and Limitations

Network maps are most effective as a tool to visualize the extent to which and how individuals, organizations, or programs are connected. Visualizing a program with a network map can inspire the exploration of direct or indirect pathways between nodes to identify promising pathways for generating change. Network maps can also help identify breakdowns in connections that can limit desired changes from taking place. For example, evaluators can benefit from using network maps to identify areas of an organization that are siloed or that are communicating well. Network maps can therefore inspire action or eliminate bottlenecks. Network maps also can help users visualize how adding and subtracting a path may affect staffing or resource needs.

While it is ideal for showing how parts of an organization fit together, a network map is unlikely to explain *why* relationships between nodes exist and *how* these relationships can best be changed. A large amount of text in a network map can interrupt the flow of the diagram since these network maps tend to leave little space for the multiple inputs, activities, and processes displayed in a typical logic model.

It also can be challenging to create a network map with large datasets if the focus is on individual relationships. Network maps with many data points more easily can show clusters of data, but the paths in between nodes can become difficult to see or would take up too much visual space.

Promoting Equity

Program theories tend to frame program participants as a homogenous group by depicting activities, mechanisms, and outcomes as uniform for all participants.

If we are to promote equity in our program theories, we need to pay more attention to how the program theory might look different for different subgroups of the program participants. We should bring the reader's attention to people who have been marginalized or excluded based on race, ethnicity, gender, age, ability, sexual orientation, and other dimensions.

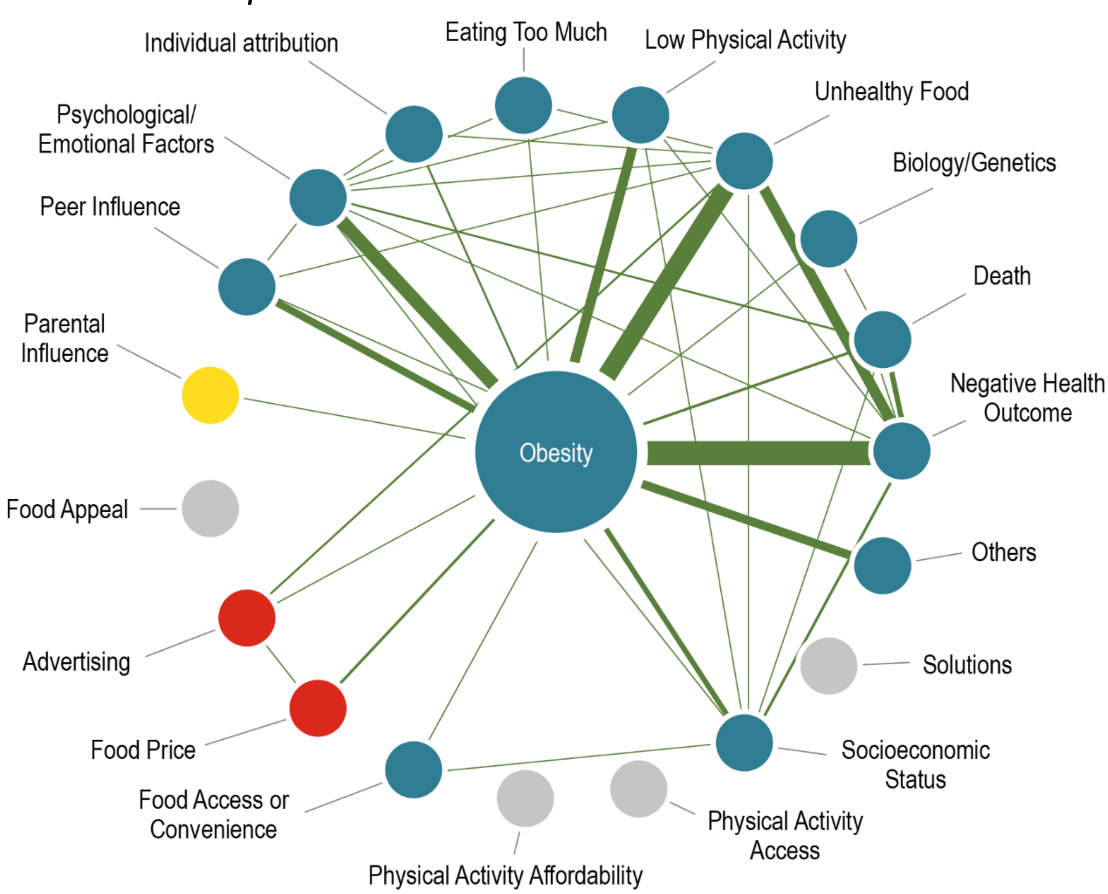
Real-World Applications

In **Exhibit 25**, the authors developed a network map of adolescents' conceptual understanding of obesity before and after participating in a series of workshops. At different stages of the program, the adolescents created mind maps of their thinking on obesity. The network extends on a concept map developed by the same authors (see **Exhibit 23** in section on Concept Maps). The authors used R statistical software and the packages *igraph* and *tnet* to calculate network measures and develop the network map.

The weight of the lines in the network map indicates the extent to which program participants were connecting specific pairs of subthemes with obesity. For example, *Negative Health Outcome, Unhealthy Food,* and *Psychological/Emotional Factors* were prevalent themes in the participants' mind maps on obesity. The color of the nodes denotes groups of cohesive concept subthemes identified by the statistical analyses. For instance, program participants often described *Food Price* and *Advertising* in proximity.

The network map provides an overview of the most salient themes and how these are connected. The authors developed several network maps at different stages of the program, allowing them to compare changes over time in the participants' thinking on obesity.

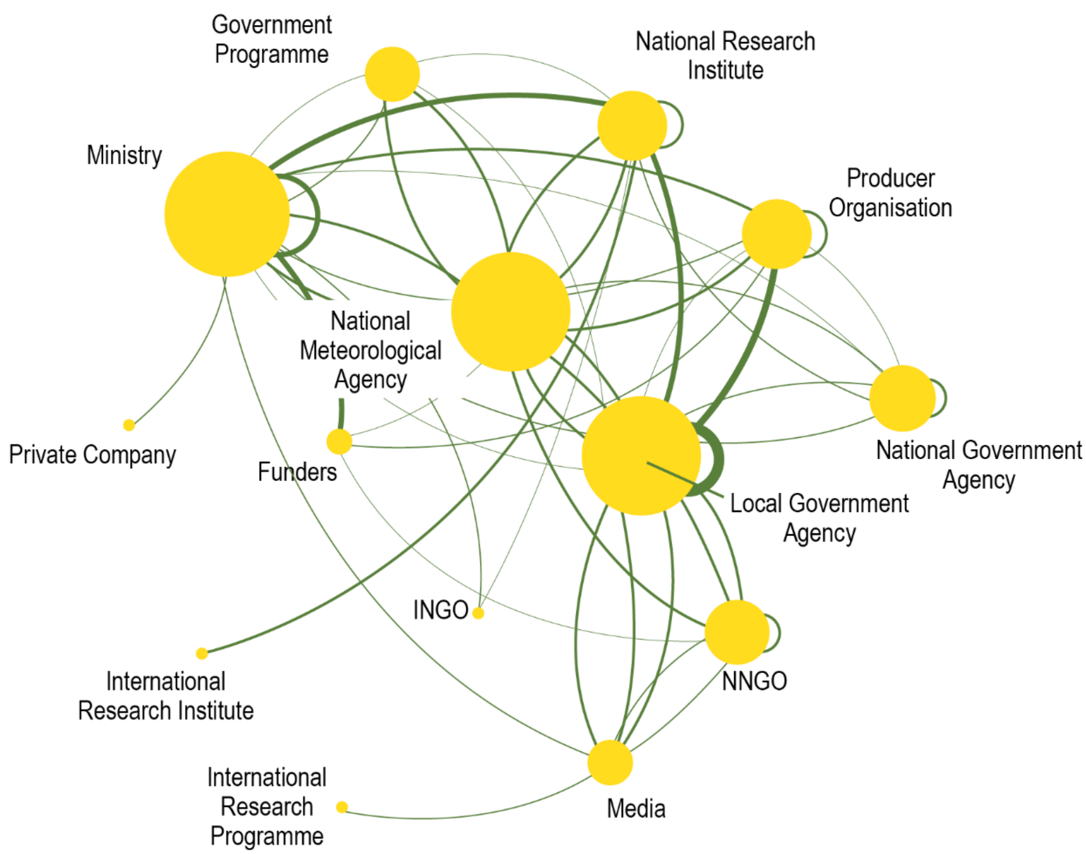
Exhibit 25. Network Map with Color-Coded Subthemes



Source: Adapted from Frerichs et al. (2018)

The network map in **Exhibit 26** depicts a network of organizations collaborating in the production, transmission, and use of weather and climate information for agriculture in Senegal. The network map focuses on degree of collaboration among the organizations: The bigger the circle representing each organization, the larger the number of other actors with which it collaborated during the program period. The green lines indicate the extent to which specific organizations collaborated. The color and thickness of the lines indicates the degree of collaboration. As the map shows, the *National Meteorological Agency* is both large and centrally located in the network, indicating its key role as a connector to local institutions.

Exhibit 26. Network Map with Color-Coded Lines for Degree of Collaboration



Source: Adapted from Blundo-Canto et al. (2021)

Resources

UCINET is an easy-to-use software for social network analysis. The program can be downloaded and used for free for 90 days. See: https://sites.google.com/site/ucinetsoftware/home.

9. PATH MODEL

Description

The purpose of a path model is to visualize how program activities are statistically associated with specific outcomes. More developed path models also include contextual conditions—within which the program is embedded—that can influence the ability of the program to generate the desired outcomes.

Structural Equation Modeling

Path models are often developed using *Structural Equation Modeling* (SEM), a set of statistical analysis techniques that examines structural relationships among variables. Stated in non-technical terms, SEM explores how variation in program activities influence variation in outcomes.

The degree to which an activity varies in correlation with an outcome is quantified and presented in the form of a *correlation coefficient*, ranging from 0 (no correlation) to 1 (perfect correlation). A high correlation coefficient indicates a strong relationship between an activity and an outcome.

Path models usually consist of the same basic building blocks as a theory of change: inputs, activities, outputs, and outcomes. These are visually presented in a diagram and accompanied by a narrative description. The path model is structured around boxes for activities, outputs, and outcomes, with line arrows (assumptions) indicating how these are connected. Path models also include correlation coefficients for the line arrows, indicating the degree which an activity varies in correlation with an outcome.

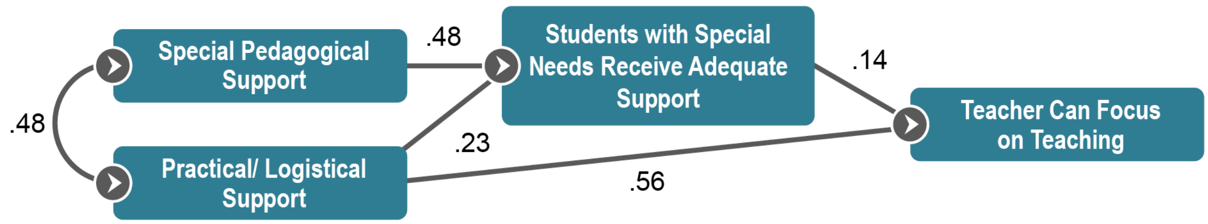
Benefits and Limitations

The main benefit of a path model is that it can help visualize the relative strength of program activities' influence on outcomes, while holding constant the relative influence of other factors, (e.g., contextual conditions). Another benefit is that path models can include *mediators*, such as an intermediate outcome that accounts for an observed relation between an activity and another outcome. In this way, path models can facilitate a better understanding of how different activities interact and influence the outcomes of interest, which may in turn inform future decisions about program design and implementation. It is important to note that path models do not support causal claims about the relationship between program activities and outcomes. As the old adage goes, correlation is *not* causation.

One limitation is that path models require specialized technical skills and software. A broad array of software can be used for developing path models (SPSS Amos is arguably the most user-friendly software). Still, developing a sensible path model requires more time, resources, and technical know-how than do other program theories.

Exhibit 27 illustrates a generic path model for the teaching assistant program. The path model depicts connections between specific activities and outcomes using arrows. Correlation coefficients indicate the strength of relationship between the different activities and outcomes comprising the path model.

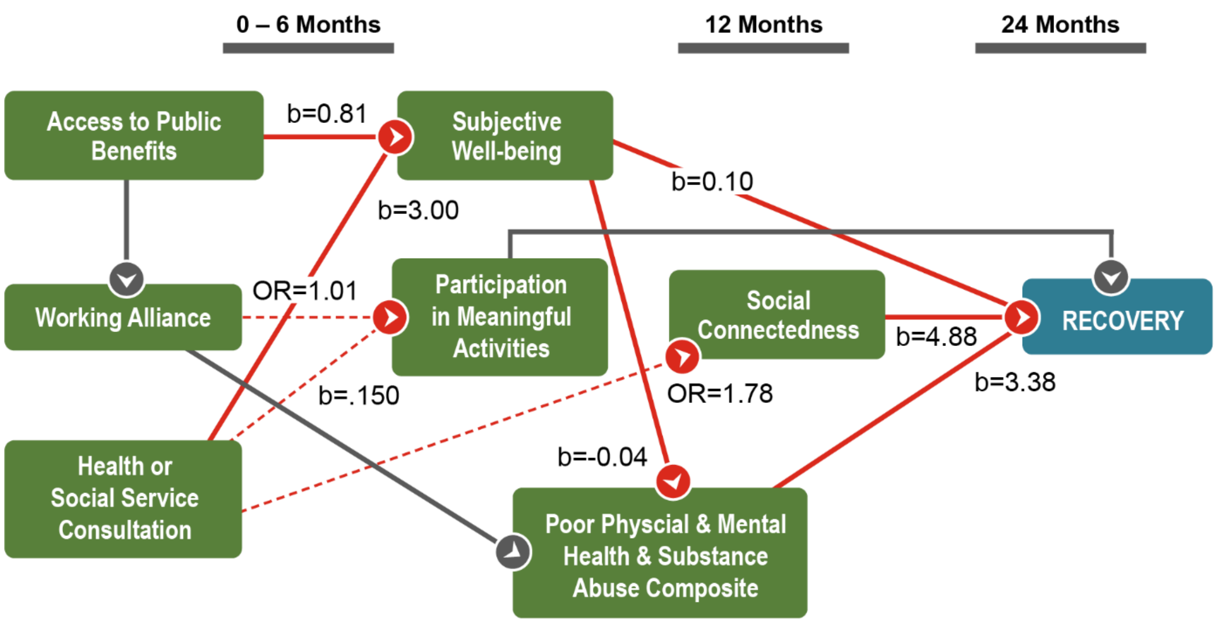
Exhibit 27. Path Model—Teaching Assistant Program



Real-World Applications

Exhibit 28 presents a path model for an evaluation of a Housing First program (At Home/Chez Soi) in Canada. The authors first developed a theory of change for the program, then used structural equation modeling (SEM) to develop the path model. In the path model below, dashed lines represent pathways to a binary outcome (e.g., *participation in meaningful activities*), and solid lines represent paths to continuous outcomes (e.g., *subjective well-being* and *recovery scores*). Informed by the analyses, the authors presented each statistically significant connection in red. They also provided unstandardized beta coefficients and odds ratios, indicating the strength of relationship between the different components comprising the theory of change. The model accounts for the temporal sequence of the outcome changes.

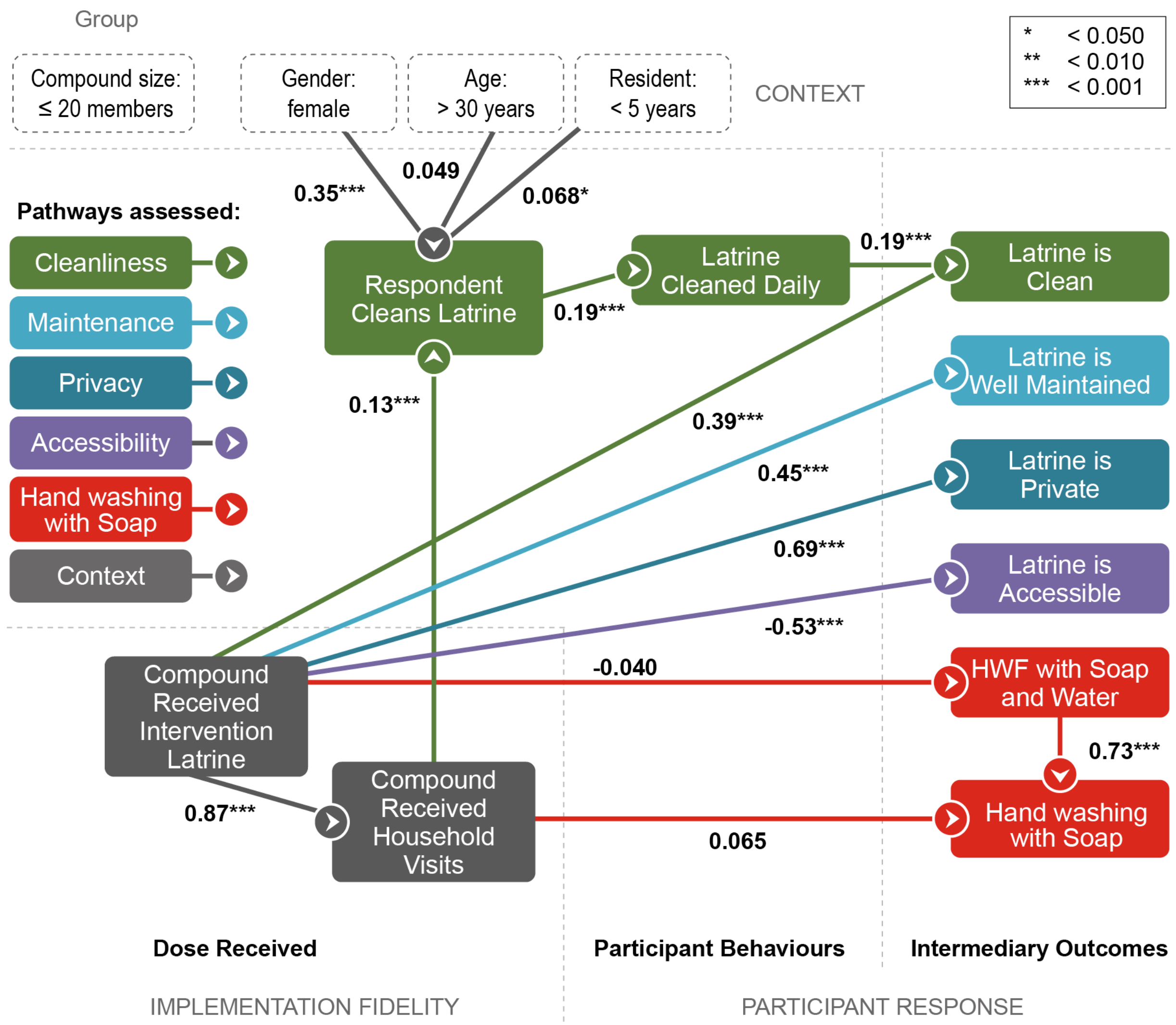
Exhibit 28. Path Model with Color-Coded Lines for Statistically Significant Connections



Source: Adapted from O'Campo et al. (2022)

Exhibit 29 presents a path model for a sanitation program (MapSan) in Maputo, Mozambique. Grounded on a theory of change, the path model summarizes findings of a quantitative process evaluation assessing program implementation (e.g., *compound received latrine*), participant behavior (e.g., *latrine cleaned daily*), and impacts on hypothesized intermediary outcomes (e.g., *latrines are clean*). The hypothesized pathways (e.g., accessibility, cleanliness, maintenance, privacy, handwashing, and context factors) are color-coded. The model includes correlation coefficients to indicate the strength of relationship between different program components (compound received household visits), short-/medium-/long-term outcomes (e.g., respondent cleans latrine, latrine cleaned daily, and latrine is clean), and contextual factors (e.g., gender, age, and years of residence). Statistical significance of the coefficients is presented using asterisks.

Exhibit 29. Path Model with Color-Coded Pathways



Source: Adapted from Bick et al. (2021)

Resources

SPSS Amos is software package for structural equation modeling (SEM). See: https://www.ibm.com/products/structural-equation-modeling-sem.

10. NESTED AND HYBRID MODELS

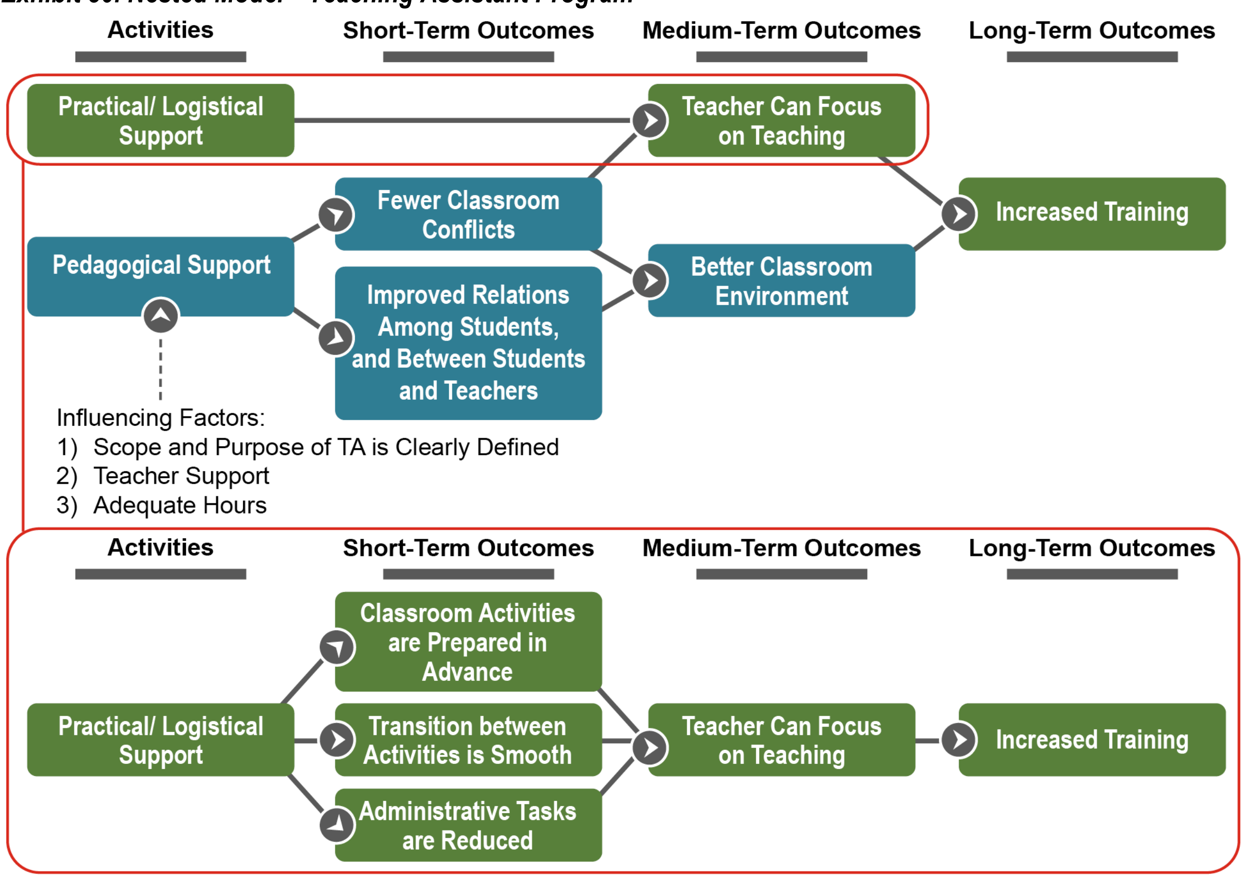
Description

In addition to the different types of models presented in the preceding pages, there is of course the possibility of mixing different types of models. Each type of program theory provides unique strengths that, when used together, can add value to program planning and evaluation. In what follows, we will consider two variants: nested models and hybrid models.

Nested (or layered or stacked) models refer to the use of multiple program theories to depict different aspects of a program. For instance, in the context of realist evaluation, it is not uncommon for evaluators to use an initial overarching theory of change as the starting point for developing one or more separate context-mechanism-outcome (CMO) configurations for specific aspects of the theory of change.

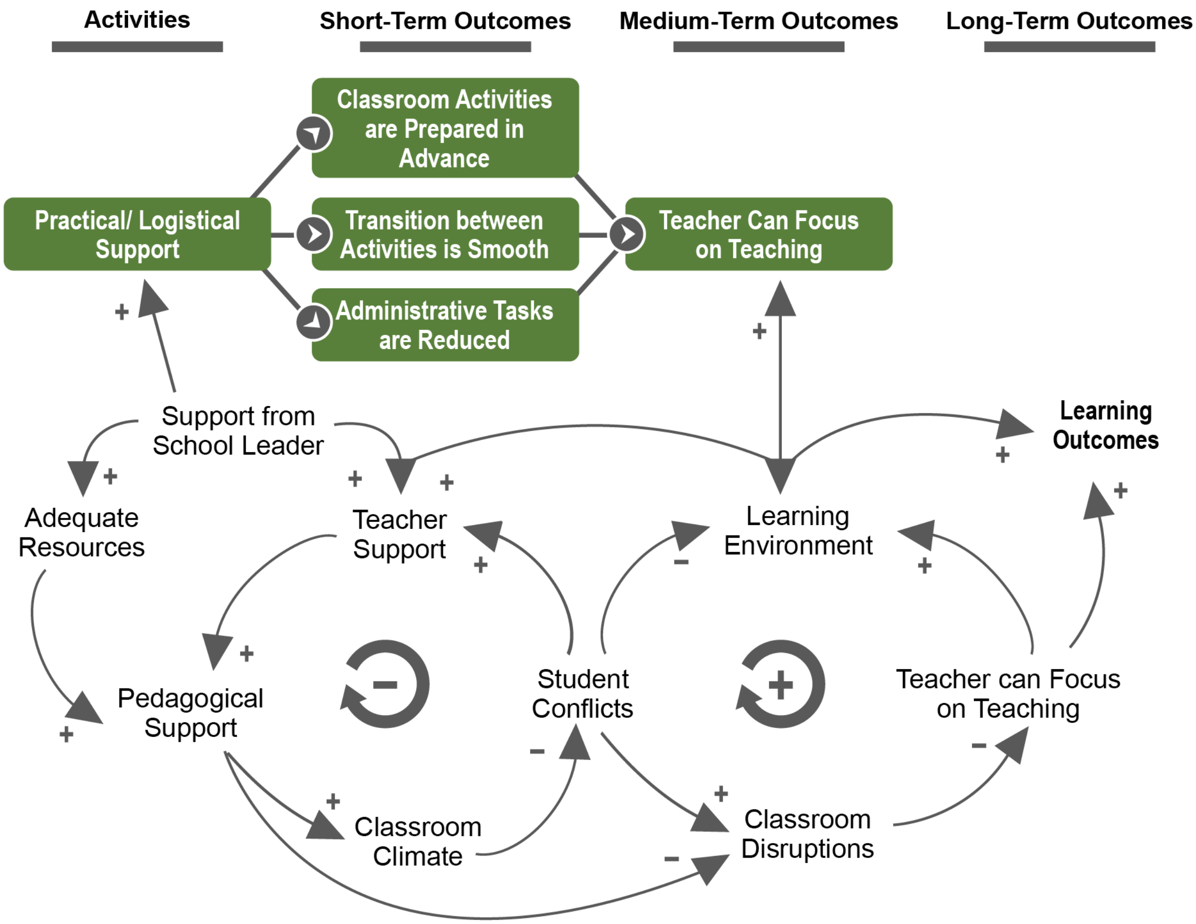
Exhibit 30 illustrates this idea for the teaching assistant program. The overarching theory of change (from **Exhibit 7**) is now paired with a new, separate theory of change (boxed in red) that brings further detail to the connection between *Practical/Logistical Support* provided by the teaching assistant and its effect, *Teacher Can Focus on Teaching*.

Exhibit 30. Nested Model—Teaching Assistant Program



Hybrid (or blended) models integrate different types of models into a single program theory. Blending different models allows us to examine and understand programs in different, but complementary, ways. As illustrated in **Exhibit 31**, integrating causal loop diagraming in a theory of change can enhance the specificity of the causal structure underlying a specific aspect of the program. In this example, the causal strand relating *Pedagogical Support* to teachers' ability to focus on teaching has been expanded on, using causal loop diagramming.

Exhibit 31. Hybrid Causal Loop/Stock and Flow Diagram—Teaching Assistant Program



Many other hybrid models can be pursued. Freer and Lemire (2019), among others, have illustrated the benefits of integrating logic models and theories of change. As indicated in the real-world examples below, there are of course many other possible hybrids to be pursued.

Benefits and Limitations

Nested and hybrid program theories allow for different levels of specificity and focus—which can be useful if you are trying to differentiate the focus of your evaluation on specific aspects of the program. Nested models are particularly useful in the context of large-scale programs—which might have multiple implementation sites, each of which could be modeled separately. den Heyer (2002) also proposes the temporal theory of change, a type of nested model where separate theories of change depict how program context changes, modifications, and both intended and unintended results change over time. The temporal theory of change is similar to the approach taken in adaptive programs, where an initial “best guess” theory of change is developed, and then periodically tested and revised based on contextual changes and additional evidence such as implementation experience or evaluation findings. Finally, nested theories of change can be used to develop theories of change for different subgroups of program participants, an important equity consideration.

Hybrid models are particularly advantageous in the context of complex or large-scale programs, where the ability to differentiate the modeling strategy facilitates a more nuanced depiction of the program. By combining different types of models, the developed program theory can be tailored to the specific needs of the program or evaluation.

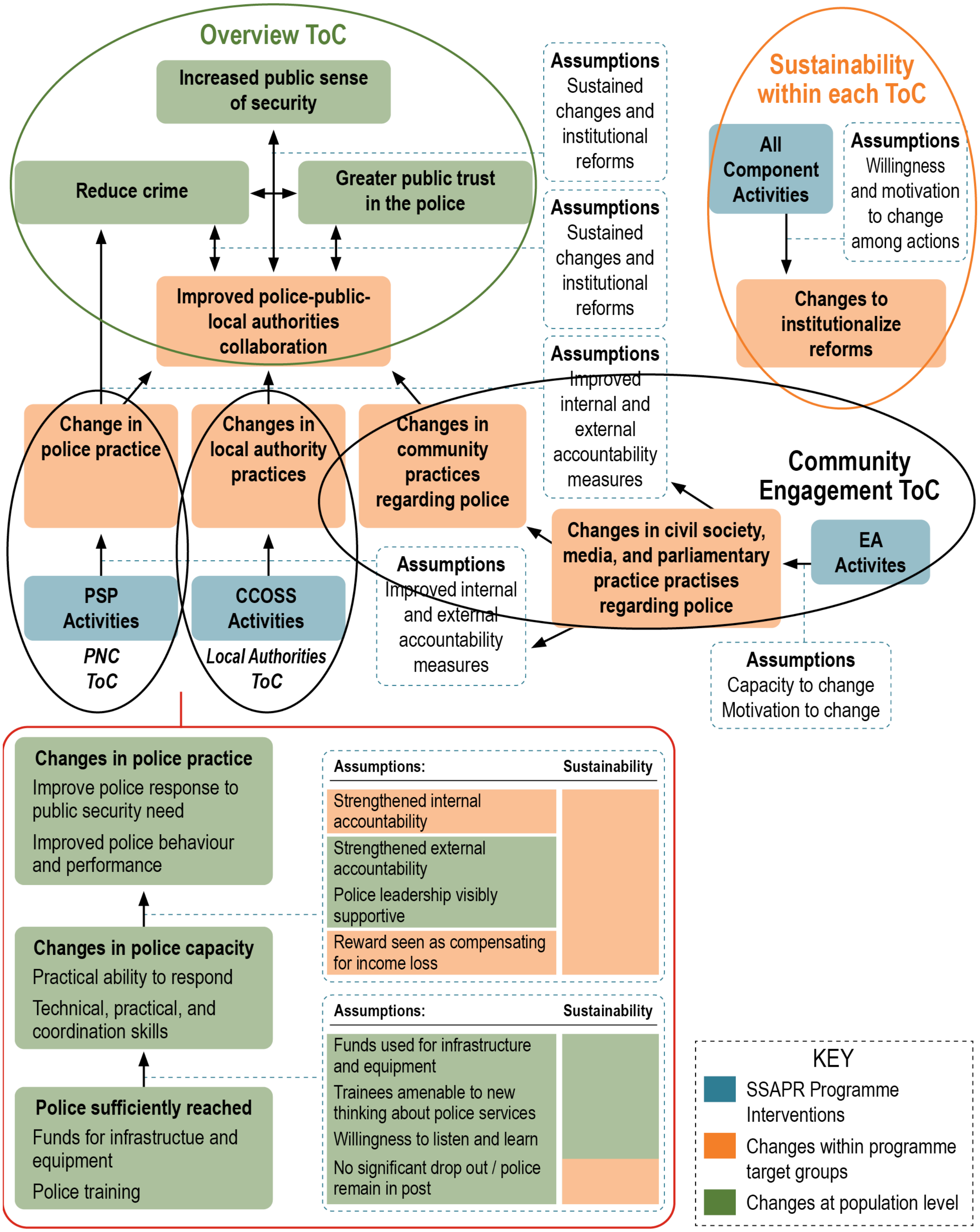
One potential limitation of hybrid models is that the level of complexity of the resultant program theories can be overwhelming to people unfamiliar with the modeling techniques, especially if the hybrid involves advanced modeling techniques such as causal loop and stock and flow diagrams (see Exhibit 33 below for an example). The trade-off between capturing program complexity and developing a program theory that is accessible to non-technical audiences should be carefully considered when using hybrid models.

Real-World Applications

Exhibit 32 displays a nested theory of change that was part of evaluating a police-reform program in the democratic Republic of Congo. The program consisted of three key activities, each of which is marked in the overarching theory of change by a black circle. The authors also identified and spelled out the most salient assumptions.

The authors developed separate and more detailed theories of change for these three individual, yet connected, program activities, to raise the level of specificity in the theory of change. An example is provided in the red box, further detailing the causal logic and assumptions underlying the *police training (PSP) activities*.

Exhibit 32. Nested Theories of Change



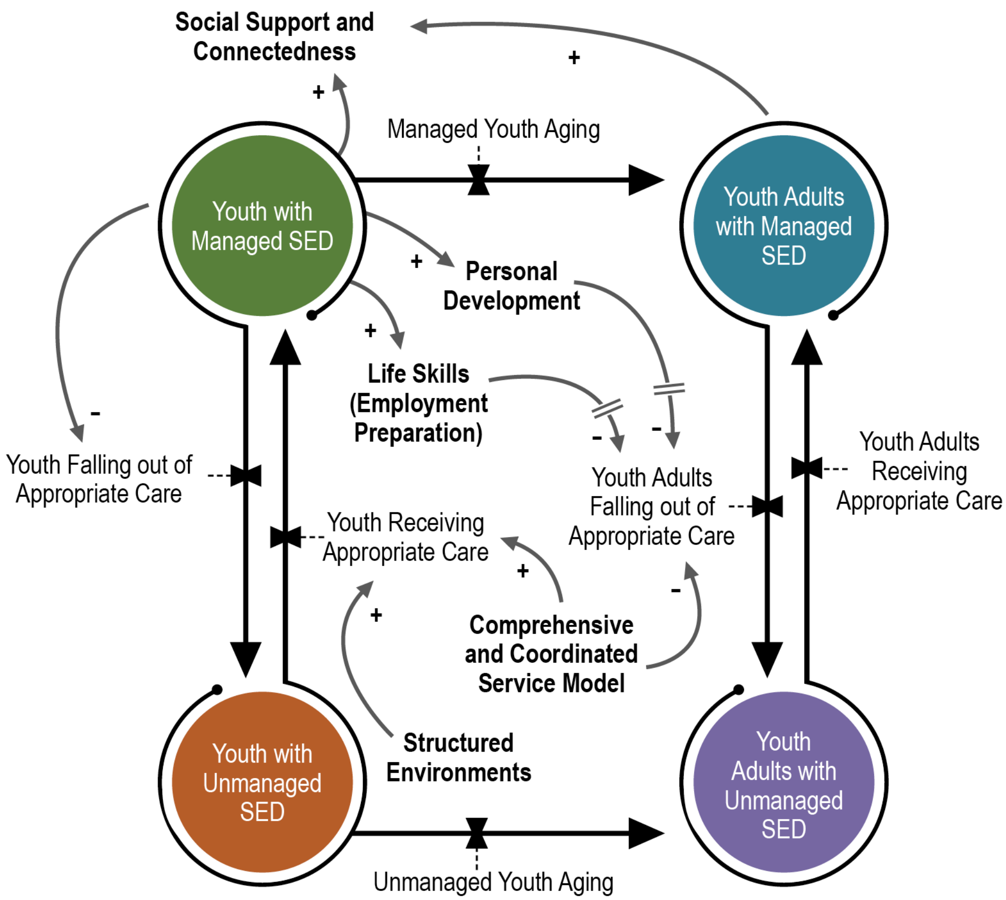
Source: Adapted from Koleros and Mayne (2019)

The hybrid model in **Exhibit 33** is a stock and flow diagram enhanced by integrating findings from a concept map, using causal loop diagramming. The original stock and flow diagram is **Exhibit 19** (Stock and Flow Diagram); the concept map is **Exhibit 22** (Concept Map).

The integrated hybrid model in **Exhibit 33** provides further detail on the original stock and flow diagram by including the five clusters of program factors affecting successful youth transition, which were identified in the concept map described in **Exhibit 22**. Below, these factors are presented in the grey boxes.

The hybrid model links these factors with stocks and flows using causal loops. The blue single-lined arrows indicate how either strengthening or degrading the factors are believed to alter flows most significantly among transition-age youth over time. Following convention, the authors use plus or minus signs to indicate the polarity of the effect. A plus sign indicates that the two variables move in the same direction—an increase (or decrease) in the first leads to an increase (or decrease) in the next. A minus sign indicates that the two variables move in opposite directions. The authors also used double bars || to indicate delayed effects.

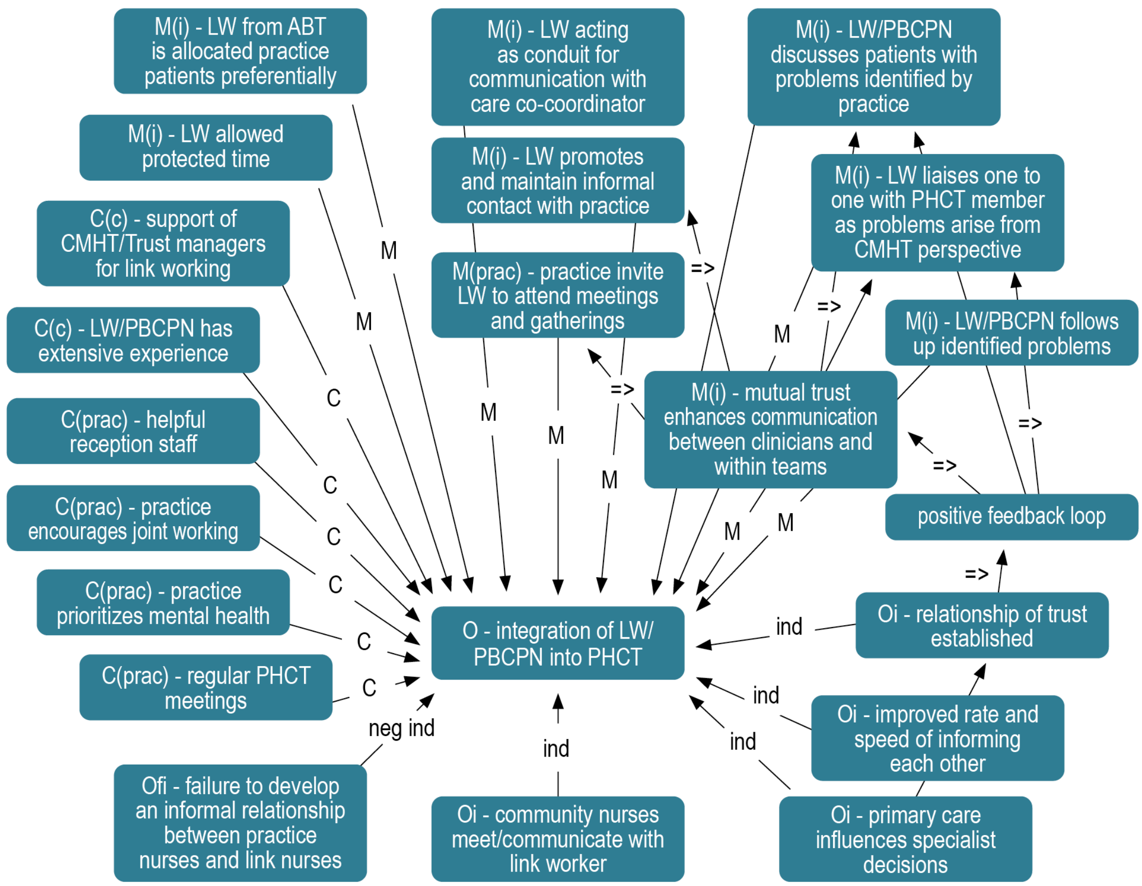
Exhibit 33. Hybrid Causal Loop + Stock and Flow Diagram



Source: Adapted from Lich et al. (2017)

Exhibit 34 illustrates a hybrid CMO–causal loop diagram of a program to improve primary healthcare for patients with long-term mental illness. The authors included feedback loops in their model and used an elaborate system of codes and labels to denote different types of context factors, mechanisms, and outcomes, as well as their links with the higher-level outcome for the program.

Exhibit 34. Hybrid CMO + Causal Loop Diagram



NOTATION FOR CODING	
C(c)	community mental-health team context
C(gp)	context relating to general practitioner
C(i)	interface context
C(prac)	practice context
M(i)	mechanism acting at interface
M(prac)	mechanism acting at practice
MM	misisng mechanism
MM(iMHL)	missing mechanism related to Mental Health Link Intervention
O	higher-level outcome
Ofi	explicit absence of indicator of outcome
Oi	indicator of higher-level
Ou	unexpected outcome

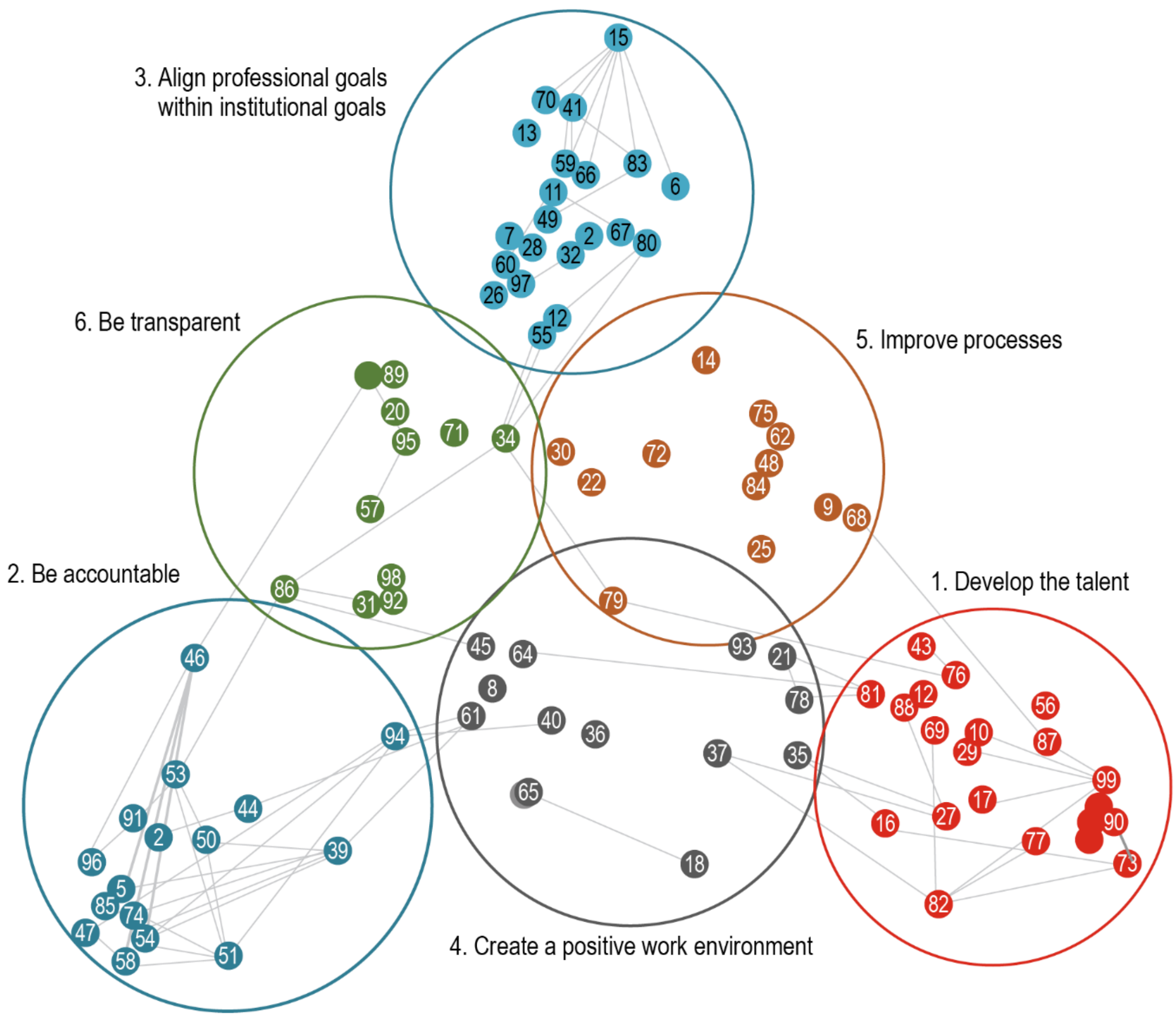
LABELS FOR LINKS	
C	context contributing to
M	mechanism contributing to
Ab-M	absent mechanism so no contribution to
Cneg	negative context contributes to failure to bring about an outcome
ind	indicator of
neg ind	indicator not achieved
=>	leads to

Source: Adapted from Byng, Norman, and Redfern (2005)

Authors of **Exhibit 35** applied social network analysis on a concept map for a leadership training program. To apply social network analysis, they imported data from the concept mapping analysis into UCINET, freeware for network analysis.

The map is structured around six clusters—or themes—related to leadership. The numbered dots correspond to specific statements about leadership. The line between a pair of dots indicates a relatively high connection between the statements. As shown in the map, clusters vary in their visual density. Some clusters have many connections among the items within the cluster and appear dense (e.g., align professional goals with institutional goals); other clusters have few interconnections and appear sparse (e.g., improve processes). The latter reflects less agreement about the connectedness of the statements among the respondents.

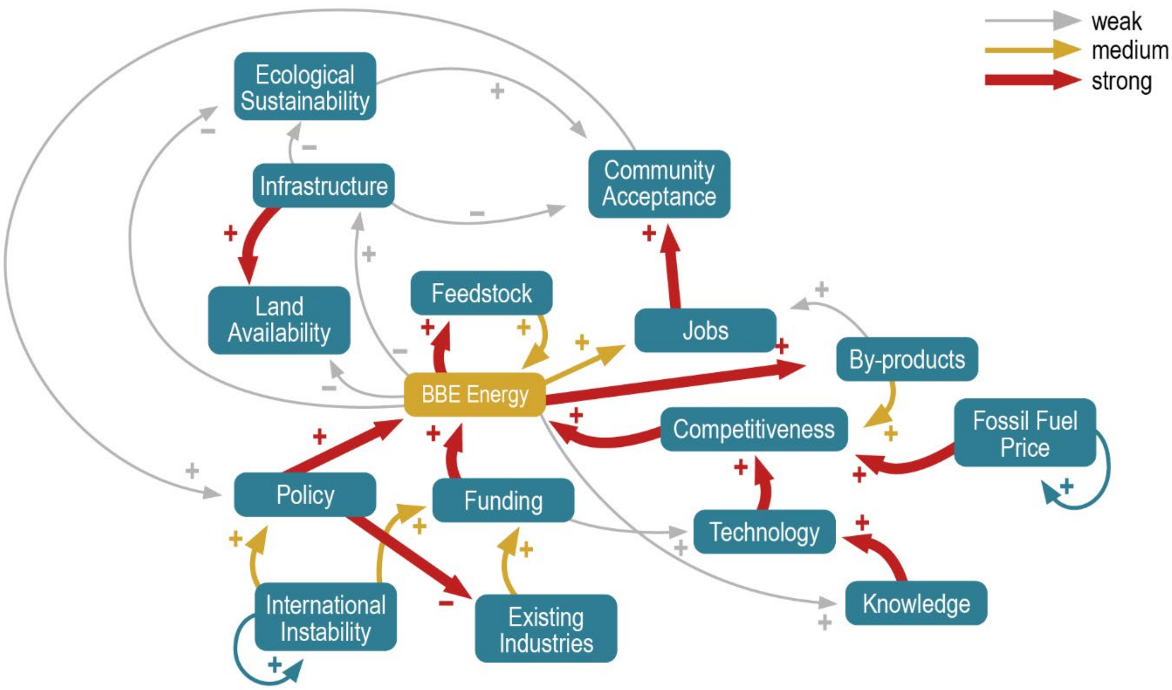
Exhibit 35. Hybrid Concept Map + Network Map



Source: Adapted from McLinden (2013)

The hybrid system map in **Exhibit 36** depicts factors (physical, political, social, economic) that stakeholders considered important in the development of a bio-based economy in England. The map combines visual elements of a theory of change and causal loop diagrams. The map was based on stakeholder input from a workshop, followed by regression-based modeling. The color and thickness of the lines denote the strength of the influence—weak, medium, strong—between the different factors.

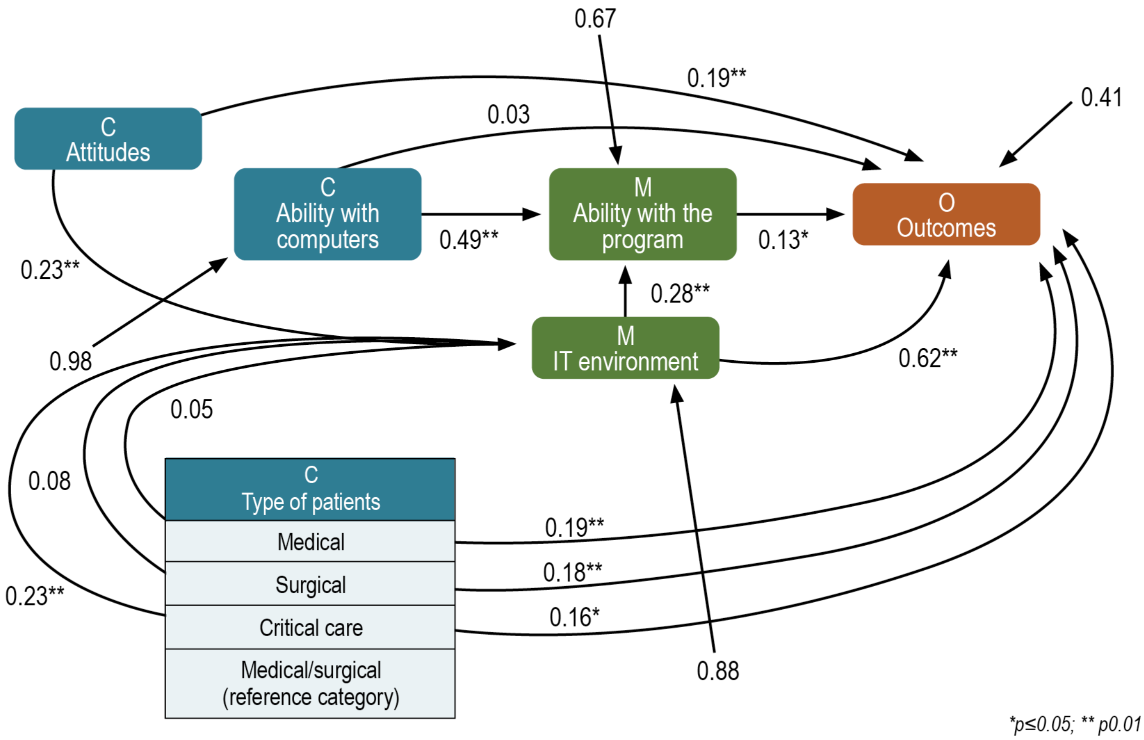
Exhibit 36. Hybrid Theory of Change + Causal Loops



Source: Adapted from Penn et al. (2013)

Exhibit 37 illustrates a CMO configuration integrated with a path model. This hybrid model was developed as part of an evaluation of the impact of a computerized information system on nurses' clinical practice. The authors first constructed a hypothetical model based on survey data and informed by significant (bivariate) relationships among context, mechanism, and outcome variables. The hypothesized model was then tested using SEM to assess the simultaneous relationships among the variables. The resultant model illustrates that in the CMO configuration, certain contextual factors (types of patients) primarily influence the program outcomes, as opposed to the mechanisms. The path coefficients (standardized regression coefficients) are numerical values that indicate the relative strength of correlation between two variables. The number of asterisks for each coefficient indicates whether the correlation is statistically significant.

Exhibit 37. Hybrid Context-Mechanism-Outcome Configuration + Path Model



Source: Adapted from Oroviogoicoechea and Watson (2009)

PRACTICAL PRINCIPLES FOR DESIGNING PROGRAM THEORIES

There is of course no single "correct" way to design program theories. The benefits and limitations of different modeling techniques presented in the preceding pages are contingent on the purpose and intended use of the program theory. With this contextual caveat as our backdrop, the position we take is that practical principles for more purposeful program theories can still be articulated and realized.

The following practical principles are informed by the existing literature on program theories as well as by the authors' collective experience developing and using program theories. Our modest hope is that the principles—individually or collectively—can guide towards more purposeful program theories.

Principles for Designing Program Theories

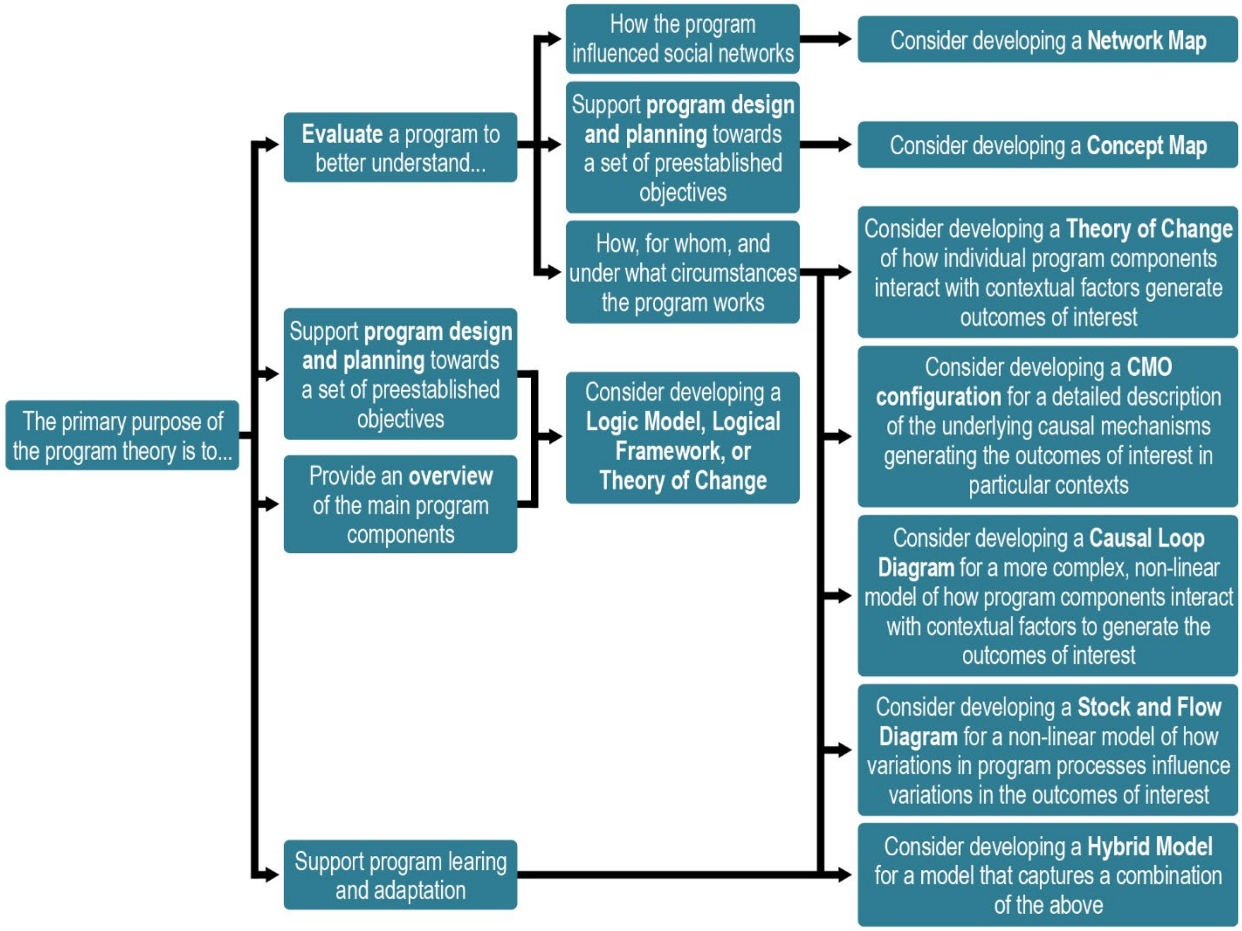
1. Be clear on purpose and use
2. Focus the program theory
3. Distinguish between complicated and complex programs
4. Consider the type of complexity to include in your program theory
5. Capture context
6. Consider outcome trajectories
7. Unpack assumptions
8. Center equity and transformative change in the program
9. Do no harm—Consider unintended and adverse outcomes
10. Mix and match visual techniques

Principle 1: Be clear on purpose and use.

As the saying goes: If you don't know where you are going, you might wind up someplace else. An important first step in developing a useful program theory is to clarify its intended purpose and use. Purposeful program theorizing entails engaging with relevant stakeholders about their information needs. For instance, if the purpose of the program theory is to capture relatively complex causal relationships between program components and outcomes, *causal loop* and *hybrid models* are likely preferable. However, if the purpose is to provide an accessible overview of the main components of a program, a *logic model* can be a good option.

Exhibit 38 provides a decision tree for selecting an appropriate program theory type. In addition to the primary purpose, deciding on the type of program theory to be developed also requires consideration of time and resources, as well as data availability.

Exhibit 38. Decision Tree for Selecting a Program Theory Type



Principle 2: Focus the program theory.

Program theories necessarily simplify programs. Depicting programs in their totality—capturing all aspects of the program and the context within which it is embedded—is a futile exercise. A central step in developing purposeful program theories is to carefully select the most salient aspects of the program to be examined.

One useful strategy for drilling down is to differentiate the level of specificity across the program theory, offering more detailed descriptions of select aspects. Towards this aim, we propose the following criteria for deciding where to drill down:

1. *Core components*. Are there program components that define what the program is and/or constitute the primary driver for program impact?

2. *Critical assumptions*. Are there critical assumptions in the program theory, such as causal connections (arrows) and mechanisms that are necessary for the program to be successful and/or vulnerable to be influenced negatively by external factors? Actively examining assumptions around social norms and belief systems is critical for promoting equity.

3. *Stakeholder/client relevance*. Which aspects of the program theory are particularly relevant for stakeholders or clients?

4. *Potential for new learning*. Which aspects of the program theory hold the greatest promise of new learning?

5. *Adverse consequences*. Are there aspects of the program theory that could potentially result in adverse or negative unintended consequences for anyone influenced by the program? This is particularly important from an equity perspective, to ensure the program is not doing harm.

These are but a few suggestions for what to consider when sharpening the focus of the program theory.

Principle 3: Distinguish between complicated and complex programs.

For the design phase of a program theory, the distinction between complicated and complex programs is useful (Funnel & Rogers, 2011, Rogers, 2018). *Complicated programs* consist of multiple components, multiple causal strands, multiple outcomes, multiple program partners and sites, and/or multiple target groups. *Complex programs* also might involve multiple components, causal pathways, and outcomes; however, the complexity emerges from the adaptive and highly interactive nature of these. Accordingly, complex programs are characterized by adaptive program components, non-linear causal pathways (feedback loops), and emerging outcomes that can be difficult to anticipate. Complexity also can be present when the definition of a positive outcome is unclear or evolving over time (i.e., different stakeholders have different perspectives on what constitutes a positive outcome). **Exhibit 39** provides examples of simple, complicated, and complex programs.

The distinction between complicated and complex programs can be useful when selecting the type of program theory to develop. A complicated program can be adequately visualized by a logic model or theory of change. A more complex program likely requires a causal loop diagram, stock and flow diagram, or a hybrid model to adequately capture the nature of its design and delivery.

Exhibit 39. Simple, Complicated, and Complex Programs

SIMPLE: Following a recipe	COMPLICATED: Sending a rocket to the moon	COMPLEX: Raising a child
The recipe is essential	Formulae are critical and necessary	Formulae have a limited application
Recipes are tested to assure easy replication	Sending one rocket to the moon increases assurance that the next will be OK	Raising one child provides experience but no assurance of success with the next
No particular expertise is required but cooking expertise increases success rate	High levels of expertise in a variety of fields are necessary for success	Expertise can contribute but is neither necessary nor sufficient to assure success
Recipes produce standardized products	Rockets are similar in critical ways	Every child is unique and must be understood as an individual
The best recipes give good results every time	There is a high degree of certainty of outcome	Uncertainty of outcome remains
Optimistic approach to problem-solving	Optimistic approach to problem-solving	Optimistic approach to problem-solving

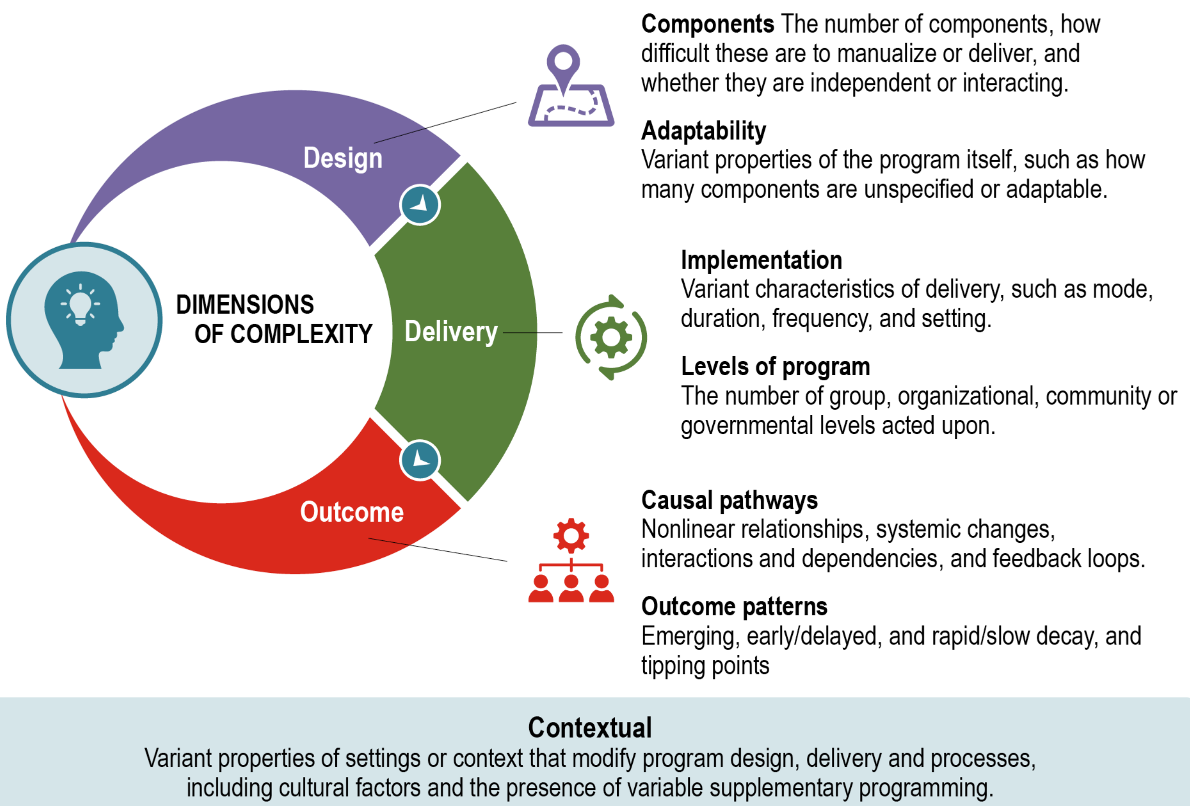
Source: Adapted from Glouberman and Zimmerman (2002)

Principle 4: Consider the type of complexity to include in your program theory.

Many evaluations involve complex programs in complex settings. When developing program theories for these, it is important to consider the types of complexity to be included in your program theory. As visualized in **Exhibit 40**, this includes complexity in program design (e.g., number of program components, adaptability of these), program delivery (e.g., implementation variation across multiple sites), and outcomes (e.g., non-linear causal pathways, varied outcome trajectories), as well as the context within which the program is implemented (e.g., political priorities, social norms).

Deciding which of these types of complexity are relevant and useful to unpack is important. Without unpacking the relevant types of complexity, it will be difficult to fully understand what is going on in a program, determine what aspects of it are working (or not), and understand whether a program is contributing to the observed outcome.

Exhibit 40. Dimensions of Complexity



Source: Adapted from Petticrew et al. (2013)

Principle 5: Capture context.

Programs never exist in a vacuum. As indicated above, context is a central dimension of program complexity. Without unpacking the context within which the program is embedded, it is difficult to provide a full account of the extent to which and how the program works (or fails to work). In line with this thinking, we propose that a useful distinction can be made between external factors and alternative explanations:

- *External factors* are contextual conditions that either enhance or inhibit the program's ability to generate the outcomes of interest.
- *Alternative explanations* are competing programs or change processes that directly influence the observed outcomes.

Further unpacking the external factors, the following contexts can be considered (Pawson, Greenhalgh, & Harvey, 2004, p. 65):

- *Individual capacities* of the program staff and stakeholders, such as interests, attitudes, motivations, skills, or knowledge

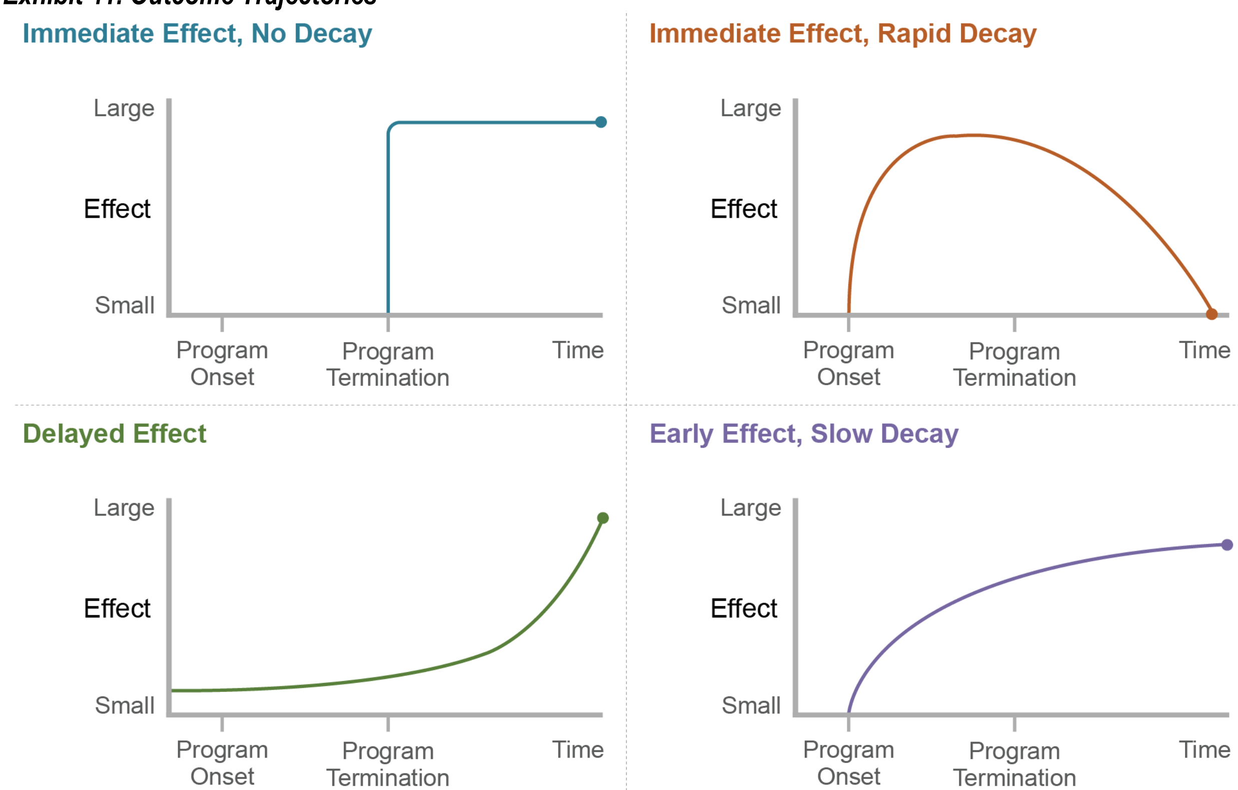
- *Interpersonal relationships* required to support the program, such as lines of communication, management and administrative support, union agreements and professional contracts, or political positioning
- *Institutional setting* in which the program is implemented, such as the culture, leadership, or rules and norms of the implementing organization
- *The wider (infra-)structural setting*, such as political support, local school system, or community support

These four levels of context can provide a structure within which to identify the most salient contextual factors to include in the program theory.

Principle 6: Consider outcome trajectories.

Program outcomes are often assumed to emerge in a linear fashion, through short-, medium-, and long-term outcome trajectories. This is a big assumption. As illustrated in **Exhibit 41**, there are of course many possible outcome trajectories, depending on the nature of the program, program participants, and outcome of interest. Some programs might even experience different outcome trajectories across outcomes or different subgroups of participants. For this reason, careful thought should be given to different types of outcome trajectories that could be relevant to depict in the program theory.

Exhibit 41. Outcome Trajectories



Source: Adapted from Lipsey (1990)

This guide's real-world examples have illustrated some of the visual techniques available to indicate *emerging* outcomes (dotted lines), *delayed* outcomes (||), or *disrupted* outcomes (┤). Different stages of the outcome chain also can be identified using color-coding. There are of course many other symbols, icons, and visualization techniques that can potentially be relevant when depicting different types of outcome trajectories.

Principle 7: Unpack assumptions.

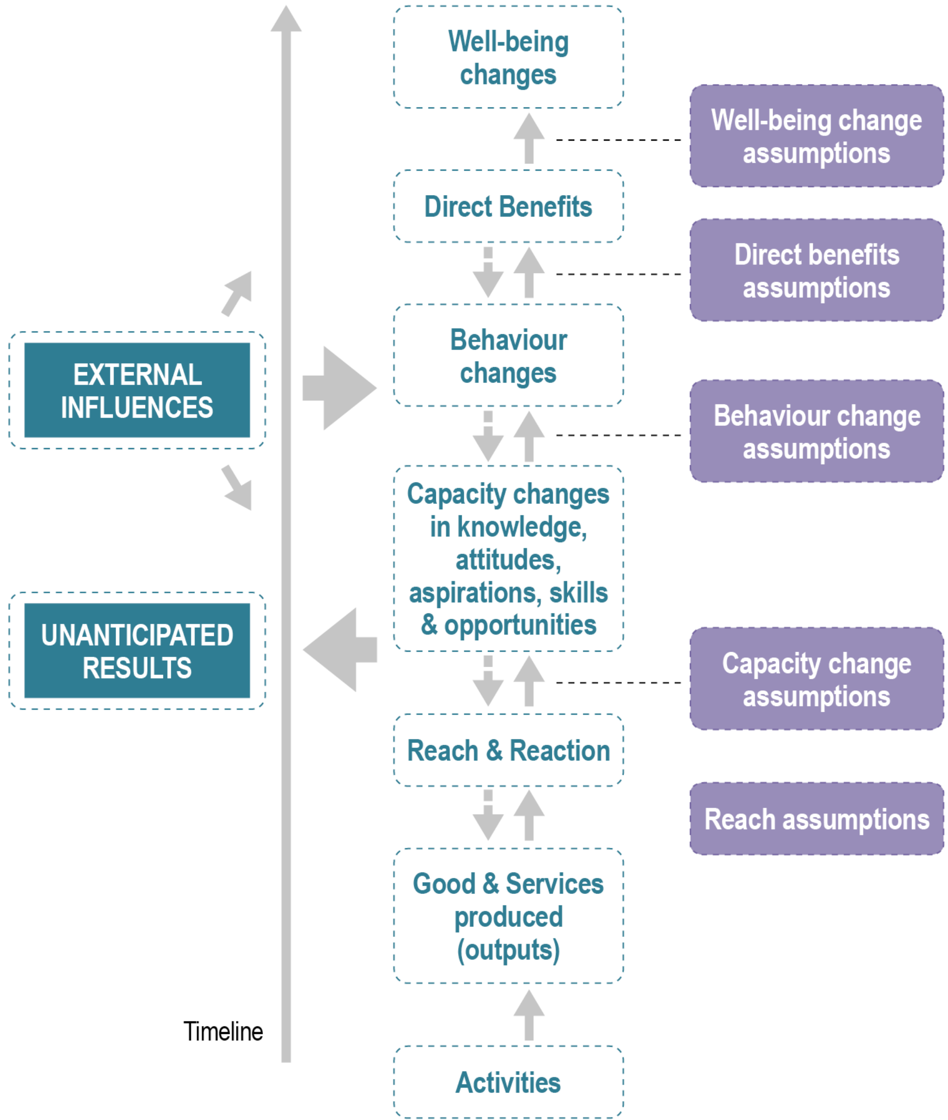
The preceding Principles 3 through 6 largely focus on the importance of capturing the often-complex nature of program components and outcomes. It is also important to carefully consider how these are connected. This involves unpacking the underlying assumptions of the program theory. As Guijt (2013) reminds us, these assumptions can be about *causal links* in the program theories (e.g., that training will lead to increased knowledge and application of that knowledge within a year), *ideology or "world-view"* assumptions about the drivers and pathways of change (e.g., that increasing economic growth will trickle down to increase prosperity for all), assumptions about the *belief systems or norms* in society (e.g., everyone we invite to a community meeting will actively and equally participate and contribute), or assumptions about the *external context* (e.g., the COVID-19 pandemic will be over in a few months and then life will return to normal). There are many types of assumptions.

As illustrated in the sections on Theories of Change and Causal Loop Diagrams, assumptions are often visualized with different types of arrows, depicting how specific program components and outcomes are hypothetically connected. For example, Mayne (2015) provides a useful typology for distinguishing different types of assumptions at various stages of a causal pathway (**Exhibit 42**):

- **Reach assumptions** are the events and conditions needed to occur if the outputs delivered are to reach and be positively received by the target group.
- **Capacity change assumptions** are the events and conditions needed to occur if the outputs that reach the target populations are to result in changes in their knowledge, attitudes, skills, aspirations, and opportunities; that is, their capacity to do things differently.
- **Behavior change assumptions** are the events and conditions needed to occur if the changes in the capacities of the target groups are to result in actual changes in their practices.
- **Direct benefits assumptions** are the events and conditions needed to occur if the behavior changes are to result in direct benefits for the target groups.
- **Well-being change assumptions** are the events and conditions needed to occur if the direct benefits are going to lead to changes in the well-being of the target group.

This is not an exhaustive list of assumptions to be considered. The types of assumptions considered should be adapted—as opposed to adopted—for the specific purpose and role of the program theory.

Exhibit 42. Types of Assumptions



Source: Adapted from Mayne (2015)

Principle 8: Center equity and transformative change in the program theory.

Program theories can be a useful tool towards promoting equity and transformative change. Towards this end, the first essential step is that program theories must surface and make visible the inequities and then focus on shifting power dynamics; reducing disparity, exclusion, and discrimination; and increasing the autonomy and voice of people who have been marginalized or excluded based on race, ethnicity, gender, ability, sexual orientation, and other dimensions. Centering the purpose of the program theory on equity is a necessary first step towards promoting equity in program theories.

Another important step is to reach beyond traditional outcomes to focus on outcomes that speak directly to the root causes of inequities. Root causes are the underlying issues that create issues or problems for individuals or communities. Root causes make those problems likely to persist even though a program might be in place to alleviate more surface-level needs of individuals and communities. To illustrate, instead of focusing the program theory on increased student academic achievement, focus the program theory on decreasing gaps in student academic achievement, or even better, on the underlying, structural reasons causing existing gaps in student academic achievement. Again, the emphasis needs to be on the root causes—the underlying causes of gaps in educational attainment, employment, and so forth. Understanding root causes requires deep analysis to understand the key dynamics; for example, in the form of a Gender Equality, Disability, and Social Inclusion (GEDSI) analysis. Any efforts to center equity should follow the principle of "nothing about me without me," meaning that inclusive approaches that seek meaningful engagement with relevant groups are critical to promote genuine transformative change and to ensure we are not doing harm.

The context factors included in the program theory should focus on structural and systemic factors that produce and sustain inequities. For example, structural racism is a very complex, dynamic system with interlinked social, political, and economic components. For this reason, program theories need to capture how interconnections between the program and existing policies, social and institutional practices, and cultural representations and narratives reinforce inequities. By capturing and closely examining these in our program theories, we will be better positioned to see how race, privilege, and disadvantage remain interconnected with and influence the program to be evaluated.

Principle 9: Do no harm—Consider unintended and adverse outcomes.

Program theories tend to focus on a select set of intended and often positive outcomes; that is, outcomes that represent a beneficial change to program participants and other stakeholders. However, programs cannot be assumed to only do good. Many programs have unintended consequences that can be positive or negative. Examples include side effects (adverse spillover effects of the program), paradoxical or counterproductive effects (opposite of the intended program effect), inequitable effects (unfair differences across program participants), and null effects (ineffective program).

Speaking directly to this issue, Bonell et al. (2015) advocate, and correctly so, for increased attention to harmful consequences when developing program theories. The authors propose:

- Reflecting on the unintended consequences that can potentially emerge from the interaction between the program and the context within which it is implemented (see also Principle 5. Capture Context)
- Comparing the logic model of the program with the logic models, program descriptions, and/or process evaluations of similar programs to identify harmful effects
- Consulting with relevant stakeholders to identify how program activities and mechanisms might be derailed, leading to harmful consequences

These are useful strategies. The position we hold is that capturing adverse consequences in program theories allows for a more complete understanding of how programs work, serves well to inform future program designs, and over time supports development of programs that are less likely to cause harm. In addition to the suggestions above, we also would encourage increased attention to the ways in which harmful consequences affect different stakeholder groups and likely fall most heavily on the most disenfranchised. Thinking carefully about who is likely to be affected by the adverse consequences is fundamentally a question of equity.

Principle 10: Mix and match visual techniques.

This guide showcases a broad range of program theory types and visual techniques. Some of the more advanced are nested and hybrid models, integrating different types of program theories. We recognize that some of these hybrid models might not be relevant or feasible in the context of many evaluations. However, we still find that using many of the visual techniques are still effective and feasible in most evaluations. Scanning across the real-world examples, the following techniques are worth highlighting:

- Color-coded lines to indicate degrees of evidence (Exhibit 36) or positive versus negative causal links (Exhibit 8)
- Line thickness to distinguish between short- and medium-term outcomes (Exhibit 14)
- Double bars "||" to indicate delayed outcomes (Exhibits 13 and 30)
- Cross bars "┤" to indicate disrupted causal strands (Exhibit 23)
- Plus "+" and minus "–" signs (Exhibit 13) or icons, such as smiley/frowny faces (Exhibit 15) to reflect the direction and polarity of causal connections or positive and negative feedback loops, or clouds (Exhibit 19) to reflect flows into and out of the program
- Differently shaped boxes for program components, context, mechanisms, and outcomes (Exhibit 11)

Mixing and matching visual techniques supports more purposeful program theorizing by visually bringing attention to the most salient aspects of the program theory.

APPENDIX

This appendix provides an overview of software for developing program theories.

Exhibit A1. Software for Developing Program Theory

NAME	DESCRIPTION	FREEWARE
CASUAL MAP	License-based qualitative coding and analysis software that facilitates development of causal models based on qualitative statements from program stakeholders. https://www.causalmap.app/	
DYLOMO	Web-based software for developing theories of change. https://dylomo.com/	✓
EDUCATION LOGIC MODEL (ELM) APPLICATION	Downloadable software that supports development of logic models and theories of change. https://ies.ed.gov/ncee/edlabs/regions/pacific/elm.asp	✓
FREEMIND	Downloadable software for developing mind maps and concept maps. https://freemind.en.softonic.com/	✓
INSIGHT MAKER	Web-based software for developing stock and flow and causal loop diagrams. The software also facilitates simulations. https://insightmaker.com/	✓
LOGFRAMER	Downloadable software for developing logical frameworks. The software comes with an integrated project planning and monitoring interface with calendar and budget. https://www.logframer.eu/	✓
SPSS AMOS	License-based software package for structural equation modeling (SEM). The software can be used to tests relationships between variables and develop path models. https://www.ibm.com/products/structural-equation-modeling-sem	
SYSTEM DYNAMICS LOOP GENERATOR	Downloadable modeling and simulation software for developing stock and flow and causal loop diagrams. http://sysdyn.simantics.org/	✓
THEORYMAKER	Web-based software for developing theories of change and causal loop diagrams. The software comes with CausalExplorer, a tool that allows the user to explore causal influence of variables on one another. http://theorymaker.info	✓
Theory of Change Online (TOCO)	Web-based software designed to develop and edit theories of change, visualizing outcomes, indicators, rationales, and assumptions. https://www.theoryofchange.org/toco-software/	
UCINET (and NetDraw)	Downloadable software for developing and analyzing network maps. The software comes with NetDraw, a network visualization tool. https://sites.google.com/site/ucinetsoftware/home	
VENSIM	Downloadable modeling and simulation software for developing and analyzing causal loop diagrams. https://vensim.com/	
VISUAL UNDERSTANDING ENVIRONMENT (VUE)	Downloadable software that can be used for development and analysis of concept maps. https://vue.tufts.edu	✓
VOSVIEWER	Web-based software for developing and analyzing network maps based on bibliometric information (citations, keywords). https://www.vosviewer.com/	✓

GLOSSARY

Activity—the concrete actions, events, and strategies implemented by the program.

Adverse effect—an undesired and harmful consequence of a program.

Alternative explanations—competing programs or change processes that directly influence the observed outcomes.

Causal loop diagram—a type of program theory that depicts more-complex outcome patterns, accounting for positive and negative feedback loops as well as emerging, delayed, and unintended outcome trajectories. The visual display of causal loop diagrams usually consists of program components and outcomes presented in text boxes and causally linked to one another using line arrows.

Concept map (also referred to as a **mind map**)—a diagram that can be used to clarify and explicate key concepts in a program theory. Concept maps often use circles or boxes for key concepts or themes, with relationships between these indicated by connecting lines.

Context—the setting and environment within which the program is embedded that can influence the ability of the program to generate the desired outcomes. (Sometimes referred to **contextual or influencing factors**)

Context-mechanism-outcome (CMO) configuration—a type of program theory that depicts the underlying generative processes (mechanisms) promoting behavioral changes (outcomes) in a given setting (context).

Equity—providing people with *tailored support* to achieve the best possible life outcomes by partnering with people most affected as decision makers to ensure that the systems that govern their lives meet their needs.

Feedback loop—a circular causal relationship in causal loop diagrams. Feedback loops are either reinforcing or balancing. *Reinforcing* loops are causal loops where a change in one direction is compounded by more change in the same direction. *Balancing* loops are causal loops where a change in one direction results in a change in the opposite direction.

Flow—processes, activities, or decisions that increase or decrease the stocks in a stock and flow diagram.

Gender Equality, Disability, and Social Inclusion (GEDSI) analysis—assessment focused on identifying and analyzing issues related to gender equality, disability, and social inclusion. GEDSI analyses are ideally conducted during the program design phase.

Hybrid model (also referred to as **blended model**)—a type of program theory that integrates different types of models into a single program theory.

Impact—long-term outcomes. Some evaluators reserve the term impact for outcomes shown to be the causal result of a program (as evidenced by a study using a control/comparison group design).

Influencing factor—a contextual factor that influences, positively or negatively, the ability of the program to generate one or more desired outcomes.

Input—program resources, funding, materials, equipment, technology, staff, or any other support services and material resources available to the program.

Intersectionality—multiple forms of inequity and disadvantage compounding themselves and creating obstacles that are not fully understood when evaluating a singular form of inequity and disadvantage.

Logical framework (logframe)—a type of program theory depicts how program activities are intended to lead to a specific set of measurable outcomes and impact. Typically in the form of a table, with rows for program components (activities, outputs, outcomes, impact) and columns for measurement information (program summary, indicators, means of verification, risks/assumptions).

Logic model—a type of program theory that provides an overview of main program inputs, activities, outputs, and outcomes, without specifying how each of these is connected. Often presented in a tabular format.

Mechanism—the underlying generative processes that bring about the outcome(s) of interest. Mechanisms include participant reactions and responses to program activities, such as increased sense of efficacy, empowerment, and motivation, as well as behavioral changes.

Nested model (also referred to as **layered model** or **stacked model**)—a type of program theory that uses separate models to depict different aspects of a program.

Network map—a diagram that displays relationships among people, organizations, themes, or other entities in the form of spiderweb-like diagrams.

Outcome—knowledge, skills, attitudinal, behavioral, and other changes among the program participants (and other stakeholders). Often broken down in terms of *short-term, medium-term,* and *long-term* outcomes (sometimes referred to as **impact**).

Output—the concrete product delivered by the program (emerging directly from the program activities).

Path model—a diagram that displays how program activities are statistically associated with specific outcomes. More developed path models also include contextual conditions—within which the program is embedded—that can influence the ability of the program to generate the desired outcomes.

Program theory—an umbrella term for any type of model of the underlying logic of a program.

Stock—accumulations of an output or outcome of interest, such as number of teachers and high-performing students, included in a stock and flow diagram.

Stock and flow diagram—type of program theory that describes how programs work in terms of variations in stocks (program outputs and outcomes) and flows (program processes). (A variant of a causal loop diagram.)

Structural equation modeling—a set of statistical analysis techniques that examines structural relationships among variables.

Structural racism—the normalization and legitimization of an array of dynamics (historical, cultural, institutional, interpersonal) that routinely advantage White people while producing cumulative and chronic adverse outcomes for people of color.

Theory-based evaluation (also referred to as **theory-driven evaluation**)—an approach that structures the evaluation around the development and refinement of a program theory. Variants of theory-based evaluation include *realist evaluation* (Pawson & Tilley, 1997) and *contribution analysis* (Mayne, 2008).

Theory of action—the program inputs, outputs, and activities, as well as how these are to be implemented. The mix of things we will do to try to affect and contribute to one or more outcomes.

Theory of change (ToC)—a type of program theory that describes how and in what way specific program activities are expected to lead to specific outputs, which in turn are expected to lead to specific outcomes. ToCs also visualize and describe the hypothesized connections among activities, outputs, and outcomes; that is, the underlying assumptions of how the program works. More-developed ToCs include contextual conditions.

Unintended consequences—unplanned or unanticipated positive or negative effects of a program. These include *side effects* (adverse spillover effect of the program), *paradoxical or counterproductive* effects (opposite of the intended program effect), *inequitable* effects (unfair differences across program participants), and *null effects* (ineffective program).

REFERENCES

Aronson, D., & Angelakis, D. (1999). Step-by-step stocks and flows: Converting from causal loop diagrams. *The Systems Thinker*, *10*(6). https://thesystemsthinker.com/wp-content/uploads/pdfs/100602pk.pdf

The Aspen Institute (2009). *Constructing a racial equity theory of change. A practical guide for designing strategies to close chronic racial outcome gaps.* https://ncwwi.org/files/Cultural_Responsiveness__Disproportionality/Constructing_a_Racial_Equity_Theory_of_Change_-_5_EXERCISES.pdf

Bick, S., Buxton, H., Chase, R. P., Ross, I., Adriano, Z., Capone, D., Knee, J., Brown, J., Nalá, R., Cumming, O., & Dreibelbis, R. (2021). Using path analysis to test theory of change: a quantitative process evaluation of the MapSan trial. *BMC Public Health, 21*: 1411.

Bickman, L. (1987). The functions of program theory. *New Directions for Program Evaluation, 33*, 5-18.

Blundo-Canto, G., Andrieu, N., Adam, N. S., Ndiaye, O., & Chiputwa, B. (2021). Scaling weather and climate services for agriculture in Senegal: Evaluating systemic but overlooked effects. *Climate Services, 22*, 100226.

Bonell, C., Jamal, F., Melendez-Torres, G. J., & Cummins, S. (2015). "Dark logic": Theorising the harmful effects of public health interventions. *Journal of Epidemiology and Community Health, 69*(1), 95-98.

Byng, R., Norman, I., & Redfern, S. (2005). Using realistic evaluation to evaluate a practice-level intervention to improve primary healthcare for patients with long-term mental illness. *Evaluation, 11*(1), 69-93.

Connell, J. P., Kubisch, A. C., Schorr, L. B., & Weiss, C. H. (1995). *New approaches to evaluating community initiatives: Concepts, methods, and contexts*. Aspen Institute.

Connell, J. P., & Kubisch, A. C. (1998). Applying a theory of change approach to the evaluation of comprehensive community initiatives: Progress, prospects, and problems. In K. Fulbright-Anderson, A. C. Kubisch, & J. P. Connell (Eds.), *New approaches to evaluating community initiatives, Volume 2: Theory, measurement, and analysis* (pp. 15-44). Aspen Institute.

Cooksy, L. J., Gill, P., & Kelly, P. A. (2001). The program logic model as an integrative framework for a multimethod evaluation. *Evaluation and Program Planning, 24*(2), 119-128.

De Lisle, J. (2015). The promise and reality of formative assessment practice in a continuous assessment scheme: the case of Trinidad and Tobago. *Assessment in Education: Principles, Policy & Practice, 22*(1), 79-103.

Freer, G., & Lemire, S. (2019). Can't see the wood for the logframe: Integrating logframes and theories of change in development evaluation. *The Canadian Journal of Program Evaluation, 33*(3), 336-353.

Frerichs, L., Young, T. L., Dave, G., Stith, D., Corbie-Smith, G., & Lich, K. H. (2018). Mind maps and network analysis to evaluate conceptualization of complex issues: A case example evaluating systems science workshops for childhood obesity prevention. *Evaluation and Program Planning, 68*, 136-147.

Funnell, S. C. (1997). Program logic: an adaptable tool. *Evaluation News and Comments, 6*(1), 5-17.

Funnell, S. C., & Rogers, P. J. (2011). *Purposeful program theory. Effective use of theories of change and logic models*. Jossey-Bass.

Glouberman, S., & Zimmerman, B. (2002). *Complicated and complex systems: What would successful reform of Medicare look like?* Commission on the Future of Health Care in Canada. Discussion Paper 8. Health Canada. https://www.alnap.org/system/files/content/resource/files/main/complicatedandcomplexsystems-zimmermanreport-medicare-reform.pdf

Guijt, I. (2013). *ToC reflection notes 3: Working with assumptions in a theory of change process.* https://content.changeroo.com/wp-content/uploads/Academy/2017/07/toc_notes_3_working_with_assumptions_in_a_toc_process.pdf

Havaei, F., & Macphee, M. (2015). The Nursing Leadership Institute program evaluation: A critique. *Journal of Healthcare Leadership, 7*, 65-74.

den Heyer, M. (2002). The temporal logic model concept. *The Canadian Journal of Program Evaluation, 17*(2), 27-47.

Higgins, A., O'Halloran, P., & Porter, S. (2015). The management of long-term sickness absence in large public sector healthcare organisations: A realist using mixed methods. *Journal of Occupational Rehabilitation, 25*(3), 451-470.

Hirsch, G. B., Levine, R., & Miller, R. L. (2007). Using system dynamics modeling to understand the impact of social change initiatives. *American Journal of Community Psychology, 39*(3-4), 239-253.

Imas, L. G. M., & Rist, R. C. (2009). *The road to results: designing and conducting effective development evaluations.* World Bank. https://openknowledge.worldbank.org/handle/10986/2699

Koleros, A., & Mayne, J. (2019). Using actor-based theories of change to conduct robust evaluation in complex settings. *The Canadian Journal of Program Evaluation, 33*(3), 292-315.

Kwamie, A., van Dijk, H., & Agyepong, A. I. (2014). Advancing the application of systems thinking in health: Realist evaluation of the Leadership Development Programme for district manager decision-making in Ghana. *Health Research Policy and Systems, 12*(29). https://doi.org/10.1186/1478-4505-12-29

Lemire, S., Kwako, A. J., Nielsen, S. B., Christie, C. A., Donaldson, S., & Leeuw, F. (2020). What is this thing called a mechanism? Findings from a review of published realist evaluations. *New Directions for Evaluation, 167*, 73-86.

Lemire, S., Nielsen, S. B., & Dybdal, L. (2012). Making contribution analysis work: a practical framework for handling influencing factors and alternative explanations. *Evaluation, 18*(3), 294-309.

Lemire, S., Whynot, J., & Montague, S. (2019). How we model matters: A manifesto for the next generation of program theorizing. *The Canadian Journal of Program Evaluation, 33*(3), 414-433.

Lich, K. H., Urban, J. B., Frerichs, L., & Dave, G. (2017). Extending systems thinking in planning and evaluation using group concept mapping and system dynamics to tackle complex problems. *Evaluation and Program Planning 60*, 254-264.

Lipsey, M. W. (1990). *Design sensitivity*. Sage.

Littlejohns, L. B., Baum, F., Lawless, A., & Freeman, T. (2018). The value of a causal loop diagram in exploring the complex interplay of factors that influence health promotion in a multisectoral health system in Australia. *Health Research Policy and Systems, 16*(1), 126. http:// doi.org/10.1186/s12961-018-0394-x

Mackenzie, M., Koshy, P., Leslie, W., Lean, M., & Hankey, C. (2009). Getting beyond outcomes: A realist approach to help understand the impact of a nutritional intervention during smoking cessation. *European Journal of Clinical Nutrition, 63*(9), 1136-1142.

Mayne, J. (2008). *Contribution analysis: An approach to exploring cause and effect.* ILAC Brief 16. CGIAR. https://cgspace.cgiar.org/bitstream/handle/10568/70124/ILAC_Brief16_Contribution_Analysis.pdf

Mayne, J. (2015). Useful theory of change models. *The Canadian Journal of Program Evaluation, 30*(2), 119-142.

McLinden, D. (2013). Concept maps as network data: Analysis of a concept map using the methods of social network analysis. *Evaluation and Program Planning, 36*(1), 40-48.

Nielsen, S. B., Lemire, S., & Tangsig, S. L. (2021). Unpacking context in realist evaluation: Findings from a comprehensive review. *Evaluation, 28*(1), 91-112.

O'Campo, P., Stergiopoulos, V., Davis, O., Lachaud, J., Nisenbaum, R., Dunn, J. R., Ahmed, N., & Tsemberis, S. (2022). Health and social outcomes in the Housing First model: Testing the theory of change. *eClinicalMedicine, 47*, 101387.

Oroviogoicoechea, C., & Watson, R. (2009). A quantitative analysis of the impact of a computerised information system on nurses' clinical practice using a realistic evaluation framework. *International Journal of Medical Informatics, 78*, 839-849.

Pawson, R., Greenhalgh. T., & Harvey, G. (2004). *Realist synthesis: An introduction.* ESRC Research Methods Programme, University of Manchester RMP Methods Paper 2/2004, pp. 7-8. http://www.ccsr.ac.uk/methods/publications/documents/RMPmethods2.pdf.

Pawson, R., & Tilley, N. (1997). *Realistic evaluation*. Sage.

Penn, A. S., Knight, C. J. K., Lloyd, D. J. B., Avitabile, D., Kok, K., Schiller, F., … Basson, L. (2013). Participatory development and analysis of a fuzzy cognitive map of the establishment of a bio-based economy in the Humber region. *PLoS One 8*(11), e78319.

Petticrew, M., Anderson, L., Elder, R., Grimshaw, J., Hopkins, D., Hahn, R., . . . Welch, V. (2013). Complex interventions and their implications for systematic reviews: A pragmatic approach. *Journal of Clinical Epidemiology, 66*(11), 1209-1214.

Proust, K., Newell, B., Brown, H., Capon, A., Browne, C., Burton, A., Dixon, J., Mu, L., & Zarafu, M. (2012). Human health and climate change: Leverage points for adaptation in urban environments. *International Journal of Environmental Research and Public Health, 9*, 2134-2158.

Renger, R., & Titcomb, A. (2002). A three-step approach to teaching logic models. *American Journal of Evaluation, 23*(4), 493-503.

Rogers, P. J. (2008). Using programme theory to evaluate complicated and complex aspects of interventions. *Evaluation, 14*(1), 29-48.

Scheirer, M. A. (1987). Program theory and implementation theory: Implications for evaluators. In L. Bickman (Ed.), *Using program theory in evaluation* (pp. 59-76). New Directions for Program Evaluation, 33. Jossey-Bass. DOI: 10.1002/ev.1446

Teskey, G. (2021, August 5). In praise of…. Logframes. *From Poverty to Power.* https://oxfamapps.org/fp2p/in-praise-of-logframes/

Wilkinson, H., Hills, D., Penn, A., & Barbrook-Johnson, P. (2021). Building a system-based theory of change using participatory systems mapping. *Evaluation, 27*(1), 80-101.

Wright, E. A. (2003). The re-design of an integrated water and pollution management programme using the systems-ware model of the log frame. *Physics and Chemistry of the Earth, 28*(20-27), 973-984.

ENDNOTES

[1] The term impact is by some evaluators used as a synonym for long-term outcomes. Other evaluators reserve the term impact for outcomes shown to be the causal result of a program (as evidenced by a study using a control/comparison group design).
[2] Logframe terminology is out of fashion in some international development circles, but the approach still underpins what is called *Performance Assessment Frameworks (PAFs)* at the Australian Department of Foreign Affairs and Trade and what is referred to as *results frameworks* in other international development circles.
[3] Some refer to a theory of change *approach* to evaluation, where the ToC is used as a sort-of qualitative counterfactual (Connell et al., 1995; Connell & Kubisch, 1998). For the purpose of this guide, we focus on theories of change as a visual model, as opposed to an evaluation approach.