Solving Virus Problems by Anti-Virus Developers-
A TRIZ Perspective

Umakant Mishra
Solving Virus Problems by Anti-Virus Developers
- A TRIZ Perspective

Umakant Mishra, Bangalore, India
http://umakantm.blogspot.in

Contents
1. Introduction to computer virus .......................................................... 1
   1.1 Illustrating virus problems faced by different people ..................... 2
   1.2 The challenges faced by an anti-virus developer ............................. 3
2. The Ideal Final Goal ............................................................................. 4
   2.1 The Ideal Final Goal of the Anti-Virus Developer ............................. 4
   2.2 Comparing IFG of end user and anti-virus developer ....................... 5
3. Problems and contradictions ............................................................... 6
   3.1 Contradiction of false positives ..................................................... 6
   3.2 Contradiction of proving virus infection ......................................... 7
   3.3 Contradiction of detecting boot sector virus .................................... 7
   3.4 Contradiction of scanning time ...................................................... 8
   3.5 Contradiction of virus scanning and system performance ............... 9
   3.6 Contradiction of prolonged scanning ............................................. 9
   3.7 Contradiction of scanning emails ................................................ 10
   3.8 Contradiction of putting anti-virus in network gateway ..................... 10
   3.9 Contradiction of behavior blocking .............................................. 10
4. Solutions adopted by anti-virus developers ....................................... 11
   4.1 Some techniques of improving an anti-virus program ..................... 11
   4.2 Some methods of controlling computer virus .................................. 12
5. Summary .............................................................................................. 12
Reference: .............................................................................................. 13

1. Introduction to computer virus

A computer virus is a manmade destructive computer program or code that is
generally loaded onto a computer system without the knowledge of a user and
causes unauthorized and unwanted changes to the components of the computer
or to the information stored on the computer. The viruses are illegitimate
programs and generally created with a destructive intention.
Computer viruses are called viruses because they share some of the traits of biological viruses. A computer virus passes from computer to computer like a biological virus passes from person to person. A computer virus increase its population, infects computers and causes damage to the computers like a biological virus does to the human beings.

1.1 Illustrating virus problems faced by different people

The first step of solving a problem is to analyze and define the problem. Many problems are solved or cease to exist when properly analyzed. When we analyze the problem of computer virus we see that the problem is not the same for different groups of people. For example, creation of virus is an undesirable phenomenon for a computer-user but is a desirable phenomenon for an anti-virus manufacturer. The same scenario of virus infection creates different problems to people at different levels, such as, a computer user, a system administrator, a corporate manager and an anti-virus developer because of their own goals and requirements.

As the problems are different for different groups of people, the solutions are also different. For example, if the user’s computer is virus infected he can use an alternative computer, but the systems administrator cannot do something similar, as his goal is to remove the virus from the computers in the network.

Besides the nature of problem may vary according to the knowledge and experience on computers and viruses. The problem faced by a novice user – who does not know how to install an anti-virus, will be different from the problems faced by an advanced user, who might have installed an anti-virus, but is facing the adverse effects of anti-virus or difficulties in removing a virus undetected by the virus scanner.
Although a large number of people use computers, their knowledge and experience on computer technology and computer viruses are different. Many computer users are experts in their own fields like graphic designing, film editing, word processing, and web designing etc. but may not be so good in dealing with computer viruses. Thus, the same scenario of virus infection may cause different problems to different users at different levels, such as, a computer user, a systems administrator, a corporate manager and an anti-virus developer because of the differences in their experience, visions, requirements and goals.

1.2 The challenges faced by an anti-virus developer

Although various groups of people dealing with computer face virus problems, each of them face different types of problems. In the previous articles we analyzed the problems faced by the end users and systems administrators. In this article will focus on the challenges faced by the anti-virus developers.

The anti-virus developers play a very significant role. They have to analyze the problems of different levels of users and find solutions for each of the problems. Their goal is not just to sell the product and make profit out of it. They undertake much greater responsibility to analyze various aspects of virus infections and remedial measures to develop their product.

The problems of anti-virus developers are different from the problems faced by the end users or the systems administrators. They are more concerned about finding fingerprints of new viruses and implementing various methods for virus prevention to improve the quality of their product. Besides they may also be concerned about sales and promotion. Some of the challenges faced by the anti-virus developer are as follows.
As numerous viruses are created on every day, it is difficult to discover and analyze every new virus, find its signature and distribute the signature database to the anti-virus users.

Different operating systems function differently. Even if the developer builds a powerful anti-virus engine the same engine may not work on different operating systems and operating environments.

The anti-virus program has to work through the framework of an Operating System. Hence, the anti-virus program can be loaded only after the OS is loaded. But there are viruses like boot-sector viruses which load before the OS is loaded and infect the OS. Once the OS is infected the anti-virus program looses its control to function in its desirable way.

There are many memory resident viruses which do not go even after repairing or removing the infected files because those viruses remain active in the main memory of the computer.

2. The Ideal Final Goal

According to TRIZ, the Ideal Final Goal is what we ultimately wish to achieve. The Ideal Final Product is no product – but only the results. The ideal anti-virus software is no anti-virus software. The ideal virus solution is “No Virus” environment without implementing any anti-virus software.

In our previous articles we saw that the IFG of an end user is a “No Virus” Environment. The systems administrator also looks forward to a “No Virus” Environment but with certain conditions. The question is whether the goal of anti-virus developer is also a “No Virus” Environment? What does an anti-virus developer ultimately want?

2.1 The Ideal Final Goal of the Anti-Virus Developer

The anti-virus developer wants a “No Virus” environments but subject to the condition that the end user uses his anti-virus product. He looks forward to have following features in his ideal anti-virus product:

- His anti-virus product should be able to detect and remove all types of viruses from the users’ computer (Note that the goal of anti-virus developer is not to destroy viruses completely out of the world as in that case he has to lose his business).

- His anti-virus product should include all (or maximum number of) virus signatures to detect all (or maximum) types of viruses. The signature database should include the signatures of new viruses as soon as they are created.
- His product should work without updating the virus database or provide easy (or automatic) methods of updating virus database.

- The product should virus scan the computer in a fast speed and consume minimum amount of system resources. The method should avoid repetitive scanning in successive sessions to save time and resources.

- His product should automatically install (or at least be least easy to install), automatically update (or at least be easy to update) and automatically survive (or at least be easy to maintain).

- His product should protect the user computer, the server, the network and all vulnerable objects from any types of virus infection.

Moving from Virus Problems to IFR

- His product should not burden the user or systems administrator for updating virus database. It should work without updating virus database or provide automatic methods of updating.

- His product should detect all the viruses with certainty and leave no scope for false negatives or false positives.

- His product should be able to remove all viruses and repair all infected files. If the system is unstable or damaged because of virus infection, the product should be able to restore the system from such disaster.

2.2 Comparing IFG of end user and anti-virus developer

Although the anti-virus developers do not really want the viruses to vanish totally out of the world, they have a great vision of their anti-virus product, which should be able to control all kinds of viruses. Let’s compare the IFG of the end users and that of the anti-virus developers.
### IFG from users’ perspective

# The computer should never get infected (no need of purchasing, installing or maintaining any anti-virus software)

# The computer and the files inside the computer should have immunity to any kind of viruses

# Ideally no anti-virus software should be required to keep the computer healthy

### IFG from developers’ perspective

# The computer should never get infected after installing their anti-virus product.

# The computers using their anti-virus product should build immunity to protect from any kind of virus under any circumstance.

# People should use their anti-virus product in order to get rid of viruses. However, their product should be fast, effective and capable of detecting all types of viruses.

When we compare the IFG of the end users and that of the anti-virus developers we find an interesting difference. While the end user wants an unconditional “Virus free” situation, the anti-virus developer targets to provide that “Virus free” situation subject to the use of his anti-virus product.

### 3. Problems and contradictions

In the above paragraphs we saw the vision of the anti-virus developer about his ideal anti-virus product. He wishes to include all such features in his product that can remove any virus from a computer and solve all virus related problems. But is it practicable to achieve all those ambitious goals?

In many situations all the Ideal final goals are not achievable to the fullest extent because of various contradicting situations. In some cases we can achieve a goal only partially because of various limitations. In some cases achievement of one desired goal may affect adversely the achievement of another desired goal, thus creating a contradiction. There are many such contradictions that the anti-virus developer faces while developing his anti-virus product. Let’s analyze some of the contradictions below.

#### 3.1 Contradiction of false positives

There are situations where the anti virus program finds a system file has been modified but is not 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 alternation then it may lead to a false negative. Both the situations are dangerous.

⇒ One solution to the above problem may be to ask the user to take a decision, but the user may not be adequately knowledgeable to give a proper decision.

### 3.2 Contradiction of proving virus infection

Generally an anti-virus program generates an alarm signal when it detects a virus like activity such as modifying a program. But in order to be sure about a viral infection the anti-virus has to wait and allow the virus to do its damages.

In order to avoid false positives an antivirus program should generate an alarm signal only when it gets sufficient proof that an operation is executed by a virus. However, by the time when an operation is detected to be done by a virus, the virus could have already made undesirable changes and damages to the program.

### 3.3 Contradiction of detecting boot sector virus

The conventional anti-virus does not work effectively for the boot sector viruses. Since the boot sector is the first item of data to be loaded in a computer system, a virus in that location is virtually without defense. The boot sector viruses are loaded before the OS is loaded and take control of the OS functions.

The conventional anti-virus programs can function only after the computer is successfully booted with an operating system. But the boot sector viruses are loaded before the operating system is loaded. Thus the anti-virus cannot work upon the boot sector virus that is already loaded prior to the anti-virus.

⇒ Solution: Exploring the possibility of loading the anti-virus program before the OS is loaded in order to ensure that the OS is never infected (Principle-13: Other way round).
3.4 Contradiction of scanning time

As the population of virus increases, the size of signature database also increases which in turn increases the scanning time. With thousands of possible virus types and hundreds of gigabytes of file storage a complete virus scan can easily take several hours which is simply not acceptable to any user.

<table>
<thead>
<tr>
<th>Contradiction</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 process will become unbearably slow. 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 scan fast.</td>
</tr>
</tbody>
</table>

⇒ One solution is to spread the scanning process into several sessions. The scanning ends as we stop it and resumes again in future from the point where it was stopped in the previous session (Principle-1: Segmentation).

⇒ Another solution is to do a partial virus scanning of the core elements like main memory, boot sector, file system etc. A partial scanning takes very less time and can be done every time the computer boots (Principle-16: Partial or Excessive Action).

⇒ Speed scanning or fast mode virus scanning- which scans the memory, system files and critical areas of the disk. The fast mode does not scan the individual files on the disk (Principle-21: Skipping).

Problem of updating virus database

Updating virus database is a boring job and we forget to do this until the virus affects us again. We want to scan for the latest viruses but we don’t want to download the latest virus signatures from the developer’s website.

⇒ Solution: Almost all modern anti-virus programs provide self-updating feature in their programs. When the computer connects to the Internet, the anti-virus program uses the standard Internet protocol to update the virus definitions from the virus database from the developer’s website (Principle-25: Self Service).

Problem of repetitive scanning

Many files are just copied to be shared by different people. When they are copied to different computers they are just scanned again and again at different computers which waste a lot of time and system resources. It would be better to avoid repetitive scanning in order to save time and system resource.
3.5 Contradiction of virus scanning and system performance

Running an anti-virus consumes system resources and thereby makes execution of other programs slow. That’s one of the main reasons why many users don’t like running anti-virus programs.

<table>
<thead>
<tr>
<th>Contradiction</th>
<th>We want to run an anti-virus to keep the computer free from virus. But we don’t want the other programs to execute 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.</th>
</tr>
</thead>
</table>

⇒ One solution is to schedule virus scanning during nights or on holidays when we are not working. The self-running feature of the program can automatically run at the pre-scheduled day and/or time (Principle-25: Self Service). However, this idea is not useful for a personal computer which is used anytime irrespective of any day of the week or any hour of the day.

⇒ Another solution is to run the anti-virus when the system is running but the user is not not working. Some anti-virus programs can be scheduled to run when the system is left idle for certain time (Principle-25: Self Service).

⇒ Another solution is to put the anti-virus on the gateway. This will protect the system by scanning all incoming and outgoing files but will not affect the system performance (Principle-2: Taking out).

3.6 Contradiction of prolonged scanning

There may be some clever viruses, which try to fool the anti-virus program and misguide it to scan wrong things for indefinite period. In such cases the anti-virus program may use excessive amount of physical or virtual memory thereby impacting the performance of the system. There is a need to time out such prolonged scanning operations.

<table>
<thead>
<tr>
<th>Contradiction</th>
<th>If a protracted scanning is not timed out, then the anti-virus system unnecessarily uses system resources and impacts system performance. But timing out a protracted scanning may inappropriately terminate a scanning operation in a slow or overloaded computer thereby compromising vulnerabilities. We want to terminate an unduly prolonged scanning but don’t want to terminate a prolonged scanning in a genuinely slow or stressed system.</th>
</tr>
</thead>
</table>
3.7 Contradiction of scanning emails

The emails in email servers like ccMail, MS Exchange, Lotus Notes etc. are stored in encrypted formats for privacy reasons. Most mail servers use proprietary encryption mechanism for email protection. The anti-virus programs cannot read these proprietary encryption formats in order to scan the attached files.

<table>
<thead>
<tr>
<th>Contradiction</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the mail servers hand over their decryption algorithm to the anti-virus program then the mail servers may not be able to maintain privacy of the mails. On the other hand, unless the decryption algorithm is known, then the anti-virus program cannot scan an encrypted mail file. We want both to scan the emails for viruses as well as to maintain privacy of the emails.</td>
</tr>
</tbody>
</table>

⇒ One option is to scan the emails at the client’s machine. But then one has to wait all the way for a virus to be sent from the sender’s computer, transmitted over the Internet, and downloaded by the recipient until it is finally opened by the client.

3.8 Contradiction of putting anti-virus in network gateway

There are two methods of administrating anti-virus software. One is to install on each individual computer and other is to install directly into the firewall. But both the methods have several disadvantages. Having a single point protection makes the scanning slow. On the other hand installing on each individual computer increases administration workload.

<table>
<thead>
<tr>
<th>Contradiction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementing anti-virus directly into the firewall prevents viruses from entering from the Internet but fails to provide protection from viruses spreading within the LAN. Implementing anti-virus on every computer within a LAN protects individual computers from virus, but increases administration work for installation and configuration of anti-virus on each computer. We need a method of administrering anti-virus so that the computers are protected from both external (Internet) and internal (within the LAN) infections without much of administration load.</td>
</tr>
</tbody>
</table>

3.9 Contradiction of behavior blocking

“Behavior blocking” is an approach of detecting virus – which runs a program on a machine that attempts to intercept malicious actions. However, the behavior blocking mechanism typically cannot view an entire log of actions in making a blocking decision. This may lead to negative consequences.
4. Solutions adopted by anti-virus developers

As we saw in the above paragraphs, there are many contradictions in developing a perfect anti-virus program. However, contradictions can be solved by applying Inventive Principles and other TRIZ methods. Many contradictions have been solved by many different ways in the past and will be solved in even more effective ways in future. We will discuss on specific problems and specific solutions in separate articles. Let’s just have a look at some of the solutions adopted by the anti-virus developers.

4.1 Some techniques of improving an anti-virus program

⇒ Many anti-virus programs implement multiple methods, such as, signature based virus detection and behavior based virus detection. Applying multiple methods help detecting a virus that is not detected by one single method.

⇒ Self test- the virus scanner first checks itself in order to ensure that it has not been infected. Similarly the programs to be executed on a computer must first undergo a self-test to determine whether the programs have been modified.

⇒ As the method of signature scanning depends on pattern recognition, better and improved methods of pattern recognition can be used for this purpose in order to speed up the scanning process.

⇒ The anti-virus should maintain the results of the previous scan so that the files that were scanned earlier to be virus-free need not be scanned again unless found altered. This method prevents repetitive scanning and saves time and system resources.

⇒ The antivirus program may create a virtual machine to run the suspect programs. If the activities of the suspect program are found to be really dangerous the program may be terminated and treated as a virus.

⇒ A good antivirus program should not simply detect all possible threats. It should also see that the false positives and unwanted detections are minimized.
4.2 Some methods of controlling computer virus

⇒ A malware is generally invited into the machine by web browsing, email downloading and using external storage like pen drive, CD, DVD etc. If we can guard all these incoming routes we can narrow down the chances of virus infection.

⇒ Users must update their anti-virus software regularly and download the codes for updating and patching the security holes. Besides the user must update the virus definition database on a regular basis.

⇒ The backup can be kept in a read only media like CD or DVD so that the backup files cannot be infected by any virus.

⇒ One should not run multiple antivirus software on the same system. As the anti-virus programs do strange activities like a virus, they are likely to suspect each other’s activities.

5. Summary

An anti-virus program typically employs a variety of strategies to detect and remove viruses. The most important methods are signature scanning, heuristic scanning and integrity checking. Various inventions have addressed the early drawbacks of each of these methods. A good solution should be holistic in approach and should not lead to any negative consequences.

The main functions of an anti-virus program may be summarized as,

- Virus prevention and file protection
- Virus scanning and detection
- Removing virus from infected files and objects
- Recovering damaged files and objects

This article focuses on the goals of an anti-virus developer and illustrates some contradictions faced while achieving the goals. Some contradictions have been resolved by various inventions but many more are still there to be resolved. The important issues on improving the efficiency of anti virus programs are as below.

- Limitations of signature scanning, heuristic scanning and other methods of virus detection and improvements thereon,
- Using behavior analysis as a method of detecting viruses,
- Improving techniques of on-access scanning,
- Managing size and distribution of virus definition files,
- Improving speed of virus scanning,
- Improving detection and removal of boot sector viruses, macro viruses, email viruses etc.
• Implementing virus scanning in networks, servers, gateways and firewalls,
• Implementing remote scanning and web based scanning,
• Improving on-access scanning and ‘on the fly’ scanning,
• Improving methods of generic detection of computer viruses,
• Implementing antivirus in internet and websites,
• Methods of repairing and recovering virus infected files,
• Trends of evolution in improving an anti-virus mechanism etc.

The methods of solving each of these problems and the inventions that have been already made will be discussed in separate articles. It is advised to read this article along with other articles in the series for better understanding.

Reference:


About the author

After working for more than 18 years in various fields of Information Technology Umakant is currently doing 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.