TRIZ Developers Summit 2014

Coordination and Integration of TRIZ tools

Using Enhanced Nested Function Models for Strategic Product Development

Author: Horst Th. Naehler, c4pi – Center for Product Innovation, Huenfeld, Germany

Co-Author: Barbara Gronauer, StrategieInnovation, Huenfeld, Germany

Abstract

TRIZ provides excellent tools for designing customized problem solving and product developing processes and algorithms. Strategic decisions can furthermore be strengthened by tools still in development and under research, as the Trends of Engineering Systems Evolution and their underlying mechanisms.

One of the difficulties when using TRIZ tools is the large number of possibilities for using and combining them for the best effect. Especially during the analysis phase, lots of information has to be gathered that lead to problem models and task definition for later problem solving.

To expand the usage of analytical tools for strategic decisions, this paper proposes an analytical approach that is based on nested function models to describe complex engineering systems.

This paper examines the interactions between components on different system levels, the use of the model in conjunction with trimming and the integration of the Multi Screen approach for developing a basis for strategic product development decisions. Furthermore the possibilities of connecting Trends of Engineering Systems with this approach are explored.

The suggested approach is aimed at creating an extensive, multi-level product map that combines the benefits of several classical TRIZ tools. It also creates a base for strategic decisions linked to problem solving opportunities on the operative level.

1. Introduction

One of the aspects that draw people to the Theory of Inventive Problem Solving is the analytical approach to inventive problems. Having a structured way of exploring an engineering system, reveal its shortcomings and extract current limits that prevent the system to develop further and thus increase the value for the user/customer, is one of the main benefits of TRIZ.

The experience from several industrial projects and feedback from practitioners brought up the wish to be able to assess a complex engineering system with Function Analysis without the need to focus on a certain system component and thus look at neighboring assemblies just as Supersystem-components or model the system on such a high level that the sub-assemblies are not assessed in greater detail. Instead of creating a number of single Function Models for each desired abstraction level an expansion of the Function Model is proposed, in which function models are nested inside each other to create a complete model of an engineering system from a high abstraction level down to each part on the lowest abstraction level.

Due to the nature of the proposed approach and the amount of work necessary to create such Nested Function Models for complex systems it is more likely to serve as a strategic tool. Thus, the combination with other strategic tools like s-curve analysis and the Trends of Engineering System Evolution (TESE) seems useful. The 9-screen-model serves as a starting point and a connecting, underlying structure for the creation of a product-map that integrates nested function models, s-curve analysis and TESE.

As a remark, this paper does not present something new. It proposes the combination and integration of well known and proven TRIZ- tools without violating the rules and directions for their application. The authors call on the expertise and experience of the TRIZ experts of the TRIZ Developers Conference to comment and constructively criticize the presented approach.

2. The 9 screen model as a generic product map

As a universal scheme to structure the composition of an engineering system in the context of its past and future, this classical TRIZ Tool is easily adopted by engineers. Its straightforward approach makes it extremely useful to accomplish numerous tasks from creating future product concepts and usage scenarios to identifying new business opportunities or make strategic decisions on an enterprise level.

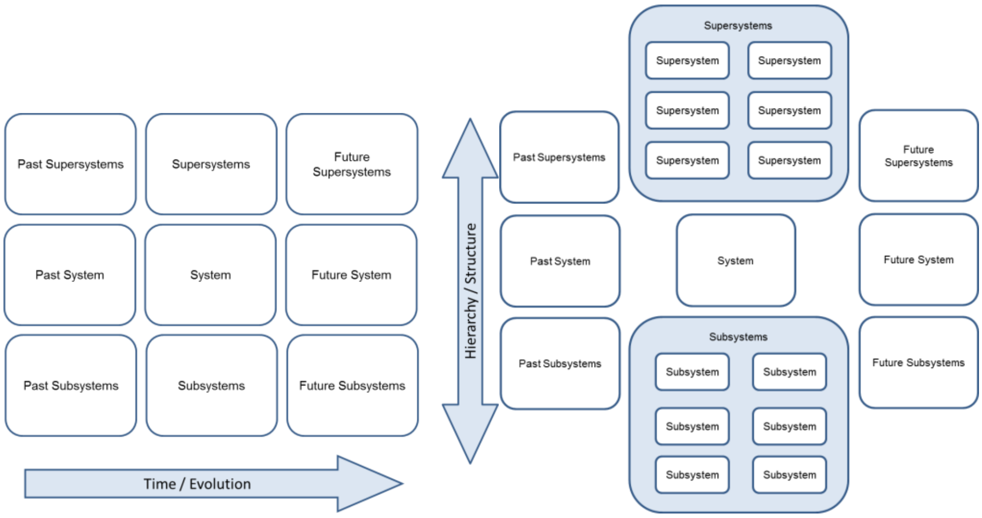

Fig.1: 9-Screen Model

2.1. The vertical Axis of the 9 Screen Model

The vertical levels of the 9 Screen model represent the structure of an engineering system with its Components (Sub-Systems) and defines it's boundaries to surrounding or neighboring systems (Supersystems).

The middle screen can be freely defined depending on the system that is to be assessed, so the 9 Screen Model very quickly gives clues when to refocus and re-define Sub-Systems and Supersystems. The system perspective and thus the definition of Supersystem and system components constantly changes:

Supersystems	A B C D
System	Product = System
Sub-System / Components	C1 C2 C3
Sub-Sub-Systems	C1.1, C1.2, C1.3…
	C2.1, C2.2,…
	C3.1, C3.2…
Sub-Sub-Sub-Systems	C1.1.1,…
	C1.2.1…
	C1.3.1…

It is obvious that the 9screen model can be expanded in the vertical to represent hierarchical structures, similar to assembly trees in CAD models. Under this aspect it can also be seen as a link to Function Modeling, for which we also need a clear definition of a system, its components and the super-systems.

2.2. The horizontal Axis of the 9 Screen Model

While the vertical axis shows the arrangement of the system , the horizontal axis represents the timeline related to either the system's history and future or the "Operating Time" during which the problem happens and the time before and after.

Using the 9 Screen model in strategic context, we can assess every system, big or small, and research and describe its history as well as the evolution of relevant super-systems and system components. So no matter how deep we go down into a product's hierarchy, each sub-assembly and part has its own history and future. This aspect enriches the approach of the Nested Function Model described below.

3. Nested Function Models

Usually the function analysis process begins with a component analysis, where the system-level and thus the abstraction level is defined. The criteria to make a decision on the choice of an appropriate abstraction level depends on the projects goals and the intended outcome. Problem Solving on the level of a sub-assembly or a single part of the system might result in a low level of abstraction and therefor a re-focus and re-definition of the term "system". Usually it is not recommended and not practical to model a complete System using all of it's single parts as components, as the resulting function model becomes increasingly cluttered and confusing.

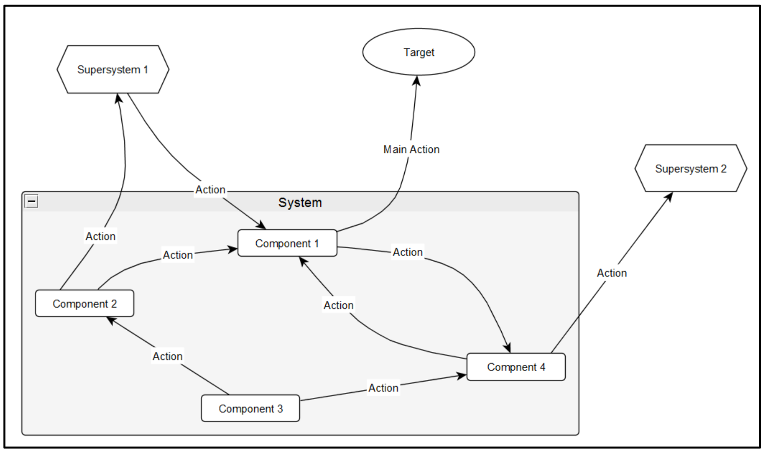

Fig.2: Generic Function Model Structure

To overcome this drawback, it is suggested to use the 9-screen philosophy and look at each component as a "box" that itself contains components, which might again consist of several sub-components. We are then able to build function models for each component and thus nesting them to generate a hierarchical structure. The use of software makes it easier to look into each component and assess the underlying function model.

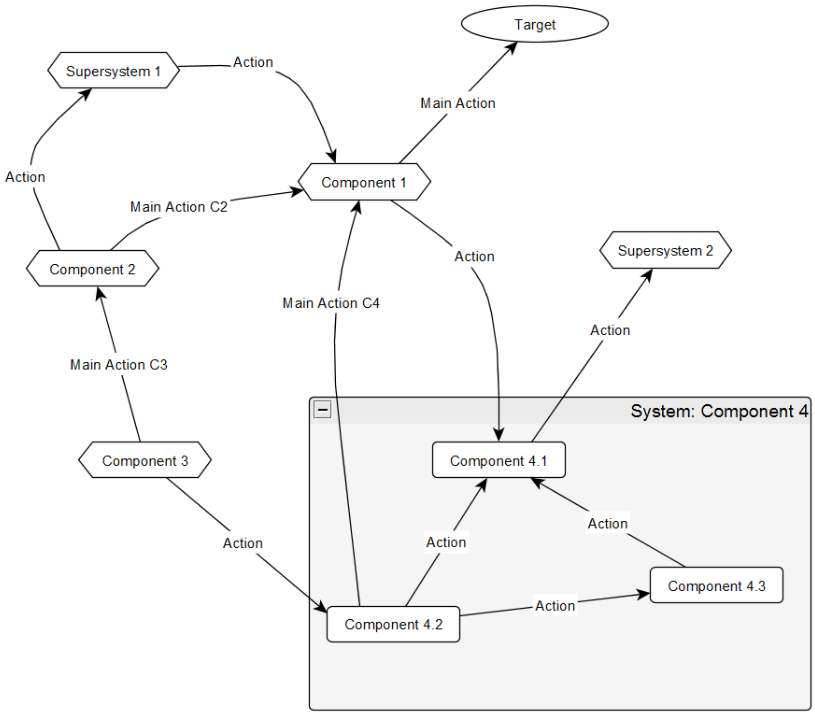

Fig.3: The Nature of Refocusing according to 9-Screen Model: When Component 4 is considered the System, each of the other Components become Supersystems and Parts of Component 4 become Components of the System

Following this process we can generate a complete product map without cluttering it, as we can chose which component or system to assess. We can also track down the components on a low abstraction level which are involved in the main function on a high product level. Furthermore it is possible to delegate the work of generating function models of sub-assemblies to different teams and then “plug” the single models inside each other to complete the product map.

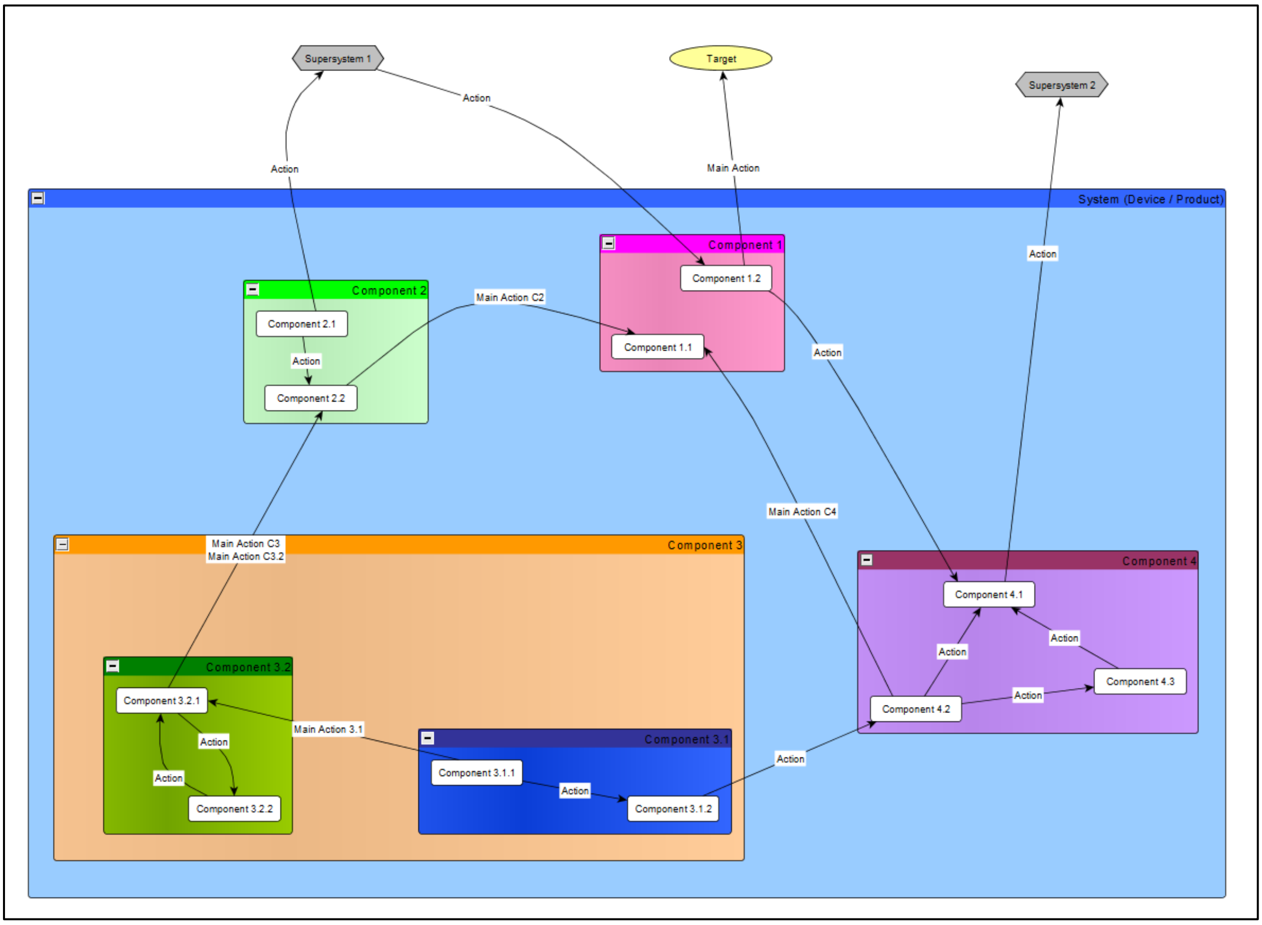

Fig.4: "Nested" Function Model for a Product with several Sub-Assemblies

3.1. Practical Considerations

When building Nested Function Models, it is not necessary to formulate new guidelines as all basic rules for TRIZ Function Modelling apply unchanged. Nevertheless, there are some aspects that are worth to be discussed:

3.1.1. Where to start the Model?

In function Analysis, the first step necessary is to define the system, it's components (Sub-Systems) and Supersystems. Of course, we can start the Nested Function Model on each hierarchical level, but it is recommended to start from the perspective of the Product, e.g. the whole product that is sold to customers. When dealing with systems on a bigger scale (e.g. a power plant) the question generally can remain the same: Which system generates an output that creates a value for the customer?

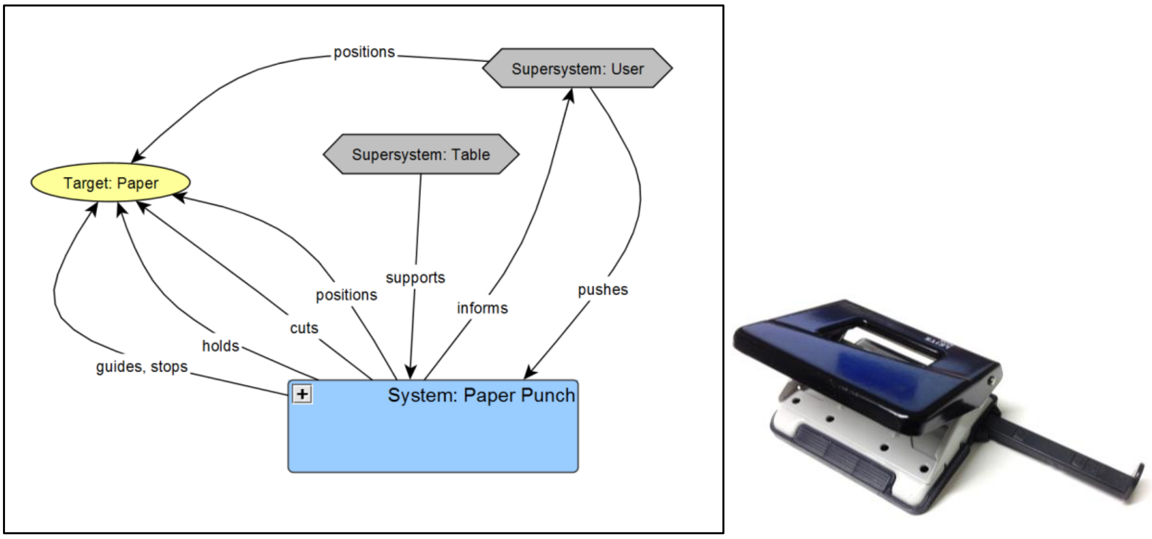

Fig.5: Starting Point on Level of Product / Device, Example Paper Punch

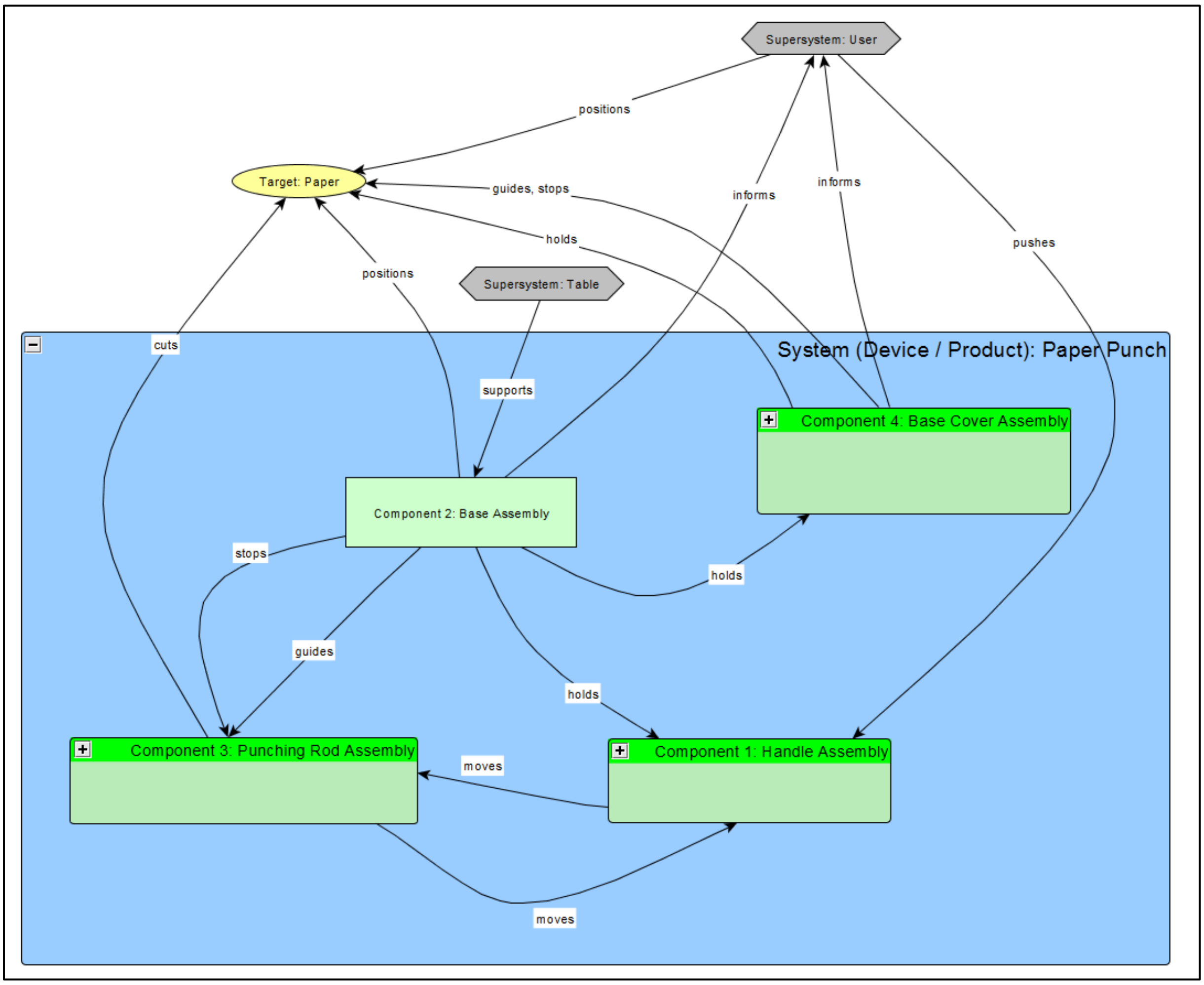

Fig.6: Function Model for Paper Punch

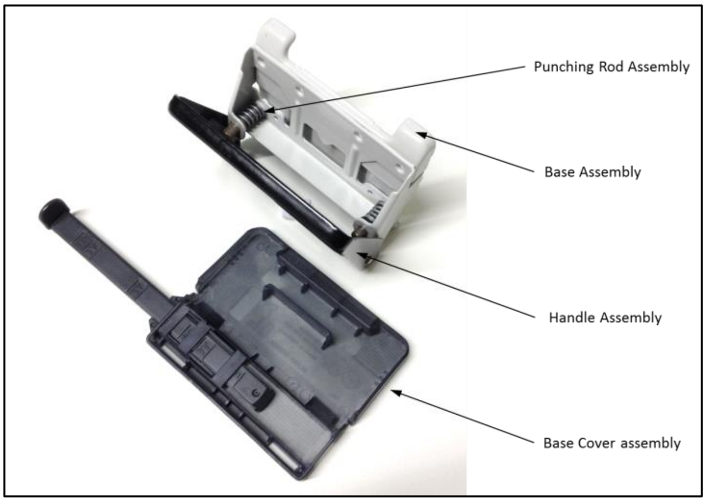

Fig.7: Sub-Assemblies of Paper Punch

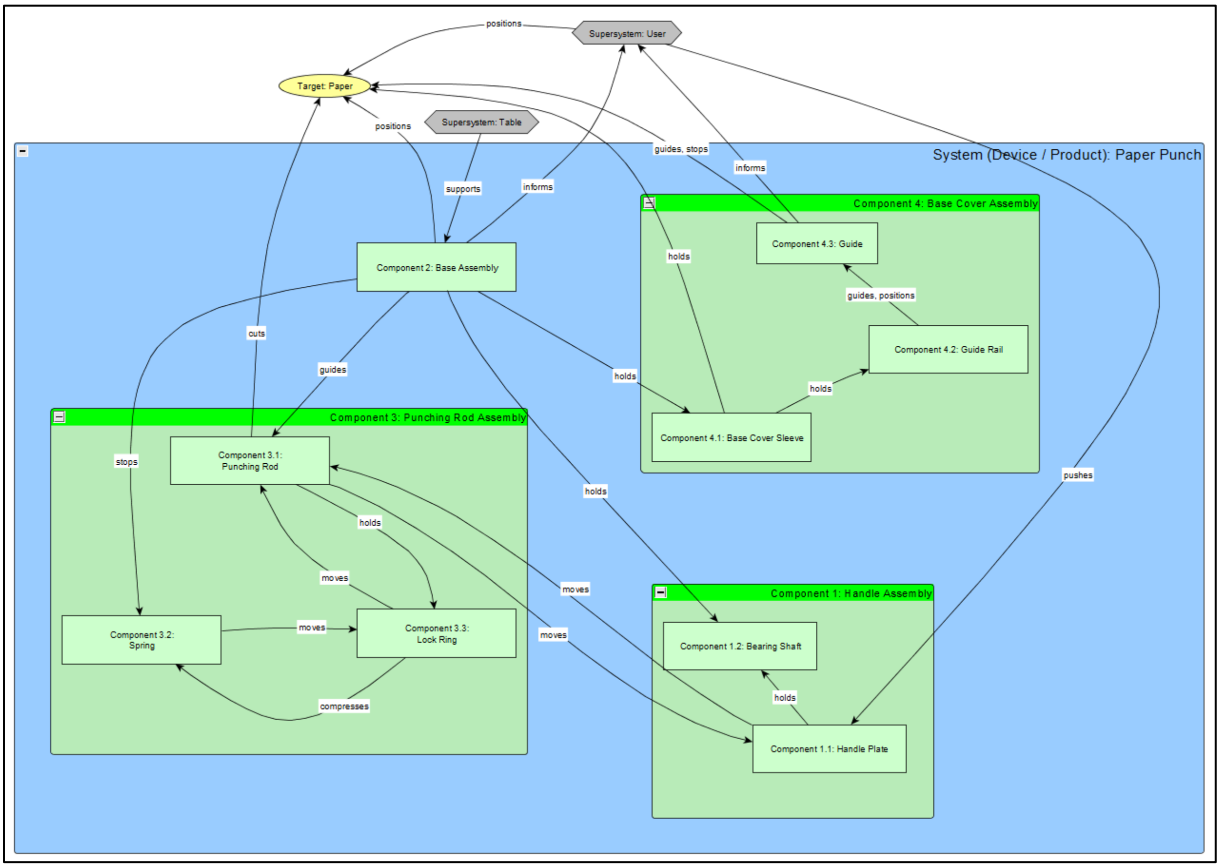

Fig.8: Nested Function Model for Paper Punch

3.1.2. Thoughts on Main Functions

During this Component Analysis it is necessary to formulate the Main Function. Following the definition, the Main Function acts on the Target and represents the “reason” why the system has been created or what the system actually does.

Following the rules of TRIZ Function Analysis, the target is always to be found in the Supersystem or the "neighborhood" of the system, so when we use Nested Function Models, the main function of each sub-assembly acts on a component of a higher system hierarchy.

Throughout the Nested Function Model we need to define the Main function of each sub- or sub-sub-assembly down to each single part.

3.1.3. Defining the timeframe for the Nested Function Model

According to TRIZ, defining the Operating Time is good practice. As a starting point it is suggested to look at the time when the system carries out or delivers its main function. As the Nested Function Model mainly deals with strategic evaluation, (exclusively) modelling problem situations is not a primary focus.

The point in time or the timeframe chosen should be consistent throughout the levels of the Nested Function Model. One problem arises when different usage scenarios are possible and the product can be used in different situations. In this cases it is useful to define those usage scenarios and build a model for each of them.

Many systems contain sub-systems that only exist to act during specific situations or usage scenarios. In Function Models we can easily identify these Components, as they are only "receiving" actions during the "normal" use of the system and are therefore only Function-Objects. Those cases might be problem-situations (e.g. emergency siren: During the normal use of a machine, the siren does not "emit" any action and is therefore no function carrier, it is only an object) or a handling robot inside a processing machine that only is active during specific operations.

Of course we can always ask the question: "What is the time when the main function of the (e.g. emergency- or handling-) sub-system is carried out" and model these timeframes, but this would lead to an inconsistent definition of the timeframe and it would clutter the Function Model as other components on higher system hierarchy might behave differently in those situations a s well.

3.1.3. How much hierarchical levels should be considered?

If we start our model on the highest level, the product considered might be made up of a huge number of components/parts. Of course we are not limited with regard to the "last" or deepest system level to model. When modelling an electronic device, we certainly can go down to the components of the capacitor or resistor, or even down to the chemical components of the dielectric. Practical considerations should guide the user here, which generally applies to the usage of all methodological tools. Usually, stopping at single parts that are bought from a supplier as the last, deepest level of the Nested Function Model, is a good practice. Also, sub-assemblies that are acquired from system suppliers, might be a good indicator not to go into further detail.

4. Enhancing Nested Function Models with the Time Axis

As described in 2.2., looking into the history of each Sub-System enables us to assess each component with regard to possible future developments. If a Nested Function Model already exists, the definition of super-systems and sub-systems is already at hand. We should then be able to research past versions of the “system” under evaluation, or if the system has been non existent in the past and has been added to the super-system as a new sub-system to increase it’s functionality. Also, we can see which components in the past have been trimmed and to which components on e.g. a higher hierarchical level the main functions have been transferred to.

Additionally, we can even model past versions of each component (system or sub-system or sub-sub-system…) and therefor create a functional evolutionary map, that gives us an overview over the structural evolution of the system.

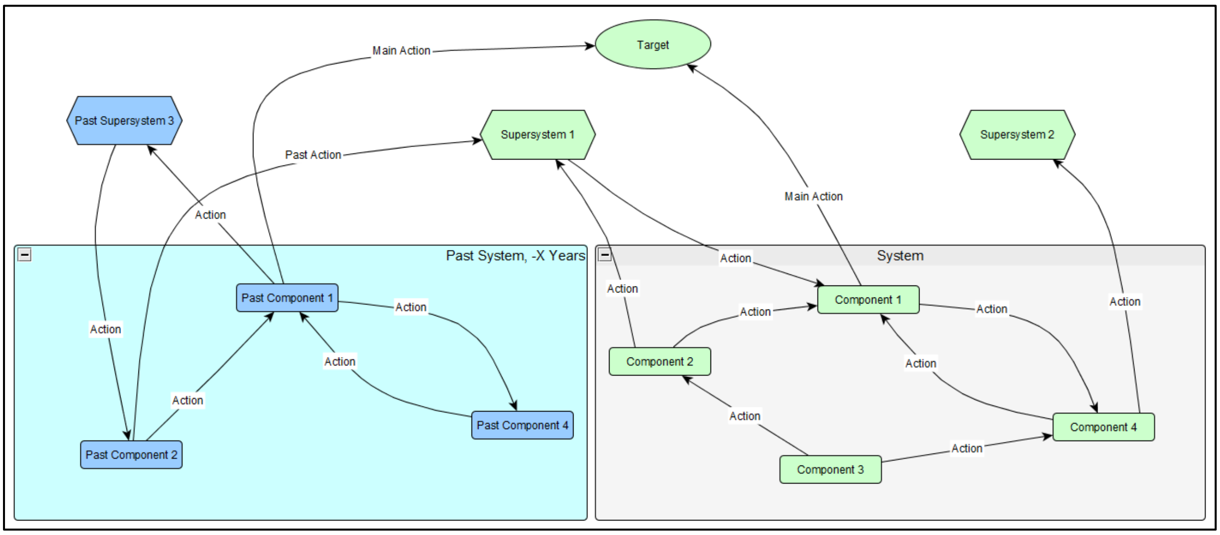

Fig.9: Function Model of Past and Present System:
Changes: Past Supersystem 3 is not existent today, Component 3 has been added
Interactions between Component 4 and Supersystem 2 and between Supersystem 1 and Component 1 have been added

For the horizontal aspect of each “box” of the Nested Function Model, we can define several key points in time representing significant development stages of the component evaluated. If modelled on all hierarchical levels, this points in time automatically synchronize throughout the hierarchical structure of the

Nested Function Model, as past components on one system level consisted of past sub-components on a lower system and at the same time Supersystems for the component have been past versions of the Supersystems today.

For each component, future considerations and scenarios can be evaluated. Future scenarios of sub-systems automatically give us clues for the system looked at, so hierarchical levels cross-feed each other with information about the development history:

-50 years	-10 years	now	+10 years
-2 system	-1 system	system	future system scenario
-2 C1	-1 C1	C1	future C1 scenarios
(----)	-1 C2	C2	future C2 scenarios
(----)	(----)	C3	future C3 scenarios
-2 C1.1	-1 C1.1	C1.1	future C1.1 scenarios
	-1 C2.1	C2.1	future C2.1 scenarios
		C3.1	future C3.1 scenarios

4.1. What about MPVs?

For each present and past system, Component, Sub-Component (…etc.) we can also identify Main Parameters of Value that have driven the development. A list of important MPVs can be attached to each “box” of an Nested Function Model. Furthermore the main functions throughout the Nested Function Model give indications of what actually are the MPVs for each component and which components are influencing the MPVs. If the MPVs are quantified over the past versions of the Systems or Components, it might even resemble an s-curve development that can be assessed for each component.

4.2. Connecting the Trends of Engineering Systems Evolution

The Trends of Engineering Evolution (TESE) are a powerful knowledge base giving an invaluable amount of known and recurring system characteristics. The Thesis is that these characteristics are objective and true for every engineering system, so they apply to a car as well as to a bolt.

If this thesis is true, we can apply the TESE and their mechanisms on each hierarchical level of the Nested Function Model. Having modelled past versions of the system, it's components and sub-components (as described in 4), we can easily compare the system's or component's characteristics with the characteristics given by the TESE. This automatically gives us possible concept ideas for the future, as we can apply characteristics logically following past developments (e.g. Mono-Bi-Poly, 1D-2D-3D…) or characteristics that have not shown up in the past. Additionally it can be evaluated, in which way the change in characteristic (or "step" of the Trend) increased the MPV. This change will most likely be quantifiable. For example, the step from a multi-hinged wiper-blade of a windscreen wiper to the current flexible flat-wiper-blades lead to a more even pressure distribution between blade and windscreen, increasing the MPV "Amount of Water left on windscreen after one wipe". So going from a system with multiple flexible links to a completely flexible system and thus following the Trend of Increased Dynamization lead to a quantifiable change in a system MPV.

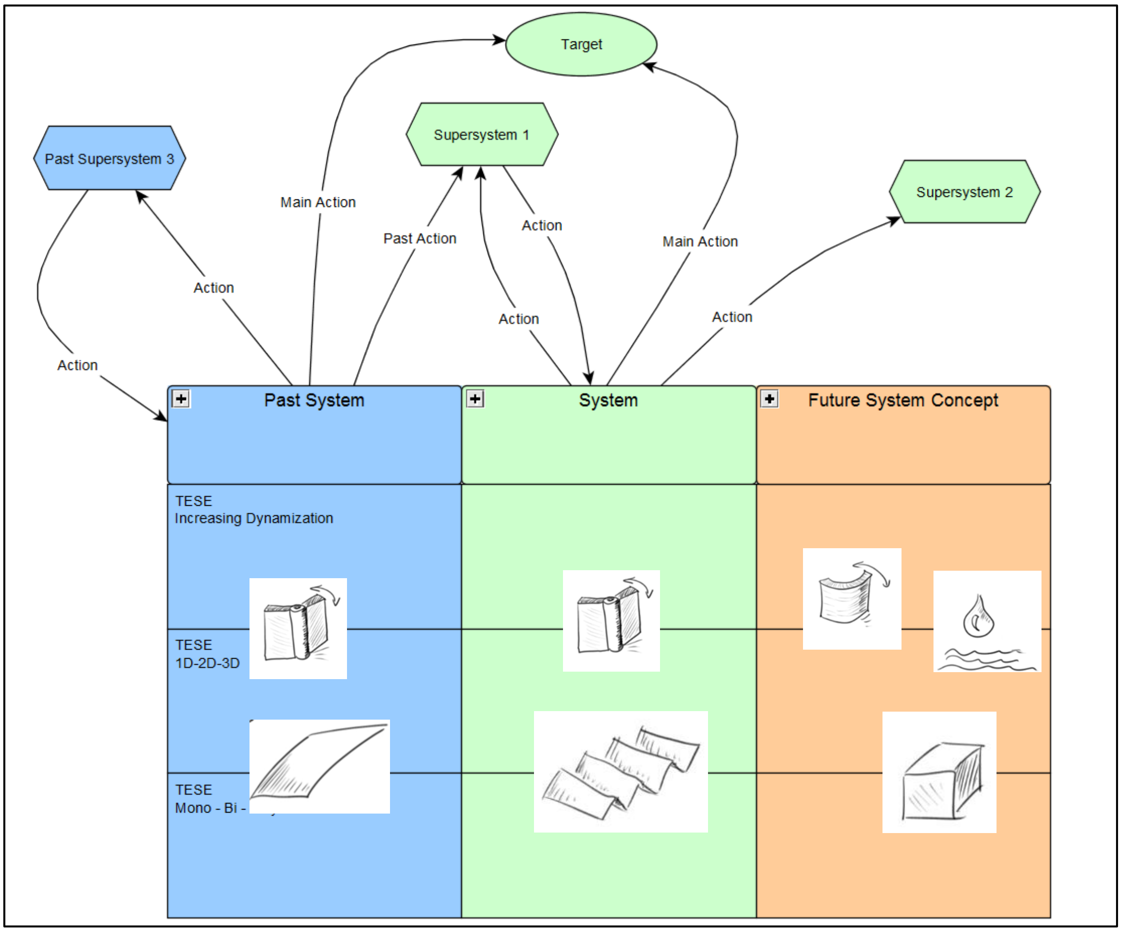

Fig.10: Identification of TESE Mechanisms
This can be repeated throughout the Nested Function Model for each Sub-System

Conclusion

The considerations presented in this paper try to combine several classical TRIZ approaches and tools to support strategic product development. The proposed Nested Function Model with its expansion through the 9-Screen Model and TESE might be a possibility to systematically drive the development of complex engineering systems.

Practical projects, which are currently being discussed, will show if the proposed approach is useful in an industrial environment. One aspect which decides on the usefulness will be the fact, that the Product Map developed is a non-stationary thing. It has to be fed with development steps and should reflect each development step. Also, the application will show if the work needed to create such a Product Map is in good relationship with the information created and decisions supported by it.

The anticipated advantages are, besides others:

1. that a complete functional and historical map of a product can be created,

2. that TRIZ based problem solving can be quickly initiated on each system level, because the function models are already at hand

3. that communication between departments (and thus “owners” of sub-function-models) can be objectified as the connection between them can be tracked down to functions connection their respective sub-system.

4. that a common and transparent understanding of the whole system is created between all people involved.

During the elaboration of the approach described, it got increasingly clear that the creation of Nested Function Models with all proposed aspects might benefit from the use of software, which then also offers a range of examination possibilities. As a starting point, the freely available software “yEd” was used to create the function models. Nevertheless, this software lacks functionality that supports the TRIZ specific aspects like evaluating MPV or TESE, or a possibility to include past function models and thus represent the evolution of a system. A TRIZ specific function modelling software could be developed in the future which takes the thoughts of this paper into consideration.

As mentioned in the introduction, the authors appreciate any constructive, critical comments from the TRIZ community.