Algorithms of Defining an Application Condition of a Functional Clue

Aleksey Pinyayev, pinyayev.a@pg.com

Abbreviations

TESE – Trends of Engineering System Evolution.
TIPS – Theory of Inventive Problem Solving

Abstract

The Application Conditions are instrumental in TESE-based technology prediction and their precise definition is important for the quality of the technical prognosis. Currently, an Application Condition, also known as the functional core of the problem, is selected from a complete functional model of an elemental technical problem. The selection approach leaves plenty of room for errors related to the variability and complexity of this model. Instead, the current article proposes defining an Application Condition step-by-step, starting from a why-why contradiction describing the elemental problem. There are two algorithms, one for the problems related to the harmful actions and another one for the problems related to the useful actions. The proposed approach is better structured, more reliable and precise than the known way.

Background

The application conditions are the essential components of the system of Functional Clues (Pinyayev 2007). In addition to that, the TESE-based technology prediction can be improved if it is done on the basis of the application condition of a key elemental problem rather than on a basis of the entire technical system (Pinyayev 2010). For both of these purposes, the precise definition of an Application Condition, also known as the functional core of the problem, is a key to a quality outcome. A known approach to defining an Application Condition is to select it from a complete functional model of an elemental problem (Pinyayev 2007). However, the very process of building functional diagrams may lead to variable, builder-dependent results. Thus, one cannot be sure that a "complete" functional diagram does not omit or distort important components and interactions. If steps are taken to ensure completeness and correctness of a functional diagram (Litvin Semyon S. 1991), the result becomes very complex and difficult to analyze. Therefore, a method of defining the Application Conditions independently from building a functional model is needed. This method, proposed in the current publication, allows to significantly improve reliability and precision of the Application Condition definition. The method is well-structured and concise and in many cases saves the time needed for defining the Application Condition compared to the method of selecting it from a complete functional model.

The proposed method requires that a why-why analysis of the initial inventive situation was completed, an elemental problem was identified and expressed as a why-why contradiction via the approach described in (Pinyayev 2007). The proposed method also refers to an updated list of the standard application conditions shown in Fig. 1. This list has been simplified compared to the original version described in (Pinyayev 2007) by eliminating conditions H5, H6 and H7. The reason for this simplification is that these Application Conditions can always be converted into condition H1. In this conversion, the information needed for the use of the Application Condition is not lost and can be recovered. Both technology prediction and problem solving via Functional Recommendations can be successfully done with H1 model.

"Functional" and "Non-functional" Problems

All elemental problems can be described via the functional Application Conditions. Sometimes, however, the description of an elemental problem in terms of parameters or properties of a certain component is preferred. A good example is a contradiction between the strength and softness of a paper towel. Such problems naturally lend themselves to techniques such as Contradiction Resolution and Principles and will not be considered in the current publication. It is important to understand that each problem described as a contradiction between parameters or properties can be also expressed as a functional Application Condition if desired since each parameter or property is only important as a means for performing certain functions.

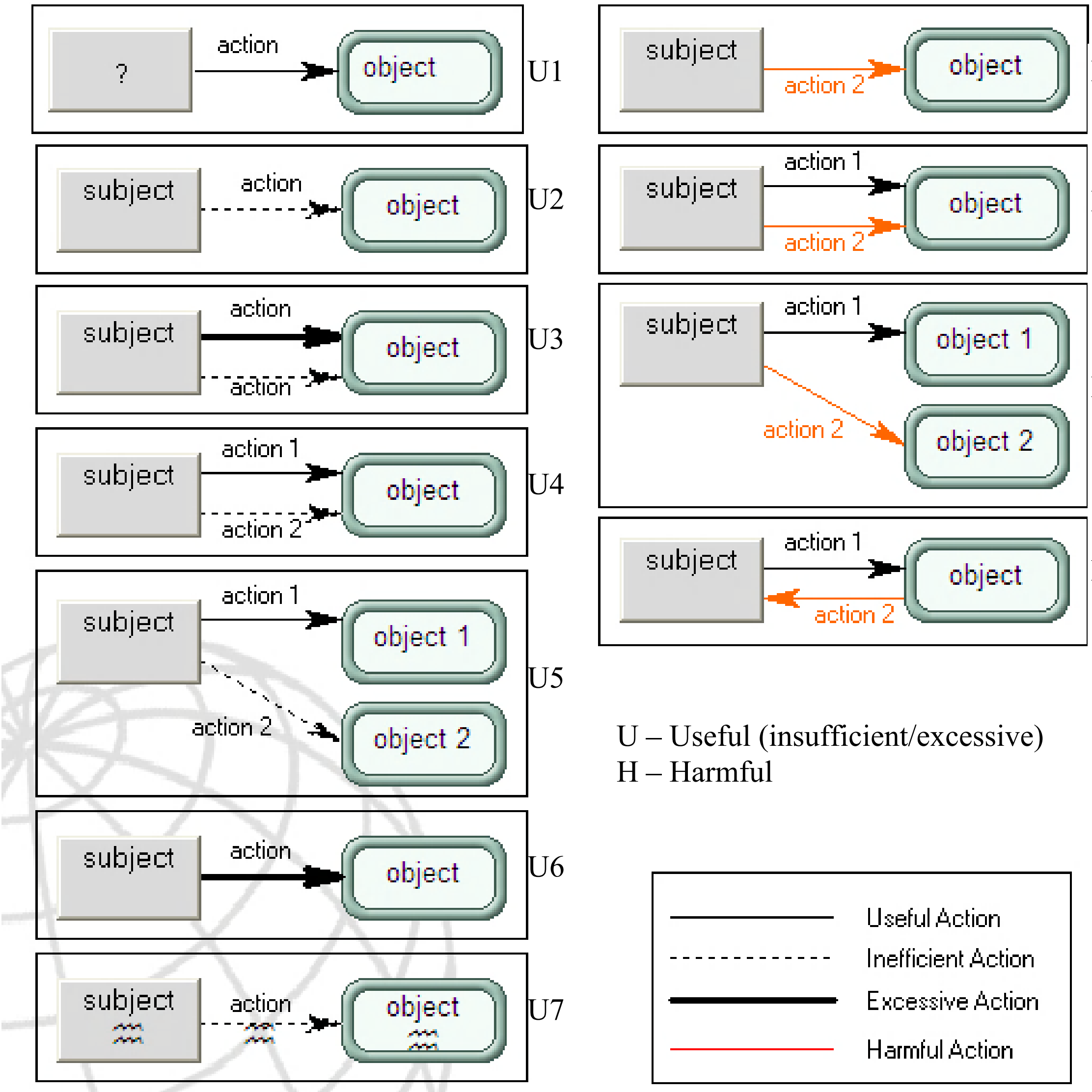
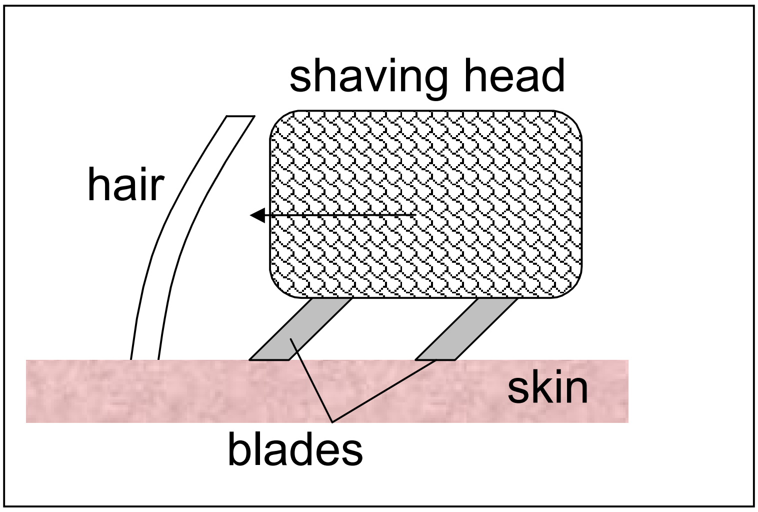


Fig. 1. Standard Application Conditions.

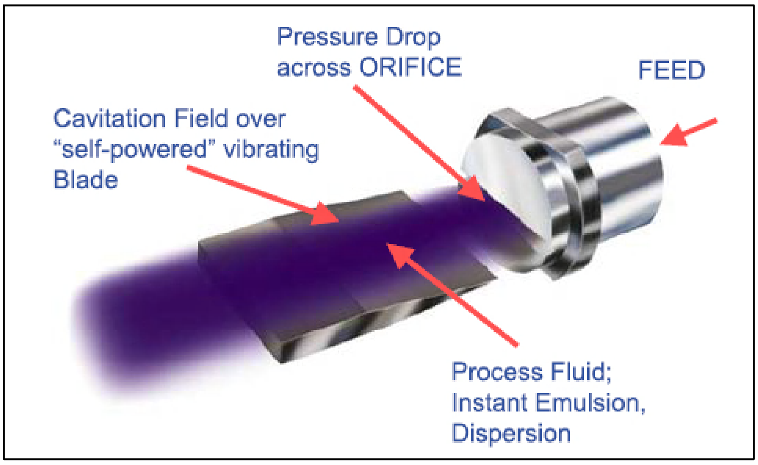
Case Studies

Below are the descriptions of the practical inventive situations which will be used as the examples in the current article. All examples are based on the actual TIPS projects.



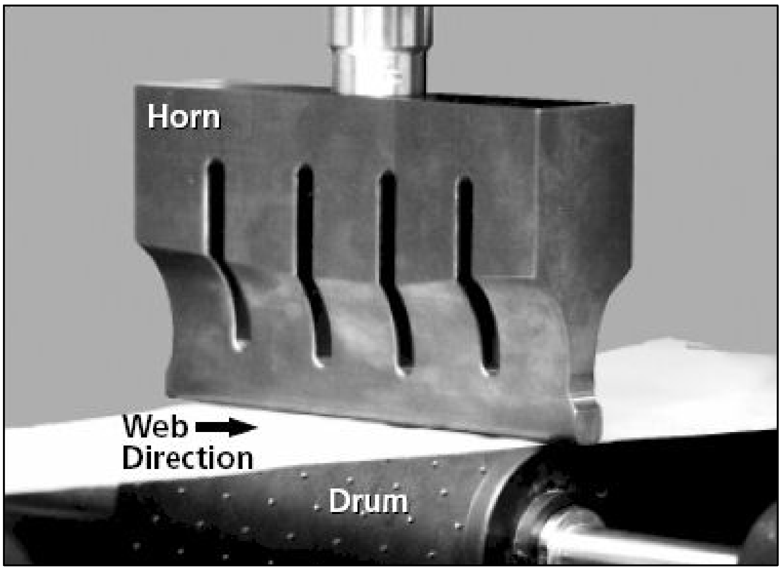
Disposable Razor Head

A shaving head of a double-blade disposable razor works as follows: as the head glides by the surface of the skin, the first blade engages and cuts the hair. When the first blade cuts, it also pulls hair out of its follicle, making the part of the hair normally hidden below the skin surface available for the second blade. The second blade cuts this part of the hair. The natural elasticity of the skin pulls the remaining hair stub back into the hair follicle and now this stub is all underneath the skin surface. This synergy of the double-blade system is supposed to significantly improve the cleanness of shaving but in fact, there is only about 10% improvement (as confirmed by a lab study). How to significantly improve the cleanliness of shaving in a double- or multiple-blade system?



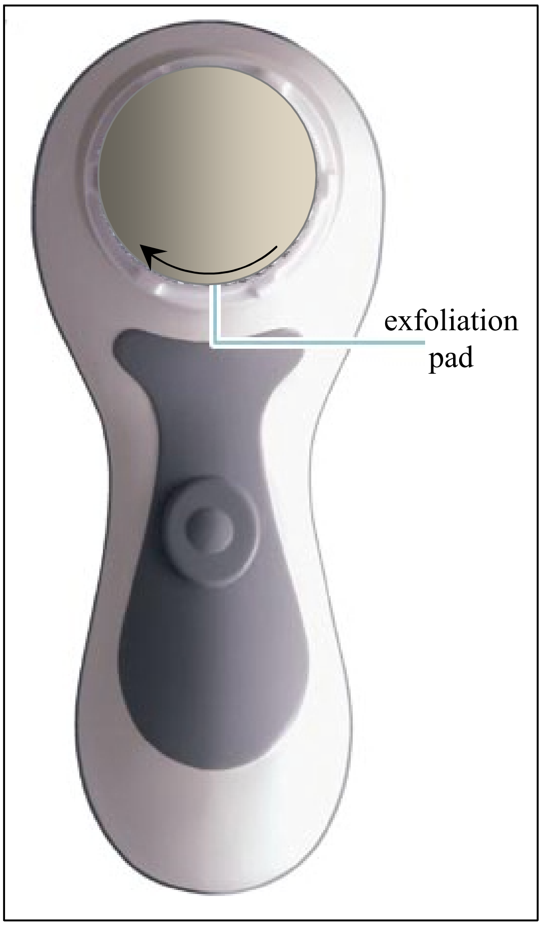
Sonolator Scale-down

Sonolator is a high-shear mixer, homogenizer and emulsifier which works via ultrasonic cavitation created when a high-velocity liquid jet impacts vibrating blade (see figure at the left). The jet is created by passing a pressurized liquid through an orifice. When jet contacts the blade, this blade starts vibrating with an ultrasonic frequency. The cavitation created by the ultrasonic vibrations thoroughly mixes, homogenizes or emulsifies the liquid. A scaled down version of the sonolator is desired in order to be able to predict the performance of the full-scale sonolator based on the results obtained with its small-scale version. However, scaling the sonolator down is not a trivial task and it cannot be achieved via simple reduction of dimensions. The most important parameter which needs to be reduced is the flow rate of the liquid through the orifice. If this reduction is achieved via reduction of the orifice size, the liquid dynamics and flow pattern change so drastically that it would not be possible to predict a full-scale version performance based on the results obtained via a small-scale version. It is desirable to scale the sonolator down without reduction of the orifice size.



Ultrasonic Bonding Process

High-speed bonding of the polymer webs can be achieved via conversion of the ultrasonic energy into heat. In ultrasonic bonding, this conversion is done by the viscoelasticity (internal friction) of the webs. Heat resulting from this conversion melts and binds the webs. For the process to work optimally, it is critical to apply significant mechanical load to the ultrasonic horn. The higher the load, the more efficient is the transfer of the ultrasonic energy to the webs. However, at the loads required by this process, there is significant drag of the web materials by the horn. The drag means localized stretching and tearing of the material. It is desired to achieve very efficient transfer of the ultrasonic energy to the webs without the drag.



Skin Exfoliation Device

Exfoliation removes top few layers of the *stratum corneum* – the uppermost layer of the skin. The skin looks younger and more beautiful as a result of exfoliation, this process also reduces visibility of fine wrinkles and lines. The methods of exfoliation fall into two categories: chemical and mechanical. Within mechanical methods of exfoliation, rotating exfoliation pad was proven to be effective. The pad is made from open-cell foam and loaded with a cream containing abrasive particles. Rotating motion, however, comes with a caveat: when the pad is in contact with the skin, the mechanical reaction from the skin pushes the pad around facial surface in a hard-to-control circular fashion. We call this phenomenon “walking”. Walking makes exfoliation process inconvenient and interferes with the useful outcome. It is desired to achieve good exfoliation without walking.

Application Condition Definition Algorithms

Action Definition Decision

The analysis begins with a decision about which of the two main types the problem belongs: problems with useful (U) or harmful (H) actions (Fig. 2). This decision is made based on the information contained in the why-why contradiction. The contradictive requirements are converted into corresponding actions and the decision is based on classifying of these actions. If at least one of the actions is harmful, the contradiction is related to the harmful action; otherwise it is related to an insufficient or excessive action.

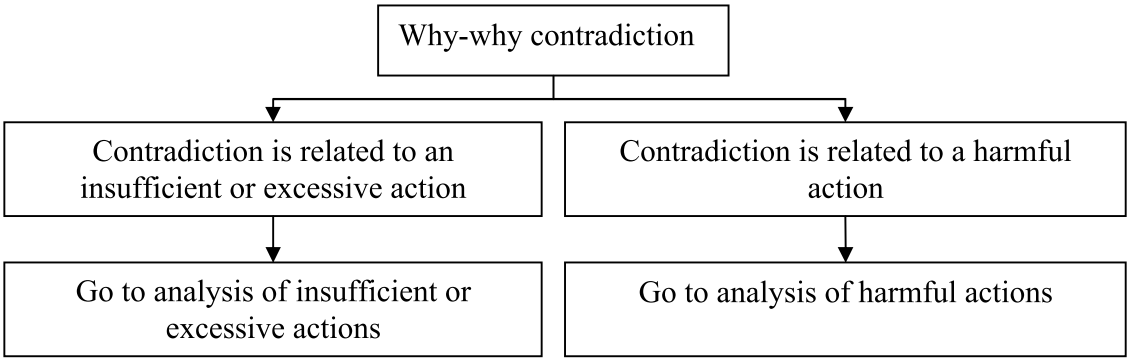


Fig. 2. Action definition decision

Below are the examples of the why-why contradictions from both U and H categories. The why-why contradictions were obtained by applying why-why analysis as recommended in (Pinyayev 2007) to the inventive situations described in the previous chapter.

Problems with useful actions (U)	Problems with harmful actions (H)
1.1. How to improve pulling action of the disposable razor blade without reducing its cutting action?	2.1. How to increase the mechanical load applied to the ultrasonic horn without increasing the drag of the polymer webs?
1.2. How to reduce the flow rate through an orifice without reducing the flow velocity and without reducing the size of the orifice?	2.2. How to eliminate “walking” of the exfoliation pad over the surface of the skin keeping good mechanical contact between the pad and the skin?

Problem 1.1. There are two actions mentioned in the contradiction: pulling and cutting, both useful, so the contradiction is classified as U.
Problem 1.2. This elemental problem deals with the flow through an orifice. Functionality of an orifice is described by two related functions [2]: *let* liquid *flow through* and *restrict the flow* of liquid. Both functions are useful, so the contradiction is classified as U.
Problem 2.1. There are two actions mentioned in the contradiction: "increase the mechanical load" (desirable) and "increase the drag" (undesirable or harmful). Because of the second action, the problem is classified as H.
Problem 2.2. There are two actions mentioned in the contradiction: "keep mechanical contact" (desirable) and ""walking" (undesirable or harmful). Because of the second action, the problem is classified as H.

Note that at this point of analysis precise definitions of the functions are not necessary. This will be done later. To make a right choice between U and H, approximate definitions of actions suffice. In most cases, we do not define objects of the functions at this first step.

Analysis of the Harmful Actions

The objective of this analysis is to define which one of the seven standard H-type application conditions describes the contradiction the best. I propose the following algorithm for this purpose.

Main Algorithm - H

1. Define the object A which is harmed.
2. Define the harmful action A on the object A.
3. Define the object B performing the harmful action.
4. Does the object B perform a useful function? (Yes/No)
   4.1. If Yes, go to the sub-algorithm for H2/H3.
   4.2. If No, go to the next step.
5. Does the object A perform a useful action on object B?
   5.1. If Yes, go to the sub-algorithm for H4.
   5.2. If No, the problem is defined as H1.

Sub-algorithm for H2/H3

1. Compile a list of the useful functions of the object B (actions and objects, including object A).
2. Do the following mental experiment with each of the useful functions in the list:
   2.1. Find practical ways to reduce (but not eliminate) the harmful action AA.
   2.2. Reduce the harmful action AA.
   2.3. See whether the reduction of the harmful action AA leads to the reduction of the useful function.

3. If the reduction of the harmful action AA *does not* weaken the useful function, delete this useful action from the list.
4. If…
   4.1. …there is only one useful action left in the list, go to step 5.
   4.2. …there is more than one useful action left in the list, go to step 5 with each of the useful functions separately.
   4.3. …there are no useful actions left in the list, go to step 5 of the main algorithm.
5. If the object of the useful function is different from the object A, the problem is defined as H3. If not, the problem is defined as H2.

Sub-algorithm for H4

1. Compile a list of the useful actions of the object A on object B.
2. Do the following mental experiment for each of the useful actions in the list:
   2.1. Find a practical way to reduce (but not eliminate) the harmful action AA.
   2.2. Reduce the harmful action AA and see whether this reduction leads to the reduction of the useful action.
3. If reduction of the harmful action AA *does not* weaken the useful action, delete this useful action from the list.
4. If…
   4.1. …there is only one useful action left in the list, the problem is defined as H4.
   4.2. …there is more than one useful action left in the list, you have identified several problems of H4 type, one for each useful action. These problems will have to be solved separately from one another.
   4.3. …there are no useful actions left in the list, the problem is defined as H1.

Analysis of the Useful Actions

The objective of this analysis is to define which one of the seven standard U-type application conditions describes the contradiction the best. I propose the following algorithm for this purpose.

Main Algorithm - U

1. Define the useful function AA (action A and object A) which needs to be improved.
2. Define the object B performing the useful function AA.
   2.1. If the object B is unknown, the problem is defined as U1.
   2.2. If the object B is known, go to the next step.
3. Is the function AA excessive? (Yes/No)
   3.1. If Yes, go to sub-algorithm for U3/U6.
   3.2. If No, go to the next step.

4. Is the action B insufficient because of the variability of parameters of the objects A or B? (Yes/No)
   4.1. If Yes, go to the sub-algorithm for U7.
   4.2. If No, go to the sub-algorithm for U2/U4/U5.

Sub-Algorithm for U3/U6

1. Is the action A excessive in some areas of the object A and insufficient in some other areas of the same object? (Yes/No).
   1.1. If Yes, the problem is defined as U3.
   1.2. If No, the problem is defined as U6.

Sub-algorithm for U2/U4/U5

1. Compile a list of the useful functions of the object B (actions and objects, excluding action A and object A).
2. Do the following mental experiment with each of the actions in the list:
   2.1. Find a practical way to improve the action AA.
   2.2. Improve the action AA.
   2.3. See whether the improvement of the action AA leads to weakening of the function from the list.
3. If the improvement of the action AA *does not* lead to weakening of the function, delete this function from the list.
4. If…
   4.1. …there is only one useful function left in the list, go to step 5.
   4.2. …there is more than one useful action left in the list, go to step 5 with each of the useful functions separately.
   4.3. …there are no useful actions left in the list, the problem is defined as U2.
5. If the object of the useful function left in the list is different from the object A, the problem is defined as U5. If not, the problem is defined as U4.

Sub-Algorithm for U7

1. The variability of which object - A or B - causes the action A to be insufficient?
   1.1. If both A and B cause the insufficiency of the action A, divide this situation into two separate problems, each defined as U7. Go to the next step.
2. Define the problem as U7, indicating the variable object defined at the previous step.

Analysis of Examples

Problem 1.1. Algorithm U.

1. Action A: pull; Object A: hair.
2. Object B: blade 1.
3. No (the function "pull *hair*" is insufficient).

4. No (variability of the hair only partially contributes to the problem. The problem is still there when the parameters of the hair do not vary significantly).

Sub-algorithm for U2/U4/U5

1.

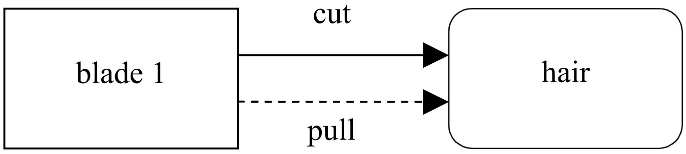
Action	Object
cut	hair
remove (from skin)	hair

2.1. Making the blade 1 duller would improve the function “pull *hair*”.

2.3. Making the blade duller weakens the function “cut hair” but does not weaken the function ”remove hair from skin”.

3. We delete the function “remove hair (from skin)” from the list.
4. There is only one function (cut hair) left in the list, so we go to the step 5.

5.1. the object of the useful function left in the list (hair) is the same as the Object A. The problem is defined as U4:

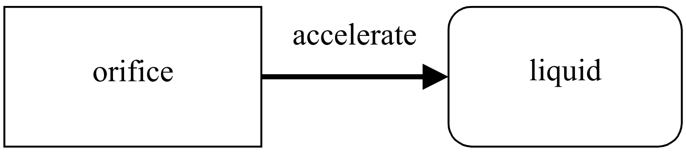


Problem 1.2. Algorithm U.

1. Action A: accelerate; Object A: liquid.
2. Orifice.
3. Yes (the volume of the accelerated liquid is too high).

Sub-Algorithm for U3/U6

1. No. The excessive action “accelerate” applies to all available liquid. We have defined the problem as U6:



Problem 2.1. Algorithm H.

1. Object A: polymer webs.
2. Action A: drag (stretch, damage).
3. Object B: horn.
4. Yes (horn perform such useful functions as “compress *polymer webs*” and “deliver *ultrasonic energy* to the webs”. We go to the sub-algorithm for H2/H3.

Sub-algorithm for H2/H3

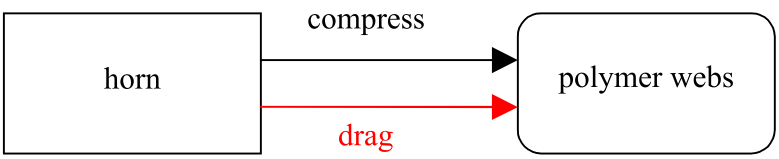
1.

Action	Object
compress	polymer webs
deliver (to the polymer webs)	ultrasonic energy

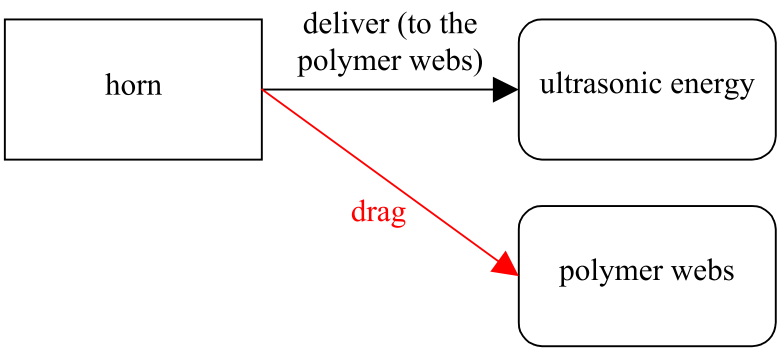
2.1. An easy way to reduce the harmful function "drag *polymer webs*" is to reduce the mechanical load applied to the horn.

2.3. The reduction of the mechanical load applied to the webs weakens both useful functions in the list.

5. a. The object of the useful function "compress *polymer webs*" is the same as the object A (polymer webs). The problem is defined as H2:



b. The object of the useful function "deliver *ultrasonic energy* to the polymer webs" is different from the polymer webs. The problem is defined as H3:



Problem 2.2. Algorithm H.

1. Object A: exfoliation pad.
2. Action A: walk (move in circles).
3. Object B: skin
4. Yes (skin supports exfoliation pad). We go to the sub-algorithm for H2/H3.

Sub-algorithm for H2/H3.

1.

Action	Object
support	exfoliation pad
protect	underlying facial tissues
sense touch of	exfoliation pad
sense applied load of	exfoliation pad
sense position of	exfoliation pad

2.1. An easy way to reduce the harmful action “walk” is to reduce mechanical load applied to the exfoliation pad.

2.3. None of the useful function in the list is weakened by the reduction of the mechanical load applied to the exfoliation pad. We go to the step 5 of the main algorithm.

5. Yes. The exfoliation pad exfoliates skin. We go to the sub-algorithm for H4.

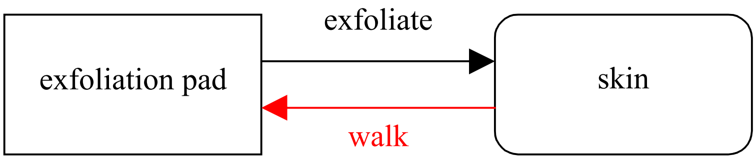
Sub-algorithm for H4.

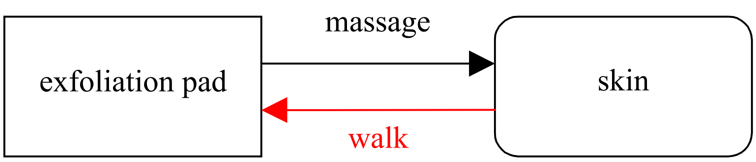
1.

Action	Object
exfoliate	skin
massage	skin
apply	exfoliating cream
carry	exfoliating cream
rotate	exfoliating particles

2.1. An easy way to reduce the harmful action “walk” is to reduce mechanical load applied to the exfoliation pad.

2.3. The useful functions “exfoliate skin” and “massage skin” weaken when the applied mechanical load is reduced. The rest of the useful functions do not weaken. We have identified two problems of H4 type:





Results and Conclusions

The algorithms suggested in the current publication allow to define an Application Condition step-by step, without reliance on a complete functional model of an elemental problem. There is an introductory algorithm, called Action Definition Decision, which leads to one of the two main algorithms: one for problems with the useful functions and another one – for those with harmful functions. The main algorithms are constructed in two layers with the first layer routing the user to one of the sub-algorithms of the second layer for a detailed analysis.

The algorithms present a well-defined, logical way of choosing one of the standard Application Conditions. They are designed to provide a reliable, precise problem definition in a form of a functional diagram. This definition can be used together with the Functional Recommendations (Pinyayev 2007) to find the solutions to the elemental problems. It can also be used for the TESE-based prediction, as described in (Pinyayev 2010). Finally, this definition can serve as a graphical model of inventive problem in the Algorithm for Inventive problem Solving (ARIZ).

References

Litvin Semyon S., G. V. M. (1991). Основные положения методики проведения функционально-стоимостного анализа. Методические рекомендации. Moscow, Inform-FSA. . http://www.triz-summit.ru/ru/section.php?docId=3952

Pinyayev, A. M. (2007). A Method for Inventive Problem Analysis and Solution Based On Why-Why Analysis and Functional Clues. St. Petersburg, Russia, International TRIZ Association. **TRIZ Master:** 24 pp. http://triz-summit.ru/ru/section.php?docId=3791

Pinyayev, A. M. (2010). Application Condition as a Starting Point of TESE-based Prediction. TRIZ Summit 2010, St. Petersburg, Russia.