May, 2014

Is Anti-Virus a Necessary Evil?

Umakant Mishra

Available at: https://works.bepress.com/umakant_mishra/124/
Is anti-virus a Necessary Evil?

Using TRIZ Ideality and Contradictions to find out what is Necessary and what is Evil *

By- Umakant Mishra, Bangalore, India

http://umakantm.blogspot.in

Contents
1. Anti-Virus Programs: a Background..............................................................2
2. Using Ideality to determine what is desirable.................................................2
3. The undesirable functions of an anti-virus program.....................................3
4. Using contradictions to differentiate what is Useful and what is Evil ..........4
   4.1 The problem of selection and procurement..............................................4
   4.2 Problem of Anti-Virus renewal.................................................................5
   4.3 Problem of updating Virus Database.......................................................5
   4.4 Problem of Scanning Time .......................................................................6
   4.5 Problem of system performance...............................................................6
   4.6 Problem of false positives.........................................................................6
5. Summary and Conclusion..............................................................................7
Reference: ........................................................................................................7

* This paper was first published in the TRIZ featured Articles of Altshuller Institute of TRIZ studies (USA) during Aug 2013, currently available on the web at http://www.aitriz.org/triz-articles/triz-features/628-anti-virus. This is a repeat publication.
1. Anti-Virus Programs: a Background

While everybody is becoming more and more dependent on computers there are some unscrupulous people who continuously try to misuse the technology and get illegal and illegitimate benefit out of this sophisticated environment. Today the Internet is like a busy street or open market place where you find almost everything you want. As there are some obvious risks when you stand or move in a busy street such as you are likely to be pick pocketed or cheated or even knocked out by a rash driving vehicle. Similar threats loom over the head of a user when he is exposed to the Internet. There are viruses, malware, spyware, spam, hackers and other malefic forces who not only damage the computer system of innocent computer users but also do more dangerous things like stealing their confidential data or stealing their identity to make bank transactions on their behalf.

An anti-virus program is generally considered as a remedy to all the above problems. The anti-virus program (including anti-spyware, internet guard etc.) not only detects the malware in your system but also guards the computer system from external attacks while you are surfing the Internet. It is supposed to detect if a spyware is stealing your passwords or a hacker is making an inward connection to squeeze your PC. But is the current age anti-virus program really giving a remedy to all these problems? Many people think that an anti-virus program along with its siblings like anti-spyware, anti-adware, internet guard etc. create a lot of burden on the user. While the computer users cannot afford to avoid using anti-virus programs most user fell it as a necessary evil.

2. Using Ideality to determine what is desirable

According to the concept of Ideality, the best anti-virus is “no anti-virus” or a “virus free environment where there is no need of any anti-virus”. However, for many practical reasons the above Ideal Final Result (IFR) is not possible to achieve in the present circumstances. When the ultimate IFR is not possible to achieve the problem solver has to take a step backward and consider a lower level IFR\(^1\). The best solution is that which is closest to the Ideal solution. The best solution is that which fulfils all the desirable functions of an anti-virus program without having any of its drawbacks.

\(^1\) Umakant Mishra, Using TRIZ for Anti-Virus Development, Chapter-6: “Using Ideality to find the Ideal anti-virus solution”.
The desired solution from different prospective

<table>
<thead>
<tr>
<th>From users’ perspective</th>
<th>From a practical perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>The computer should never get infected (no need of purchasing, installing or maintaining any anti-virus software)</td>
<td>The computer should never get infected after using some kind of protection like installing an anti-virus product.</td>
</tr>
<tr>
<td>The user should not spend money on buying anti-virus products nor waste time and energy on installing and maintaining those products.</td>
<td>The anti-virus product should be free or at least cheap and should be extremely easy to install and maintain.</td>
</tr>
<tr>
<td>The user should not waste valuable system resources for running anti-virus that could have been used for other purposes to increase productivity.</td>
<td>The anti-virus should run fast and consume minimum system resources.</td>
</tr>
</tbody>
</table>

As a “no virus environment” is practically not possible, using an anti-virus is the most accepted solution in the current scenario despite of its various drawbacks.

**3. The undesirable functions of an anti-virus program**

Any system is built to achieve its Main Useful Functions (MUF). The concept of Ideality in TRIZ helps us to determine what are the useful (or desired) functions of a system and what are the harmful (or unwanted) functions. If we apply that concept on an anti-virus system we can find the actual anti-virus requirements of an end-user and how much of it he is getting from an anti-virus program.

A user gets an anti-virus program with a view to get rid of one type of problem, i.e., virus infection. But unfortunately after installing the anti-virus he gets into more and more problems of different types. The problem begins at the time of searching for a good anti-virus and continues throughout the life afterwards during scanning, updating, upgrading, renewing, reinstalling etc.
The concept of Ideality tells us that the Ideal anti-virus system should consist of all useful functions and should be void of any harmful functions. But this feature of ideality is not easy to achieve. While improving the anti-virus system in this direction we come across various different contradictions. When we try improving one part (or aspect or functionality) of the anti-virus system it results in worsening another part (or aspect or functionality) of the system. These situations are called contradictions. Defining the contradictions help us clearly visualize what is desirable and what is not desirable in a system.

We will not discuss the drawbacks and limitations of anti-virus systems as we have discussed it earlier in separate articles. We will just illustrate a few contradictions solving which can eliminate the evilness of an anti-virus program.

4. Using contradictions to differentiate what is Useful and what is Evil

Contradictions are conflicting situations in a system where improving one parameter of the system affects another parameter of the system which results in a deadlock situation. For example, “scanning all types of viruses requires a lot of system resources thereby makes a system slow.” In this example, “scanning” is desirable but “affecting system performance” is undesirable.

Before solving a problem it is important to define a problem. Formulating the right contradiction defines the exact nature or technicality of the problem and helps to solve the problem in the right way. The following are some of the problems faced by users presented in the form of contradictions. These contradictions clearly differentiate as what is desirable and what is undesirable in an anti-virus system.

4.1 The problem of selection and procurement

Selection and procurement of the right anti-virus product is a difficult job for an ordinary computer user. The user has to compare various aspects of the product like the price of the anti-virus, renewal costs, user friendliness, performance, reliability, load on computer resources etc. An ordinary user cannot be sufficiently knowledgeable to evaluate various aspects of an anti-virus software in order to choose the right product.

---

2 Refer to other articles on “contradictions” in the reference.
The common user wants to install an anti-virus program that is best in its performance but cheapest in price. But choosing the best anti-virus is not easy even for an experienced technocrat. Even specialized organizations engage full time professionals for doing anti-virus evaluations. The common user wants to use the best anti-virus but does not want to spend time on evaluating anti-virus products.

4.2 Problem of Anti-Virus renewal

Even if a person buys an anti-virus product his difficulties are not over. In order to ensure continuity of anti-virus service, the current day anti-virus venders renew the product by automatically taking money from the credit card numbers given by the customers. Although the problem of forgetting about renewing of the anti-virus is solved, this mechanism leads to other problems as below.

If the customer does not opt for automatic renewal then he may forget to renew the anti-virus product at the end of the year which may result in discontinuance of the anti-virus service. On the other hand if he opts for automatic renewal then the money will be taken from his credit card even if he has stopped using that product. The customer wants an automatic renewal but he doe not to pay money if he has already discontinued using that product.

4.3 Problem of updating Virus Database

With the current mechanism of virus detection, installing an anti-virus program is not enough to prevent viruses. One has to update the virus definition database on a regular basis. But updating the virus database is a boring job and the user often avoids to update until the virus affects him again.

We want to scan for the latest viruses but we don’t want to download the latest virus database from the Internet because it is a boring and time consuming job. Even if we update the virus database we don’t want to spend time and resources for the purpose.
4.4 Problem of Scanning Time

The continuous increase in the population of viruses increases the size of signature database which in turn increases the required scanning time. With hundreds of thousands of possible virus types and hundreds of gigabytes of file storage a complete virus scan can take an enormous time which is simply not acceptable to any user.

| Contradiction | If a scanner includes less number of signatures or less number of algorithms then there is possibility of some viruses being escaped. On the other hand if a scanner includes all available signatures and all possible algorithms then the scanning will take very long time. We want to apply more scanning methods to detect all types of viruses, but at the same time we want to apply less scanning methods to finish the scanning fast. |

4.5 Problem of system performance

Running an anti-virus consumes significant amount of system resources and affects the system performance negatively. An anti-virus loaded system takes more time to boot, more time to shutdown and runs slow while executing other programs. That’s why many users don’t like running an anti-virus.

| Contradiction | We want to run an anti-virus to keep the computer free from viruses. But we don’t want the other programs to run slow because of the burden of anti-virus on the computer. In other words, we want the anti-virus to scan the computer, but we don’t want it to affect the performance of other programs. |

4.6 Problem of false positives

There are situations where the anti virus program finds a file to be infected because of insufficient heuristics. Some device drivers are stopped because of false positives. In other cases the anti-virus finds that a system file has been modified but it cannot be sure about whether the file has been modified by a virus or by the user. In such a situation, if the program generates a virus alarm it may lead to a false positive.
If the anti-virus program is not definitive about a suspicious alternation in a system file and raises a virus alarm then it may lead to a false positive. On the other hand if it ignores such a suspicious alteration then it may lead to a false negative. Both the situations are dangerous.

5. Summary and Conclusion

While the anti-virus has become a necessity, it has many negative impacts on our day-to-day computer usage. The anti-virus frequently connects to internet to download its updates and patches and consumes our Internet bandwidth. When the anti-virus scans the computer, it consumes significant amount of system resources thereby making all other programs running slow. An anti-virus program consumes more memory and processor than an average harmful virus. Hence, the anti-virus program is considered as a necessary evil by almost all users.

While there is no problem of it being necessary, there is a need to make it free from its evil characteristics. This is possible by finding and eliminating contradictions within the anti-virus system. The TRIZ method of defining contradictions clearly points out the conflicts in the system, showing what is necessary (or useful or desirable) and what is not necessary (or harmful or undesirable). However, there are also contradictions in the super-system of an anti-virus system. While the end user wants a virus-free environment and does not want to see any computer virus in the world, the anti-virus developer does not want the same. The anti-virus developer wants some virus to survive for the survival of their business. Solving contradictions at higher level in super-systems can lead to tremendous results.

Reference:


   http://papers.ssrn.com/abstract=1916673

4. Umakant Mishra, “Methods of Virus detection and their limitations”,


11. Umakant Mishra, Overcoming limitations of Signature scanning - Applying TRIZ to Improve Anti-Virus Programs,
    http://papers.ssrn.com/abstract=1980629

12. Umakant Mishra, Improving Speed of Virus Scanning- Applying TRIZ to Improve Anti-Virus Programs, TRIZsite Journal, Apr 2007
    http://papers.ssrn.com/abstract=1980638


About the author

After working for more than 18 years in various fields of IT in different organizations Umakant is currently carrying out independent research on TRIZ and IT since 2004. He last worked as Director and Chief Technology Officer (2000-2004) in CREAX Information Technologies (Bangalore). Before that he worked as IS/IT manager (1996-2000) for ActionAid India (Bangalore).

Umakant is a Master in Philosophy (MA), Master in Business Administration (MBA), Bachelor in Law and Logic (LLB), Microsoft Certified Systems Engineer (MCSE+I), Certified Novel Engineer (CNE), Master Certified Novell Engineer (MCNE), Certified Intranet Manager (CIM), Certified Internet Professional (CIP), Certified Software Test Manager (CSTM) and holds many other global IT certifications.