Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2014) 000–000

www.elsevier.com/locate/procedia

TRIZ FUTURE CONFERENCE 2014 - Global Innovation Convention

Method for Transferring the 40 Inventive Principles to Information Technology and Software

Hartmut Beckmann[a]*

[a]*Giesecke & Devrient GmbH, Prinzregentenstrasse 159, 81677 Munich, Germany*

* Corresponding author.

Abstract

The 40 Inventive Principles are one of the best known and most used tools of TRIZ. Originally, the 40 Inventive Principles were focused on solving problems of physics and chemistry. Caused by the increasing impact of software solutions, there were some activities in searching for analogies of the 40 Principles in software environments since the year 2000. Unfortunately, these efforts only had limited success. TRIZ and software is a difficult topic until today.

This paper takes a look at past searches for software analogies of the 40 Principles. As a result, the creation of analogies is regarded as very useful. In the past, the analogies were limited to one-to-one transfers and new examples for the existing 40 Principles. But information technology is very different from physics and chemistry. This causes lateral thinking to be necessary when applying the current analogies of the Inventive Principles, thus often reducing quality and number of the ideas found. For avoiding lateral thinking, the transfer of the Inventive Principles to information technology has to be done in a more flexible way.

For achieving this objective, information technology is separated into its three characteristics: objects, data, and algorithms. In the first step, each original Principle is applied to each of these three characteristics. Then, all results are put together. In the second step, the found results are put back into groups based on the original Principles. Thereby, groups are created, dropped, modified, or split. Thus, the new groups can be very different in comparison to the original Inventive Principles. Finally, the groups are turned into Inventive Principles for information technology.

Peer-review under responsibility of The International Scientific Committee ETRIA.

Keywords: TRIZ, Inventive Principles, Information Technology, Computer Science

1. Introduction

TRIZ without the 40 Inventive Principles would not be TRIZ. Altshuller derived these Principles by analyzing 40.000 patent abstracts in the fields of electrical engineering, acoustics, optics, mechanics et cetera as well as chemistry [1]. Each Inventive Principle provides an abstract solution model. When applied to a problem, a solution model forces human mind to think in unusual ways, thereby creating uncommon and innovative ideas. This is called "overcoming the psychological inertia" [2].

In 1973, when Altshuller published the 40 Inventive Principles [3], information technology just began to evolve. Thus, the very most patent abstracts analyzed did not tackle problems of information technology, so the solution models provided by the Inventive Principles were not designed for handling information technology problems.

But this circumstance is no drawback. As shown in this document, information technology is very different from physics and chemistry. Thus, including information technology in the original 40 Inventive Principles could have caused a mix of different worlds becoming the Inventive Principles to be less useful.

Nevertheless, due to the dramatic emergence of the Internet in the last two decades, there is a huge demand also to apply the Inventive Principles to information technology and software. This document describes a method how this transfer can be done and gives examples of the results created thereby.

2212-8271
Peer-review under responsibility of The International Scientific Committee ETRIA.

2. Previous "TRIZ and software" publications

By the end of 1999, ARIZ was used to create solutions for a software concurrency problem [4]. Two years later, TRIZ helped enhancing the usability of a software GUI [5]. Also, Su-Field Analysis was applied to software problems [6]. Much later, in 2013, there was a proposal of how to use Function Analysis on software and information technology [7] without changing any single rule of Function Analysis.

There are many publications which discuss the application of the 40 Inventive Principles [8] to software problems. In April 2001, the contradiction between energy consumption and performance of a processor was examined using the 40 Principles [9]. In this article, the author describes that "many of the 40 Principles do not directly map onto our software system as this system has virtual components as well as physical". Because of this, there is a "(...) need to do some lateral thinking about what this (Principle) might mean (...)".

Also in 2001, software analogies were built for most of the Inventive Principles [10, 11]. These analogies refer to internal software design principles. For example, the analogy for Principle #26 "Copying" is to perform a shallow copy on data structures instead of a deep copy, thus reducing memory usage and computing time. In August 2004, [12] added analogies for some missing Principles. In January 2006, [13] provided a complete list of Principle analogies based on [10, 11]. Some analogies given are very detailed. For example, the analogy for Principle #5 "Consolidation" suggests to run processes in parallel. Other analogies given are very generic. For example, the analogy for Principle #6 "Universality", suggests "Make a technical system support multiple and dynamic classifications based on context".

In October 2004, there was an article discussing adaption of the Technical Parameters, Contradiction Matrix and the 40 Inventive Principles to software [14]. The results were published in [15]. Regarding the Inventive Principles, for each Principle many software examples are given, the Principles itself remained almost unchanged.

3. Transfer process of the previous publications

Regarding the previous transfers of the 40 Inventive Principles, the method used is shown in figure 1.

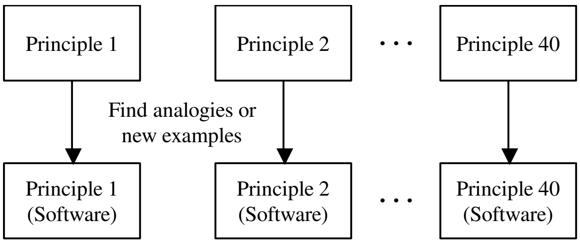

Fig. 1. Earlier transfer methods used on the 40 Inventive Principles.

Basically, by building analogies, each principle was transferred one-to-one to software. A major drawback of this approach is that it relies on at least one implicit assumption. This assumption is that each original Principle exactly has one equivalent Principle in information technology. When looking at Principle #8 "Anti-weight" or Principle #18 "Mechanical vibration", it could be possible that sometimes there is no equivalent existing in information technology. On the other hand, perhaps one original Principle may result in two or more Principles for information technology. Thus, it has to be avoided to rely upon a possibly wrong assumption, so another kind of transfer is necessary.

Caused by the assumption described above, an additional drawback occurred. The one-to-one transfer generated usable results but as mentioned in [9], lateral thinking is still necessary when using most of these transferred Principles. This lateral thinking prevents the user from focusing on applying the solution model to the problem and increases psychological inertia causing suboptimal results. For better results, the Inventive Principles have to be transferred in a way so the necessary lateral thinking is reduced to a minimum.

4. Information technology is very different from physics

When looking at the details, information technology is very different from physics, although it is realized by physics. Figure 2 shows the abstraction between information technology and physics.

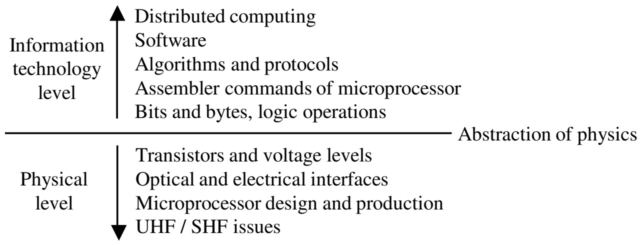

Fig. 2. Abstraction between information technology and physics.

In terms of this document, information technology is anything of bits and bytes and above. Thus, the underlying physics is completely hidden in most cases and of very little interest when setting up an information technology system or creating software. The abstraction of physics is the fact which makes information technology so powerful: Everyone can use it without any knowledge of physics. It is much easier to create and implement software than to create and produce the processor executing it.

Furthermore, information technology was created by humans. This means, the exact rules of information technology are known and relatively simple. In contrary, physics was built by nature, so many of the exact rules, relations between rules, and effects are not known to mankind yet, because everything is very complex.

Altshuller created the 40 Inventive Principles by analyzing patent abstracts in the field of physics and chemistry. Information technology is much simpler than physics and chemistry, so it can be expected that the Inventive Principles for information technology differ from the original Inventive Principles. As a result, instead of a one-to-one transfer of the Inventive Principles (see figure 1), a more flexible transfer method should be applied. Furthermore, the transfer results have to be as close as possible to information technology to minimize lateral thinking.

5. The characteristics of information technology

When doing a transfer of the Inventive Principles to information technology, it is important to focus on the characteristics of information technology for achieving best results. Basically, the characteristics of physical or chemical systems are:

- The objects the system consists of. In physics, also "fields" may be regarded as some kind of objects.
- The characteristics and the condition of each object.
- The interactions between the objects of the system. In chemistry, these interactions are often part of processes.

By modifying one or more of these characteristics, the system is also modified. This is what the Inventive Principles do. They provide solution models of how to modify the system for solving problems. Thus, it is important that the Inventive Principles for information technology also apply to the characteristics of an information technology system. In table 1, the analogies for these characteristics between physics and chemistry and information technology are shown.

Table 1. Analogies of system characteristics.

Physics and chemistry	Information technology
Any objects and fields	Objects – any equipment able to provide or process binary data
Characteristics and status of an object or field	Data – any information available in binary format
Interactions between objects and processes inside the system	Algorithms – any sequence of well-defined steps performed by objects using data

The first characteristic of an information technology system are the objects the system consists of. In contrary to physics and chemistry, the objects of an information technology system are required to handle some binary data for being a valid object. The reason is that an object which does not at least provide any binary data is not "visible" inside an information technology system. Therefore it has no influence on the information technology system. So, if the object would not provide any binary data, it does not make sense to modify it using Inventive Principles because a modification would have no effect inside the information technology system.

As an example, a normal coffee machine is not part of an information technology system because it is not "visible" to other computers. The same coffee machine comprising a WiFi module for controlling it from somewhere else however is part of an information technology system.

The second characteristic of an information technology system is the data. For being processed by an information technology system, data is required to be available in binary format. This definition means that data also comprises the binary program code used for defining an algorithm.

The third characteristic of an information technology system are the algorithms. Algorithms are sequences of well-defined steps to create a result. It can be argued that in information technology, the data already defines the algorithms, because data also represents program code. This argument is right in principle. But Inventive Principles are thought to overcome psychological inertia, so handling the characteristic "algorithms" in a special way makes much sense. It helps the user to distinguish between bits and bytes level (data) and process level (algorithm) in an easy way.

As a result, the characteristics of information technology are defined as shown in table 1. For best usability, the Inventive Principles for information technology have to provide solution models for each of these three characteristics.

6. Applying the Inventive Principles to the characteristics of information technology

In this step each of the original Inventive Principles is applied to objects, data, and algorithms according to the definitions shown in table 1. Then, all results found are put together. The process is shown in figure 3.

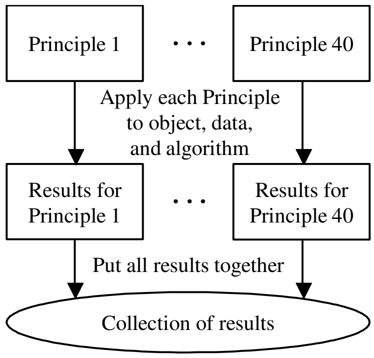

Fig. 3. Applying Inventive Principles to objects, data, and algorithms.

The original 40 Inventive Principles provide abstract solution models. The characteristics objects, data, and algorithms applied to these Principles are also abstract. Therefore, many of the results generated are tiny abstract solution models. Table 2 and table 3 show results of applying Principle #14 "Spheroidality – Curvature" to data and algorithms.

Table 2. Results of applying Inventive Principle #14 to data.

Result	Target group
Encode data non-linear, e.g. logarithmic.	#16
Introduce random access to data.	#15
Change order of data.	#15
Work using approximated data.	#16
Encode extreme values using special codes.	#14
Turn linear parameters to non-linear.	#14

Table 3. Results of applying Inventive Principle #14 to algorithms.

Result	Target group
Transform a linear algorithm into a non-linear one.	#12
Split an algorithm in its special cases. Handle each one separately.	#1
Execute parts of the algorithm at different speed.	#21
Introduce branches into algorithms. Further actions become different.	#1
Just process parts of the data and in random order.	#15

When processing a list of data, increase the indices in a non-linear manner.	#14
Segment an algorithm into its linear parts.	#1
Change parts of the algorithm each time executed.	#23
Use logarithmic output of data.	#14
Switch communication channels.	#28

Originally, Principle #14 just suggests using “non-linear” shapes and motions. But as shown in table 2 and table 3, this “non-linearity” triggers many ideas for data and algorithms. According to the process shown in figure 3, it is not important whether the idea matches the Principle it was created from. All ideas are put together anyway, so it is impossible to create “wrong” ideas. Every idea is useful.

The result of this first step is a huge collection of possible tiny solution models. Many of these tiny solution models share the same basic concept, so it makes sense to put them together to groups again. This procedure eliminates duplicates and simplifies usage.

7. Grouping the results of Inventive Principle application

Creating meaningful groups from some hundred results is very difficult and time-consuming. For simplifying this task, the original 40 Inventive Principles are used again. This time, the original Principles are used as proposals for the groups the results are put into. This is shown in figure 4.

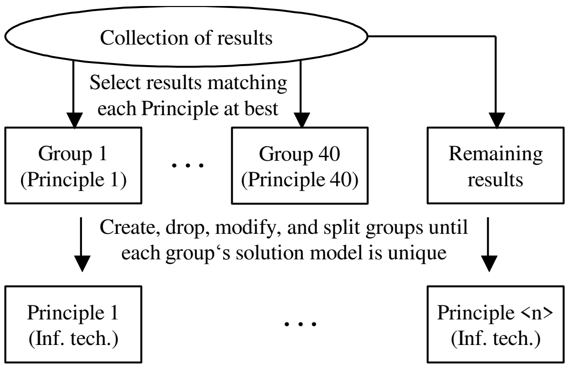

Fig. 4. Creating the Inventive Principles of information technology based on the collection of results.

When assigning the results to the preliminary groups, some difficulties are to be solved:

- If a group gets no or only less interesting results assigned to: Drop the group and its solution model.
- If a group gets too many interesting results assigned: Split the group and create separate solution models.
- If there is more than one group the result can be assigned to: Separate the solution models of the conflicting groups, so the result can be assigned to one group. If separation is not possible, drop one group.
- If there is no group matching the result: Put the results to “remaining results” and search new solution models for these results afterwards.

By using this process, number and meaning of the generated Principles are not bound to the original ones. Nevertheless, this process benefits from the experience provided by the original 40 Principles. Thus, the process is more flexible than just building analogies for every single Principle while keeping the experience at the same time.

An example for dropping a group is Principle #14 “Spheroidality – Curvature”. Looking back in table 2 and table 3, the second row shows the number of the group each result was finally assigned to. In the end, group #14 had not enough useful results assigned to, so it was dropped for information technology.

In some cases, it was not possible to specify a single group matching a result at best. Nevertheless, assigning a result to more than one group should be avoided because the new groups and therefore Principles would become redundant and therefore inefficient. Table 4 gives some examples for these conflicting groups in terms of information technology.

Table 4. Examples of conflicting groups in terms of information technology.

Group	Conflicting group
#16 Partial or excessive actions	#2 Taking out
#18 Mechanical vibration	#19 Periodic action
#25 Self-service	#6 Universality
#26 Copying	#2 Taking out
#27 Cheap short-living objects	#34 Discarding and recovering
#38 Strong oxidants	#16 Partial or excessive actions
#38 Strong oxidants	#22 Blessing in disguise
#39 Inert atmosphere	#34 Discarding and recovering

In short, group #26 “Copying” and #2 “Taking out” conflicted in the fact that #26 suggests to use a simple and inexpensive copy. But this modification is part of #2 “Taking out”. Thus, the transferred Principle #26 for information technology just suggests to “introduce many clones” without any modification. Also, group #38 “Strong oxidants” conflicted with #16 “Partial or excessive actions” and #22 “Blessing in disguise”. The reason is that in terms of information technology, #38 suggests using drastic actions. This is an excessive action as already suggested in #16. If the drastic actions of #38 are harmful and thought to cause useful effects, this is identical to #22. As a result, #38 was dropped.

Finally, all conflicts were solved and all results were assigned to the remaining and modified groups as well as “remaining results”. Table 5 and table 6 give examples for the results assigned to group #15 “Dynamics”. The second row indicates the original Principle the results were generated from.

Table 5. Results for data matching group #15 “Dynamics”.

Result	Source principle
Introduce random access to data.	#14
Change order of data.	#14
Expand and shrink data as necessary.	#15
Make constants variable.	#15
Use data structures easily changeable.	#27
Change size of datagrams.	#35

Now in each group less inventive results are dropped. The remaining results are simplified and formulated in a more generic way, so they are as intuitive as possible to use. Afterwards, the group becomes the respective Inventive Principle of information technology.

Table 6. Results for algorithms matching group #15 "Dynamics".

Result	Source principle
Just process parts of the data and in random order.	#14
Extract implicit data from algorithms.	#15
Change algorithm's resources during runtime.	#15
Make parts of the system replaceable.	#24
Introduce loose coupling between algorithms.	#28
Use self-modifying algorithms.	#34

In practice, 30 Inventive Principles for information technology were created. For more intuitive usage, the groups were neither renumbered nor renamed, so the correlation between each original Principle and the Principle of information technology is easy to recognize.

Table 7 gives an example of the final description of the Inventive Principle of information technology #15 "Dynamics".

Table 7. Inventive Principle of information technology #15 "Dynamics".

Identifier:	Principle #15 "Dynamics".
Abstract:	Let things change.
Objects:	Enable objects for adding, modifying or removing functions while in use.
	Enable the system to add or remove objects at any time.
Data:	Make static data dynamic.
	Make implicit data explicit, then dynamic.
	Make static data parameters dynamic.
Algorithms:	Enable an algorithm to execute steps in a random order.
	Fit an algorithm to resources changing during runtime.
Example:	In Internet television, the video stream bandwidth is dynamically adapted to the current bandwidth of the user device. By doing so the video quality may get worse but the video does not stop.

The transformation of the Inventive Principles to information technology ended up with some results not matching a single Principle yet. These results were collected in "remaining results" (figure 4). The most interesting result was "Exchange data by algorithms and vice versa" generated from Principle #36 "Phase transitions". In computer science, there is at least one application of this method: white-box cryptography.

Nevertheless, the question whether there are "new" Principles is very difficult to answer. During the steps shown before, some abstracts of the original Inventive Principles were modified considerably. Examples are given in table 8. Are these "new" Principles or not? For daily work, this is not important. The main thing is that the Inventive Principles for information technology help to solve problems and improving systems. According to daily usage of the found Inventive Principles for information technology, this goal was achieved.

Table 8. Examples of Inventive Principles becoming modified considerably when applied to information technology.

Original Inventive Principle	Description as Inventive Principle of information technology
#28 Mechanics substitution	Change the connection.
#29 Pneumatics and hydraulics	Make things fuzzy.
#34 Discarding and recovering	Do it quick and dirty.
#36 Phase transitions	Analyze the changes.

8. Summary

In this text, a heuristic method for transferring the Inventive Principles to information technology was presented. The method uses the power of the original 40 Principles in both steps. In the first step, the original Principles are applied to the characteristics of information technology, objects, data, and algorithms generating results. In the second step, the original Principles provide groups as a starting point for the new Principles. When matching the generated results to the groups, groups can be created, dropped, modified or split. Thus, the final groups can differ from the original Principles considerably. This means, this transfer method is able to generate much more flexible new Principles in comparison to the one-to-one transfers done in the past.

For checking the method, a complete transfer of the 40 Inventive Principles to information technology was performed. Thereby, 30 Inventive Principles for information technology were created. In daily work of the author, these Principles became an essential tool for improving information technology and software systems.

References

[1] Altshuller, G., 40 Principles: TRIZ Keys to Technical Innovation, Technical Innovation Center, 1997.
[2] Altshuller, G., Creativity as an Exact Science, CRC Press, 1984, p. 10-15.
[3] Altshuller, G., The Innovation Algorithm, Technical Innovation Center, 2007.
[4] Rea, K., Using TRIZ in Computer Science - Concurrency, TRIZ Journal, August 1999.
[5] Schlueter, M., TRIZ for Perl-Programming, TRIZ Journal, May 2001.
[6] Rea, K., Applying TRIZ to Software Problems - Creatively Bridging Academia and Practice in Computing, TRIZ Journal, October 2002.
[7] Beckmann, H., Function Analysis integrating Software Applications, TRIZ Future Conference, October 2013.
[8] Tate, K. and Domb, E., 40 Inventive Principles With Examples, TRIZ Journal, July 1997.
[9] Rawlinson, G., TRIZ and Software, TRIZ Journal, April 2001.
[10] Rea, K. TRIZ and Software - 40 Principle Analogies, Part 1, TRIZ Journal, September 2001.
[11] Rea, K. TRIZ and Software - 40 Principle Analogies, Part 2, TRIZ Journal, November 2001.
[12] Fulbright, R., TRIZ and Software Fini, TRIZ Journal, August 2004.
[13] Tillaart, R., TRIZ and Software - 40 Principle Analogies, a sequel., TRIZ Journal, January 2006.
[14] Mann, D., TRIZ For Software?, TRIZ Journal, October 2004.
[15] Mann, D., Systematic (Software) Innovation, IFR Press, ISBN 978-1-906769-01-7, 2008.