FLYING LOGIC

Welcome to Flying Logic

FlyingLogic.com

Sciral

The Elevator Pitch

Flying Logic is a revolutionary tool for proactive thinkers, whether you want to change your personal life for the better, or grow a world-class business that impacts thousands or millions of people. **Flying Logic** is a highly visual, easy to use software application that does for reasoning what spreadsheets do for numbers. Just as you would not do detailed financial projections in your head, you need **Flying Logic** for all nontrivial planning and reasoning endeavors. Project management software won't help you when you have no idea what part of a complex system needs improvement, or what that improvement might look like, or how to cause that improvement. Answering these questions is a critical and often-overlooked part of the planning process, and this is where **Flying Logic** shines. **Flying Logic** diagrams are not static images— they are working models of your reasoning that you test as you build and that stay "live" as you share your reasoning with others.

Critical Thought More Critical Than Ever

Rational thinking, planning, and communication skills are key to every field of human endeavor ranging from personal growth to building viable, competitive businesses. Even the ability to successfully articulate non-rational concepts such as emotion and intuition ultimately depends upon the mind's power to reason. And anyone from young children, to governments, to the CEOs of multinational corporations can benefit from well-developed reasoning abilities *and* a common language in which to reason. **Flying Logic** is a revolutionary tool that fills a widespread need for software that supports, simply and beautifully, *the practice of reasoning*. However, no software can do your thinking for you, so Flying Logic was inspired by the **Theory of Constraints**.

The Theory of Constraints (TOC)

The Theory of Constraints is an overall management philosophy founded on the idea that all real-world systems; whether personal, interpersonal, or organizational; have at least one **constraint**: something that holds the system back from accomplishing more of its primary purpose, or **goal**. The rate of this accomplishment is called **throughput**. If a system had absolutely no constraints, it would be capable of infinite throughput. But though infinite throughput is impossible, amazing throughput gains *are* possible through the careful identification and management of the system's key constraints. The purpose of the TOC, originally developed by Eliyahu M. ("Eli") Goldratt and first popularized in his bestselling business novel *The Goal*, is to give individuals and organizations the tools they need to manage their constraints in the most effective manner possible. Originally applied to manufacturing lines, TOC principles have been successfully adapted for areas as diverse as supply chain, accounting and finance, project management, health care, military planning, and software engineering.

TOC claims that a real-world system with more than three constraints is extremely unlikely, and in fact usually only one constraint is key. Perhaps counter-intuitively, this is because the more complex a system becomes, the more interrelationships are necessary among its parts, which results in *fewer* overall degrees of freedom.

A major implication of this is that managing a complex system or organization can be made both simpler and more effective by providing managers with few, specific, yet highly influential areas on which to focus — maximizing performance in the areas of key constraints, or **elevating** the constraint (making it less constraining.)

The Five Focusing Steps

To accomplish this, the developers of TOC developed the Five Focusing Steps, which define a process of ongoing improvement. (Step Zero was later added for additional clarity.)

0. **Articulate** the goal of the system. *How do we measure the system's success?*

1. **Identify** the constraint. *What is the resource limiting the system from attaining more of its goal?*

2. **Exploit** the constraint to its fullest. *How can we keep the constraining resource as busy as pos-*
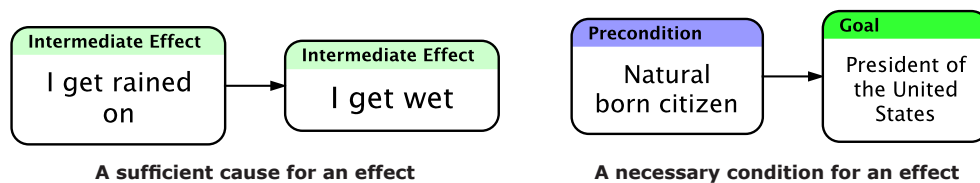
sible, exclusively on what it does best?

3. **Subordinate** all other processes to the decisions made in Step 2. *How can we align all processes so they give the constraining resource everything it needs?*

4. **Elevate** the constraint. *If managing the constraining resource more efficiently does not give us all the improvement we need, then how can we acquire more of the resource?*

5. **Avoid inertia.** *Has the constraint moved to some other resource as a result of the previous steps? If so, don't allow inertia itself to become the constraint: go back to step 1.*

The Thinking Processes (TP)

The Thinking Processes emerged as TOC practitioners worked with organizations that needed to identify their core constraints and how to manage or elevate them. They needed the answers to three deceptively simple questions:

- **What** to change?
- **To what** to change?
- **How to cause** the change?

The Thinking Processes are based on the scientific method, to which is added a simple visual language for describing and reasoning about situations, arguments, and plans using the language of cause and effect. There are two basic kinds of reasoning: **sufficient cause** and **necessary condition**.

Intermediate Effect		Intermediate Effect
I get rained on	→	I get wet

A sufficient cause for an effect

Precondition		Goal
Natural born citizen	→	President of the United States

A necessary condition for an effect

From these, the practitioners developed several methodologies called **application tools** designed to answer the three questions. The application tools provide the ability to develop a complete picture of a system's core constraints and how to overcome them.
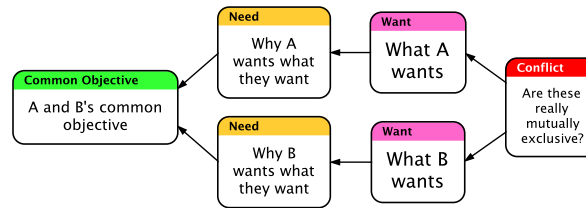
Current Reality Tree

Used to move from an articulation of the undesirable symptoms (or desirable strengths) present in a system to the core cause that has the most influence over them (i.e., the constraining resource or core issue.)

Current Reality Tree
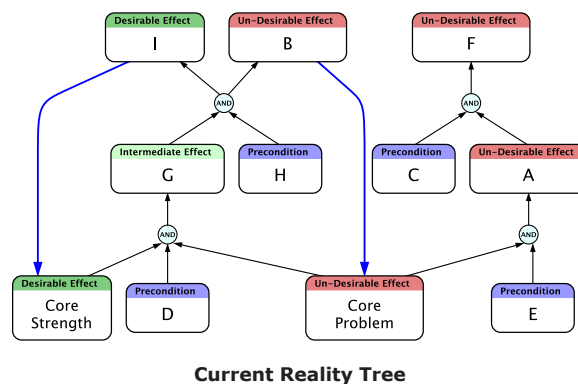
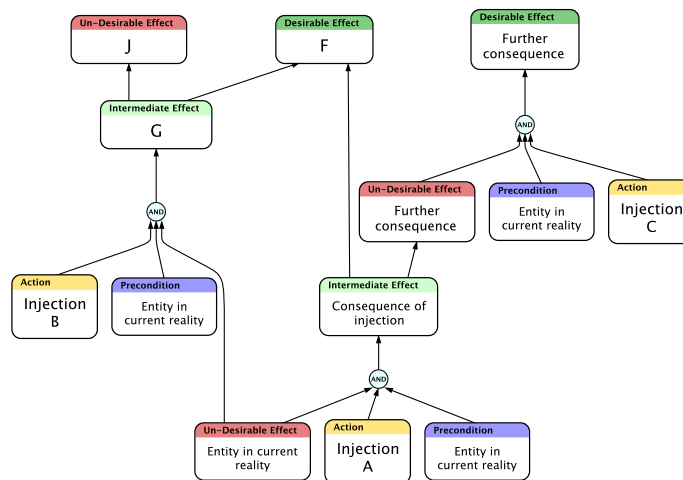## Evaporating Cloud (Conflict Resolution Diagram)

Used to move from apparently mutually-exclusive options to creative, win-win solutions by surfacing and "breaking" the assumptions that underlie the supposed conflict.

Need — Why A wants what they want

Want — What A wants

Common Objective — A and B's common objective

Conflict — Are these really mutually exclusive?

Need — Why B wants what they want

Want — What B wants
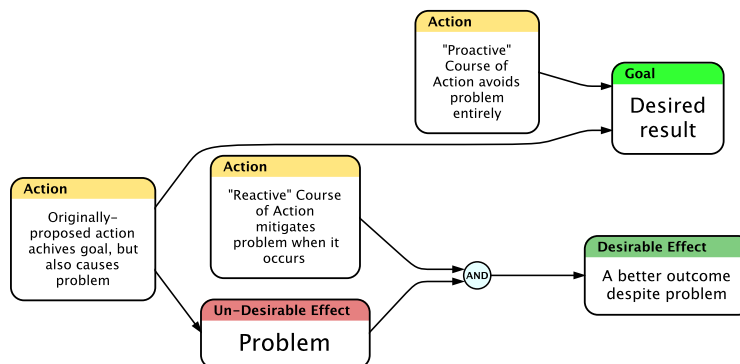
Evaporating Cloud

Future Reality Tree

Used to move from a sketched-out plan to a fully-realized plan by thoroughly examining the consequences of various proposed actions applied to the current reality.

Un-Desirable Effect — J

Desirable Effect — F

Desirable Effect — Further consequence

Intermediate Effect — G

AND

Un-Desirable Effect — Further consequence

Precondition — Entity in current reality

Action — Injection C

Action — Injection B

Precondition — Entity in current reality

Intermediate Effect — Consequence of injection

Un-Desirable Effect — Entity in current reality

Action — Injection A

Precondition — Entity in current reality

Future Reality Tree

One of the principles that distinguishes the TOC-TP is its emphasis on seeking and blocking the undesirable effects of actions (UDEs.) This is an essential activity when constructing Future Reality Trees and Transition Trees.

Action — "Proactive" Course of Action avoids problem entirely

Goal — Desired result

Action — Originally-proposed action achives goal, but also causes problem

Action — "Reactive" Course of Action mitigates problem when it occurs

Desirable Effect — A better outcome despite problem

Un-Desirable Effect — Problem

Undesirable Effects: Proactive and Reactive Responses

Transition Tree

Used to move from a set of predetermined goals to a set of actions that both achieve the goals and that deliberately avoid or mitigate undesirable effects.
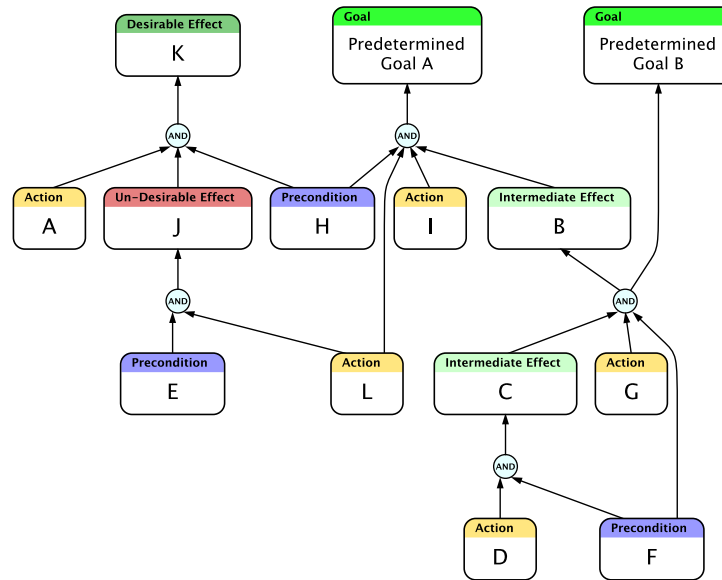
**Transition Tree**

Prerequisite Tree

Used to move from a set of objectives and attendant obstacles that stand in the way of accomplishing each objective, to a set of feasible intermediate objectives (milestones) that when fully implemented overcome every obstacle and thus achieve every objective.

Prerequisite Tree
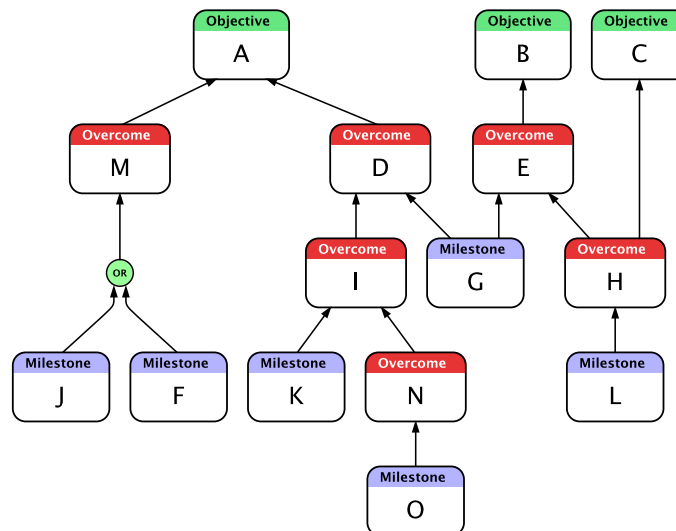
The latest addition to the TOC-TP application tools, the Strategy and Tactics Tree is used to move from the highest-level organizational goals to a comprehensive, multi-tiered, fully-justified set of implementation steps.
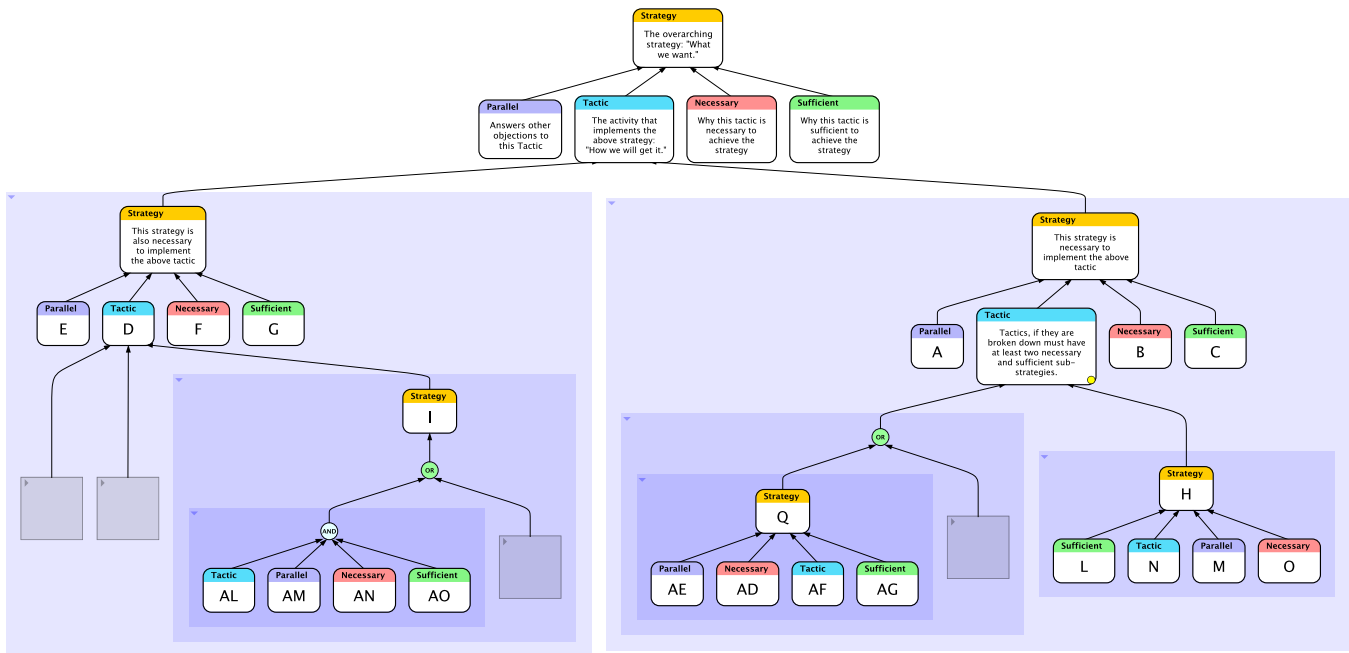
Strategy & Tactics Tree

The Need for Software Support

Although there are many success stories of organizations that have implemented TOC using the Thinking Processes, so far there have been two problematic approaches to working with the application tools:

- Groups work with a facilitator in front of a white board to capture their thinking. This has the drawback that as they grow, whiteboard diagrams quickly become disorganized and difficult to understand.

- Individuals create their own plans or memorialize plans created by other groups using tools such as Visio™. The primary drawback to this approach is that the planner often becomes bogged down in diagram layout details: fonts, sizes, colors, placement, styles, etc., that have nothing to do with the actual planning process.

- Either way, there is no easy way to test the logic inherent in the diagram.

Flying Logic was designed to eliminate all of these constraints by allowing groups or individual planners to create Thinking Process diagrams using an intuitive interface that requires no attention to layout issues, and by producing presentation-quality output as a by-product of the planning process.

By providing a clear, visual language of causes and effects, Flying Logic encourages detached, rational thinking. By removing constraints around re-working the diagram, Flying Logic encourages people to painlessly consider every factor that really matters, and address every blind spot as it

> **More than TOC!**
>
> The TOC-TP is a sophisticated system and Flying Logic supports all of its methodologies. But perhaps you find other techniques useful such as **Argument Maps**, **Concept Maps**, or probabilistic **Belief Networks**. Flying Logic works great for these techniques and many others— or you can use Flying Logic to invent your own.

comes up. A good analogy is spreadsheet software— before spreadsheets, people still did financial projections, but they were laborious, error-prone, and there was a great deal of resistance to rework. After spreadsheets, people take many more factors into consideration and easily explore many more alternatives in their financial planning.

Just as a spreadsheet allows you to play "What if…?" with numbers, Flying Logic lets you play "What if…?" with plans, arguments, and ideas.
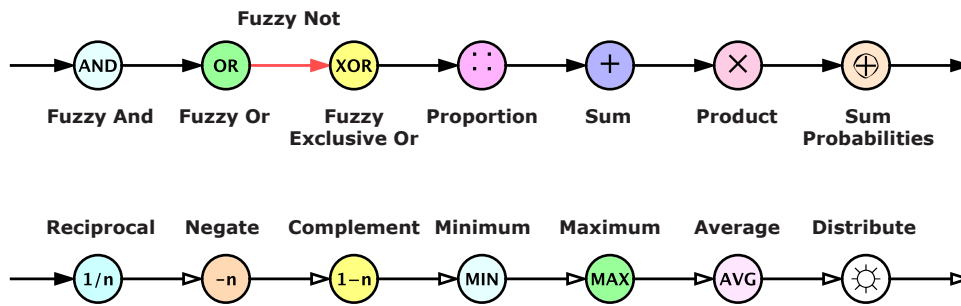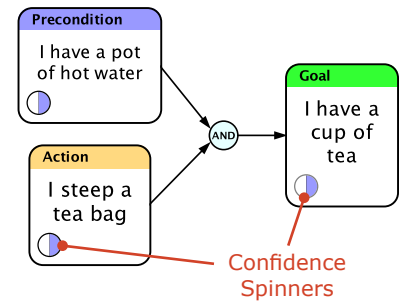
The Flying Logic Interface

Flying Logic uses a simple, gesture-based interface. Drag from the list of entity types on the left to create your diagram. Each time you place an entity, it "snaps into place" with arrows that represent the causal relationships automatically appearing where you want them. Drag from entity to entity to create additional causal relationships as desired, or drag from entity to arrow to combine causes into necessary condition relationships. Relationships representing AND, OR, and NOT are easily created.
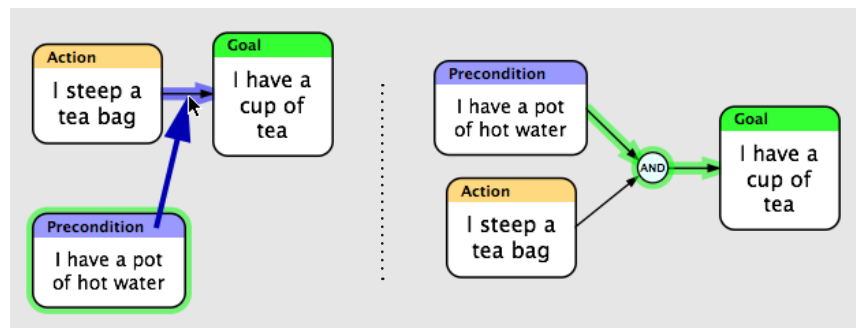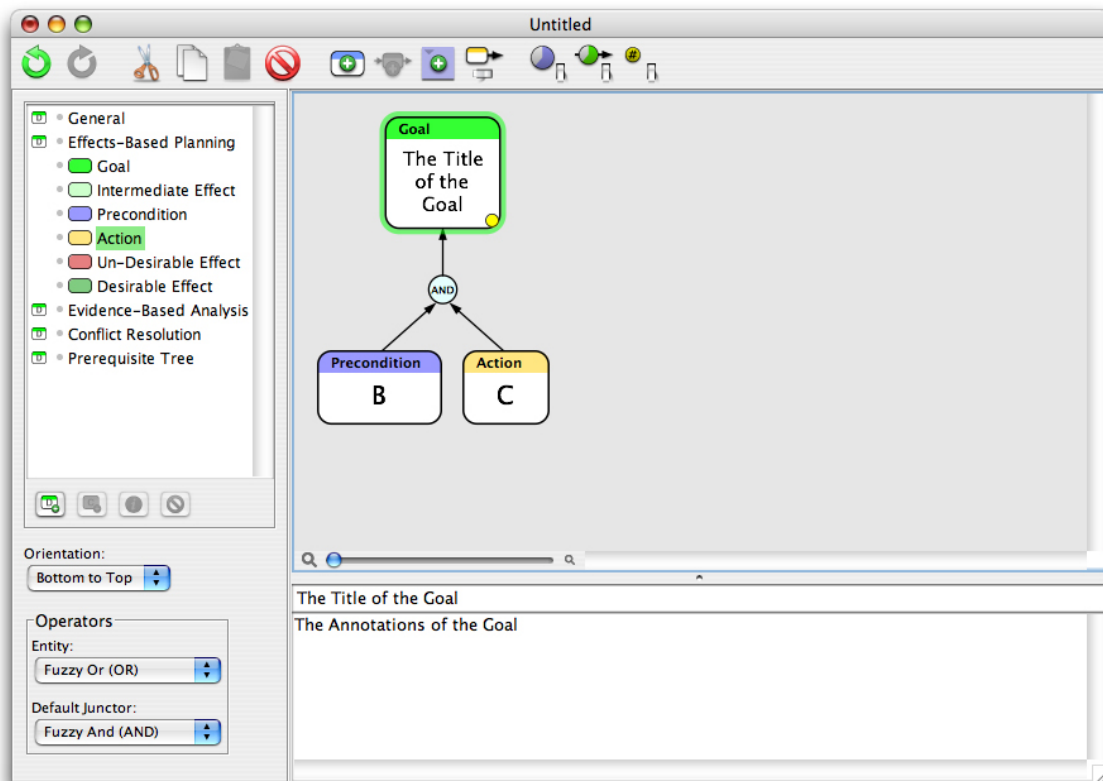
A Working Model

As you build your diagram, or when it is time to present it to others, you can step through the actual *model* of your logic by using the confidence spinners attached to the entities. As each spinner is adjusted, the other affected spinners move in tandem. The various operators you use in the diagram determine how the data flowing through it is combined along the way.

If you need sophisticated operations beyond AND, OR, and NOT, **Flying Logic Pro** supports many *advanced operators*:

Fuzzy Not

AND	OR	XOR	∴	+	×	⊕
Fuzzy And	**Fuzzy Or**	**Fuzzy Exclusive Or**	**Proportion**	**Sum**	**Product**	**Sum Probabilities**

| **Reciprocal**	**Negate**	**Complement**	**Minimum**	**Maximum**	**Average**	**Distribute**
1/n	−n	1−n	MIN	MAX	AVG	☼

Annotations, Groups

Every diagram element can be annotated using textual annotations which appear in the diagram as yellow bullets. In the TOC-TP methodology, annotating the causal relationships is particularly important, as they explain *why* you believe that one or more causes are sufficient to produce particular effects. Multi-level grouping of entities is also supported, and groups can be collapsed to help manage even very complex diagrams. Groups can also be given titles and annotated.

Free Your Mind to Focus On Your Problem

As you create your diagram, Flying Logic takes full responsibility for the layout of the entities and relationships that connect them. Because the sophisticated graph layout engine built into Flying Logic makes many decisions about how to keep the diagram compact, the number of line crossings minimal, etc., Flying Logic displays smooth, animated transitions each time the diagram is modified. The animations preserve your "mental map" of the diagram even though Flying Logic may sometimes completely reorganize the layout. This technology frees you to concern yourself entirely with what the diagram means.

Professional-Quality Output

Flying Logic outputs print-quality PDF files, as well as JPEG and PNG files suitable for use in presentation software. Flying Logic Pro can also export diagrams to Microsoft Project™ when you are ready to move to traditional project management techniques.
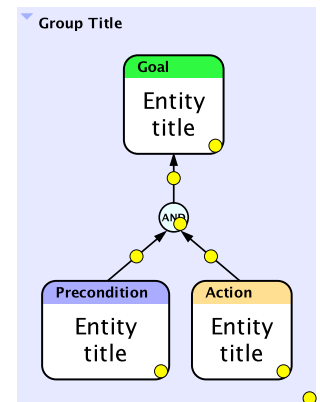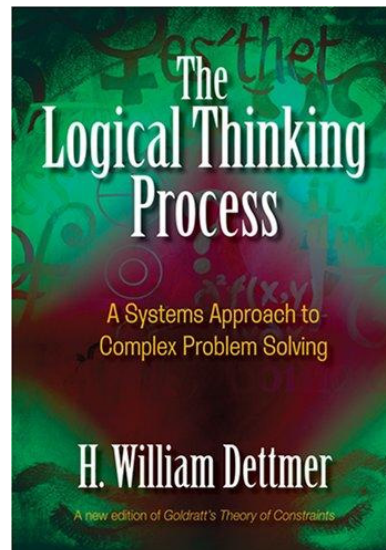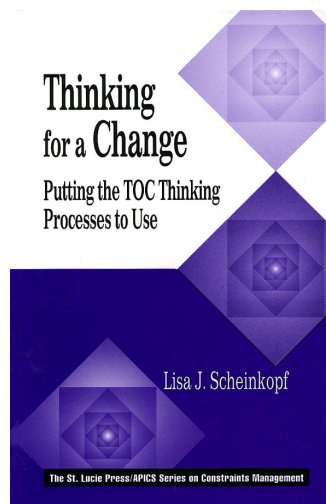
FLYING LOGIC

For downloadable demos, purchasing information and more visit

FlyingLogic.com

Recommended Reading

Get the most from Flying Logic! For detailed information on the Theory of Constraints and its Thinking Processes, we recommend these books. Click the links to order now on Amazon.com:

• **Thinking for a Change: Putting the TOC Thinking Processes to Use** by Lisa J. Scheinkopf.

• **The Logical Thinking Process: A Systems Approach to Complex Problem Solving** by H. William Dettmer.

And don't miss these **FREE** documents, included with Flying Logic, or available for separate download:

• **Flying Logic User's Guide**, covering all operational aspects of using the software.

• **Thinking with Flying Logic**, an introduction to the essentials of the Thinking Processes, and how to use them with Flying Logic.