“How Function is Realized in Problem Solving”

TAKAHARA Toshio
takahara-t@m.ieice.org

0. Preface

“TRIZ” is the Russian acronym for “Theory of Inventive Problem Solving” [1]. “ASIT” (the acronym for “Advanced Systematic Inventive Thinking”) [2-8] is the thinking method derived from TRIZ.

I discussed the basic outline of “function” and “structure” in the thinking tools or the problem solving tools in a previous paper [11]. I try to show more detail expansion of “function” in the thinking tools or the problem solving tools in this paper. I make clear the way how function is achieved in the real world with examples. The correspondence between the tools and several types of function realization in problem solving is shown.

1. “System object”/“process object” and “function”/“structure” [9-11]

In the area of operation on the outside world, “technology” born by the “technical means” or “technical system” and “institution” born by “common notion” or “institutional system” are made and made use of.

We can grasp that “object” is not only “system object” consisting of the element of technical system or institutional system to make but also “process object” consisting of the element of process of system action or human action. [9]

We have three phases as to direction between “technology/institution” and the human being.

1) “make” phase
2) “manage” phase
3) “use” phase

In “use” phase, we perform simply in real-time domain.

In “manage” phase, we manage to go well according to the situation, in other word we collect information about the situation and decide the images of “object” by which we should act in advance then immediately realize that image in the real world. In this sense we perform in semi-real-time domain in “manage” phase.

On the other hand “make” phase is to select and decide the image of “object” as to “technology” and “institution” before their realization in the real world in advance. This making action is done in semi-real-time domain or in batch environment.

Ordinarily making “technical system” consists of designing the content of “object” in mind and afterwards manufacturing it physically in batch environment. These making actions are followed by their realization in the real world in real-time domain as a “use” phase. On the other hand usually making “institutional system” is in the semi-real-time domain soon followed by “manage” phase also in the semi-real-time domain.

The relations between three phases and “system/process object” are shown in table 1. [10]

Table 1 Applied area, three phases of “technology and institution” and “object” [10]

Area	Phase	Time Domain			Used Object		Changed Object	
		Batch	Semi-real -time	Real -Time	System Object	Process Object	System Object	Process Object
Technical	“Make”	X (usually)			X	X	X	X
	“Manage”		X		X	X		X
	“Use”			X	X	X		(X)
Institutional	“Make”		X (usually)		X	X	X	X
	“Manage”		X		X	X		X
	“Use”			X	X	X		(X)
Personal	“Make”		X (usually)			X		X
	“Manage”		X			X		X
	“Use”			X		X		(X)

After examining the situation around the human behavior and object, I give an outline of “structure” and “function” considering formal contents of “object” which have statics and dynamics.

In general “object” itself has

1. Statics: Attribute, and

2. Dynamics: Property to change.

Between objects and between attributes, there are

3. Static relations and

4. Dynamic relations which include

4.1 Causality and

4.2 Interaction.

A viewpoint of human being or institutional system specifies the “granularity” of “object”. “Granularity” is time range, space range and level of abstraction.

“Object” has “attribute”(in broad sense).

An “attribute”(in broad sense) consists of quality attribute and quantity attribute.

A “state” is the attribute that can be easily changed.

An “attribute”(in narrow sense) including “ability” is the attribute that can not be changed easily.

“System object” can take “action”. An action is taken by “system object” via process or a “process object” itself. And both “system object” and “process object” have attributes.

“Action” can change

1) “attributes and/or states” of the object itself,

2) “action and/or attributes and/or states” of the other object,

3) static relations between “attributes and states” of the object itself, and

4) static relations between “attributes and states” of the object and those of the other object.

A “function” is the “positive meaning” of “object” to the outside world. “Object” has a “positive meaning” to the outside world when its action itself is useful or its “attribute” or “state” is useful. [11, changed a little]

A “structure” is the whole things that consist of elements and relations between them. A set of “objects” have the structure of “objects”. [11]

Operation of ASIT and “Enhancement of ASIT (ASIT+)” are shown in table 2. [11]

Table 2 Operation of tools and objects [11]

Classification		The name of tool	Case	Apply to	
				ASIT	Enhancement
Add	Add Object	Multiplication: introduce a slightly modified copy of an existing object into the current system or process		X	X
	Add Function	Unification: assign a new use to an existing object	Add action	X	X
			Add/activate attribute/state	X	X
Remove	Remove Object	Object Removal: remove an object from the system or process		X	X
	Remove Function	Function Removal: remove function of existing object from the system or process	Remove action		X
			Remove/deactivate attribute/state		X
Change	Change Structure	Division: divide an object and reorganize its parts		X	X
	Replace Object	Object Replace: replace or substitute existing object with adding object obtained by multiplication			X
	Change Attribute	Uniform Attribute Change: Change attribute uniformly			X
		Breaking Symmetry: change a symmetrical situation into an asymmetrical one		X	X

x： available

In table 2 we can
#1 add “object” using multiplication tool *1,
#2 add function of existing “object” using unification tool *2,
#3 remove “object” using object removal tool *3,
#3+ remove “function” using function removal tool *3+,
#4 change “structure” of “object” using division tool (including the case of changing object using “breaking symmetry” tool) *4,
#4+ replace “object” using object replace tool *4+,
#5+ change attribute of “object” using uniform attribute change tool *5+, and/or
#5 change attribute of “object” using breaking symmetry tool *5.

Here,
*1: Multiplication: Solve a problem by introducing a slightly modified copy of an existing object into the current system or process.
*2: Unification: Solve a problem by assigning a new use to an existing object.
Case 1: adding an action
Case 2: adding/activating attribute or state.
*3: Object Removal: Solve a problem by removing an object from the system or process and assigning its action to another existing object.
*3+: Function Removal: Solve a problem by removing function of existing object from the system or process.
Case 1: removing an action
Case 2: removing/deactivating attribute or state
*4: Division: Solve a problem by dividing an object and reorganizing its parts (including the case of changing object using "breaking symmetry" tool) .
*4+: Object Replace: Solve a problem by replacing or substituting existing "system object" or "process object" with adding object obtained by "multiplication" tool.
*5+: Uniform Attribute Change: Solve a problem by changing attribute uniformly.
*5: Breaking Symmetry: Solve a problem by changing a symmetrical situation into an asymmetrical one. [11]

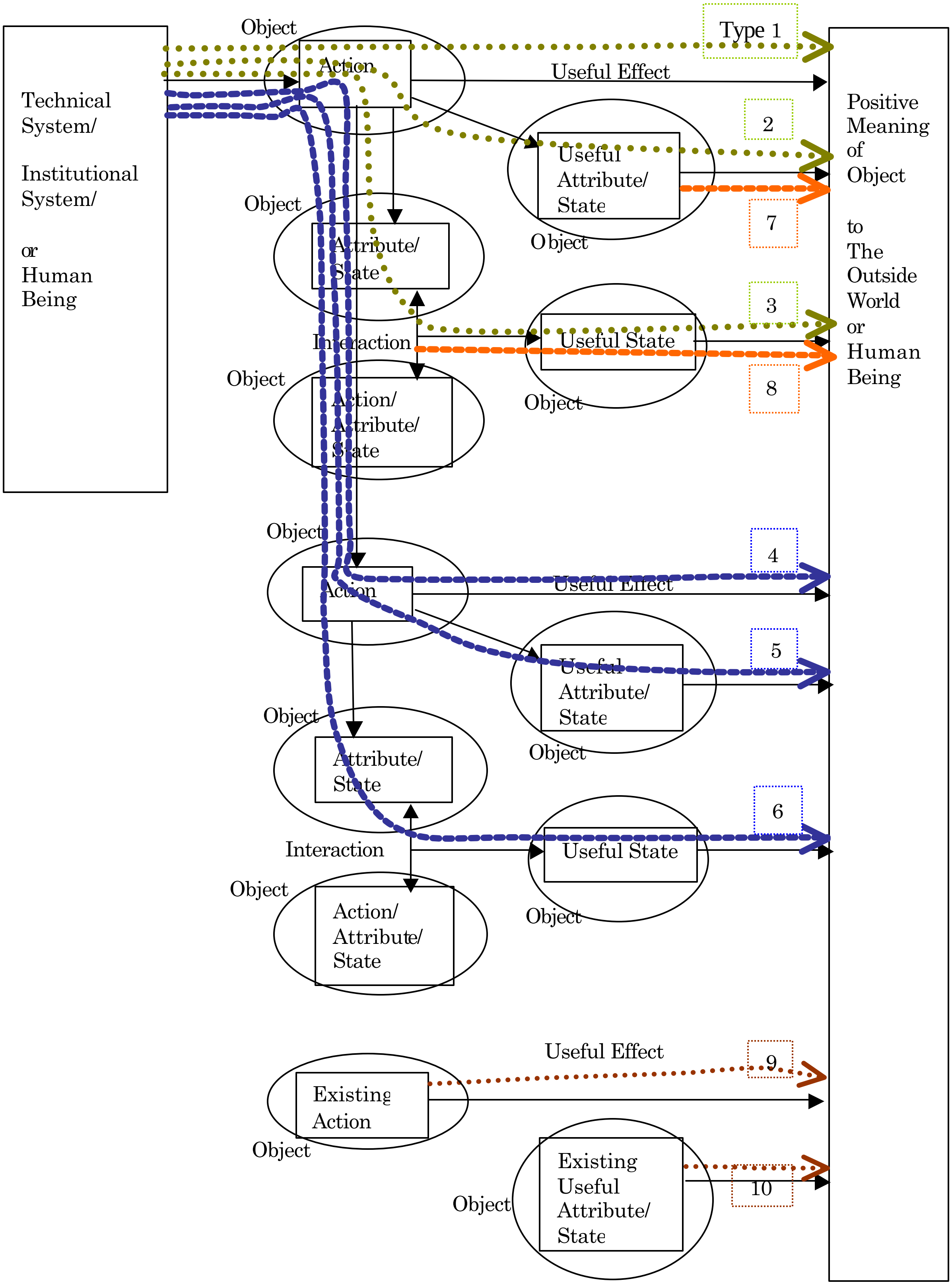
2. Realization of function with examples

A "function" is the "positive meaning" of "object" as I said before. "Object" has a "positive meaning" as a "function" to the outside world when
1) its action itself is useful or
2) its "attribute" or "state" is useful.
Its "attribute" or "state" is useful, when
2.1) its action activates a useful "attribute" including "ability" or "state",
2.2) its action makes interaction between its "attribute/state" and the other "attribute/state" which makes up a useful "state" (In this case we say both existing "attributes" are "activated"). Examples of these interactions are shown in [12],
2.3) change of structure or object directly make a useful "attribute" including "ability" or "state", or
2.4) change of structure or object makes interaction between its "attribute/state" and the other "attribute/state" which makes up a useful "state".
3.1) An existing useful "action" is also able to have a positive meaning.
3.2) An existing useful "attribute/state" is also able to have a positive meaning.
Problem solving tools can help the effort to make them more useful or to prevent them from expiring.

"Function" is realized in two steps. First we decide the contents of object beforehand in batch environment or in semi-real-time domain, by which we should act or object should be in process. To decide the contents of object contains decision of structure as relations between objects and attributes of objects. These structure and attributes can make activate an ability to become real functions. The second step is to put these contents into real world to realize function in real-time domain, physically in many cases.

Fig. 1 shows several types of causal relation to realize the function.

Fig. 1 Realization of Function



In fig. 1 realization of function by action in real-time domain is shown as follows.

⇢ show action-oriented functions.

Type1. Action→ Useful Effect: 1) Action itself is useful.

Type2. Action→[Useful Attribute/State]: 2.1) Action activates a useful "attribute" including "ability".

Type3. Action→[Attribute/State]→Interaction→[Useful State]: 2.2) Action makes interaction between its "attribute/state" and the other "attribute/state" which makes up a useful "state".

⇢ show action-action-oriented functions.

Type4. Action→ Action→ Useful Effect: 1) Action itself is useful.

Type5. Action→ Action→ [Useful Attribute/State]: 2.1) Action activates a useful "attribute" including "ability".

Type6. Action→ Action→ [Attribute/State]→Interaction→[Useful State]: 2.2) Action makes interaction between its "attribute/state" and the other "attribute/state" which makes up a useful "state".

Change of structure or object make a useful "attribute" including "ability" as follows in fig. 1.

⇢ show attribute/state-oriented functions.

Type7. [Useful Attribute/State]: 2.3) Change of structure or object directly make a useful "attribute" including "ability".

Type8. [Attribute/State]→Interaction→[Useful State]: 2.4) Change of structure or object makes interaction between its "attribute/state" and the other "attribute/state" which makes up a useful "state".

Existing action/ attribute works in fig. 1.

⇢ show functions of existing action/ attribute.

Type9. Existing Action→ Useful Effect: 3.1) An existing useful "action" has a positive meaning.

Type10. [Existing Useful Attribute/ State]: 3.2) An existing useful "attribute/state" has a positive meaning.

(Cause→ Effect)

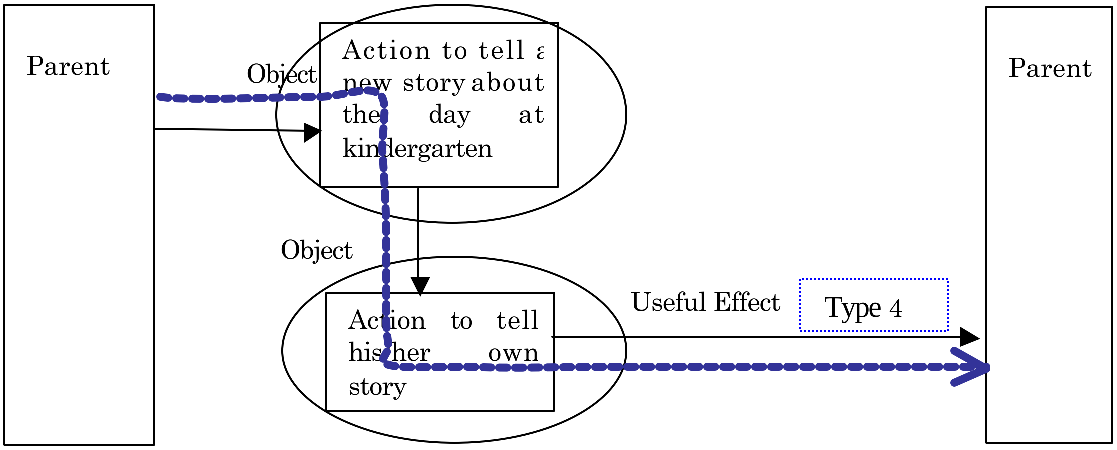
In type 1, 4, useful action is needed.
In type 2, 3, 5, 6, action that brings useful attribute is needed.
In type 7, 8, useful attribute is needed.

Some examples are shown as follows. They are all in the articles on ASIT

Example 1: The kid does not want to tell the story about the day at kindergarten to the parent. To tell a new story about the day at kindergarten will make the kid tell his/her own story. [3]

Area: institutional--home
Phase: manage
Object: process object
Tool: multiplication
Function type in "manage" phase: Type 4(fig. 2)

Fig. 2 Realization of function in example 1



Parent
Object
Action to tell a new story about the day at kindergarten
Object
Action to tell his/her own story
Useful Effect
Type 4
Parent

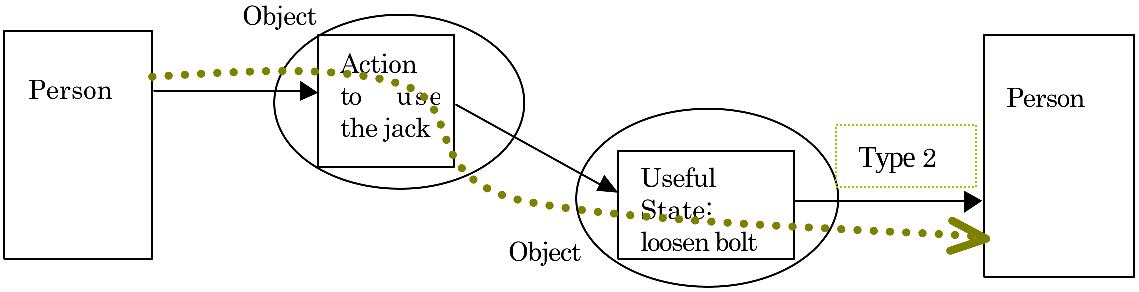
Example 2: Wheel game (part 1 tight bolt): I have a flat tire. I loosened three bolts of this tire. But one bolt is too tight to loosen. I must loosen this bolt by wrench and jack. [6]

Area: technical
Phase: make and use
Object: system object
Tool: unification

In "make" phase I make the system to put the wrench on the problematic bolt and to place the jack under the handle of the wrench using the power of the jack to unscrew the bolt.

Function type in "use" phase: Type 2(fig. 3)

Fig. 3 Realization of function in example 2



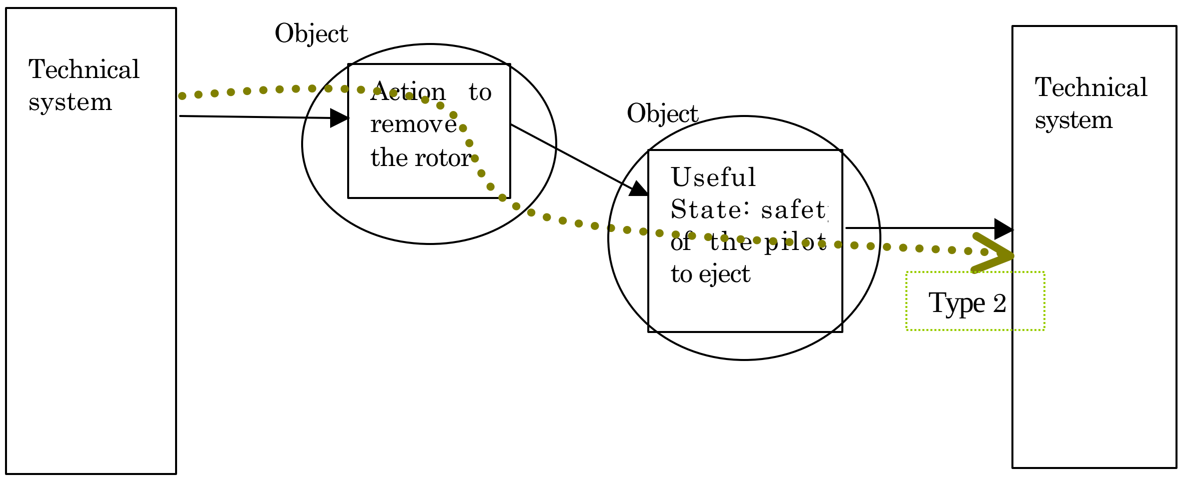
Example 3: “Currently, helicopter pilots are unable to escape the helicopter in case of technical problems. A good solution would be to eject the pilot upward before he parachutes down. But this is impossible because of the danger of being hit by the rotor. (Solution: The rotor will be removed just before the pilot is ejected)” [3]

Area: technical
Phase: make and use
Object: system object
Tool: object removal

In “make” phase, the engineer design the image of “object” including a detector of technical problem and ejector of pilot seat logically, and afterwards it is manufactured physically in batch environment. These making actions are followed by their realization in the real world in real-time domain as a “use” phase.

Function type in “use” phase: Type 2(fig. 4)

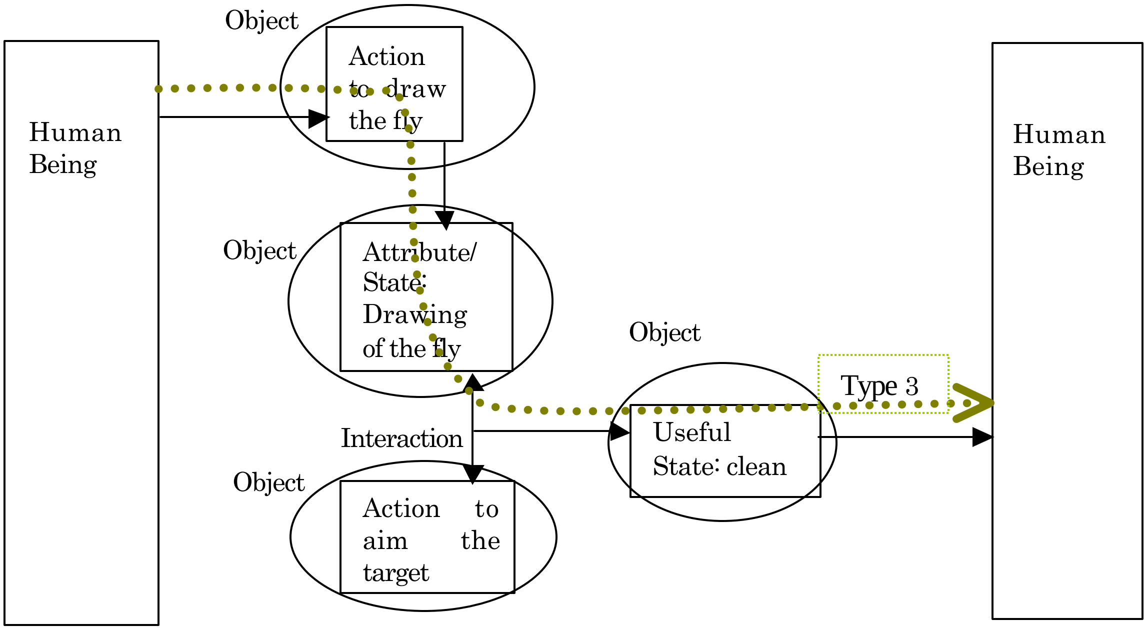
Fig. 4 Realization of function in example 3


Technical system
Object
Action to remove the rotor
Object
Useful State: safet of the pilot to eject
Type 2
Technical system

Example 4: The men's restrooms were not so clean at Schiphol airport in Holland. “Many men weren't aiming very well and were missing the target… The problem was solved by drawing a small fly on the walls of the urinals to attract the attention of the men using the restrooms. It turns out that when a man concentrates on a certain spot, he naturally also aims in that direction” (During the European Soccer Championships that took place in Holland and Belgium, the flies at Schiphol airport were replaced by miniature orange plastic soccer goals!) [7]

Area: technical
Phase: make and use
Object: process object
Tool: unification and multiplication
Function type in “use” phase: Type 3(fig. 5)

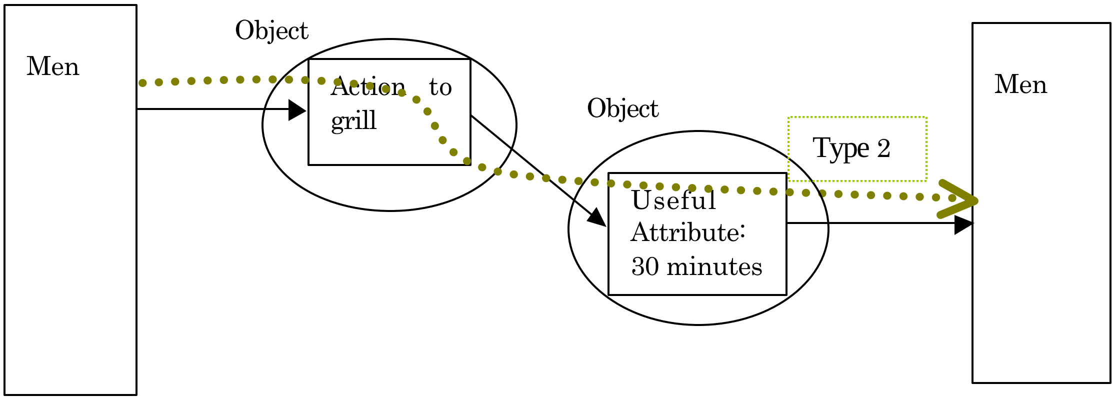
Fig. 5 Realization of function in example 4



Example 5: Three men decided to have a barbecue. Their grill was only big enough for two steaks, and of course they wanted to barbecue three. Grilling each side of the steak takes 10 minutes. How is it possible to grill three steaks in 30 minutes? [4]

Area: technical
Phase: manage
Object: process object
Tool: division
Function type in "manage" phase: Type 2(fig. 6)

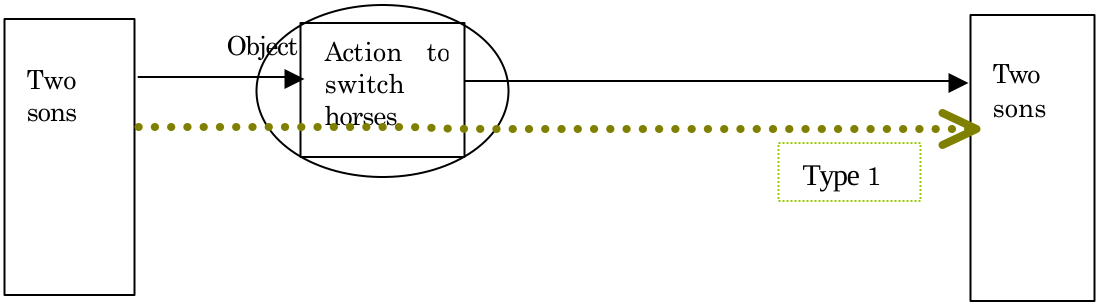
Fig. 6 Realization of function in example 5



Example 6: “A wise Chinese Emperor decided to divide his legacy between his two sons in an unusual way. He called his two sons, both excellent horse riders, and this is what he told them:
"One of you will inherit the largest portion of my legacy. You will both take part in a riding contest to determine who this will be - the winner will inherit the most. Oh... I forgot to tell you one of the conditions. The winner of this contest is the one whose horse comes last in the race. And if a winner is not announced by the end of the day tomorrow, neither of you will get the inheritance."
Both of his sons were utterly confused. "What kind of contest is that?" they thought. It was obvious to both of them that each would try to ride as slowly as possible, and this way the race would never end.
So they went to consult a wise old Chinese sage.
(The old sage told the two sons to switch horses. By doing this, each of them would try to ride as fast as possible to make their own horse come last.) ” [8]

Area: institutional--group
Phase: manage (or “make and use”)
Object: process object
Tool: division
Function type in “manage” phase: Type 1(fig. 7)

Fig. 7 Realization of function in example 6



3. Problem solving method

In “ASIT’s Five Thinking Tools with Examples”, Roni Horowitz explains the method of problem solving. “Applying each tool is a five-step process. In the first step we define the problem world by listing the problem objects and the objects in the environment. In the second step we prepare for the application of the tool by collecting relevant data and making a few simple decisions. The third step is where we actually apply the tool and where the idea is born.” [3]

In this first step we specify a useful effect of action or a useful “attribute” or “state” to be obtained. In the third step of application of the tool (we have five tools in ASIT, eight tools in “Enhancement of ASIT” [11]), we can consider that the types of causal relation in function realization correspond to the tools as shown in table 3 taking into consideration that “object” involves “function” (action or attribute). Then we can specify the cause to make the effect.

Table 3 Operation of the tools of ASIT+ and the types of function realizing

Classification		The name of tool (ASIT tools)	Case	Types		
				1,4	2,3,5,6	7,8
Add	Add Object	Multiplication		X	X	X
	Add Function	Unification	Add action	X	X	
			Add/activate attribute/state			X
Remove	Remove Object	Object Removal		X	X	X
	Remove Function	Function Removal	Remove action	X	X	
			Remove/deactivate attribute/state			X
Change	Change Structure	Division				X
	Replace Object	Object Replace		X	X	X
	Change Attribute	Uniform Attribute Change				X
		Breaking Symmetry				X

x： available

4. Conclusion

I made clear the way how function is achieved in the real world with examples to find out several types of function realization in problem solving. The correspondence between the tools and several types of function realization was also shown. These examinations will help the TRIZ/ASIT practitioners to solve the problems in some ways.

References

[1] The TRIZ journal, http://www.triz-journal.com/

[2] Roni Horowitz, "From TRIZ to ASIT in 4 Steps", The TRIZ journal, http://www.triz-journal.com/archives/2001/08/c/index.htm, Aug.2001.

[3] Roni Horowitz, "ASIT's Five Thinking Tools with Examples", The TRIZ journal, http://www.triz-journal.com/archives/2001/09/b/index.htm, Sept.2001.

[4] Roni Horowitz, "Introduction to ASIT", http://www.start2think.com/, Mar. 2003.

[5] "ASIT Premier online course", http://www.start2think.com/

[6] "Creative Thinking" CD-ROM Professional Edition, http://www.invention-highway.com/, Compedia&BCI software, 1999.

[7] Roni Horowitz, "ASIT newsletter e-book 1", http://www.start2think.com/, 2002.

[8] Roni Horowitz, "ASIT newsletter e-book 2", http://www.start2think.com/, 2002.

[9] Takahara Toshio, "Application Area of Thinking Tool or Problem Solving Tool",
The TRIZ journal, http://www.triz-journal.com/archives/2003/06/e/05.pdf, Jun.2003.
[10] Takahara Toshio, "How People Interact with Objects Using TRIZ and ASIT", The TRIZ journal, http://www.triz-journal.com/archives/2003/08/d/04.pdf, Aug.2003.
[11] Takahara Toshio, "Logical Enhancement of ASIT", The TRIZ journal, http://www.triz-journal.com/archives/2003/09/e/05.pdf, Sept.2003.
[12] Nakagawa Toru, "Commentary on 'The Picture Hanging Kit Problem' ", Source Reference: Ed Sickafus: "Unified Structured Inventive Thinking: How to Invent", Ntelleck, 1997, pp. 439-442, http://www.osaka-gu.ac.jp/php/nakagawa/TRIZ/eTRIZ/epapers/eUSITE9Nakagawa010726/eUSITE9Nakagawa010803.html, Jul. 2001 [in Japanese]; English translation by Nakagawa Toru, Aug. 2001.

About the Author

TAKAHARA Toshio received his B.E. degree in electric-communication engineering from Waseda University, Tokyo, Japan in 1968. He joined FUJITSU Limited where he mainly engaged in river management as a design engineer and a consultant. Before retiring an affiliated company of FUJITSU in 2002, he engaged in quality management concerning ISO9001 for a few years. In the study of quality management and design method at the affiliated company of FUJITSU, he met TRIZ in 2001.
His today's interest includes culture theory, design method and problem solving tools.

His e-mail address: takahara-t@m.ieice.org
His homepage URL: http://www.geocities.co.jp/WallStreet/2744/ (almost in Japanese)