

Navigating the I-TRIZ Knowledge Base Using Hyperbolic Trees

Ron Fulbright
University of South Carolina Upstate
Spartanburg, SC 29615

rfulbright@uscupstate.edu

ABSTRACT

The I-TRIZ knowledge base consists of over 400 operators and several hundred case studies. At its current size, the knowledge base is overwhelming to browse and use effectively without the assistance of some kind of software tool and future plans include greatly expanding the content in the body of the operators and adding several thousand case studies. Currently available software assists practitioners in utilizing the knowledge base to synthesize innovative improvements to systems by focusing attention to a subset of the operators—those the software determines to be applicable to the system under study. Even with the software's assistance, it is easy to get lost in "operator space." We seek a method of displaying the content of the knowledge base allowing users to interactively navigate through operators and case studies while retaining the sense of where they are in the knowledge base. We also seek an interface inviting users to explore a knowledge base that will continually grow in size and complexity.

1. INTRODUCTION

TRIZ (the Russian acronym for "the theory of inventive problem solving," pronounced "trees") is a methodology, knowledge base, and tool set designed to assist practitioners in generating innovative solutions to problems [1]. Began in 1946 by Russian engineer and researcher Genrich Altshuller and colleagues, TRIZ went unknown to the Western world until some of his writings were translated into English in the 1980s. Some of the contributions from this, the so-called "classical" era of TRIZ, include [2] [3]:

- 40 principles of innovation
- Lines of Evolution
- 72 standard solutions
- Substance/field analysis
- ARIZ (algorithm to solve an inventive problem)

The "contemporary" era of TRIZ began in the late 1980s and early 1990's. This era began in Russia but moved to the USA with the start of a new company, Ideation International which moved most of the principle TRIZ scientists to the USA. The result of development by this group of scientists is called "I-TRIZ" and features:

- Over 400 operators (superseding the 40 principles)
- Methodology for use by the masses
- Family of software applications
- Problem Formulator diagramming tool
- Documentation of case studies
- Compilation of knowledge base

The I-TRIZ knowledge base consists of over 400 operators, 1300 illustrations, 4000 individual screens, and 14,500 links [4]. Whereas each of the original 40 principles represents a general innovative approach, each I-TRIZ operator encapsulates a specific transformation. Associated with each operator are one or more case studies describing an example of the application of the operator, including an illustrated description of the problem solved. Operators are associated with other operators both by direct linking ("see also" links), and also by grouping operators into logical associations:

- Universal
- General
- Specialized
- Auxiliary

Operators are also arranged hierarchically. Some operators subsume, and "point to," a collection of lower-level operators. In some cases, three or four

levels of hierarchy need to be traversed to arrive at a “primitive” operator (one that contains no links to a lower-level).

The I-TRIZ knowledge base represents an astounding distillation of human innovative thought but is too expansive and not in a form facilitating easy navigation by users. The purpose of the Ideation software applications is to assist the user in determining which operators, or operator sets, are applicable to the problem under study. The software selects chosen operators and presents them to the user who may then explore them one-by-one.

While this is an effective method, we feel that much more could be gained if users were able to browse the knowledge base directly. Currently, the knowledge base comprises thousands of HTML files connected together via hyperlinks. While this is a convenient form, it does not afford a “global” view of the knowledge base and it does not encourage browsing and self-discovery on the part of the user. Over the last two years, we have been exploring different display and user interface technologies in hopes of identifying a new presentation style for the I-TRIZ knowledge base that would afford this flexibility. This paper discusses some of the alternatives we explored and concludes that a hyperbolic tree is the best method.

2. Information Presentation Mechanisms

This section describes some of the other information presentation mechanisms we explored. For each, we give a brief description and example followed by a list of arguments for and against the mechanism.

2.1 Desktop Metaphor

Current Microsoft Windows versions allow a significant amount of customization of the appearance and contents of the desktop and even allow multiple desktops for each user. The customization can be under programmatic and user control. We envision using the desktop itself as the knowledge base navigation tool by arranging icons and other graphics to visually organize the contents of the knowledge base. The desktop would also serve as a “project space” allowing users to add their own files and arrange them



as they desire. Mechanisms for sharing the Windows desktop among users and with others over the Internet exist too so the desktop could be used as a collaborative platform.

Pros

- Metaphor is familiar to users
- Windows supports the desktop well
- User has control over arrangement and visual appearance (without custom programming)
- Drag-and-drop applications can be used
- Existing HTML files could be used

Cons

- Does not provide a searchable “database-style” interface to the knowledge base (although the Windows indexing service could be used)
- The Windows platform may change over time with future versions yielding consequences we cannot anticipate
- The desktop does not lend itself easily to hierarchical information other than the traditional folder/subfolder representation
- Interaction with the desktop causes multiple windows to open which hide the desktop itself. The user will have to constantly move through his or her work to access the I-TRIZ knowledge base lying on the desktop underneath

2.2 Folder Trees

The folder tree is the standard representation for files and is readily understandable and usable by most computer users. The folder tree features expandable subtrees (the +/- icons) and a icon/text format. This is a common control in Visual Studio, so is easily programmable and quickly makes any application look and feel like standard desktop applications.



Pros

- The nested folder metaphor is familiar to users
- Handles hierarchical information well to any number of levels
- Common control in programming environments
- Easy to implement right-click and drag-and-drop interactivity

Cons

- Does not represent associative information well (information requiring inter-linking)
- Impossible to fit knowledge base on one screen. A folder tree implementation is likely to extend across several screens requiring a very long scroll bar action to move from one side of the knowledge base to another
- Only the portion of the knowledge base that fits on one screen is visible at any time

2.3 The Brain

The Brain [5] is a commercial information visualization product allowing the user to associate an item to any other item in a dynamic network representation. Each node in the network is a “thought” and can represent an idea, a document, Web site, etc. Thoughts can be associated with folders in a file system or a Web site. A setting allows The Brain to automatically index the information source and build a graphical representation of the source.



Pros

- Handles associative information well
- Handles different file types
- Handles hierarchical information well
- Employs attractive animation when clicking on a node to bring it into focus
- A software developer’s kit is available so one can build their own application with Brain-like associative interfaces

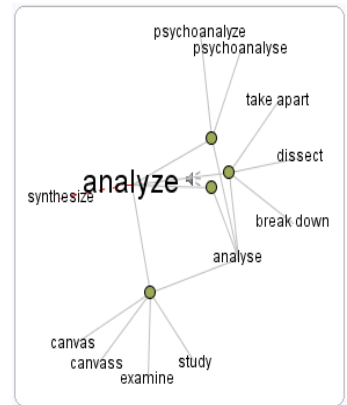
Cons

- Not easily data-driven (as in XML-driven) but this may be easier in future versions
- A large number of nodes quickly renders the screen unreadable or too cluttered (labels become truncated)
- There is no way to “focus” attention on one portion of the knowledge base without clicking on a node and navigating away from your current point of interest
- For thousands of nodes, one would get lost after a few clicks (essentially has multiple screens)

- Software positions nodes automatically so users are not able to customize the appearance
- Opening a document attached to a node opens a new application which hides The Brain behind it. After some time, the desktop becomes cluttered with orphaned windows

2.4 Spider Diagrams

A very effective user interface is the spider diagram used in the Visual Thesaurus [6] product. When the user looks up a word, synonyms for each meaning are displayed as branches of a spider diagram. The hubs on the diagram are interactive as are the words. When a new word is clicked, it becomes the center of the diagram and the other nodes move to occupy the screen, automatically spacing themselves. The display engine behind Visual Thesaurus is the Thinkmap [7] visualization component. This component is sold as a software developer’s kit to OEMs so would facilitate writing custom applications.



Pros

- Handles associative information well
- Attractive animation, professional look and feel
- Can be data driven via an XML file

Cons

- Does not handle different file types well
- Screen becomes cluttered and labels run together or become truncated (designed for one or two word labels)
- There is no global view of the entire knowledge base. Clicking several levels into the hierarchy quickly disorients the user

2.5 Self-Scaling Spider Diagrams

An interesting variation on the spider diagram is a diagram that automatically expands as the user explores nodes. The Thinkmap engine described above shoves every node onto the same screen regardless of the cluttering it may cause. With a self-

scaling diagram, some nodes will be visible while their children nodes remain hidden until the user clicks on the node. At this time, the children nodes are displayed and remain visible. The effect is that the user controls the “growth” of the diagram by exploring it.

Pros

- Prevents cluttering the screen
- Invites the user to explore
- All advantages of The Brain and Thinkmap

Cons

- After nodes are made visible, they remain on the screen and become accessible by panning and scrolling the screen. This causes the diagram to quickly exceed a single screen.
- Unless the user explores each node, they do not know if something of use is attached to that node. While this interface invites exploration, it also requires exploration to expose the entire information space.
- No known SDK exists for this style of diagram (although a more exhaustive search may discover one)

2.6 Zoomable User Interfaces

Researchers at the University of Maryland have developed a zoomable user interface (ZUI) [8] named Piccolo. A piccolo screen looks like any other window in Microsoft Windows with the exception that the user can zoom in on the contents of the window.



This makes it possible to display an arbitrary amount of information on the screen, and then let the user zoom into the area he or she wants to focus on. We envision a screen affording a global view of the I-TRIZ knowledge base inviting users to zoom in on a selected area. From the global view, any details of a specific operator or case study would not be readable. However, as the user zooms in, more detail about a smaller subset of the knowledge base is made visible.

Ultimately, a single operator and/or case study would fill the screen.

Pros

- Affords a global view of the knowledge base while giving the user control
- Invites the user to explore
- A software developer’s kit is available along with OEM licensing

Cons

- Once the user has zoomed in, anything else on the screen is lost (pushed away by the zoom)
- Other than zooming back out and back in again on another section, there is no easy way to navigate to another portion of the knowledge base.
- Currently, only objects in the Piccolo object model can be displayed and zoomed. This means that a significant amount of custom programming will be required to replicate user interface interactivity that will be expected by the users (and is readily available in a “normal window”)

3. Hyperbolic Trees

The previous section described several of the user interface technologies we considered. This section describes the *hyperbolic tree* and explains why we chose it as the best user interface for the I-TRIZ knowledge base browser.

A hyperbolic tree continually displays all information on a single screen but brings into legible resolution only the area focused on by the user. The user moves information in and out of



resolution by dragging with the mouse. As the user drags, some parts of the diagram come into resolution while other parts fall out of resolution. The effect is that of sliding the information along the surface of a 3-dimensional orb (hence the name “hyperbolic”). However, even as it slips out of resolution, the user can always orient oneself with the overall structure of the information. For example, no matter how far the diagram is skewed, the root node is visible. Therefore,

it is very easy to quickly move from one point in the knowledge base to another. This invites exploration without the risk of getting lost. The nodes are interactive and clicking on a node can be handled programmatically so any amount of interactivity can be implemented.

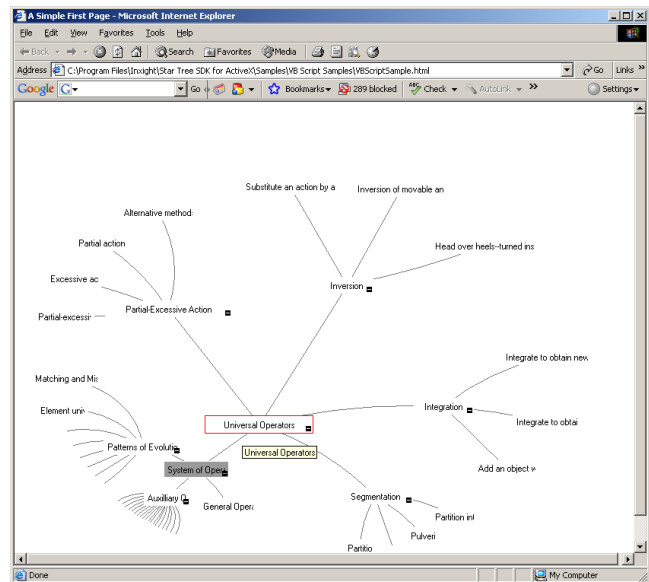
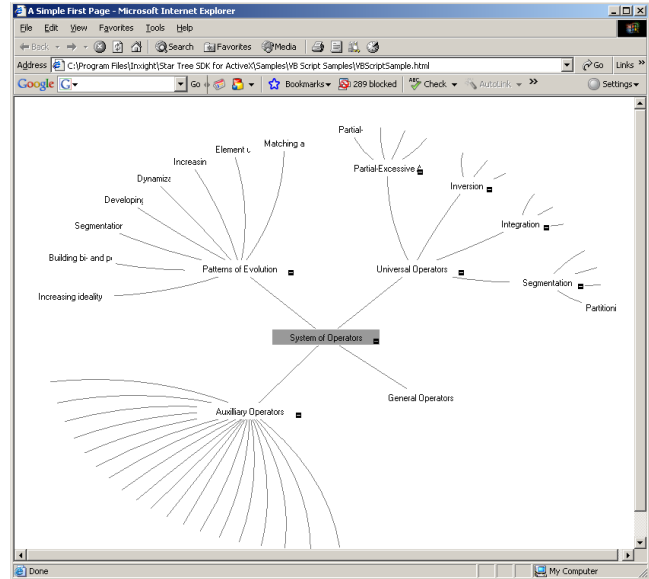
Currently, the hyperbolic tree is a 2-dimensional graphical representation. An interesting future enhancement would be for the graphical rendition to be a true 3-dimensional surface around which the user can navigate. In fact, work on this has already been done by a Stanford graduate student [9]. In private communications with this graduate student, the software is apparently free of legal entanglements and available for use which provides us with a possible course of action for the future.

Hyperbolic space (non-Euclidian plane) was first described by Poincare and used for information visualization by researchers at the Xerox Pal Alto Research Center (PARC) [10]. InXight was a PARC spin-off that commercialized hyperbolic trees research into a product named Startree. InXight has now been acquired by Business Objects who sells the Startree product as well as other information visualization tools [11].

The power behind a hyperbolic tree is its ability to orient the user (“context”) within a large information space, but permit one to quickly drill down to a specific piece of information (“focus”). The combination of context and focus is what makes hyperbolic trees interesting.

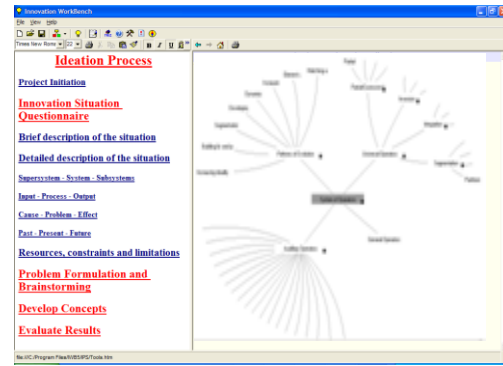
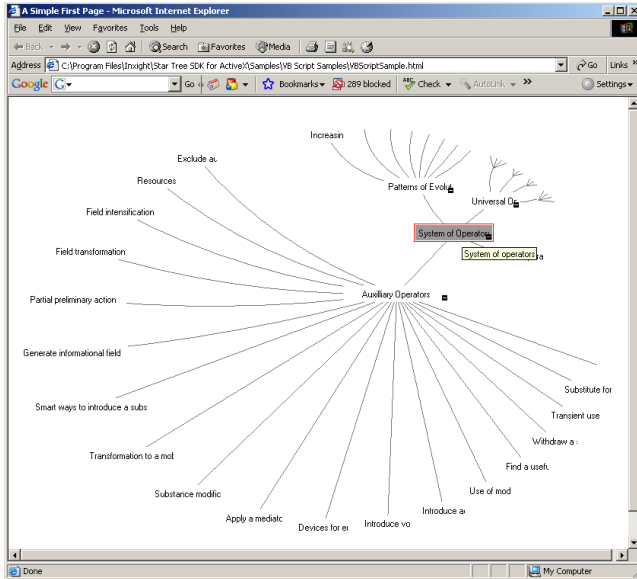
4. Structuring the I-TRIZ Knowledge Base

We have developed a proof-of-concept model using Business Objects’ StarTree product (trial version). The following figures show how one might use a hyperbolic tree to navigate I-TRIZ operator space. The first image shows the root node “System of Operators” with branches leading to major categories “Patterns of Evolution,” “Universal Operators,” “Auxiliary Operators,” and “General Operators.” The second image shows the node “Universal Operators” slid into focus thereby exposing second-level categories “Segmentation,” “Integration,” “Inversion,” and “Partial/Excessive Action.”



The final image shows an expansion of the “Auxiliary Operators” node. Note how no matter how the image is manipulated, the overall structure remains visible without screen changes.

Clicking on a node would display more information about that operator. For the lowest-level operators, the description of the operator would be displayed along with available case studies for that operator. The case studies could be displayed either as additional branches in the hyperbolic tree.



An important feature of the StarTree product is that it can be driven programmatically or by an Extensible Markup Language (XML) file. The latter makes the display data-driven and is how the proof of concept was done. Therefore, any distributed product with StarTree embedded can be updated simply by downloading a new XML file. Through this mechanism, the knowledge base can expand indefinitely.

4.1 Uses for a Hyperbolic I-TRIZ Browser

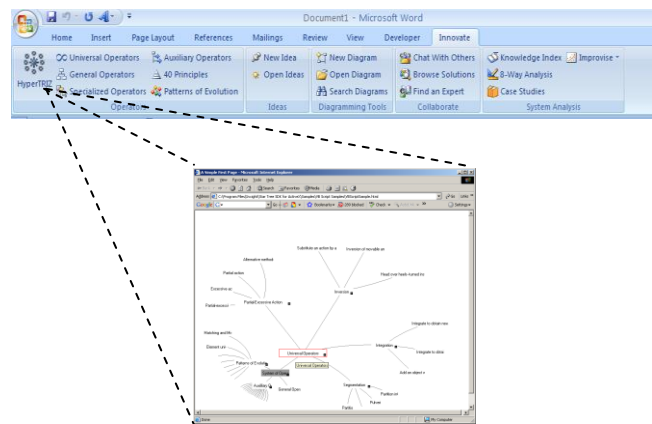
We are recommending the use of a hyperbolic tree user interface to make the I-TRIZ knowledge base more usable, open, and inviting in future software implementations. This section describes some specific implementation ideas.

Existing Applications

The Ideation software suite currently consists of several standalone desktop applications, running on IBM-PCs, all using the same HTML-based operator display mechanism described earlier. Certainly, a hyperbolic tree-based browser could be included in any future software release.

Ribbon-Based User Interface

In a separate document, we describe a vision for a ribbon-based set of I-TRIZ tools [12]. A hyperbolic tree I-TRIZ knowledge base browser would be an ideal tool to include as one of the tools available on the Ribbon



The HyperTRIZ Ribbon would appear in any standard Microsoft Office 2007 application and would bring I-TRIZ tools to the user's standard work area without having to launch a new applications.

Web-Based Applications

The StarTree product is Web-ready, so it could be used on any Web page or Web-based portal or user interface. We can envision placing it on a company's Web site or on a Web site supporting the I-TRIZ community.



Vertical Software Applications

One strategy being discussed for future I-TRIZ software is to “verticalize” for a specific application domain. Instead of a generic application like the Innovation Workbench (IWB) that is intended to be used on a problem in any domain, a verticalized application, employing I-TRIZ methodologies, tools, and knowledge base would be customized for a particular domain. A hyperbolic browser could be employed in such a customized application and give the suite of applications a common look and feel.

5. Works Cited

1. About TRIZ. *Altshuller Institute for TRIZ Studies*. [Online] [Cited: May 8, 2008.] <http://www.aitriz.org/ai/index.php?page=triz&article=about>.
2. **Shulyak, Lev**. Introduction to TRIZ. *Altshuller Institute for TRIZ Studies*. [Online] [Cited: May 12, 2008.] <http://www.aitriz.org/ai/articles/40Ptriz.pdf>.
3. **Alla Zlotin, Boris Zlotin**. History of TRIZ and I-TRIZ. *Ideation International*. [Online] [Cited: May 12, 2008.] <http://www.ideationtriz.com/history.asp>.
4. **Zlotin, Alla Zlotin and Boris**. An Integrated Operational Knowledge Base (System of Operators) and the Innovation Workbench System Software. *Ideation International*. [Online] September 22, 1992. [Cited: May 12, 2008.] http://www.ideationtriz.com/paper_Integrated_Operational.asp.
5. The Brain: Visual Information Management. *The Brain Technologies*. [Online] [Cited: May 12, 2008.] <http://www.thebrain.com/>.
6. Visual Thesaurus home page. *Thinkmap Visual Thesaurus*. [Online] [Cited: May 12, 2008.] <http://www.visualthesaurus.com/>.
7. Thinkmap home page. *Thinkmap*. [Online] [Cited: May 12, 2008.] <http://www.thinkmap.com/>.
8. A Structured 2D Graphics Framework. *Human Computer Interaction Lab*. [Online] University of Maryland. [Cited: May 13, 2008.] <http://www.cs.umd.edu/hcil/piccolo/index.shtml>.
9. **Munzner, Tamara**. Interactive Visualization of Large Graphs and Networks. *Stanford University*. [Online] June 2000. [Cited: May 13, 2008.] http://graphics.stanford.edu/papers/munzner_thesis/.
10. *A focus+context technique based on hyperbolic geometry for visualizing large hierarchies*. **J. Lamping, R. Rao, and P. Pirolli**. s.l. : Proceedings of the Conference on Human Factors in Computing Systems, 1995.
11. Inxight is now part of Business objects. *Business Objects*. [Online] [Cited: May 13, 2008.] <http://www.businessobjects.com/company/acquisitions/inxight.asp>.
12. **Fulbright, Ron**. *Embedding I-TRIZ Using the HyperTRIZ Ribbon*. Spartanburg, SC : University of South Carolina Upstate, 2008. DOI-RDF-051608A.