FAST for TRIZ

Kai Hiltmann

FAST for TRIZ

Hiltmann, Kai[0000-0002-0700-3146]

Coburg University of Applied Sciences and Arts, Friedrich-Streib-Str. 2, D – 96450 Coburg,
kai.hiltmann@hs-coburg.de ✉

Abstract. The functional framework of FAST, which C. Bytheway originally published in 1971, has been widely employed in several variants in Value Engineering, but is not common in TRIZ. However, the original, logically consistent and unabbreviated form Bytheway originally proposed can be an instructive way to model a landscape of functions that can be used in TRIZ for a variety of purposes. Among these are landscaping the functions of the system and its super systems, including neighbouring systems and subsystems, thus depicting a system in new coordinates. In contrast to Function Analysis, possible ways of functional integration and evolution of a system can be modelled and explained; moreover, the technique constitutes a creativity tool and a strategic tool indicating disruptive innovations.

Keywords: Functions, FAST, Map of Functions.

1 Introduction

The notion of function was coined by L. Miles in 1947 [1, p. xxi], so G. Altshuller does not seem to have known it before about 1980 [2]. Altshuller instead used the term of field to express the influence of one object onto another, even though the concepts of field and function are not equivalent [2]. Moreover, he used the term of ideality to describe the main function of an engineering system (**ES**) and to relate a function to the cost and effort required for its realization. Highest ideality thus means pure fulfilment of the main function at no cost [3, ch.2, p.83].

In TRIZ, Gerasimov et al. first employed the term of function in 1990[1] in the context of Functional Value Analysis [2, 5] with the definition „*function is the manifestation of the properties of a material object, consisting in its action (influence or interaction) on the change in the state of other material objects*". In 1992, Gerasimov and others introduced function analysis, which models the interactions between the components of an engineering system and its environment – the super system – [6]. The functional model thus enlarges the substance-field model from the conflicting pair to the complete system and connects the notions of field and function. In 1996, Terninko and Zusman used the term in a generalized meaning of events or states of a system in their tool „Problem Formulation" [7, 8].

---

[1] Souchkov [4] mentions the year 1988 but we could not identify the source

As will be shown, there are different concepts of functions and of their relations with different models addressing different objectives. This often causes difficulties in adequately formulating functions.

At the task of product planning and development, it is necessary to analyse the functionality of products in order to define new products, to seek new applications, and to match market needs with the uses that products offer. A further objective in TRIZ itself is to understand the mechanisms of the so-called laws or trends of system evolution. The functional models in use do not support this objective.

C. Bytheway developed his „*Function Analysis System Technique, FAST*" in 1963 [9, sec. A2.3.2], which he originally intended as a support for the value analyst in correctly composing the hierarchy of functions in a functional tree [10]. To achieve this, he introduced a logical scheme of „*how? vs. why?*" into the functional tree, corresponding to the direction of hierarchical level. In doing so, he additionally attributed a new meaning to the tree structure. In addition, the two questions can trigger creativity to find alternative ways of performing a function or finding new uses to it. The questions of how and why do not have to end at the system boundaries, though. If the logic is extended consistently, a functional structure is achieved that functionally connects a product with its environment and indicates chances for further evolution. This produces a functional map of needs and their fulfilment – extending from basic human activities and natural resources until basic human needs – in which the borders of the product are outlined.

A part of this work was presented earlier on TRIZfest conference 2018 [11] and has been complemented since.

2   Function modelling

Gerasimov's definition has largely purported the definition in standard VDI 4521-1 [12] as „*influence from a system or a system component upon one or more others which changes, eliminates, or maintains a parameter of the other component or system*". The TRIZ Function Model is a directed graph that models the components as nodes and their functions – i.e. relations between the components – as edges [13]. Standard EN 1325 for value engineering (**VE**) also defines a function as an „*action of a product or of a component of its*" [14], even though value analysts in practice attach more importance to formulations that inspire phantasy than describe measurable effects [15].

Other function models are [16]:

- the flow-oriented model, i.e. a block diagram with flows of energy, substance, and information as system quantities

- the relation-oriented model, which describes interrelations between functions as the problem formulation does,

- and the functional tree as a hierarchical system of functions.

Reports on function models have been compiled by Ponn [17], Stetter [18], and others.

The different models picture different relations, with only the two TRIZ models considering the interrelation of a system with its super system.

3   The FAST model

Bytheway's method proved its value to determine the basic (main) function of the ES. It was quickly modified by others: Ruggles introduced „*the notions of superordinate functions, essential functions, secondary functions, permanent functions, simultaneous functions, design objectives, and problem domains*" [1, pp 11]. Snodgrass determined a fixed format, „*refers to superordinate functions as "tasks" and below them places essential functions and four supplementary functions. He designates these four supplementary functions as assuring convenience, reliability, satisfaction, and appeal, and writes that their purpose is to secure customer "acceptance" of the product.*

The Ruggles and Snodgrass versions of FAST, with their patterned, formulaic diagrams, represent interpretations not seen in Bytheway's FAST. We can fairly say that FAST has changed course as its significance and emphasis have shifted from the application process to the diagrammed result." We are reporting these modifications in order to justify a use of FAST differing from its standard application in VE. The current formats used in VE are reproduced in fig. 1 for standard EN 12973 (preliminary version) and in fig. 2 for German standard VDI 2803.
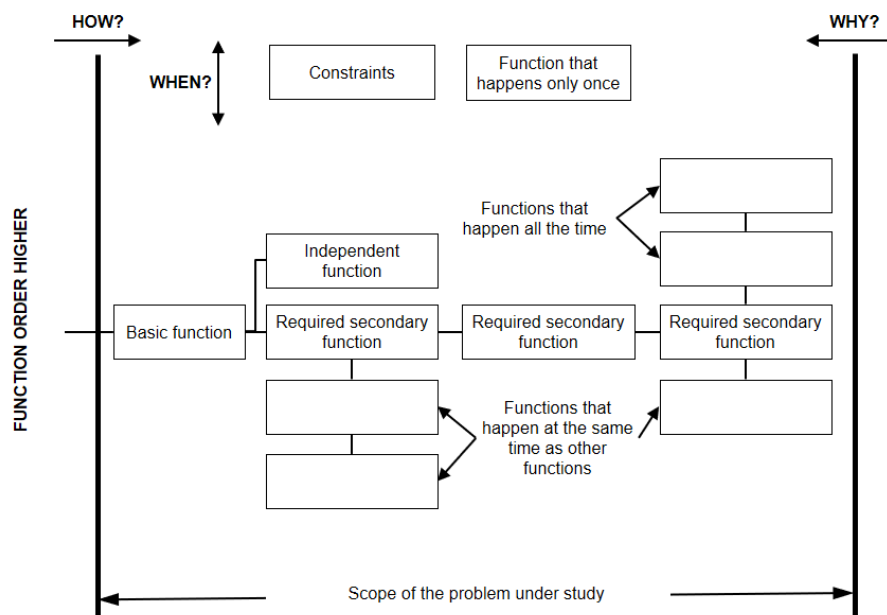

Fig. 1. FAST Diagram acc. to prEN 12973:2017. Note constraints and rarely occurring functions inserted on top and without connections. "Functions that happen at the same time as other functions" are also called "simultaneous functions".
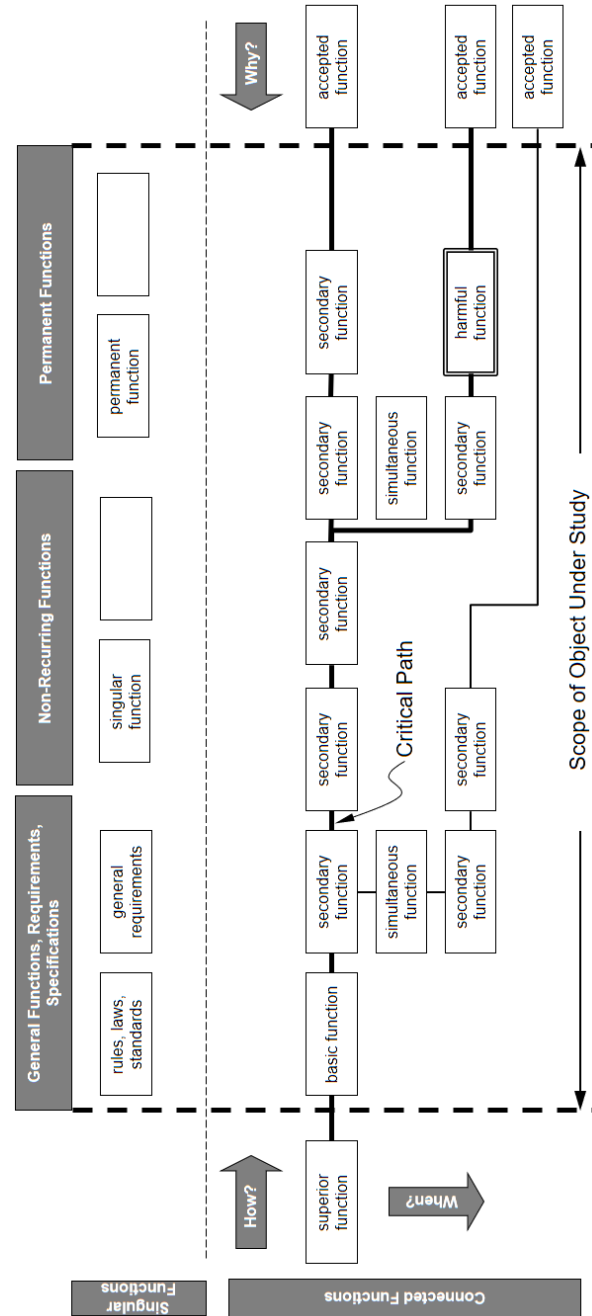
4

Fig. 2. FAST Diagram acc. to German standard VDI 2803: 2017. Note the "accepted functions" delivered by the super system on the right. Translation by the author; "harmful function" is called "problem domain" by Ruggles

4 Modifying FAST for TRIZ

4.1 Basic concept

The modification we suggest for use in TRIZ is just based on Bytheway's original logics of "why" and "how". To make it compatible with TRIZ conventions, we formulate functions according to the definition given above. The basic model is shown in fig. 3: The ES performs the functions enclosed in the system boundary (replacing the Scope Lines in VE). With the "how" direction extending to the right (the coordinate system may actually be oriented arbitrarily), the first function right to the left boundary is the system's main function (or basic function in VE). Ikovenko gives a training example of a projectile hitting a soldier's helmet and asks for the function of the helmet [19]. This may be to "deflect bullet". Asking "how is 'deflect bullet' performed?" one may find several subfunctions ("secondary functions" in VE) to be required. It may be, however, that a helpful answer is "accelerate bullet [sideways][2]". In this case, subfunction 1 would be a mere reformulation of the function ("deflect bullet" = "accelerate bullet"). In the second step, asking "how is 'accelerate bullet [sideways]' performed?", a few subfunctions may be identified, one of them being "transfer IMPULSE[3] [to bullet]". Transferring impulse means simultaneously transferring a counter-impulse, in this case to the head, which constitutes a *harmful simultaneous function*.

Often, functions require secondary functions that are performed not by the system itself but by super system components; these are marked as "*accepted functions*". Typical accepted functions are "control system", "provide electric energy[4]" or "position helmet" as in the example.

On the other hand, one will ask "*why is 'deflect bullet' performed?*". The answer is the purpose of the ES: the main function is performed in order to fulfil its super function, which should be formulated in a user-related way [14], i.e. "protect head". One may continue to ask "why should 'protect head' be performed?". This will lead to a secondary super function "keep body alive". At this point, a basic boundary is reached: A human being is not an engineering system and thus has no purpose to meet, so no reason is required for staying alive and the functional tree has reached its root point.

Please note in the diagram of fig. 3 that in contrast to VE, a situation mentioning solely one single subfunction corresponds to a mere reformulation. Sometimes, however, more subfunctions may exist but are not considered relevant and therefore are neglected. In the example, "protect head" and "keep body alive" require more functions than just deflecting projectiles. We mark this situation with an additional line ending in a bullet (as an abbreviation for "…"). In the diagram, this situation is also indicated after subfunction 3.

---

[2] Since correct function formulations are difficult for beginners, we use to write the attributes of the function in [brackets].

[3] We use to write FIELDS in a different font

[4] We regard "provide energy" a correct formulation because (a) electric energy has parameters voltage and amps, can influence components and be influenced, and (b) "providing" changes these parameters.
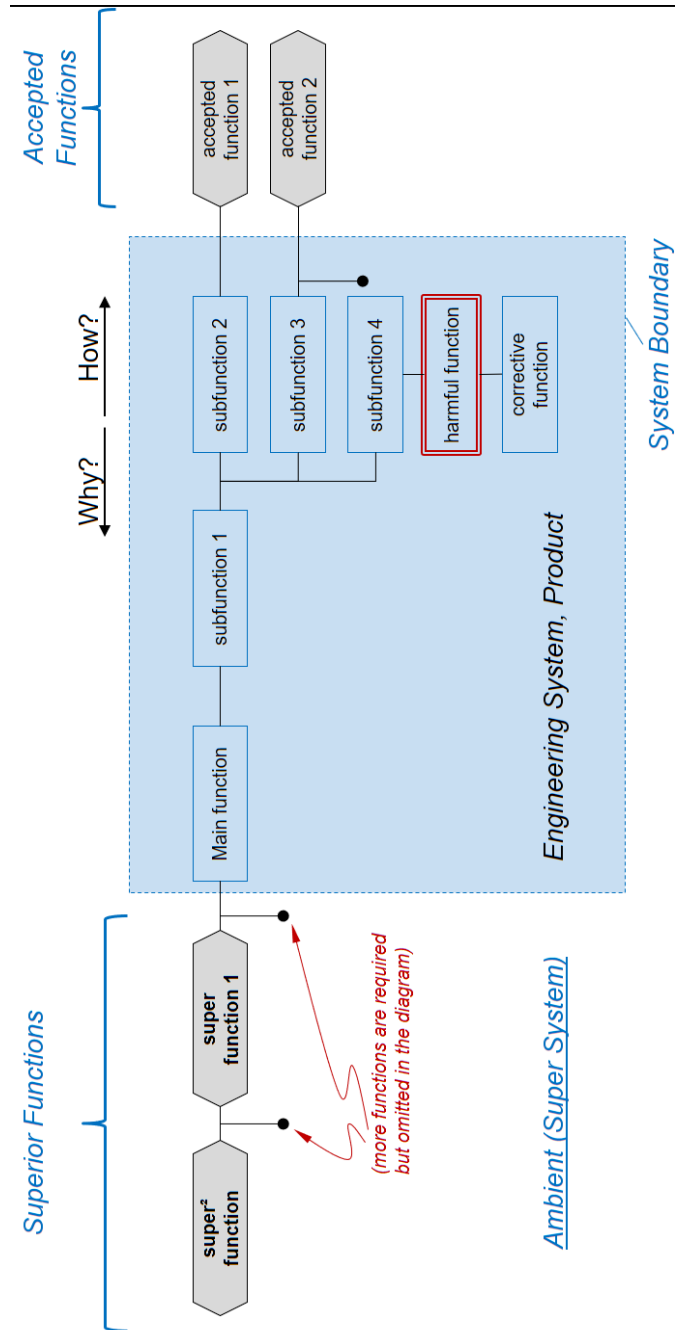
6

Fig. 3. Basic concept of FAST for TRIZ. We use to draw supersystem functions in hexagons.

As a rule, every function in the ES is supposed to possess a super function, i.e. a neighbour to its "why" side. Otherwise, the respective function would be evidently useless. Therefore, we do not suggest to insert any general, singular, recurring etc. functions in this diagram as isolated boxes. The only exception to this rule are simultaneous functions, first of all harmful ones, which of course do not serve a useful purpose (they may possess harmful functions as their super functions, though, see fig. 8). We have also marked a corrective function associated to the harmful one in fig. 3. At least, the fact that corrective functions lack super functions is both plausible and agrees with trimming rules for processes [19].

Please also note that the configuration of the system modelled may not be constant over time – depending on the phase in its life cycle or its momentary use, super functions, accepted functions, and even the internal functional structure may change. The diagram must therefore be redrawn for differing situations or this variability should be indicated by suitable means.

4.2   Assistance in formulating functions

Identification of functions is required for various reasons. These include TRIZ Function Analysis [20], which, in turn is used for system improvement, problem solution, or cost reduction, for description and understanding the objectives of the customer as well as of an ES's structure and operation, and for analysis of a projected design task for subsequent search for partial solutions [16, 21].

When working with functions, users often encounter difficulties finding the correct formulation. Despite the definition of a function seems simple and logical, even experts get into predicaments in practice. Indeed, there are different concepts of functions as Pulm points out [22, 16]. The official VDI 4521 definition regards functions as relations between system components. Correspondingly, in his lessons Ikovenko asks for the function of an open door [19]. The answer is "none" since the open door does not interact with persons travelling through. If however doors had no function, there would be no reason to build them[5]. The concept of functions is thus closely related to the specific functional model chosen [22]. It therefore seems appropriate to translate formulations from one concept to another, which can be done easily in the FAST model. In the helmet example, "protect head" would then be a reasonable *user-related formulation* which is translated via secondary function into the *system-related* formulation "deflect bullet".

Another difficulty lies in interface functions: If the functionality of a hammer is only described as "drive nails", its usability for the worker will be neglected – a mistake that has been made by engineers for long times. Practitioners therefore use to formulate "hammer permits / enables / supports use", which is not correct from the standpoint of system theory since "use" is not a component and "permit" is not an influence. The FAST model though can easily consider interface functions: Due to the separation in

---

[5] a formal argumentation in TRIZ might be to formulate a harmful function of a wall "block persons" and to counterweigh it with the corrective one of the door "remove wall".

8

system and super system functions, there is no more need to formulate "hammer supports use" but "move / accelerate / guide / … / hammer" in FAST is a function of the super system.

By asking "how is <function> performed?", the structure of the system is detailed further until a satisfying level of detail is reached. Subfunctions of this level are called "elementary functions" [23 pp. 30, 24 p. 423]. It depends on the system and its user to decide which degree of refinement is elementary, as is the rule in TRIZ function analysis [20]. A machine designer will consider a motor and its function "transform electrical energy [to mechanical]" as an elementary component whereas a motor producer has to consider the components of the motor and their respective functions.

In the process that standard VDI 2221 recommends for mechanical design, function analysis among others serves to identify subordinate problems. After these have been found, their corresponding requirements ought to be conveyed to the specification sheet. We do so by first inserting comments into the diagram, Fig. 4, and then collecting the comments in the specification sheet.
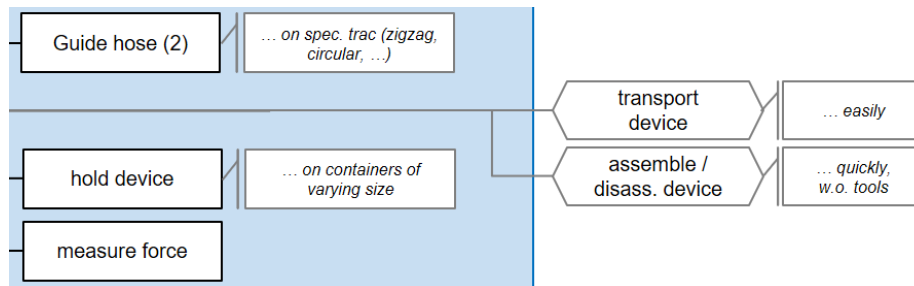

Fig. 4. Part of a FAST diagram for a fire-fighting device with comments

It may further be helpful to indicate interface parameters, especially if functions are to be performed by differing function carriers as in highly modular systems. These parameters can be entered into the model as shown in Fig. 5:
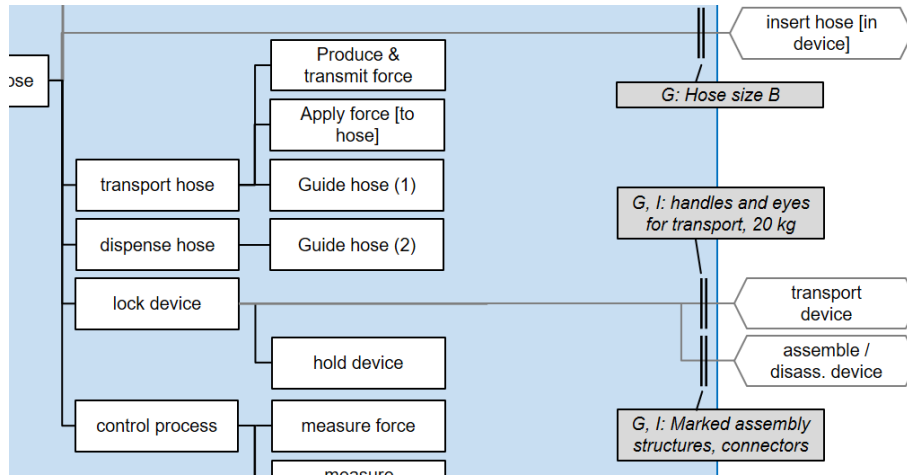
Fig. 5. Interface parameters (**G** geometric, **I** information, **E** energetic) entered in the model

4.3 Landscape of Functions / Functional Map

Typically, a system fulfils not just a single basic function, i.e. an effect on its super system, but several, including harmful ones and aesthetical functions. For instance, an electric drilling machine serves for drilling workpieces. Besides this, it can also be used to produce chamfers, for threading etc., but also to stir paint and for countless other applications. We call these functions "*Additional*" ones, fig. 6. Which of these basic functions is the main function is arbitrary and again depends on the user and the considered phase of the product life cycle. The argument, the main function was the one the ES was built or bought for does not hold since depending on the situation and the life cycle phase, the main function may change [25, 24 p. 423].

A question that comes up when searching for additional functions is "which exactly are all the functions, also unknown ones, a system can perform?" One tool to identify possible additional functions is Anticipatory Failure Determination [26, 27], another Reverse Function-Oriented Search [28] but more work is needed on this subject.

An ES always serves a superior function[6]; since this is a user-related function, usually a use aspect and an aesthetic aspect must be distinguished [14]. Each ES therefore uses to serve *at least two superior functions* but the ES is not sufficient in fulfilling these. If it was, the ES's basic function would be just a reformulation of the superior function. For instance, a hammer serves to impact on an object and the user needs this action to drive a nail into a wall. Obviously, two more ES's are required: The nail and the wall (and the worker). All these three (four) systems together with their respective functions can fulfil the user's desired function which requires mutual matching.

Functions that are fulfilled externally from the ES and support the ES in serving a common superior function we call "*Neighbouring Functions*". Neighbouring functions may be performed by one or more systems in the neighbourhood of the system, i.e. in

---

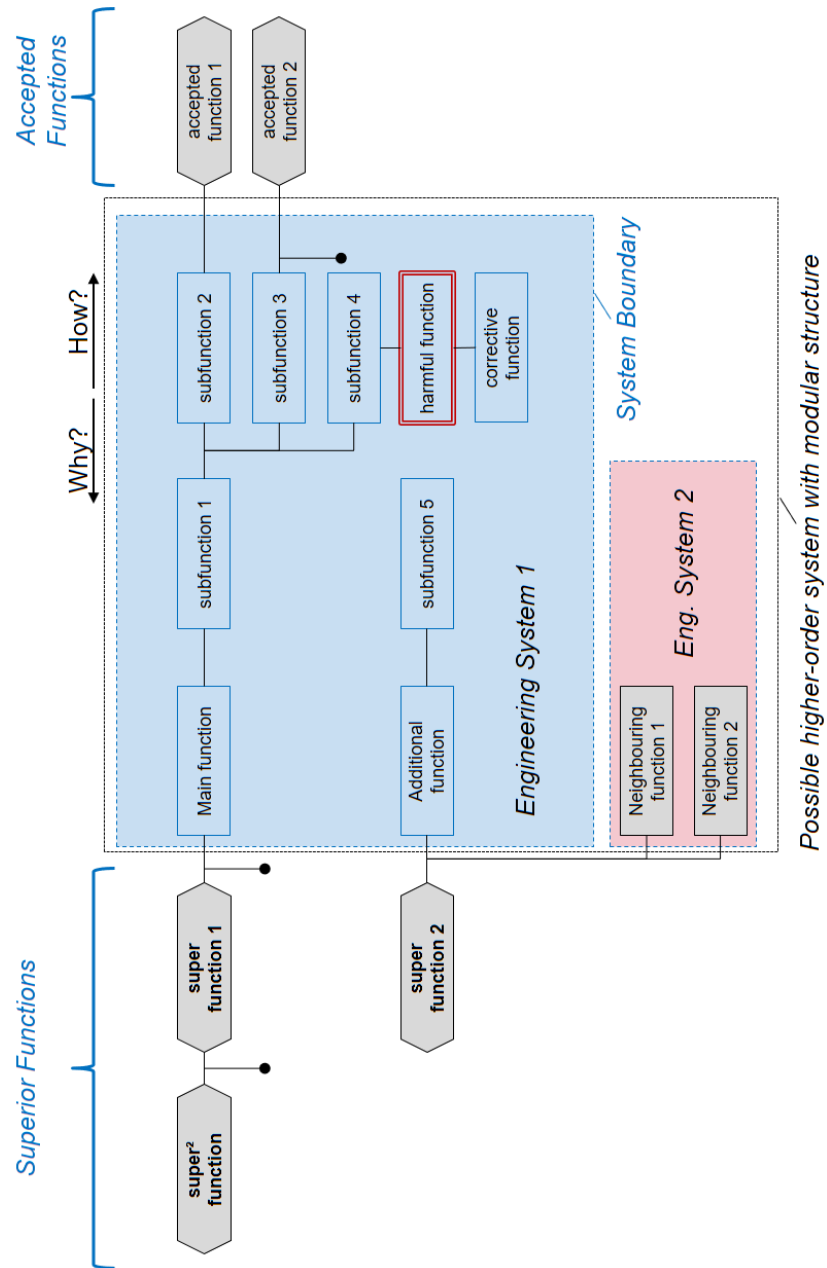[6] … which follows from its definition „man-made assembly […] which meets a purpose" [11]

Fig. 6. A system performs several functions on its super system. On the other hand, it contributes only partially to super functions

its super system. The concept of neighbouring functions is important for matching their parameters, in order to increase the functionality of the ES (neighbouring functions are prone for integration into the ES), and to organize ES's into systems of ES's. This latter point is required when a product portfolio shall be defined or when an ES is to be organized into modules.

One tool of "classical" (Altshuller's) TRIZ is the system operator. The system operator compares a system at different times and may be used to estimate the future evolution of the system. In addition, several levels in system hierarchy are considered: The system itself, its subsystems, and its super system(s). We suppose that the diagram drawn for different situations may help clarify the interrelation of ES and super systems. However, more experience is needed on this subject.

4.4   Illustrating and explaining the Laws of Evolution of Engineering Systems

G. Altshuller formulated laws that governed the way engineering systems evolve over time. According to Schollmeyer [29] and Karasik [30], these laws resulted from Altshuller's understanding of Soviet books on the history of technology (first of all aviation technology) of his time and reflect the principle of dialectics in the system of Marx and Engels. To our knowledge, Altshuller does not explain why, in what limits, and how exactly he expected these laws to apply.

The first of these laws is the "*law of the completeness of parts of the system*" stating that "*each technical system must include four basic parts: an engine, a transmission, a working organ and an organ of steering*" [3].[7] In the scheme of GenTRIZ, an engineering system acquires these functional elements in the order of *Operating Element – Transmission – Energy Source – Control* [19].

Obviously, the first stage in the evolution of an engineering system (tool) is characterized by a situation without this tool, i.e. naked man who provides solely of his body to perform the functions he desires, fig. 7. Improving the system will include first absorbing the operating / working function, and consecutively the remaining ones.

As an example, fig. 8 reproduces Hodler's painting of a lumberjack who constitutes in combination with his axe a complete (although, as a human, not necessarily an engineering) system. Please note that the axe itself – though incomplete – is a functioning system as well, provided that it can borrow the missing functions from its super system. We consider this observation important since it draws a link to well-adjusted systems of ES's and touches the aspect of how much functionality an optimally fitted system should contain.

---

[7] V. Hubka mentions in "Theory of Technical Systems" [31] that J.-A. Borgnis in 1818 [32] distinguished six classes of machine elements (engines, transmissions, transformers, supports, control, and working organs). K. Marx, whom Altshuller might have referred to, mentions only three elements (motor, transmission, and working organ) [33 p. 261].
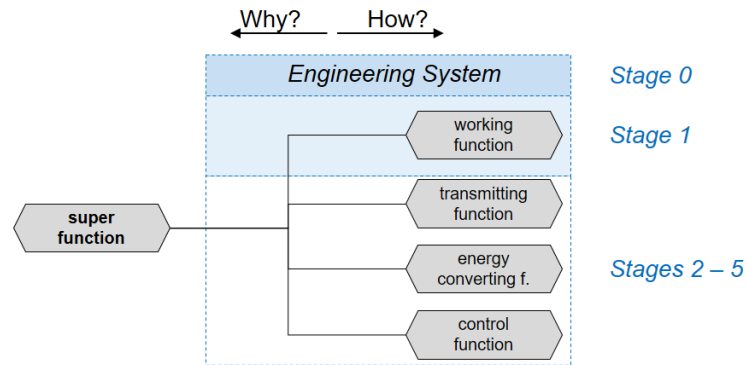
Fig. 7. An engineering system evolves from a situation before its creation (man performs all required functions with his body) until a complete system, gradually absorbing all the functions required.
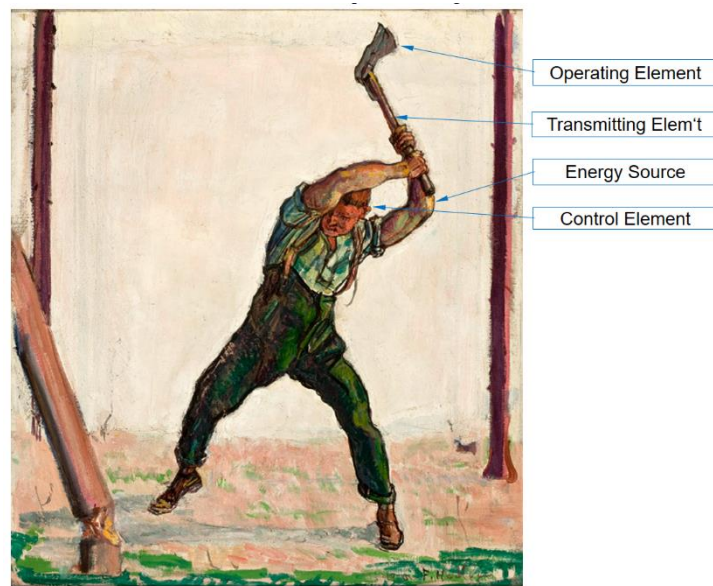
Fig. 8. Complete system consisting of axe + lumberjack. The axe itself will be a working system if it borrows the functions of energy conversion and control from its super system [using 34].

4.5 Increasing Product Functionality, Updating the Product Portfolio

One way to increase the value of an ES is to increase its functionality as is recommended especially in the first phases of the S curve [35 p. 168; 19; 36]. V. Prushinskiy recommends to merge (hybridize) the ES with an alternative one that performs the same

main function [37]. We would enlarge the field of search for hybridization partners to systems contributing to the same super function, i.e. systems that perform neighbouring functions (neighbouring systems).

With value V = total functionality F / total cost C [19], cost must be kept low in spite of rise in functionality, so synergy must be used [38] and the new products to be integrated into the portfolio must be similar to the existing ones. From the viewpoint of an engineer, similarity means technical or technological similarity. From the marketer's viewpoint however, similarity means similarity of market. The highest similarity of market is achieved with products serving the same super function, i.e. the same need.

Likewise, new functions to be integrated into a product must use technical synergies and/or serve a known market – if possible, the same marked as covered before.

The network of functions displayed by the FAST diagram supports both identification of neighbouring functions serving the same need / super function and technical synergies. We have therefore varied the functions and extended the notion of neighbouring ones, Fig. 9. This diagram uses concepts of systems with varied parameters and competing systems that Ikovenko presented with the GenTRIZ system of evolutionary laws, first of all the *law of transition to the super system* [19]: According to the first subtrend, the parameters of the integrating system become increasingly different from those of the Engineering System.

Let us explain the scheme given above using again the example of a hammer, Fig. 10. With "attach board" being the super function, "detach board" is the anti super function. Both are required to fulfil the secondary super function. However, the simple hammer does not serve "extract nail" which a gripper is needed for. Other functions required for attaching boards are neighbouring function "hold nail", driving a small nail with lesser force (varied parameter, lesser hammer), and "drive screw" (competing function) together with their respective anti functions of "let nail go" and "extract screw".

On the input side, the user will accelerate and direct the hammer at which activity he mail fail, requiring a correcting function, or which he may perform in an opposite way.

In accordance with the law of evolution, the diagram now suggests extending the functionality of the hammer "horizontally" by including the functions of extracting nail, holding the nail, driving with reduced force, or driving screws. "Vertical" integration would mean to have some of the accepted functions included, e.g. accelerating and directing the tool. Vertical integration may as well extend to the "why" direction, realizing a complete system with "attaching boards" as its main function.

Search for technically similar products is done by regarding the functionality of the ES and searching for alternative uses, e.g. using Reverse Function-Oriented Search.
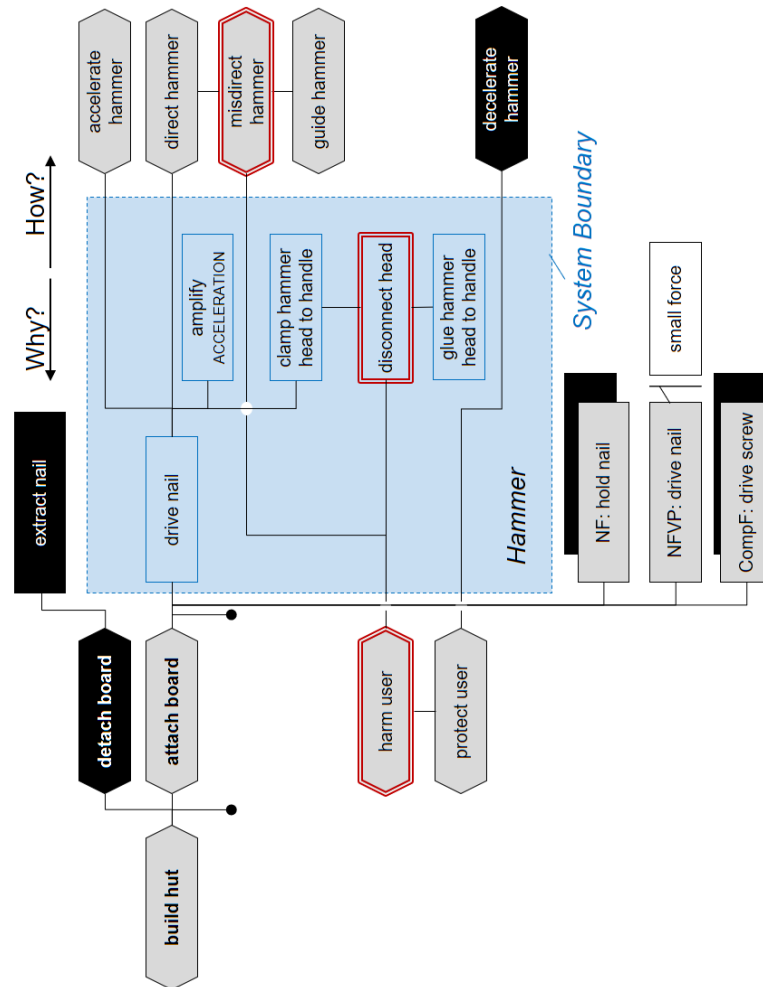
14

Fig. 9. FAST diagram with varied functions, see text

Fig. 10. Varied functions on hammer example. Example is shown for main function, but can be applied to any subfunction.

4.6 Finding Threats of Substitution

An ES is a means to fulfil a need (its super function) and it is created in order to achieve the need using the available resources efficiently [39 p.23]. The FAST model illustrates this situation by arranging the ES between the needs on the "why" side and the resources (via accepted functions) on the "how" side. One may imagine a flow of resources streaming from "how" to "why" into the fulfilment of needs. Since there use to be competing systems, the main functions of which deliver an alternatively viable

means to fulfil the need, the ES may lose the race in favour of its competitor, and vanish. Examples for this are the steam engine or the mechanical typewriter.

It must be kept in mind that the FAST model represents first of all a model of functions into which we draw the allocation with systems. Basically, the model is self-similar: What has been said above may be applied to any super or subsystem, as big or small as it may be. Therefore, in general, every subfunction has its neighbouring, competing etc. functions, which may replace it, thereby eliminating all its direct subfunctions. Likewise, a super function may vanish as it is replaced by a competing one[8]. The model thus permits consideration of alternative ways to fulfil a need and to monitor threats of substitution. What is still missing however, is the assessment of the risk, see discussion.

5   Conclusion and Discussion

We have proposed for application in TRIZ a variant of the FAST model and diagram that omits changes which had been introduced for specific needs of Value Engineering compared to Bytheway's original approach. On the other hand, we added concepts of function which result from considerations within TRIZ. The resulting scheme depicts a landscape of functions which is associated with the boundaries of engineering systems. The scheme is self-similar and applies to all hierarchical levels of functions.

The advantages of the model are seen primarily in product development: Similar to the flow-oriented model it permits to enter specifications of functions and interface parameters as well as demarcation of system levels – at least as long as they correspond with the functional hierarchy. In addition, it can present a map of needs and their fulfilment (a market-related view) and correlates this with the functionality of a product. The optimal set of functionality can be checked using a list of potential functions:

- standard functions (acc. to Altshuller [3], Borgnis [32], Linde [40, p 264], Bettencourt [41])
- anti functions
- competing functions
- parameter-varied functions
- harmful together with failure functions and their corrective ones
- neighbouring functions
- higher superiour functions

The developer should decide if incorporating these functions or trimming and borrowing them from super system would increase the value of the ES.

The aspects of trimming the model and optimal design of functionality, linking to and optimizing the super system as well as possible mechanisms of laws / trends of system evolution have not yet been treated in this paper and remain for later research.

---

[8] it is an open question if the „final" super function can vanish or not. Since it constitutes a basic human need, it is expected to last as long as mankind.

What we mentioned are changing conditions in the life cycle of a system. Souchkov pointed out that due to changes in the super system – super functions vanishing or resources running dry – systems may face becoming obsolete [42]. Evidently, a time-dependent format of the model would be desirable.

The FAST model as described here displays functions, however for strategic system planning, ideality or value is a needed information to assess future development. A task to be solved in the future therefore is how to integrate the aspect of cost. A vague idea may be to compare the flow of resources through the system with an electrical circuit or "potential problem" with cost being represented by the respective electrical "function resistances".

References

1. Akiyama, K.: Function analysis. Cambridge, Mass.: Productivity Press, 1991.
2. Zlotin, B. and Zusman, A.: [Introduction of the concept of function in TRIZ; private email]. Sent: 2018-06-06.
3. Altshuller, G.S.: Creativity as an exact science. New York: Gordon and Breach, 1987.
4. Souchkov, V.: A Brief History of TRIZ. In: The TRIZ Journal (2015). Available from https://triz-journal.com/a-brief-history-of-triz/ [accessed 2018-06-01].
5. Gerasimov, V.M. et al.: ТРИЗ-ФСА - Методические рекомендации 1990 [TRIZ-FVA – Methodologic Recommendations 1990, online]. St. Petersburg: MATRIZ. Available from http://triz-summit.ru/triz/metod/204080/204081/, updated 2006-05-01 [accessed 2018-06-01; in Russian].
6. Gerasimov, V. and, Litvin, S.: Main principles and postulates of the methodology for the performance of the TRIZ based value engineering analysis. In: *The TRIZ Journal* (1992), 3.2, pp. 7–45.
7. Zusman, A. and Terninko, J.: TRIZ / Ideation Methodology for Customer Driven Innovation. [cited after [8]]. In: 8th Symposium on Quality Function Deployment, The QFD Institute, Novi, MI, June 1996.
8. Terninko, J., Zusman, A.V. and Zlotin, B.L.: Systematic innovation. Boca Raton: St. Lucie Press, 1998. Available from http://www.gbv.de/dms/hbz/toc/ht008711909.pdf [accessed 2018-06-01]
9. Standard prEN 12973:2017.: Value Management. Berlin: Beuth, 2017
10. Bytheway, C.W.: The creative aspects of FAST diagramming. In: Society of American Value Engineers (ed.): Proceedings of the SAVE Conference 1971. Miami, Florida, May 24-26, 1971. Smyrna, Ga., 1971. (SAVE proceedings, 6), pp. 301–312.
11. Hiltmann, K.: The Fast Diagram for TRIZ. In: Mayer, O. (ed.) TRIZfest - 2018. The 14th International Conference. Lisbon, Portugal, 2018, p. 136-147. ISBN 978-0-942442-95-3.
12. Standard VDI 4521-1: 2016-04: Inventive problem solving with TRIZ - Fundamentals, terms and definitions. Available from www.vdi.eu/4521 [accessed 2018-06-01]
13. Bruns, M.: Systemtechnik [Systems Engineering]. Berlin: Springer, 1991.
14. Standard EN 1325: 2014-07.: Value Management - Vocabulary - Terms and definitions. Berlin: Beuth, 2017
15. Bolton, J.D. et al.: Function Analysis Guide: A Supplement to the SAVE Body of Knowledge. Saline, MI: Interspiral Press, 2016.
16. Standard VDI 2221-1: 2018-03.: Entwicklung technischer Produkte und Systeme – Modell der Produktentwicklung [Development of engineering products and services – model of product development]. Available from http://vdi.eu/2221 [accessed 2018-06-01]

18

17. Ponn, J. and Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte [Concept development and design of engineering products]. 2nd ed. Berlin: Springer, 2011.
18. Stetter, R., Assmann G. and Viertlboeck, M.: Functional Product Modelling – New Methods for the Generation of Product Functions". In: Culley, S. (ed.): Design research, theories, methodologies, and product modelling. 13th International Conference on Engineering Design: ICED 01: 21-23 August 2001, Glasgow, UK. Bury St Edmunds: Institution of Mechanical Engineers (WDK publications, 28), pp. 493–500.
19. Ikovenko, S.: MA TRIZ Level 3 Training 2013/14. AT – Wolfsberg: Jantschgi C&R, 2014.
20. Standard VDI 4521-2: 2016-12: Inventive problem solving with TRIZ - Description of objective, problem definition, and prioritisation of solutions. Available from www.vdi.eu/4521 [accessed 2018-06-01]
21. Standard VDI 2222-1: 1997-06.: Konstruktionsmethodik - Methodisches Entwickeln von Lösungsprinzipien [Methodics of design – methodic development of basic solutions]. Berlin: Beuth, 1997.
22. Pulm, U. and Lindemann, U.: Enhanced Systematics for Functional Product Structuring. In: Culley S. (ed.): Design research, theories, methodologies, and product modelling. 13th International Conference on Engineering Design: ICED 01: 21-23 August 2001, Glasgow, UK. Bury St Edmunds: Institution of Mechanical Engineers (WDK publications, 28), pp. 477–484.
23. Koller, R.: Konstruktionslehre für den Maschinenbau [Design theory for mechanical engineering]. 3. ed. Heidelberg: Springer, 2011.
24. Ehrlenspiel, K. and Meerkamm, H.: Integrierte Produktentwicklung [Integrated Product Development]. 5. ed. München: Hanser, 2013.
25. Hentschel, C.: Unorthodox Use: TRIZ for Non-intended Product Use. *The TRIZ Journal*. Available from http://www.triz-journal.com/archives/2008/02/03/, updated 2008-02-04 [accessed 2018-06-01].
26. Visnepolschi, S.: How to Deal With Failures (The Smart Way): Anticipatory Failure Determination". Farmington Hills, MI: Ideation International, 2008. Available from http://where-innovationbegins.net/build/publication.asp [accessed 2018-06-01].
27. Standard VDI 4521-3: 2018-x: Inventive problem solving with TRIZ - Problem Solution. Available from www.vdi.eu/4521 [to appear by 2018-12]
28. Feygenson, O.: Tutorial. Part #2: Elements of Advanced Function Approach. Pre-conference training session at TRIZfest 2017. MATRIZ. Krakow, 13.09.2017.
29. Schollmeyer, J.: Wie kam TRIZ zur Dialektik? [How did TRIZ get dialectic?]. [Lecture] Hamburg, University of Applied Science, 2019-02-01.
30. Karasik, Y B.: Why was TRIZ born in Baku? [Editorial] [online]. In: The Anti TRIZ-Journal, 2010, 9 (2). Available from http://www3.sympatico.ca/karasik/why_was_triz_born_in_ baku.html
31. Hubka, V and Eder, W.E.: Theory of Technical Systems. 4. print. Berlin, Heidelberg: Springer, 1988. ISBN 978-3-642-52123-2.
32. Borgnis, J-A.: Traité complet de mécanique appliquée aux arts. Vol 1: Composition des machines. Paris: Bachelier, 1818.
33. Marx, K.: Capital. Moscow, USSR: Progress Publishers. Volume 1 Book One. Available from https://www.marxists.org/archive/marx/works/download/pdf/Capital-Volume-I.pdf, [accessed 2019-07-28].
34. Hodler, F.: Lumberjack. 1910, São Paulo Museum of Art [online], [accessed 2016-05-10], Available from https://commons.wikimedia.org/w/index.php?curid=7018049
35. Fey, V. and Rivin, E.I.: Innovation on demand. Cambridge: Cambridge University Press, 2005.

36. Abramov, O.Y.: Product-Oriented MPV Analysis to Identify Voice of the Product. In: The 12th MATRIZ TRIZfest-2016 International. July 28-30, 2016, Beijing, China, pp. 1–12.

37. Prushinskiy, V.:Tutorial: Part 1 - Basics of Hybridization. Pre-conference training session at TRIZfest 2017. MATRIZ. Krakow, 13.09.2017.

38. Welge, M.K., Al-Laham, A. and Eulerich, M.: Strategisches Management [Strategic Management]". 7th ed. Wiesbaden: Springer-Gabler, 2017.

39. Hubka, V. and Eder, W.E.: Theory of Technical Systems. 4th ed. Berlin: Springer, 1988.

40. Linde, H and Herr, G.: World Opening Innovation Strategy and the Contributions of Altshuller. In: C. Rizzi [ed.]. Proceedings of the 10th ETRIA World TRIZ Future Conference. Bergamo: University of Bergamo, 2010, S. 257-264.

41. Bettencourt, L.A. and Ulwick, A.W.: The Customer-Centered Innovation Map. In: Harvard Business Review, 2008, 86 (5), 109-114.

42. Souchkov, V.: The Trend of Functionality Evolution. In: Aoussat A. et al. (ed.). Proceedings of the 13th ETRIA World TRIZ Future Conference 2013. Paris, 29th - 31st Oct 2013. ETRIA. Langres (FR): La Manufacture (No. Imprimeur 13814), p. 779–788. Available from http://www.xtriz.com/publications/Souchkov_TrendFunctionalityEvolution.pdf [accessed 2018-06-01]