

Hands-on Logic for Inventive Problem Solving – I. Formulating Technical Contradiction from Initial Situation

Mi Jeong Song Ph.D.; V. Leniachine, V.; Sung Cheol Kim; S. Antonov, Doctor

Six Sigma Innovation Team, Samsung Advanced Institute of Technology

San 14-1, Nongseo-ri, Giheung-eup, Yongin-si, Gyeonggi-do, 449-712, Korea

Abstract

A method to formulate technical contradiction from initial situation, as a starting point of inventive problem solving, has been introduced. This logic has been developed as one part of DFSS roadmap and designed to apply even in very ambiguous situation. Starting from clarifying the passive knowledge about what we have now, this logic helps us to describe indispensable information to formulate sets of technical contradictions, so called induced technical contradiction in real technical system. Transition action to eliminate single undesirable action is the key to formulate induced technical contradiction. This logic transforms usual information to definite problem models, sets of technical contradictions, which includes physical contradiction in it. This logic has shown excellent performance to identify initial situation and problem models for SAIT initiation project.

Introduction

We started our series of publications from the article, which contain two initial steps of suggested methodology for Technical Contradiction formulation – namely, “Needs analysis” and “Benchmarking Prototype systems” (9). We consider this article as the continuation of first one, but for readers who have known well about the subject of first part from another sources of information (for instance, DFSS, etc) this article can have an independent sense.

The achievements of many authors (1-8) help us to formulate hands-on logic of inventive problem solving presented in our previous work. (9)

After preparation of step 1 and step 2, we can select appropriate object for analysis, analysis criteria and moreover right analysis scope (9). The goal of next stage is to select appropriate problem model(s) in very complicated real technical system.

If we have very simple system, technical contradiction and physical contradiction is obvious to pick up. Unfortunately modern system is not so easy to pick up appropriate problems models at once because it is complicated enough. That's why we should

develop specific methodology to select problem models from technical system as following.

- Step 1. Needs analysis: described already (9)
- Step 2. Benchmarking Prototype systems: described already (9)
- Step 3. Formulating prototype system model
- Step 4. Determining cause/effect of undesirable function(s)
- Step 5. Choosing Transition Action to fix up 'single undesirable function'
- Step 6. Formulating effects of transition actions to system/supersystem
- Step 7. Formulating induced technical contradictions
- Step 8. Evaluating importance of induced technical contradiction

Figure 1 summarizes whole scheme of inducing technical contradiction from initial situation. After formulating induced Technical Contradiction, we can go ahead to formulate Physical Contradiction, then to apply typical methods of Contradictions resolving – these processes are shown on Fig.1 as the separated block – “Concept generation phase”. To disclose the detail of this block is a task, which exceeds the scope of this article and will be discussed next series of our articles.

This article will present detailed description from step 3 to step 8. This logic follows the big logic stream of DFSS described in previous article (9).

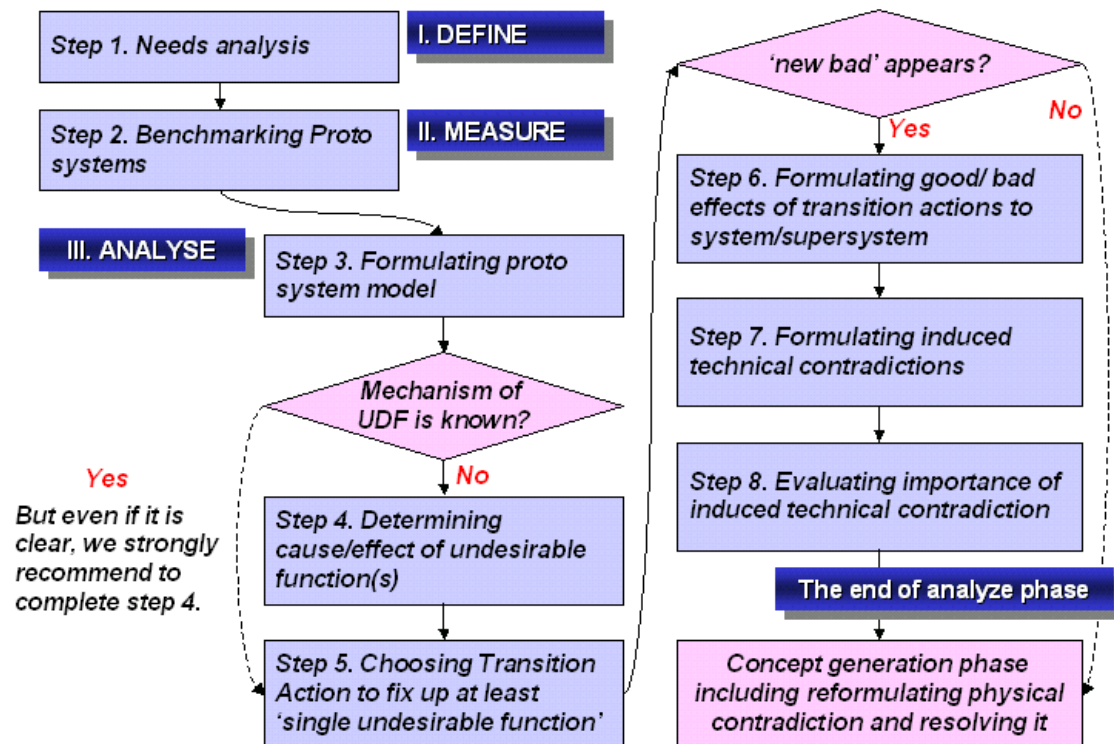


Figure 1. Whole scheme of formulating induced Technical Contradiction

Detailed description of formulating problem models

Step 3. Formulate system model

This step progresses following 3 stages:

Step 3.1. Symbolizing prototype system

At the first stage, we have to symbolize (i.e. make a model) for chosen prototype system for further analysis.

Step 3.2. Analyzing the value of function/elements

We must evaluate the value of each function/elements and generate valuable problem lists from the result of function analysis

Step 3.3. Constructing list of elements in system

At this stage we have to make a list of elements in the system, supersystem and moreover, to know the reason why we use such specific elements in the prototype system. A reason why any element exists in the system is defined as 'Function'. If any element of a Technical System with no function then such elements has no reason to exist in the Technical System.

According to the general scheme of TRIZ problem solving logic, at the first time we formulate problem models from initial situation. But we suggest rewriting it as following:

Initial situation -> initial situation description -> problem model -> partial concept solution -> whole concept solution -> real solution

Almost all activities of classical TRIZ were devoted to clarify the process from problem model to partial concept solution. The scope of this article is to clarify the hands-on process from initial situation to initial situation description and from initial situation description to problem model.

According to our vision, at the very beginning of problem solving, we should formulate **situation description** from **real situation based on step 1 and 2**. Only after the first situation description is defined, we can formulate the problem model.

Knowledge exists just in our brains. Nobody could recognize it before it is described in appropriate language and/or figure. Real problem solving starts from the situation that the knowledge is submerged in the brain. It is necessary to stress how it is important to

symbolize this ‘hidden’ knowledge to ‘visual’ i.e., ‘explicit’ knowledge.

As a method of this 1st order transformation from initial situation to initial situation description, we could use well-developed methods such as:

- Drawing ‘figure’
- ‘Product analysis module’ (TechOptimizerTM) i.e. Function analysis
- ‘Process analysis module’ (TechOptimizerTM) i.e. Process flow chart, with function analysis of major steps

We have two different kinds of information. One is future CTQ from needs analysis (step 1). The other is main prototype system from the Benchmark analysis (step 2). *From the viewpoint of future customers we can evaluate the details of current prototype systems.*

To satisfy future customers we have to define ‘UDF (undesirable function)’ according to previous analysis of model and element’s functions. Many authors (2,3,6,10,15 and more) have used function analysis because it is very useful to identify the elements of prototype system/supersystem and their functions. We can further clarify the ‘UDF (undesirable function)’ from the function analysis model. Every function can be classified as following:

1. Useful function
2. Insufficient function
3. Useless function
4. Harmful function

In common, among above 4 different kinds of function, functions 2~4 are UDF for customers (Fig. 2). It is necessary to find out a way to eliminate these UDF’s. Note, that in case of process we can apply the similar approach.

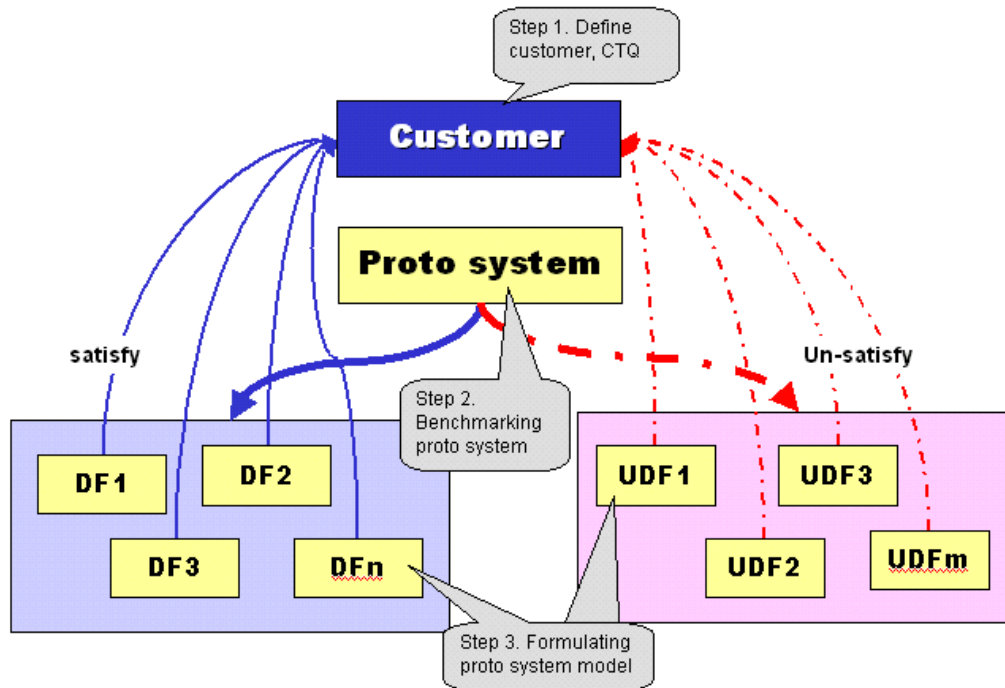


Figure 2. The scheme of formulating system model

TOC recommends drawing “Undesirable Effects” (UDE) tree as initial description to make “Current Reality Tree” (CRT) (16). This kind of description would be good enough for initial situation description in usual case, but to describe “Technical System”, the function-based description seems to be more appropriate than CRT or Root Causes Analysis (RCA).

Before describing problem situation, the technical system should be analyzed in detail because problem situation is different in every prototype systems. Step 3 corresponds also to the process from initial situation to initial situation description.

Novel solution space locates out of existing prototype system (“unknown region”). That’s why we should describe current prototype system and problem so definitely.

Step 3 helps us to identify what we’ve had, including prototype system elements with their properties. We strongly recommend making a list about the properties of elements that we’ve had (Fig. 3).

These elements with their properties will be one of most important resources when we try to make a concept solution. Really, description about current used properties of elements, which take part in carrying out of DF/UDF, is enough at this stage to define ‘known region’ more definitely. But idle (“free”) properties of the same elements could be very helpful in future to change ‘unknown region’ to ‘region of solution’. The element description strategy is very important issue of real problem solving, but

exceeds the scope of this article– it will be discussed later as another form of articles.

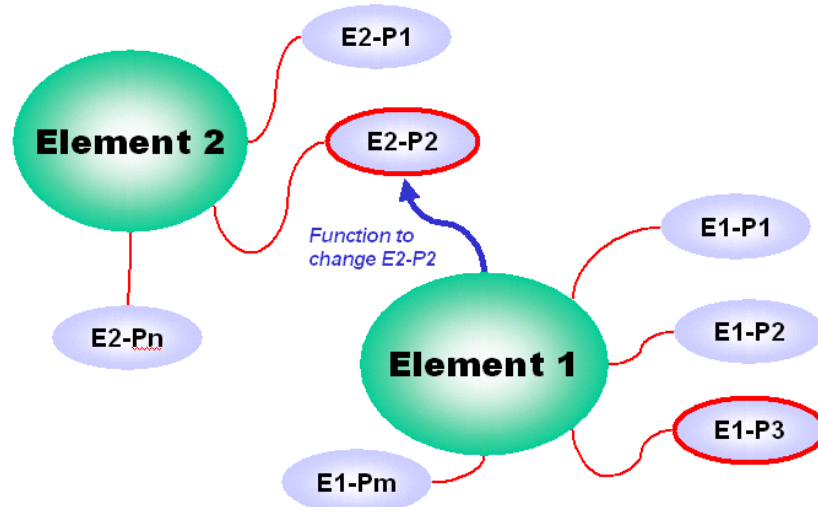


Figure 3 Description of Element in the system/supersystem

After formulating function model and element description, their importance could be evaluated item by item, which will not be discussed further now.

Step 4. Determining cause/effect of undesirable function(s)

Root Cause Analysis (RCA)(8) is well-known methodology, which is proven useful to clarify an ambiguous situation. If there are undesirable functions, there exist reasons why such undesirable functions appear and what effects could be expected from the results of such undesirable functions. Root cause analysis is easy to learn as well as easy to do (comparing to table based methodology, but really, doing RCA well enough is not an easy task). Function analysis could include very restricted elements in the system/supersystem, but root cause analysis could include almost every elements, effects, and features in/out of the system, which exists along the logic link of the problem (Figure 4).

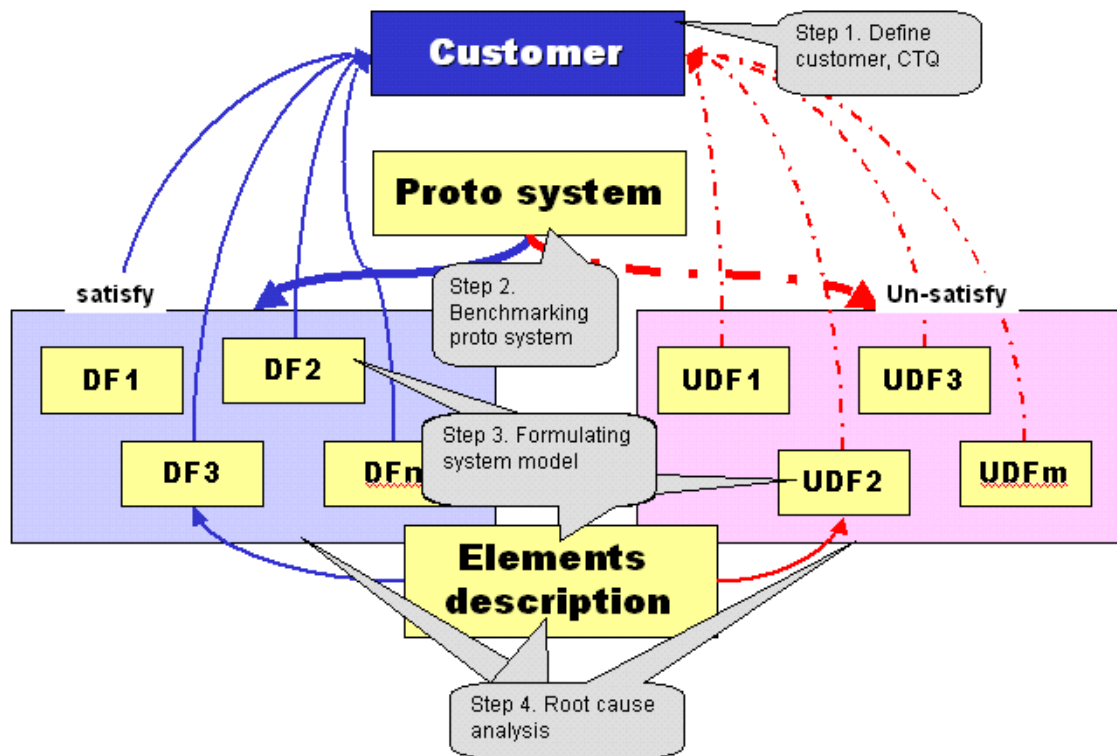


Figure 4. The scheme of formulating system model after Step 4

It is necessary to mention the reasons why we try clarifying the causes of undesirable functions. Why do we elucidate the reason of undesirable function so enthusiastically? In real modern systems there are very complicated cause and effect networks of undesirable functions. If we don't know the cause and effect network, we can't suggest any probable concept to 'fix up' undesirable function. The concepts should correspond to each node of cause and effect network. Cause and effect of problem could be considered as one of objective law we should obey.

Rule of thumb: Description of function, element, cause and effect is the most important activity to change 'hidden' knowledge to 'explicit' knowledge to elucidate 'known region'.

Step 4 is closing stage of initial situation description (we named it as "a passive stage"), after which we would try to resolve problems or formulate problem model more definitely at the same time. Really, till now we only did some "observations" (e.g., wish-list of customers, studied system, collected information/knowledge and so on). And next step starts "active stage" – according to former observation we have to launch "active actions".

Step 5. Choosing Transition Action to fix up 'single undesirable function'

If researchers follow the step 1~4, they have two kinds of knowledge sets. One is system information about elements and their properties, and undesirable function that is logically linked to the CTQ. The other is cause and effect information. Only after these two sets of information are ready, we can suggest appropriate action to eliminate undesirable functions in the system.

In the usual case, almost all researchers know the possible action to eliminate at least 'single undesirable function'. And only after researchers suggest the possible action to eliminate at least 'single undesirable function', we can define technical contradiction.

The following discussion will clarify this situation. The following is typical discussion between **TRIZ tortoise** (some researcher calls TRIZ consultant as "**tortoise**" because TRIZ thinking is so definite and slow comparing to other methods, for example, brainstorming, etc.) and researchers whose name is "**Achilles**" (because every researchers hurry to resolve their problems at once like Achilles). (13)

TRIZ tortoise: "Do you know the method to eliminate this undesirable function B in the system? I know this undesirable function is a critical obstacle for you to meet the CTQ."

Achilles: "Yes, I know. If we 'try action A' we can eliminate absolutely this undesirable function B."

TRIZ tortoise: "Ok. It's good news. Please 'try action A' to eliminate undesirable function B."

Achilles: "I can't."(More often—"It is impossible!")

TRIZ tortoise: "Why? You know where you have undesirable function B and you know why undesirable function B appears and you know the method to eliminate undesirable function B. You know almost everything. Why can't you 'try action A'?"

Achilles: "Because if I 'try action A', undesirable function B may disappear but another 'new bad C' appears in the system and supersystem. It is too complicated to avoid. That's why we can't 'try action A'."

TRIZ tortoise: "Yes. It is your contradiction. Right?"

Achilles: "Right!?"

TRIZ tortoise: "Congratulation! Now we succeed to formulate one of technical contradiction. Let's do our best to resolve it by the help of classical TRIZ tool sets!"

Achilles: "Oh my goodness. You're crazy as well as slow. What technical contradiction did we formulate? Let me see the technical contradiction you did. You

just messed up everything.”

We suppose every TRIZ consultant is frequently faced with situations like this. But very few people recognize what important information is buried in above discussion. A ‘real key of contradiction’ is hidden between the lines in above discussion.

We should concentrate on the fact that “before the researcher *Achilles* suggested to “try action A”, the Technical Contradiction was absent. If we don’t “try action A”, “undesirable function B” still exists without any change and “new bad C” doesn’t exist and will not appear itself in any way. Only after we “try action A”, Technical Contradiction starts existing. This definitely corresponds to the definition of Technical Contradiction according to “classical TRIZ” (1,17).

When we deal with real problems in existing system, root cause analysis is very powerful tool to elucidate the presence of Technical Contradiction. But in case of initial situation of project, there exists **no Technical Contradiction** in the world before we analyze prototype system and **choose an action to eliminate undesirable function** of the prototype system. From now on, we will call **‘try action A’** as one of **Transition Action**. If we didn’t do any action to eliminate undesirable function then we have not Technical Contradiction but just discrepancy (dissatisfaction) between what we’ve had and what we would like to have. (Note, that very often the discrepancy between wishes and abilities someone introduces as “Technical Contradiction”. We agree, maybe such situation could be named as “Contradiction” in everyday meaning, but never “Technical” in TRIZ terminology).

We **induced**, i.e. created, Technical Contradiction by introducing any **Transition Action** (‘try action A’). You should concentrate on the results of above discussion – it is very important for understanding of Technical Contradiction’s essence.

After needs analysis (step 1), the prototype system will be chosen for deep function analysis (step 2). As a result of function analysis, undesirable function (UDF) can be identified according to the needs of our future customers (step 3). We identified the cause and effect of undesirable function using root cause analysis (step 4).

To eliminate UDF’s of technical system, we can try ‘transition action’ to the selected proto system. If new bad functions/effects appear owing to the transition action, the transition action generates good and bad effect at the same time, which will be one side of technical contradiction.

To do a transition action, we can select from 4 different well-known directions.

1. Ideal ways (3):

2. 76 Standard solutions (17)
3. Parameter change of element according to STC operator (1)
4. Well-known solution derived by other methods (e.g. technology tree, patent map, articles, etc.)

These 4 kinds of directions can provide us right direction to resolve ‘at least’ one UDF. From the single UDF, really, we can identify lots of valuable technical contradictions, which will leads us to the innovative solutions. But what direction of Transition action should be done? The choice of the direction may be done according to all previous analysis (needs, prototype system, Law of Technical System evolution and so on) for every particular case, of course, with taking into consideration of all restriction of real problem. In any case, every Transition Action will cause good/bad features simultaneously, but we will choose the one, which will corresponds to the needs of our customers more closely and common trends in technical system evolution.

To do this choice more reliably, a few persons could do it independently. If we found the ‘transition action’ with minimal/admissible level of bad effect, we can introduce this ‘transition action’ directly into existing system as a concept of solution (Figure 5). But if transition action causes a critical bad effect, it means transition action contains the real “tool” of contradiction model in it(Figure 6).

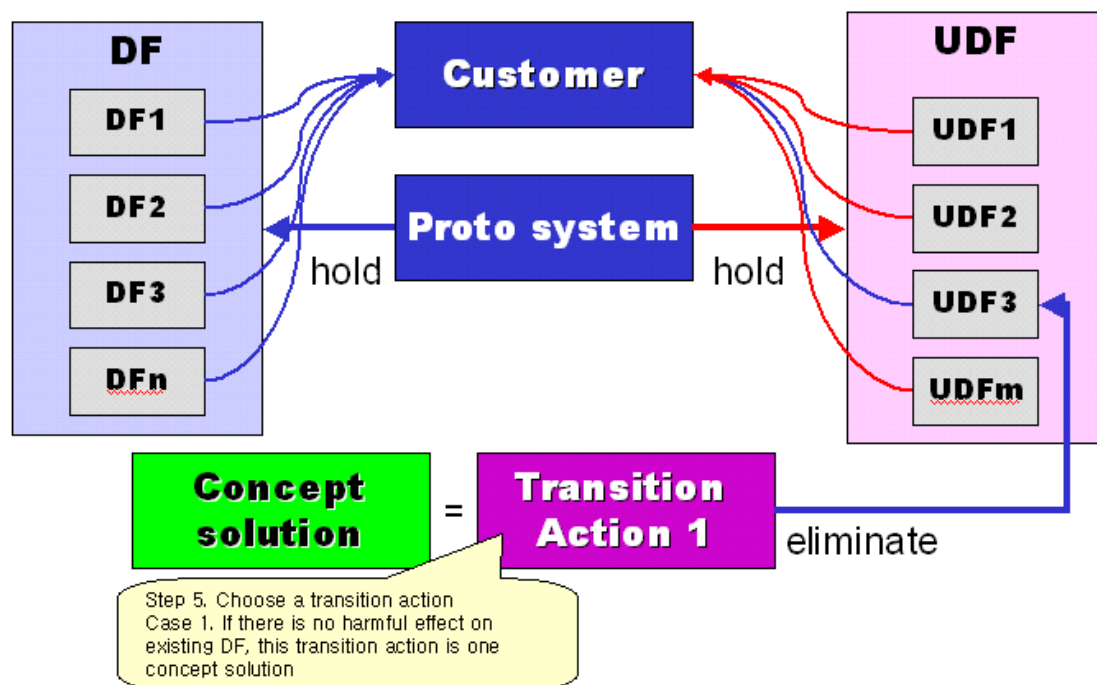


Figure 5. Transition action as concept solution itself

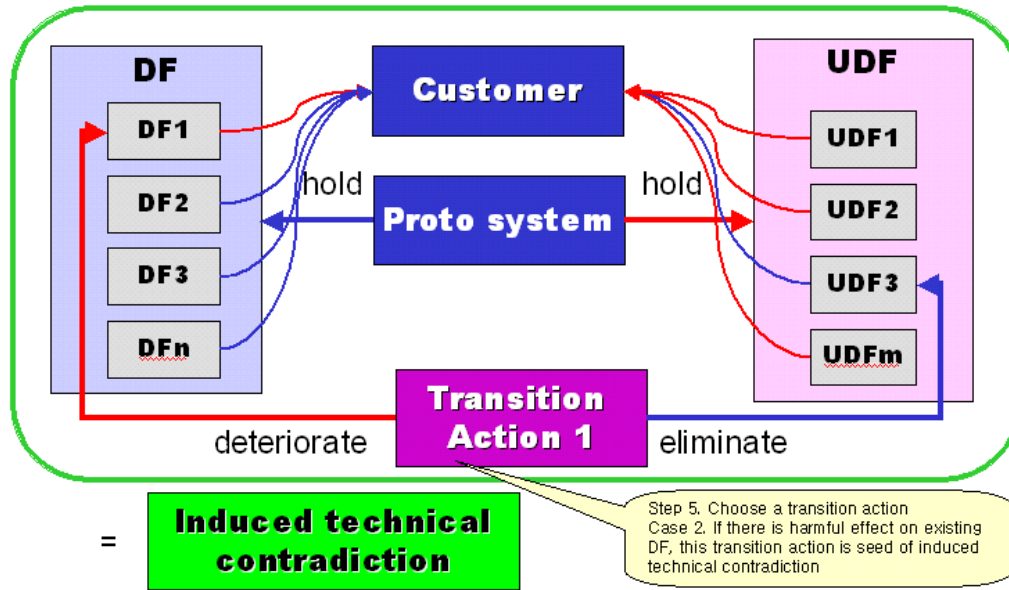


Figure 6. Transition action as tool of induced technical contradiction

Figure 6 shows induced technical contradiction as a result of transition action, it shows transition action as a generator of conflict situation. From the framework of formulating induced technical contradiction, we can identify macro level physical contradiction automatically because ‘Transition Action’ contains a tool of which property should be in two different states at the same time to meet the both requirements.

Even if Figure 6 shows the induced Technical Contradiction between DF and UDF, but in real case Technical Contradiction could appear between DF_i and DF_j and/or between UDF_k and UDF_l .

Step 6. Formulating effects of transition actions to system/supersystem

In real case very often we have a lot of ‘new bad networks’. Single table or single conflict model can’t denote the whole structure of effect of ‘transition action A’. We suggest a simple logic in the form of diagram to formulate this situation as following:

Step 6.1. Choose undesirable function from function analysis result.

e.g. element X ---insufficiently deform--> element Y

Step 6.2. Formulate ‘mission statement’ in deep blue box as following

e.g. Eliminate “element X ---insufficiently deform--> element Y”.

Step 6.3. Choose a ‘transition action A’ in purple blue box and make a link between

‘mission statement’ and ‘transition action’”

Step 6.4. Denote the good effect of ‘transition action A’ to the node of CTQ in blue box because it is useful for us.

Step 6.5. Denote the bad effect of ‘transition action A’ to whole system, supersystem including manufacturing process, operating process and environments in red box because it is harmful for us.

Figure 7 shows typical example of transition effect analysis. This logic is so simple that almost all our researchers could draw down the whole story as a form of transition effect diagram only after 5~10 minutes instruction in consulting meeting. It is a kind of ‘visual thinking’ method followed by multi-screen scheme. Note that the quality of transition effect diagram depends on the knowledge depth of researcher who creates the transition effect diagram. Only well-trained researchers in specific field can create such network of transition effect that includes valuable information set. Just TRIZ consultants cannot create detailed network of knowledge.

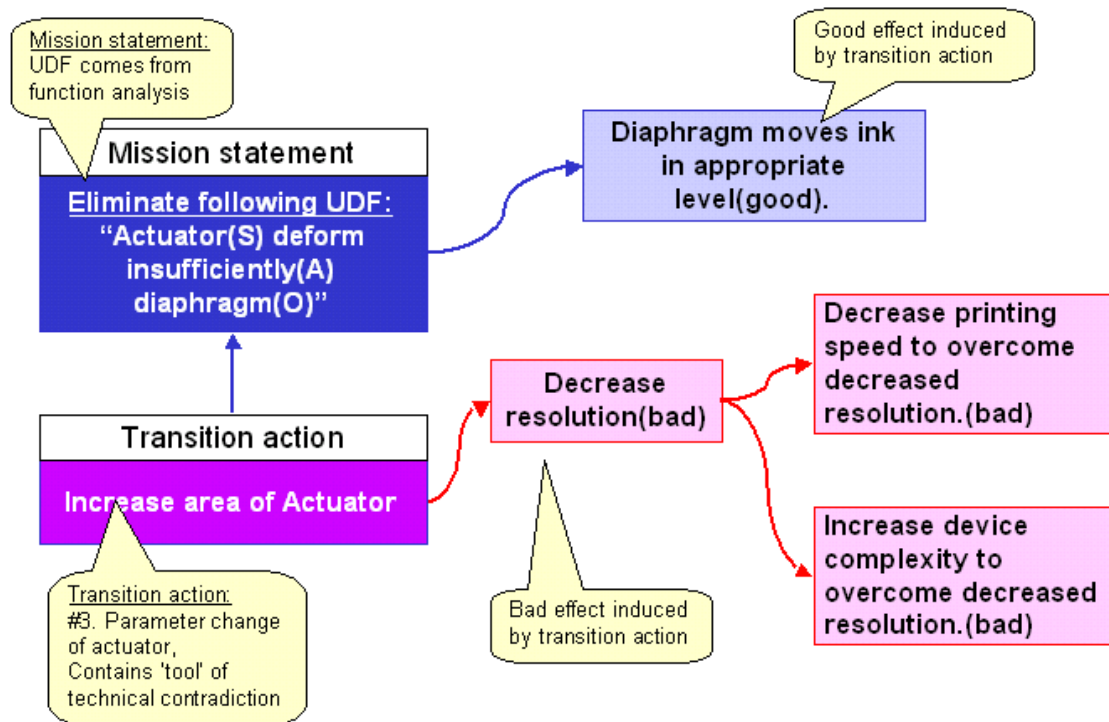


Figure 7. Typical example of transition effect analysis

The Problem Formulator in IWB (Innovation WorkBench™)(4) could easily construct this transition effect diagram. Sometimes, simple drawing tool in any word processing SW could be used to construct transition effect diagram. Altshuller suggested a couple of

methods to clarify technical contradiction in the system; especially parameter change is denoted in ARIZ 77 (1). We collect more methods to clarify technical contradiction and organized as the name of transition action which create contradictory situation.

Step 7. Formulating induced technical contradictions

This step looks very similar to step 1.1 ARIZ-85C (2), but we can do it only after analysis of above steps (not immediately after situation description as recommended in ARIZ).

In our terminology we can introduce it as following:

TC-0

***If I don't 'try a transition action A'(element A is <state 0>),
'An undesirable function B' will exist (bad),
but there is no 'new bad C'(good).***

TC-1

***If I 'try a transition action A' (element A is <state 1>),
'An undesirable function B' will be eliminated (good),
but there appears 'new bad C'.(bad)***

TC-0 corresponds the initial situation without 'transition action A'.

TC-1 corresponds the situation after 'transition action A' has been applied.

We have chosen TC-0/TC-1 instead of traditional convention TC-1/TC-2 because '0' means 'no change' in usual case '1' means alternative case for '0'. We consider that '0', '1' notation seems more appropriate for current digital society.

Step 8. Evaluating importance of induced technical contradiction

We can create numbers of technical contradictions following step 1~7. But every technical contradiction doesn't have the same importance. For the practical purpose, it is necessary to evaluate the importance of each technical contradiction. According to the corresponding CTQ and the severity of problems, researchers can evaluate the importance like the case of FMEA (7)

After evaluation, classical TRIZ methodology can be applied to generate concept solution for each selected 'induced technical contradiction'. Table 1 shows typical example of evaluation results. This template works effective to summarize induced technical contradictions and to evaluate the importance of each problem models.

Every row is filled after appropriate analysis step along step 1~8. If new concept has

been developed in the problem solving procedure, this new concept will be filled in the next blank line of the table and will be evaluated whether new bad appears or not. This table used for preliminary evaluation of the concepts, before turning it into the final solution.

The first and second rows show us that we could succeed to find out important technical contradictions. In this case, contradiction matrix or ARIZ could be appropriate solving strategy.

Table 1. Problem model table with importance

CTQ or Strategy	Old bad	Mechanism of Old bad	Transition Action to remove old bad	New Bad	Importance
Needs of Supersystem	Undesirable function	Cause of UDF	Transition action to eliminate UDF[step3]	New bad due to transition action	Importance of problem
Increase efficiency		Bad adhesion	Press TiO2 after slurry deposition (for good adhesion)	Electrolyte penetrates into TiO2 insufficiently (because there are little pores)	10
Increase efficiency		High defect density in TiO2 particle	Heat treatment at high temperature during TiO2 particle manufacturing(to increase TiO2 crystallinity)	Decrease surface area	10
Increase efficiency		Surface resistance of ITO	N/A		5
Increase efficiency/stability		Total reflection in glass	N/A		10
Increase efficiency/stability		Absorption of ITO itself	N/A		4

But the 4th row tells us even though this undesirable function is so serious; we have failed to suggest an appropriate transition action yet. In this case, 76 standard solutions or effects module in TechOptimizer™ (use of scientific effects from other disciplines) should be considered as the first solving method. Well known articles or parameter change could be another type of transition action also.

Before we can formulate technical contradiction, we are not ready to solve real problem. We would like to note that we consider the Technical Contradiction formulation only as an important part (but not the unique one!) of the total process of the problem statement and resolving. Some authors (18) provided analysis of using Technical and Physical Contradiction and their relationship in practice, based on statistics.

Is it possible to resolve the problem just after Technical Contradiction formulation with using of Altshuller's matrix (40 Principles)?

We don't consider so – in reality it happens very seldom. Technical Contradiction formulation is necessary but not sufficient action for problem solving. We consider TC formulation only as “a control milestone”: if we are able to formulate the Technical Contradiction that it means we understood problem well (at least, the right way).

The fact of the Technical Contradiction existence denotes presence of a Physical Contradiction, which is “hidden” in the Tool of Technical Contradiction. Moreover, we consider that the term “To solve the Problem” really means, “To resolve Physical Contradiction”. But resolving of Physical contradiction usually happens on abstract level, irrelative of System under consideration. We can apply the inventive principles recommended by Altshuller's Matrix Principles while practical Technical embodiment of Physical Contradiction resolving for our system. In other words, Principles applying is a Technical realization of main idea of Physical Contradiction resolution.

The residual part of problem statement and concept generation will be presented soon in our future publications.

Intermediate Verification of the logic

Since Jan. 2004, this logic has been applied to resolve more than 30 field projects in SAIT. SAIT develops many projects for various technologies including electronics, optics, materials, tele-communications, SW etc.

They have usually more than one existing proto system in the initiation stage. After selecting most promising proto system, they could excavate valuable problem model sets as induced technical contradiction and generate high-level concepts for each prototype systems. After generating high-level concepts, they could evaluate the value and risk of each concept, which is important input data to settle down technological strategy for project execution.

Our logic is very effective to generate problems models iteratively, which will lead us from high level concept design down to detailed concept design for system, manufacturing process and probable failure of system. Induced technical contradiction is proven to be an effective key for iterative problem solving, i.e. problem flow technology (2).

Table 2 summarizes the results of induced TC logic for initiation project. Our researchers have taken 40 hours lecture to deliver whole logic to analyze technical contradiction and resolve contradictions.

After 40 hours education we had proceeded consulting meetings with the researchers for 10 weeks. Every week we had a deep discussion about CTQ, UDF(undesirable function of selected proto system), the cause and effect of UDF, probable transition

actions and so on. We could formulate more than 10 different valuable induced TC's per single TRIZ projects and more than 15 different valuable real concept designs (not just an idea but realistic concept solution) per projects. Originally our logic has been designed to formulate technical contradictions for the project planning stage, but it works strong and effective for other stage of project with little trouble.

Note that comparing to normal existing project, situation description and problem modeling are more important part for initiation project. That's the reason why we consider our logic is one of key success factor of TRIZ activity for initiation project.

Now some members of our colleagues in other division of SAMSUNG group decided to apply this logic for their own projects. Some members of DFSS team ask us to make an independent educational course of our logic from conventional TRIZ coursework as 'problem identification methodology' for DFSS Black Belt program also. They say our logic is one of most effective and strong logic to identify clearly important features of problems: what is gap between what we want and what we have, what exact element should be changed, why it should be changed, why it couldn't be changed.

For effective application of this logic with concept generation activity, SW will be most effective interface. We have also devoted our efforts to align TRIZ into DFSS workflow, which was introduced in TRIZCon2004 in Seattle (11); the result of our efforts could be summarized and presented soon as another article.

Table 2. Verification result of suggested logic

Method of logic evaluation	Step 1. Educate 40 hour lecture of classical TRIZ and problem formulation logic to the participants Step 2. Consult real projects of participants for 10 times*1.5 hours Step 3. Participants prepared reports for their TRIZ activity by themselves
Period of logic evaluation	Jan. 2004 ~ July , 2004
Number of projects	More than 30 projects(characteristic of projects as following : Electric, MEMS, Semiconductor, LED, Biochip, Polymer device, IT/SW, etc.)
Number of created ideas	489
Number of filed patents	50

The logic for concept generation and verification will be present as a series of publication soon. We hope our hands-on logic can help many people who really want to solve serious problems and/or who want to teach real hands-on inventive problem

solving to other people.

Conclusion for Part I

A logic chain to formulate technical contradiction as a starting point of inventive problem solving from initial situation has been provided and verified. After defining appropriate project scope, we analyze the element and function of selected proto system. Since information of UDF(Undesirable function) and root cause of UDF are ready, we can suggest transition action to eliminate at least single UDF. Transition action is defined here as ‘direction of eliminating at least single UDF’ chosen from well-known resolving ways. If there is minimal disadvantage according to transition action, this transition action could be considered as one of partial concept solution. But if there is critical disadvantage generated by transition action, we can induce exact sets of technical contradictions in which the corresponding physical contradiction is underlying. Selecting transition action provides us short-cut way to formulate technical contradiction. Transition action has a tool of conflict model in it so we can formulate physical contradiction from it relatively easy. Only after we are able to formulate the Technical Contradiction(s), we can say we are ready to resolve inventive problems. Seven months verification period has shown that this logic is successful for almost all field problem of R&D initiation/execution stage.

Acknowledgement

Special thanks to Dr. N. Shpackovsky and all colleagues in six sigma innovation team in SAIT.

References

1. <Creativity as an Exact Science>, G.S.Altshuller, Translated by A. Williams, 4th printing, Gordon and Breach Publishers Inc., 1998
2. <Seminar of Introduction of OTSM-TRIZ>, N.Khomenko, SAIT, Sep., 2002
3. <TRIZ technology of conceptual design>, Presentation material of 5-day TRIZ seminar, Z. Royzen, in Samsung Electronics in Korea, 1998.
4. Innovation WorkBench™ v.2.8.0, Ideation International Inc.
5. TechOptimizer™, v.3.0, 3.5, 4.0, Invention Machine, Inc.
6. “Ideation TRIZ methodology One-day orientation”, A.Zusman, B. Zlotin, TRIZCon2004, Seattle, April, 2004
7. <Potential failure mode and effects analysis – reference manual, 2nd edition>, Chrysler corp.,

Ford motor company, General Motors Corp., February, 1995

8. "TRIZ in the Process Industry", Gert Poppe and Bart Gras, Presented at ETRIA World Conference TRIZ Future 2001, Held at Univ. of Bath, Bath, UK on November 7-9, 2001.
9. "Hands-on inventive problem solving-0: Prologue", Mi Jeong Song et al., TRIZ-Journal, September, 2004.
10. <Effective Innovation>, D.Clausing and V.Fey, The American Society of Mechanical Engineers, New York, pp27-103, 2004.
11. "TRIZ in Samsung R&D Process", M.J.Song, V.Leniachine, S.H.Cheong, TRIZCon2004, Seattle, April, 2004
12. "Mapping the Innovation Space One: Novel Tools for Problem Definition in Product Innovation", By: Barry Winkless, Dr. John Cooney, TRIZ-Journal, July, 2004
13. <Gödel, Escher, Bach: an Eternal Golden Braid>, Douglas R. Hofstadter, Basic Books, 1979.
14. <DFSS guideline> (in Korean), company private document in SAIT, 2004
15. "Advanced Topics on Substance-Field Modeling", S. Ikonenko, TRIZCon2004, Seattle, April, 2004.
16. <TOC: Password for Enterprise survival> (in Korean), Choi Kwang Shick, Haneun, Korea, 2001.
17. <TRIZ: the right solution at the right time>, Y. Salamatov, Insytex B.W., Netherlands, 1999