ABSTRACT
In order to stay competitive, today’s businesses need to be able to innovate and problem-solve. Specialized software tools such as I-TRIZ are being developed to give practitioners the ability to leverage the total sum of human innovation knowledge. This paper explains what I-TRIZ is, detail the operator structure, and present 20 case studies of how the I-TRIZ operators apply in the realm of information technology.

Keywords
TRIZ, I-TRIZ, operators, operator space, innovation, computer-aided innovation.

1. INTRODUCTION
As we increase our ability to gain knowledge from stores of electronic information, the importance of being able to innovate with that knowledge will become increasingly important. As Fulbright explains [4]:

Human history teaches us how new types of workers are continually created to keep pace with technological advances. As an outgrowth of the information age, companies are spending increasing amounts on innovation in order to remain competitive in the globalized marketplace and this is leading us into a new era—the innovation age.

Innovation workers need specialized tools. Ideation International, the company that produces I-TRIZ software, was established in 1992 to fulfill the need of the natural market for problem solving [8]. I-TRIZ is based on the original TRIZ concepts that were developed in the 1950s and 1960s by Genrich Altshuller, and that have been extended today by various other researchers.

2. TRIZ
TRIZ is a Russian acronym for “Teoriya Resheniya Izobretatelskikh Zadatch” which when translated stands for the “Theory of Inventive Problem Solving” [7][2]. Altshuller and his colleagues studied more than 200,000 patents and discovered that the same fundamental problems had been addressed by a number of inventions -- in different areas of technology. He also observed the same fundamental solutions were used over and over again, often separated by many years [8]. He came to the realization that there are algorithmic approaches that can be utilized in the process of solving technical problems [1].

The aim of TRIZ is rooted in the belief that universal methods of problem solving and innovation can be developed based on discovered patterns of how systems evolve. [4] Among the many contributions of Altshuller and his colleagues in the realm of TRIZ are the 40 principles of innovation. A very basic idea of the 40 principles is that after going through the systematic analysis of a problem, one can pinpoint specific issues to work on and then use the list as a guide to trigger the innovative process. For a quick example, consider this problem:

*Painting a complex item with a paint sprayer does not work well because the paint fails to cover 100% of the parts.*

Looking at this problem based on the principles of innovation leads to several solutions. One principle is Segmentation:

<table>
<thead>
<tr>
<th>Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Divide an object into independent parts</td>
</tr>
<tr>
<td>• Make an object easy to disassemble</td>
</tr>
<tr>
<td>• Increase the degree of fragmentation or segmentation</td>
</tr>
</tbody>
</table>

This leads to the idea that the item should be divided into smaller parts and then painted individually. This suggests painting the individual parts *before* they are assembled, which falls under the category of the principle of Preliminary Action. [7] Another solution may come about by considering the principle of Inversion:

<table>
<thead>
<tr>
<th>Inversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Invert the action used to solve the problem</td>
</tr>
<tr>
<td>• Make movable parts fixed and fixed parts movable</td>
</tr>
<tr>
<td>• Turn the problem or process upside-down</td>
</tr>
</tbody>
</table>

Thinking of the problem in the terms of inversion (i.e. ‘turning the problem on its head’) might lead the user to abandon the process of spraying paint on the item, and instead dip the item into the paint [7]. Using the principles in this fashion takes away the trial-and-error aspect of innovation and allows the process of problem-solving to build on what has been done before. In fact, [2] states that one of the principle tenets of TRIZ is that “Somebody someplace has already solved this problem, or one very similar to it. Creativity is finding that solution and adapting it to this particular problem.”

3. I-TRIZ
The development of the TRIZ methodologies finds its current incarnation in the I-TRIZ software, which includes four different modules [7]:

1. **Segmentation**
   - Divide an object into independent parts
   - Make an object easy to disassemble
   - Increase the degree of fragmentation or segmentation

2. **Inversion**
   - Invert the action used to solve the problem
   - Make movable parts fixed and fixed parts movable
   - Turn the problem or process upside-down

3. **Operations**
   - Solve the problem using the operator structure

4. **Structured Analysis**
   - Analyze the problem using matrices and diagrams

By utilizing this software, practitioners can quickly and efficiently find innovative solutions to complex problems.
3.1 Operators

The operators are what the IPS module is built upon. They are a collection of over 400 suggestions for incremental changes, based on the original 40 principles of innovation. However, the I-TRIZ operators differ in that they are more detailed and tend to focus on thinking toward a specific change to the system [4]. For example, I-TRIZ replaces the TRIZ “segmentation” principle with five operators:

- Make an object dismountable
- Partition into simple shaped parts
- Pulverize
- Link degeneration during partitioning
- Partitioning followed by integration

“Operator space” or the “operator network.” Is arranged into four groups:

- Universal applicable to any problem
- General applicable to many situations
- Specialized applicable to a specific situation
- Auxiliary improve in terms of ideality and feasibility

However, with over 400 individual operators, it is difficult to intellectually grasp the complex array of interrelationships between them. One of the ways that Ideation helps the user decide where to start is with a tool called the Problem Formulator (PF). The PF allows the user to map out the problem using a cause-effect analysis of useful and harmful functions of the system. Based on the types of relationships between useful and harmful aspects of the system, the PF generates the subset of operators most likely to be beneficial to the system [7]. Currently the user interface for navigation through the operators is a simple explorer database using HTML pages. However, this method only allows the user to link backward and forward through pages, showing only a linear path through the operator space. For instance, when navigating through the operators, one might start at the universal level and see this (arrows added for emphasis):
While that particular link only leads to one more page, other operator page links lead to many more paths through the operator space. The continual branching and looping back within the space of the operators is one of the shortcomings of the software but with a linear design, the dynamics of the relationships between the operators would be lost.

Some suggestions would be that the future design of the software could include a UI that features spider-diagram and branching functions that more adequately depict the operator space. For instance, the diagram shown in Figure 1 represents the chain of operators connecting from one of the universal operators, “Partial/Excessive Action.”

Although it goes five levels deep many of the end nodes are connected to still deeper levels. This particular diagram is a small snapshot of the system; it has only a little over 50 of the 400+ operators. It is meant to demonstrate how quickly the connections become overwhelming. A true representation of the operator space, with every operator within the network is what is needed.

Intuitiveness in a UI is a relative term; i.e. each user has his or her opinion of what is easy to use. However, an increasingly important factor in all interface designs when it comes to intuitiveness is known by the industry buzz phrase, “the cool factor.” The newest generation of IT users has grown up in a world dominated by IT. In order to make complex ideas like I-TRIZ accessible and accepted, the interface has to be dynamic and engaging. This can be accomplished in a couple of ways.

Navigation can be done using a zooming user interface (ZUI). This way the entire network of operators can be depicted as nodes and their interconnections in the network. The user could then zoom out to see the scope of the connections, and zoom in to examine the details of an operator and its neighboring operator nodes.

One example of a scalable interface is InXight’s StarTree organizational software [15]. With this software LexisNexis is used to map out their online resources. Clicking on a node zooms in to that region of the graph and magnifies the nodes connected to the selected node.

Another option is to use a 3-D model. A 3-D model would have the benefit of being able to represent an operator’s relative usefulness to the user by way of its distance in space.

A feature that both would need to have would be the option to make the nodes self-branching. This way the user wouldn’t clutter the interface with irrelevant nodes. The user could click on a node and instead of being taken to a new page the node could branch into the linked operators. Then the user could click on the new nodes and they would branch in to more new nodes, and this could be repeated as much as needed while still showing the overall organization and pathway the user took.

### 3.2 case Studies

Once the user has decided upon an operator to investigate, he or she can choose from one or more case studies associated with the operator. For example, here is one of the case studies from the I-TRIZ library [8] is:

**Operator:** Abandon Symmetry:

*If an object is symmetrical consider reducing its weight by abandoning the symmetry. Consider, for example, excluding a part of the object that does not bear the main load.*

**Case Study:** Designing asymmetrical mounts

For aesthetic reasons, motor and generator mounts are often designed with symmetrical shapes. But because the machines rotate, the load on the mounts is actually asymmetrical.

To reduce the weight and conserve material, mounts for non-reversible units should be designed to support only the loads they must actually bear.

![Figure 3. Motor mount case study](image-url)

Again, because one of the most important tenets of TRIZ is that somewhere, somebody has solved a similar problem before, the key to being creative is to take that knowledge and apply it to the new problem.

Currently the case studies are mostly in the realm of physics, chemical engineering, and chemistry. It is interesting that today’s heightened need for TRIZ for innovation is greatly due to the information technology explosion, yet there is no focus in the I-
TRIZ software for IT! There is a dual need to study how the I-TRIZ principles relate to IT:

1. Tomorrow’s “innovation worker” will be one who works with stores of electronic information. An IT specialist may have the innovative process stifled by only seeing examples in a field of science unfamiliar to them. The addition of case studies that fall within their “comfort zone” of thinking will benefit the software by giving it a much broader user-base.

2. The current TRIZ thinking is that no new principle can be discovered that is not in some way covered by another principle. However, it has been decades since the vast majority of research was done on the patents. Technological innovation since then has been mainly in the IT field. It is possible that researching IT and TRIZ principles could yield the discovery of new principles that only apply in the IT domain. [5]
4. APPLICATIONS TO INFORMATION TECHNOLOGY

To illustrate how the operators can relate to the realm of IT Table 1 shows 20 operators and their IT applications. This is followed by several examples of representative case studies. All 20 case studies have been documented in this way [14]; however only the first 5 case studies are examined here.

4.1 Confidential Objects

**Operator:** Partitioning followed by integration:

*Divide or partition an object into parts, then integrate the parts in a different way. Search for integration variants that will produce new results.*

**Case Study:** Confidential Objects

In a network consisting of several users, the notions of fault tolerance and information security in data transmission are very important. One of the ways to ensure that accidental and intentional damage to data can be fixed, as well as keeping data confidential, is to create confidential objects using the Fragmentation Redundancy Scattering (FRS) method [3].

This method takes sensitive data and breaks it down into fragments (confidential objects) in such a way that each fragment means nothing on its own. Then each portion is given a redundancy to allow for errors in transmission. The fragmented data is only allowed to be put back together by a trusted node within the network. All other nodes only have access to the fragments (Diagram 1 in the Appendix).

This method is not only applicable to data transmission, but it can be used in other areas of computing such as permanent data storage as well.

4.2 Radio Frequency Identification

**Operator:** Integrate to obtain a new property:

*To obtain a new property for a system or process, integrate it with a component or subsystem that has the desired property.*

**Case Study:** Radio Frequency Identification (RFID)

Inventory management is a critical monetary concern in every business. It is imperative that items be tracked throughout the shipping process and are properly checked in. An item in a warehouse or storeroom can not tell the inventory management system if it arrived in the last shipment. One of the newer methods for inventory tracking involves integrating RFID into the inventory process.

RFID tags are placed on pallet shipments, and in some cases on individual items. This way, hand held scanners can log the items into the system quickly and accurately. Any variances in inventory counts are promptly found and can be fixed by tracing the shipping path. Also, another use of RFID is to create “smart shelves” in the store that can alert employees when merchandise is running low on the sales floor.

Integration of RFID tags has occurred in many areas of business. Retail stores are one of the first examples that come to mind, but the need for accurate information tracking does not stop there. Some hospitals use RFID to track medication for patients. Each patient has a RFID tag in their wristband that identifies them, what medications they have been prescribed, and the proper dosages. The bottles of medication also have identifying tags. Automotive companies use RFID technology to create “smart keys” that can talk to the vehicle. If the correct key is not used, the car won’t start, thus decreasing the likelihood of auto theft. According to [13] the current vertical industries involving the use of RFID are:

- Retail
- Manufacturing
- Automotive
- Health-Care
- Pharmaceutical
- Distribution/Logistics

Table 1. Summary of I-TRIZ Operators with IT Examples

<table>
<thead>
<tr>
<th>Operator</th>
<th>IT Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Partitioning followed by integration</td>
<td>Confidential objects [6][11]</td>
</tr>
<tr>
<td>2. Integrate to obtain a new property</td>
<td>Radio Frequency Identification</td>
</tr>
<tr>
<td>3. Use multiple inexpensive objects</td>
<td>Rapid-prototyping in software design [6][12]</td>
</tr>
<tr>
<td>4. Concurrent operations</td>
<td>Dual-Processor systems and Dual-Core processors</td>
</tr>
<tr>
<td>5. Vaccination</td>
<td>Anti-virus and backups</td>
</tr>
<tr>
<td>6. Add a marker</td>
<td>Markers in programming</td>
</tr>
<tr>
<td>7. Add an object with a required property</td>
<td>Power supply</td>
</tr>
<tr>
<td>8. Disable an undesired action</td>
<td>Limits on password attempts</td>
</tr>
<tr>
<td>9. Eliminate a stressful operation</td>
<td>Computer cooling methods</td>
</tr>
<tr>
<td>10. Integrate to obtain an opposite property</td>
<td>NOT gates</td>
</tr>
<tr>
<td>11. Make an object dismountable</td>
<td>Hardware components</td>
</tr>
<tr>
<td>12. Partition into simple-shaped parts</td>
<td>Object-oriented software design</td>
</tr>
<tr>
<td>13. Preliminary placement of an object</td>
<td>Buffers</td>
</tr>
<tr>
<td>14. Pulverizing</td>
<td>Altering images in photo software</td>
</tr>
<tr>
<td>15. Removal</td>
<td>Recycle bin and uninstallation</td>
</tr>
<tr>
<td>16. Use the reverse side</td>
<td>Circuit board design</td>
</tr>
<tr>
<td>17. Upside-down or inside-out</td>
<td>Inverted views of lists in data management</td>
</tr>
<tr>
<td>18. Use an auditory field</td>
<td>Sound notifications</td>
</tr>
<tr>
<td>19. Specialization</td>
<td>Specialization of and within computer chips</td>
</tr>
<tr>
<td>20. Exclude duplicate elements</td>
<td>Removal of redundancy in database design</td>
</tr>
</tbody>
</table>
4.3 Rapid Prototyping in Software Development

Operator: Dispose:

Use multiple inexpensive objects.

Case Study: Rapid Prototyping in Software Development

When software is being designed, software engineers often employ rapid prototyping to quickly create prototypes that can be given to a user to test and provide feedback that is essential to making the software do what is needed. This is sometimes more efficient than creating the software in its entirety, letting users use it and then going back to fix all the problems they identify.

There are several kinds of prototypes that software engineers use. For instance, when designing a user interface, the UI concept might be shown to the users on paper or on PowerPoint slides. This is known as “paper and pencil prototyping” because no actual programming or design needs to be shown on a computer. The user can provide suggestions on the paper about how the UI might be changed to be more efficient and user-friendly.

Another method is called “parallel design.” This might involve different design groups coming up with separate user interfaces without any interaction or influence from the other groups. This way a diverse array of ideas can come to fruition, with the final product coming from a combination of the designs, taking the best features from each. Since many ideas are generated without having to spend the money to implement them, it is a very cheap way of exploring a range of possible concepts before selecting the optimum [10].

4.4 Dual-Processor Systems and Dual-Core Processors

Operator: Concurrent Operations:

Try to combine homogeneous or neighboring operations and replace a step-by-step operation with simultaneous, parallel operations.

Case Study: Dual-Processor Systems and Dual-Core Processors

Although a CPU is capable of handling thousands of instructions a second, it still has to execute them in order, one at a time. One of the newer ways to allow computers to process instructions is to do them in parallel using either a dual-processor system or a dual-core processor.

A dual processor system is one in which two separate processing chips are connected to the motherboard. Conversely, a dual-core processor is a single processor with two CPU “cores” that can process information independently of each other.

Either setup is capable of processing two sets of instructions at the same time. Some software applications can be written so that they split their instructions into threads that can be processed in parallel instead of serially. This is currently not the norm due to how complex the process is to create such software. The main advantage to dual-processor and dual-core systems is their ability to run multiple applications at the same time.

For instance a dual-core system might be running a virus scan and playing a movie at the same time. Instead of having to juggle the computational requests of the two separate applications, a dual-core processor can handle them both simultaneously, allocating one core to the virus scan and the other core to the movie (Diagram 2 in the Appendix).

4.5 Backups and Anti-Virus Software

Operator: Vaccination:

Consider decreasing the sensitivity of a system or process to a harmful effect, or creating immunity to the harmful effect.

Case Study: Backups and Anti-Virus Software

Now that the world is connected via the Internet and World Wide Web, as well as local area networks and wide area networks, the need for vaccination against computer viruses has seen a great increase. There are several levels of threat when it comes to viruses. Viruses can do something relatively harmless such as display a message such as “Ha Ha you are infected,” at a certain time, they can maliciously delete files, or they can log keystrokes in order to steal passwords and account information. Even so-called “good viruses” that attempt to optimize settings and seek out bad code can use up system resources, slowing the computer down or inadvertently delete or alter important system files.

One method to look at vaccinating is to decrease the sensitivity of the computer to viruses. This can be done by storing sensitive data in areas other than the hard drive and by running programs from an optical or USB drive. There are many types of digital storage that can be used to back up files such as external hard drives, CDs, DVDs, floppy disks, flash memory devices, etc. Backing up files in various sources is not a method for preventing viruses; in actuality it can spread them if one of the backed up files is infected. It also makes the information more vulnerable to being stolen. However, having extra copies of sensitive information decreases the sensitivity of the system to the viruses.

The other method of vaccination is to introduce immunity to the computer. This is accomplished by anti-virus software that actively seeks out viruses and deletes them.

5. CONCLUSION

The I-TRIZ software will become an increasingly important tool for tomorrow’s innovation worker, particularly those of us in IT. The current version of I-TRIZ does not have the interface and IT focus that it needs to be accepted and understood by those who will become innovation workers. These shortcomings can be fixed by looking at adding a more intuitive interface for navigating and representing the operator space. Also, because the need for innovation is increasing due to the amount of knowledge gained by the information technology explosion, the software needs to be more IT focused. Further study into how the I-TRIZ operators apply to IT can only serve to add a missing portion to the TRIZ knowledge base, and open the principles up to a broader range of innovative minds.
6. ACKNOWLEDGEMENT
The work described here was done at the University of South Carolina as part of the Research Experiences for Undergraduates in Multidisciplinary Computing project and supported in part by National Science Foundation Award # 0353637. I would like to thank Dr. John Bowles, Dr. Caroline Eastman, Dr. Ron Fulbright (USC Upstate) for their help in conducting this research.

7. REFERENCES
[14] Camara, Chad, IT-Focused Case Studies for I-TRIZ, REU research report, [web address to be supplied]