Solving problems with TRIZ and AIPS-2015

Hans-Gert Gräbe, InfAI Leipzig
Middle German Competence Center on Systematic Innovation
https://infai.org/systematische-innovationsmethodiken

October 25, 2021

Contents

1 TRIZ as Strong Inventive Methodology

1.1 TRIZ and the World of Contradictions

The TRIZ[1] world is about contradictions in real existing technical systems or technical system to be designed. A technical system has always to be considered in the unity of its (mental) description and (real world) operation. This unity is of dialectical nature in the sense of [2] – in the description form unity in diversity plays the central role, in the execution form the diversity from unity has prevalently to be recovered. This abstract formulation is to be understood as follows: From diversity abstract *technical principles*[2] are derived and condensed in techno-scientific *procedures*. In the executive form, on the other hand, *several* such procedures are used in interaction to solve a real-world technical problem. However, the latter is also accompanied by a description form, a *description form of second kind*, which is different and has to be distinguished from description forms of first kind for domain-specific technical principles, since they are about the *interplay* of the domain-specific technical principles in a real-world technical solution and hence more of *methodological* nature or at a *framework level*.

This distinction carries through up to professional profiles – specific technical procedures are developed by specialists, real-world technical solutions by generalists, see in detail [2, section 9]. However, the world is not so dichotomously structured, but rather fractal. Specialists in one perspective may as well be generalists in another perspective, when they are faced with a domain-specific problem that requires cooperation with specialists from other domains to solve it.

In this sense, TRIZ is a methodology for generalists, who consult or support specialists on individual questions. This team-player approach is very important for the successful practical application of the methodology. The ability to work in a team counts as an important soft skill in modern co-operate actions. It's one of the weaknesses of TRIZ that such aspects are hardly elaborated. TRTL[3] as a part of the overall TRIZ Body of Knowledge (even if not included in [6]) is based on the central concept of a *creative personality* and sees her in an important leading position for the whole process of problem solving. In TRTL conflicting situations with other stakeholders are considered more from a private-psychological («heretical») and less from a structural perspective.

1.2 TRIZ as Problem-Solving Methodology

The TRIZ methodology is above all a problem-solving methodology where standard solutions or simple engineering approaches do not work. The reason for failure is usually an *obstacle* that stands in the way of such a simple solution and can be formulated as *contradiction* (Figure 1).

[1]TRIZ is an abbreviation of the Russian notion «Теория Решения Изобретательских Задач» – Theory of Inventive Problem Solution.

[2]This term *principle* is not to be confused with the same word in the connotation as *TRIZ principle*, which is an unfortunate translation of the Russian word «Прием», that would be better translated as *procedural pattern*.

[3]This acronym is in use also in English TRIZ literature and stands for «Теория Развития Творческой Личности» – Theory of the Development of a Creative Personality.

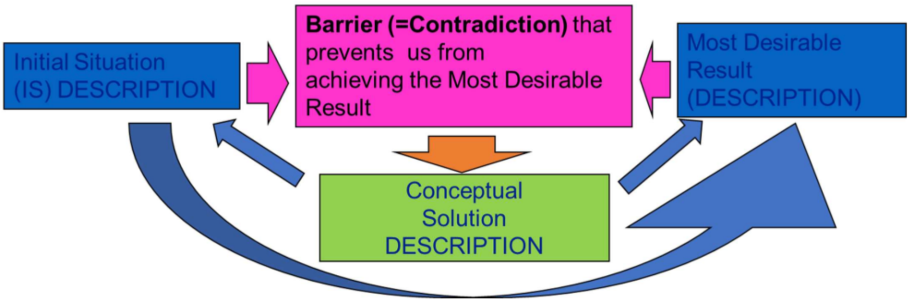

Figure 1: The TRIZ Process Model

In the course of modelling, this contradiction is often identified as a *conflict* where a beneficial (from the point of view of the system's purpose) effect cannot be achieved without a harmful accompanying effect, as well as the *operational zone* (in space and time) in which the conflict occurs. In the TRIZ approach, the attempt is not to resolve such conflicts by compromises ("or ... or ..."), but to arrive at principal innovative approaches ("... as well as ...").

Example: A tea glass with hot tea. Useful effect: Hot tea in the glass has good taste. Harmful effect: When yuo touch the glass you burn your fingers. The compromise solution *lukewarm tea* does no one satisfy. With TRIZ we analyse where the conflict occurs (on the wall of the glass when lifting the glass to drink). Here as typical solution approach the Separation Principle can be applied: Can the whole system be separated in space or time? Function of the glass: it is a container for the tea, so something can only be changed with the hand. Find an additional instrument (X-element) with appropriate properties. Touch the glass with a glove (which has a insulating effect), or with a pair of grill tongs (keeps distance). Or use a tea glass holder (perfects the idea with the grill tongs). Or paste the handle of the tea glass holder to the tea glass (more perfect spatial separation on the tea glass itself; "trims" the tea glass holder). Or analyse more precisely: the glass wall must be *inside* hot and *outside* cold. Make the glass as container out of heat-insulating material – the Coffee-To-Go-Cup is invented.

The TRIZ methodology is based on the Hill Model (Figure 2).

1) Modelling of the given problem (structure and processing of the system, spatio-temporal delimitation of the problem region in the model as *operational zone*, identification of the contradictory structure of the problem).
2a) Identification of the abstract problem structure, the available resources and selection of suitable TRIZ tools for a solution (*Problem Model*).
2b) Application of the tools to develop an abstract solution (*Solution Model*).
3) Instrumentation of the abstract solution with appropriate resources and deriving one or more concrete proposals for a solution.

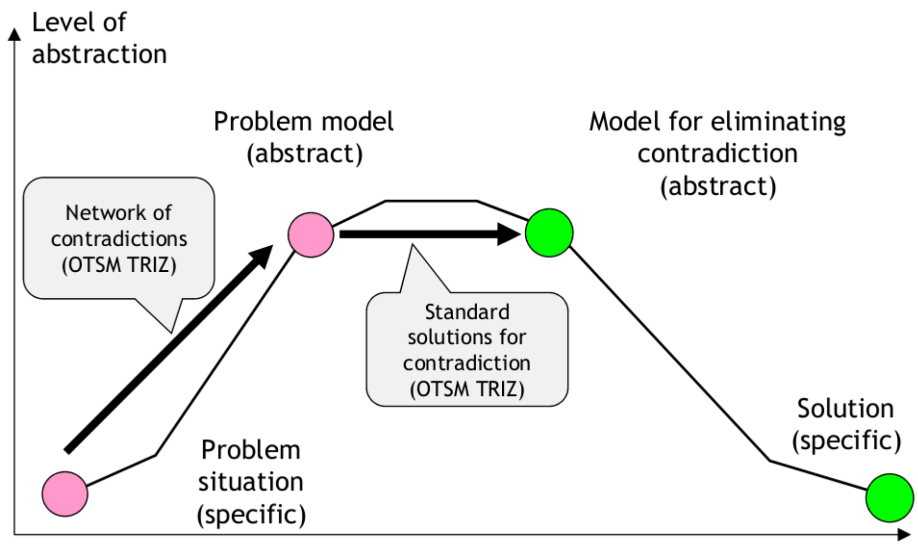

Figure 2: The TRIZ Hill Model

1.3 OTSM-TRIZ

OTSM-TRIZ[4] as a *General Theory of Powerful Thinking* goes beyond this and develops methodological tools for thinking in terms of development concepts in which contradictions can be dialectically resolved as we move forward. Thus contradictions get solved, that seem unsolvable in a fixed, conservative-static world. This kind of thinking is an important skill for dealing with in situations of disruptive changes, be it the *digital transformation* or more comprehensive processes of changes towards *more sustainable modes of production and living.*

1.4 The TRIZ System Concept

An appropriate concept of a system is central to understand the methodological approaches of TRIZ: it is used to *delimit an analysable part* of the complex all-connected reality and to make this part accessible to analysis by *reduction to essentials*. Both – demarcation and reduction to essentials – are not arbitrary, but a highly sophisticated task that is oriented at the concept of steady-state equilibria in the Theory of Dynamical Systems. The aim of systems analysis from such a perspective is to investigate the interplay of *viable components* in a *delimited context* in order to produce a new, *emergent* functionality that results from this interplay only. Such a functionality is also called *synergy effect* and demonstrates why the whole is more than the sum of its parts. The context is essential for the viability of the whole system by guaranteeing a *throughput of substance, energy and information* that drives the synergy effect.

The prerequisite of the existence of viable components is part of the reduction to essentials. In many cases it turns out that problems in the system have their root in the problematic behaviour of such a component. Since the description concept is self-similar, in such a case

[4]OTSM is an abbreviation of the Russian ≪Общая Теория Сильного Мышления≫ and marks a special branch of TRIZ theory that we follow. See [5], [3] for more information about OTSM-TRIZ.

the problematic component can be analysed in the same way as a system, with the previous system stepping into the role of the context as upper system.

In the TRIZ Trainer [9] these relationships are reduced to the informal determination of the *purpose* and *main useful function* (MUF) of a previously delineated problematic technical system as determination of its specification using a black-box approach. Following an approach of autonomy of the technical system under investigation, its (external) operating conditions are largely left out. The (methodical) approach can therefore not necessarily be transferred to components, since according to the evolution law of *energy conductivity*[5] the flow of substance, energy and information through all of its components is an essential property of the viability of a technical system.

The tasks of the TRIZ Trainer therefore basically refer to a technical system with a certain autonomy status (boat in water traffic, truck in mining, racing driver in a desert rally, etc.). On the other hand, since TRIZ works with a precise localisation of contradictions, the component structure of the technical system (*main system*) has to be analysed in more detail, including the analysis of sub-components of different hierarchical levels up to the localisation of the problematic component and the operational zone.

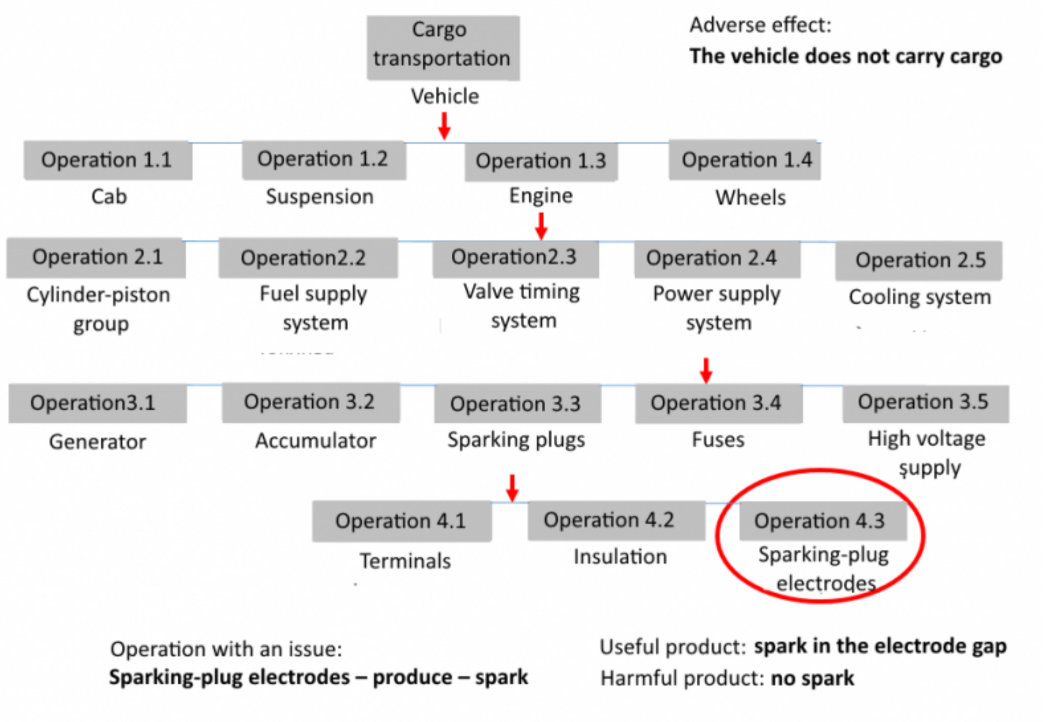

Figure 3: Identification of the problematic component in the hierarchy of system components (from [9])

[5] "The continuous flow of energy through all parts of the system is a necessary condition for the basic viability of a technical system" [1, p. 125]. "The basic prerequisite for the viability of a system is the free flow of energy through all its parts." [4, p. 172]. Somewhat differently [7, p. 86]: "The trend of Flow Enhancement is much more sophisticated, and it also takes substance and energy into account." Recent versions also take flows of information into account.

Furthermore, the processual organisation (*How the machine works*) of both the technical system and the problematic component have to be analysed. The processual analysis of the main system shows which resources are occupied and thus are primarily available or can be reallocated for problem solving. The processual analysis of the problematic component shows the structure of the conflict and is the primary target of a "classical" TRIZ analysis. The processual analysis of the main system is, however, also helpful to develop an appropriate notational framework for the analysis of the problematic component, especially if, in the course of the workflow analysis of the technical system, it turns out that there are different states (operating state, maintenance state, etc.), which appear as contextualising conditional patterns in the determination of the operational time and thus *separation by change of conditions* [4, p. 111] can be applied.

2 TRIZ Trainings

TRIZ training is about methodical support to these generalists at work. In **advanced TRIZ training** you learn to handle complex requirement situations with a larger number both of components as well as contradictions. Tools as *Functional Analysis*, *SuField Modelling*, *Cause-Effect-Chains* (CECA), *Root Conflict Analysis* (RCA+), *Trends of Engineering System Evolution* (TESE) etc. are to be applied in a complex landscape of contradictions to identify a starting point and transformation ideas for the solution of a given problem, where the skills of problem analysis acquired in the TRIZ basic training are applied and extended.

Basic TRIZ training, the training objective of the TRIZ trainer [9], addresses the general task to eliminate contradictory behaviour through a precise analysis of a *single* critical system, without clear rules, *how* this critical system was identified in the first place. This (main) system has to be analysed along the methodological rules and patterns with the goal to propose one or several sound ways have to transform it into a system without the problematic behaviour.

Common to both forms of training is the perception of a pre-existing world of strongly interdependent technical systems. Such a picture is based on description and execution forms of *direct* interaction between such systems and largely ignores higher forms of abstraction such as framework models. This is due to the fact that the TRIZ methodology itself does not use abstractions of interactions at such a level of analysis. See [8] for basic considerations on higher abstraction forms of "re-use".

3 TRIZ Trainer – the Solution Process

In the TRIZ trainer [9], the algorithmic version AIPS-2015[6] of TRIZ is used, which is described in more detail in the *Help System* of the TRIZ Trainer. According to this methodology, the solution process is divided into three steps.

[6]AIPS is a Russian acronym for ≪Алгоритм Исправления Проблемных Ситуаций≫ – Algorithm for Problematic Situations Transformation.

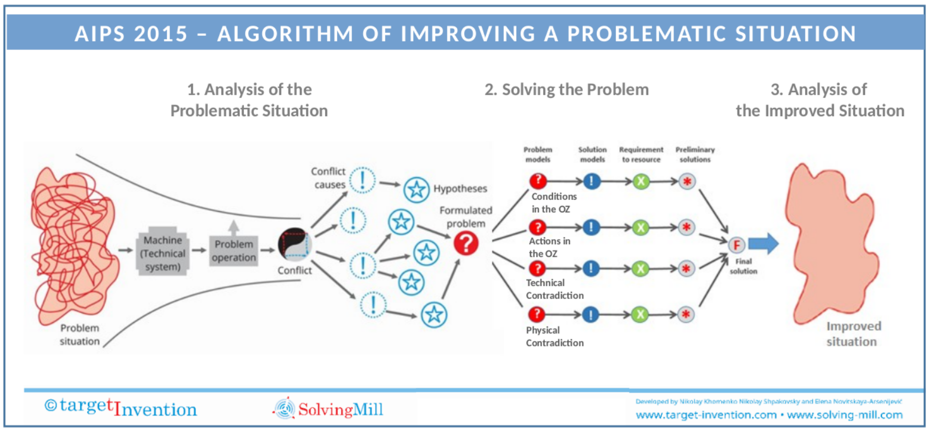

Figure 4: The AIPS-2015 Phase Model

1. Analysis and Modelling of the Problematic Situation. This phase includes

- the delineation and classification of the *problematic system*,
- the analysis of the *structural* and *operational* dimension of the problematic system (the "machine" in the terminology of the TRIZ Trainer),
- the delimitation of the *problematic operation* in the *operational zone*,
- the *analysis of the conflicts*, contradictions and causes that give rise to the problem, and the formulation of the *Ideal Final Result* (IFR),
- The formulation of *general solution hypotheses*, from which the one is selected for further investigation that can be expected to come closest to the IFR with minimal modification of the problematic system.

2. Analysis of the Selected Solution Hypothesis. This phase includes

- selection of one of the four *problem models* according to the specificity of the solution hypothesis,
- the selection of the *TRIZ tools* recommended for this problem model,
- the development of the *general solution model* through problem-specific application of the selected TRIZ tools,
- the derivation of the *requirements for the resources* needed for this solution,
- the formulation of one or more detailed *solution sketches*,
- the development of the most promising solution sketch into the *plan of the final solution.*

3. Analysis of the Improved Situation. This is an evaluation of the solution found with regard to the following questions

- Does the implementation of the solution plan solve the original problem?
- What effort is required to implement the solution in practice?
- Do new problems arise that cannot be accepted?

For didactic reasons, AIPS-2015 assigns only little importance to the third phase, since both the non-implementability of the solution and the occurrence of new problems usually require a return to the first analysis phase of the algorithm and thus the algorithm must be applied iteratively several times. Other TRIZ variants such as IDM therefore work with *networks of problems* and *partial solutions*. Such approaches are only addressed in advanced TRIZ courses.

3.1 First Phase of the Solution Process

In the **first phase**, the modelling of the problematic situation, it is required to identify

1. the *main system* (using a "speaking name"), its *purpose*, the *MUF*, the required *operating conditions* and the *problematic behavior* to be eliminated (section *Specification of circumstances*),
2. the *structural organisation* of the main system (according to the pattern shown in figure 4) – which components and which resources are used, where is the problem concentrated, recursive analysis of the structural organisation of sub-components as in figure 3 up to the *operational zone* where the problem manifests itself – (section *Machine*),
3. the *processual organisation* of the main system (as preliminary work to identify resources that are available in the system to solve the problem) and of the problematic sub-component (section *How the Machine operates*).
   The processual organisation leads in many cases to a clear distinction of different *states*, which should be taken into account for optimal solutions as different modes of operation of the system. These states are to be clearly conceptualised and delineated in this part of the solution process.

Identification of purposes and system delimitation. The proper delimitation of the main system to be analysed in more detail is largely heuristic and accompanied by a first analysis of the internal functionality of this system as a *White Box*. At the end of this analysis, the useful as well as the inadequate or harmful effects are to be listed, if possible, as formalised statements using the template "tool – acts-on – object" (section *Conflict Structure*) and on this basis the conflict (place, time, structure) has to be described in more detail as a basis for further planning of a transformation of the system that solves the problem.

Examples:

- Ship mast: Purpose – transport on water, system – ship.
- Starting a heavy train: Purpose – freight rail transport, system – train.
- Tipper in mining: Purpose – transport of ore from the mine, system – tipper.

At the end of this first phase (in computer science also called *requirement analysis*) we have an exact model of the system. Further (section *Generation of Hypotheses*), on the basis of this model, a *precise abstract task* has to be formulated, the implementation of which would solve the problem.

In contrast to analysis methods such as *Design Thinking*, which are stronger focused on the needs of the customer but less on technical conditions for their realisation, systematic modelling develops a good understanding of the technical conditions and thus focuses the solution space to be analysed in a *justified way* in contrast to brainstorming, which leads in many cases to an *unjustified* focus on "traditional" solutions without inventive potential. This strong focussing effect of a TRIZ analysis on *practical* feasibility is repeatedly emphasised by users as its great advantage. With the formulation of the task, the *direction* of the solution is already focused at this point, even if the details still have to be worked out in the further process.

How "radical" a solution should be? As a rule, in the given use cases for a solution the system can be modified in such a way that the transformed system continues to fulfil its MUF as specified or only minor modifications have to be made, i.e. the transformation of the process organisation can be *locally limited* (eingrenzen) and restricted to the context of the system itself.

In other tasks, a temporary additional function of the system has to be designed, which is to be executed in a special state of the system. In this case, the analysis of the MUF is important because this function supports the identification of the resources available to the system that can (and should) also be used for the additional function.

How "arbitrary" is the delimitation of the main system? The delimitation of the main system represents a certain moment of arbitrariness of a separation of "inside" and "outside". It may turn out during further modelling that a different level of detail is more appropriate as main system. Then the modelling should be repeated at that level. More on this can be found in the *AIPS-2015* section of the Help System.

See also the explanations in [4, chapter 4.3] on the connection between technical and physical contradictions and the reconstruction of a technical contradiction from an (occasionally obvious) physical one in order to understand how the physical contradiction is embedded in the overall modelling of the system.

Of course, elementary knowledge of natural law concepts and technical interrelationships is assumed – e.g. interrelationships between different forms of energy, forces, moments, motion quantities, etc. – which you should be familiar with from school.

3.2 Second Phase of the Solution Process

For the specified hypothesis, in the **second phase of the solution process** one or more *solution ideas* are to be found by a precise analysis of the available resources. At the end, one of the solution ideas is to be elaborated in more detail as *final solution* and to be checked whether the solution also works.

This second phase includes the following parts:

(1) Selection of a *Problem Model* that fits the conflict structure identified in the first phase.
(2) Identification, as comprehensively as possible, of *resources* which fit the Problem Model and the conflict structure.
(3) Selection of an appropriate TRIZ tool for the *transformation* of the Problem Model to the *Solution Model*.
(4) Configuration of the tool according to the specific conflict structure.
(5) Instrumentation of the Solution Model with appropriate resources.

While (1) includes an essential methodological decision, steps (2)-(5) are closely related. A coherent picture of the instrumented Solution Model often emerges only after multiple steps back and forth, when later new insight requires the revision of earlier decisions. In most cases, it is discovered that the former modelling was too coarse or that essential aspects were overlooked. In such a case, the modelling must be revised from that point on.

It may also turn out during these refinements that the first phase was insufficiently elaborated or that the Problem Model is inappropriate. In this case, the deeper insight should be used to return to the first phase, the modelling there should be made more precise and the context, which is constitutive for the second phase, should be adjusted.

This part is described in very detail in the Help System.

3.3 The Third Phase of the Solution Process

At the end, in the **third phase**, one of the solution ideas is to be elaborated in more detail as the *final solution* and to be checked whether the solution works.

Evaluate the own Solution. TRIZ consultants usually leave the selection of the most appropriate solution to the client, as the decision often includes additional requirements that arise from internal company processes. Therefore, the consultant does not focus on finding a single strong solution that is likely to be difficult to implement under certain conditions, but offers a range of (proven) solutions from which the client can combine the most suitable, locally ideal solution.

This approach is also followed in the TRIZ trainer, even though here we do not have a *client* as the source of the problems and to evaluate the solutions. Accordingly, the best solution can only be selected from several to a limited extent – the one that appears to be the best based on general experience and logic. If you arrive at several proposed solutions, in the last steps of the submission (conclusion, final decision) you can highlight the solution that you think works best and mark it (with justification) as the most effective one. The evaluation criterion should be the extent to which the solution requires changes to the existing system – optimal solutions are those that require only minor changes at the place of the conflict or its vicinity.

How precisely must the final solution be worked out? The final solution must be worked out to such an extent that secondary problems are also solved. One cannot be satisfied with a solution that works only in principle, but which, if one tries to implement it, would run into further obstacles, which is expensive or complicated or to 99% unrealistic in terms of resource requirements. In this case, the solution is returned for revision with the following two options:

- You remain in phase 2 and go through the solution cycle once again with the extended knowledge and the same hypothesis (iteration).
  Modify the Problem Model or use a different one, perform the solution steps again and submit the new solution for evaluation.
- You return to phase 1, approach the modelling differently, formulate a new hypothesis or select another one from the multiple hypotheses already formulated earlier, and then

go through phase 2 again with the new approach.

4 Additional Methodological Remarks

4.1 On the Basic Structure of a Problem in TRIZ

The central objective of the application of TRIZ is to solve a problem in a technical system that results from a contradictory constellation.

A *technical system* is an interaction of components, which (using resources) achieves a *purpose*, an *emergent function*, which is neither rooted in one of the components nor in the sum of all components, but results from the *interaction of the parts* of the TS ("The whole is more than the sum of its parts").

Of course, each component contributes its part, provided that the operating conditions of the component are guaranteed by the system. But decisive for the understanding of the functionality of such a system is the *modelling* of the interaction of its components. The modelling assumes that the components do their work, i.e. they behave in accordance with the specification, if their operating conditions are guaranteed by these interactions.

Problems with a component (or the whole system) require a more detailed analysis of its inner working. For this purpose a component is itself considered as technical system. The (methodological) modelling and analysis scheme is thus self-similar, but the analysis presupposes that the context of the component's operating conditions is given and fixed. The modelling of a system is thus always done against an *external context* of its purposes and operating conditions as depicted in figure 3.

4.2 Analysis of the Problem Situation

When searching for the cause of a problem, an experienced engineer (and also any digital diagnostic programme) starts from the main system level by analysing individual components and subcomponents step by step until the problem is localised (see Figure 3). This modelling is to be carried out as *analysis of the problem situation* in the first phase of the problem solving process in the TRIZ-Trainer. Both the main system and its problematic component are to be analysed. This means that, as a rule, *two* system analyses are to be carried out.

First, the main system must be delimited and its purpose determined. In a first step the main system is roughly delimited as a black box giving it a *name* (use a "speaking name") and its purpose is defined as *main useful function* (MUF). The MUF results from the reason why the system under investigation exists at all. This reason becomes apparent when the supersystem or neighbouring systems are identified for which the service of this system is significant or necessary. In many cases, the MUF is evident without such an extended analysis. As a third component, the *operating conditions* of the system are to be roughly fixed as context. In this way, the *specification* of the main system is (roughly) captured.

In order to identify the core of the problem, a more detailed analysis of the system as *White Box* is required. TRIZ starts from the concept of a *Minimal Technical System* and its basic structure

The tool — acts on → the object.

In the course of this action the *object* is transformed into a *useful product*. The (general) purpose of a technical system is thus to transform an object of work (Arbeitsgegenstand) into a useful thing.

This transformation, however, requires an energetic and controlling influence on the tool or, more generally, the means of work (Arbeitsmittel). For simple tools this influence is exerted directly by a human operator. In more complex contexts supply of energy and control is exerted by other components. In the TRIZ-Trainer the term *Machine* is used for such a more complex context in a generalised sense (see the Help System), whose *standardised structural organisation* is shown in Figure 4.

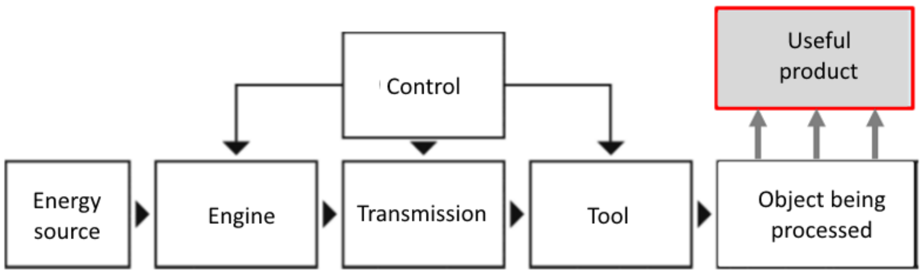

Figure 4: Basic structure of a “Machine”

For a system, in addition to the *structure* (static model), its *processual organisation* (dynamic model) is also important. Corresponding diagrams such as sequence diagrams, state diagrams, state transition diagrams, process chains etc. are helpful to represent such processes. The processing in the system connects process steps of the individual components, which are considered as *elementary* at the system level (and moved into subdiagrams in diagram representations). In addition to calling up functionality such process steps can cause *state changes* to processed common *objects*. In this sense, *components* are system parts with their own active functionality, *resources* and *objects* are passive targets or inputs of functional transformations. The concept of an object thus differs from that of OO programming and is close to the concept “beyond object oriented programming” as developed in [8], for example.

According to [7, p. 39], a reasonably complete system comprises the following functionalities (see also Figure 4):

- the functionality of the operating agent (working organ, tool),
- the energy supply (energy source or storage),
- the engine (conversion of stored energy into working energy),
- the transmission (transfer, adaptation and transport of working energy to the working organ) and
- the control.

4.3 External Components and Resources

When modelling the technical system under investigation as white box, it happens that it uses services from other systems via their interfaces. Typically, these are components of the technical system that are considered as a black box within the system modelling, the ability of which is described by an interface specification and whose practical performance is given by a specification compliant *throughput* through the system, which is constitutive for the viability of the technical system. Hence the *external* throughput keeping the technical system alive comes partially from the *inside* of the technical system.

When searching for resources with certain properties, for example as X-component, it is possible that external resources are subsequently to be integrated into the system modelling (in the sense of the evolution law of *increasing system completeness*). In a world of technical systems, these are neighbouring components that were not previously part of the technical system. Hence, in the course of system modelling, there is a natural process of transformation of neighbouring components into system components to be taken into account. Here, the structural similarity (descriptions of the of neighbouring components are available as black boxes in the same way as of internal components; and like them they can only be addressed via interface specifications) substantiates a merging of the two “concepts” – internal and neighbouring component – to a common notion of *potentially available components*, at least for modelling purposes.

Since, on the other hand, the modelling of the internals of the system in the section *Conflict Structure* of the first phase in the TRIZ-Trainer is anyway restricted to *essentials* (essential components and essential relations) according to a given methodological pattern (energy source, transmission, tool, ..., control), there is no problem to include *potentially all* neighbouring systems as components in the modelling of the technical system under investigation and to include the relations between them as relations between components. The originally largely arbitrary delimitation of the main system reads in this way as (initial) weighting of components and relationships according to a principle of “essentiality” given by the modelling purpose.

This is, of course, a formal step that is largely transparent to the TRIZ novice, which starts with the selection of those components and relationships in APIS-2015 that are *essential* according to the inner logic of the system. However, the methodological advantage is, that the system to be modelled *in such a modelling context* has no longer an outside and thus is more homogeneous.

In APIS-2015, a shell model is proposed for the search for resources to instrumentalise the solution model that includes step by step

1. the operative zone with tool and processed object,
2. system components in the environment of the operative zone,
3. system components at all (i.e. those already identified as part of the system in the previous modelling),
4. readily available resources from the environment (the *supersystem*),
5. machine components (which is redundant in a certain sense) and
6. environmental components.

The distinction between resources and components remains vague in that respect, resources being the more general term and also include “natural” resources, although every resource being useful in the system has a (even rudimentary) *description* of its “usefulness” properties in the form of a black-box model. In this respect, the terms differ only gradually.

For the search for resources, a good functional analysis [4, sec. 4.4] is important in order to describe the required characteristics of the resource as precisely as possible. In advanced applications, the opposite is also helpful, namely the exact knowledge of the properties of resources in an *effect database* as explained in [4, sec. 8.2] in more detail as well as the potential of a *function-oriented search* as described in [4, sec. 4.14] which specifically searches for more precisely specified functionalities of the same type in other technology areas.

References

[1] Genrich S. Altshuller (1980). Wings for Icarus. German translation: *Flügel für Ikarus.* Berlin 1983.

[2] Hans-Gert Gräbe (2020). People and their technical Systems (in German). LIFIS Online, Mai 2020. http://dx.doi.org/10.14625/graebe_20200519

[3] Nikolai Khomenko, John Cooke (2007). Inventive problem solving using the OTSM-TRIZ "TONGS" model. http://www.informatik.uni-leipzig.de/~graebe/Material/tongs-en.pdf.

[4] Karl Koltze, Valeri Souchkov (2017). Systematic Innovation methods (in German). Hanser, Munich.

[5] Andrei Kuryan, Dmitri Kucharavy (2018). The OTSM-TRIZ Heritage of Nikolai N. Khomenko. A General Theory of Powerful Thinking. Slides of a talk given at TDS 2018 in St. Petersburg. http://www.informatik.uni-leipzig.de/~graebe/Material/OTSM-Folien.pdf

[6] Simon Litvin, Vladimir Petrov, Mikhail Rubin, Victor Fey (2012). TRIZ Body of Knowledge. https://matriz.org/auto-draft/triz-body-of-knowledge-final/

[7] Alex Lyubomirskiy, Simon Litvin, Sergei Ikovenko et al. (2018). Trends of Engineering System Evolution (TESE). TRIZ Consulting Group. ISBN 9783000598463.

[8] Clemens Szyperski (2002). Component Software. Addison Wesley, Boston.

[9] TRIZ-Trainer. https://triz-trainer.com.