機能とプロセスオブジェクト概念を基礎にした差異解消方法 その2

高原利生

概要

問題解決，新機能実現，理想化という差異解消の全体が，現実世界の認識，操作,変更の内容である. 差異解消は, 目的設定, オブジェクト操作で行われる. 目的設定がオブジェクトの言葉で行われ, プロセスオブジェクトを含む全てのオブジェクト操作が, 三種のオブジェクト変更の論理型毎に形式化して行われることにより，差異解消の全体が，統一した共通のインプットーアウトプット関係によって行える．理想化に二つの面がある.

1. はじめに

前稿[1][2]と本稿の関心は，**現実世界を認識し操作, 変更する形式的理論**を作ることである．この理論の使用者と適用領域に関する要件について[1]で述べた．[2]でこの形式的理論成立の要件を明らかにした.

このうち認識についての要件は，認識可能なあらゆるものがオブジェクトであることを[1]で述べた.

[2] で，操作,変更についての要件を満たすための整備を行い，全操作の型の網羅を試みた．本稿の第一の内容は，これについての修正である.

また[2]で，従来の「問題解決」を含む差異解消過程の構造の枠組みを明らかにした．差異解消は現実世界の認識，操作,変更の内容である．本稿の第二の内容は，これについての修正である.

以上について本稿は，前稿[2] ２項,３項をベースにして，本稿のみで把握可能なための必要最小限の内容の繰り返しと[2]に対する追加,修正分を記述する.

2. 理論の要件と準備

2.1 認識と操作の理論の要件

現実世界を認識し変更する理論の要件は, 認識可能なあらゆるものを扱う対象とすることができること，それに対してあらゆる操作可能な変更の型を適用できることである[2].

2.2 オブジェクトと機能

オブジェクトを「**認識可能なもの**」ととらえる[1].

認識可能なオブジェクトの内容は，**物**,「**観念**」(認識可能な物質的実体に担われた誰かの観念内容および私が主体である場合の私の頭脳の中にある観念内容)という二つの**存在**に対応する**システムオブジェクト**および**運動**に対応する**プロセスオブジェクト**である．**操作,変更可能なオブジェクトはオブジェクトのサブセットである**．[1]

属性とは，オブジェクトの内容を各**粒度**（空間的, 時間的大きさ），各**密度**（細かさと抽象度）において，具体的に規定する全てのものである．具体的な粒度と密度もオブジェクトの属性である.

上位のオブジェクトの属性は，1. 全体属性，2. オブジェクトの内部構造(下位のオブジェクトそのもの，その数, その間の構造)である. [1] 属性を変化しにくい(狭義の)属性と変化しやすい状態に分ける.

運動(プロセスオブジェクト)が私にプラスの意味を持つとき，この意味を**機能**という. また副次的に存在, 運動(オブジェクト)の属性がこのプラスの意味に資すると考えられるとき，この意味も副次的に機能と扱う. [1][2]

操作,変更可能なプロセスオブジェクト，システムオブジェクトの中から機能に関与するオブジェクトが漏れなく枚挙されなければならない．**主たる機能の実現はプロセスオブジェクトを生成することであり，多くの問題はプロセスオブジェクトの削除で解決される**.

2.3 オブジェクト操作,変更

[2] 233 項「運動過程の構造」では，図-１「運動過程の構造モデル」から４項のオブジェクト操作表を導いていた．本項は,オブジェクトの操作,変更という運動に限定して運動の構造を再検討する.

2.3.1 オブジェクト変換の原理

次の図はTRIZに限らず機能モデルに良く使われるものである．オブジェクト1 がプロセスオブジェクトの運動によって オブジェクト2 を変化させることを表現する. この“オブジェクト1- プロセスオブジェクト- オブジェクト2 モデル”は，現実の多対多の双方向関係のいくつかを縮退し，現実の今の私にとって意味のある変化を伴う

一対一の一方向リンクだけを残したものである[2]．酸素(オブジェクト1)が燃焼という運動によって可燃物(オブジェクト2)を変化させる．

図 1.　“オブジェクト 1- プロセスオブジェクト - オブジェクト 2 モデル”

人とシステムの関係に「変更モード」と「利用モード」の区別がある．[3]「変更モード」では我々は技術システムや制度システムの作成，変更を行う．「利用モード」ではシステムの利用，運用を行う．この「利用モード」における如き変化の原理を原理 U で表す．

オブジェクト変更の原理 U：オブジェクト 1 とプロセスオブジェクトがオブジェクト 2 を変化させる．

オブジェクト1 とプロセスオブジェクトがオブジェクト2 を変化させる限り，「変更モード」と「利用モード」双方に適用できる原理である．この中でオブジェクト2 が変化する．

しかしオブジェクト 1 や プロセスオブジェクトを変化させようとするとき，図 1 は使えない，そこでもともとは双方向性を捨象して作られたこのモデルから双方向性を復活させる．

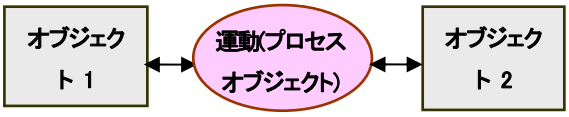

図 2.　双方向“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”

この図は，場合によってはオブジェクト 1 とオブジェクト 2 がプロセスオブジェクトを変化させることを示している．酸素(オブジェクト 1)を取り去っても可燃物(オブジェクト 2)を取り去ってもが燃焼という運動は消滅する．こうして原理P を得る．

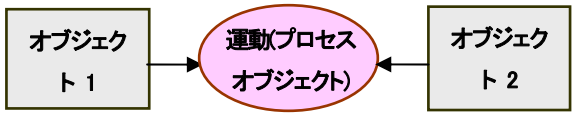

図 3.　オブジェクト 1 と オブジェクト 2 が プロセスオブジェクトを変化させる

オブジェクト変更の原理 P：オブジェクト 1 とオブジェクト 2 がプロセスオブジェクトを変化させる．このためには，オブジェクト 1 とオブジェクト 2 はシステムオブジェクトに限定される．

ただし図 1 や図 2 で常にオブジェクト 1 とオブジェクト 2 がシステムオブジェクトに限定されると考えることは認識の粒度を限定してしまう．

さらに弁証法の法則による変更原理がある．オブジェクトの内容すなわちオブジェクトの属性の変化はオブジェクト自身を質的に変化させうる．オブジェクトの属性とは，全体属性とそのオブジェクトの構造(下位のオブジェクトそのもの，その数，その間の構造)であった．

オブジェクト変更の原理 D：全体属性,とそのオブジェクトの構造の変更が，新しいオブジェクトの生成や自身の消滅をもたらす．

原理 U と原理 P の結果はそれぞれ原理 D に作用しうる．

2.3.2　人によるオブジェクト操作の原理

次に人がオブジェクトを操作する型を検討する．最初の型はありふれたものである．“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”における全てのオブジェクトを操作しうる．

これは空間粒度によって二つに分けられる．

オブジェクト操作の型 AM：人は“オブジェクト1- プロセスオブジェクト- オブジェクト2 モデル”におけるオブジェクト1，オブジェクト2 を直接操作しうる．そのためには，オブジェクト1とオブジェクト2 はシステムオブジェクトに限定される．

例：可燃物（オブジェクト2）を取り去る．

オブジェクト操作の型 AO：人はモデルの外部から“オブジェクト1- プロセスオブジェクト- オブジェクト2 モデル”における全てのオブジェクトを他のオブジェクトを媒介にして操作しうる．

例：酸素（オブジェクト 1）と可燃物（オブジェクト 2)の間にこのモデルの外部から不可燃物を挿入すると燃焼という運動は消滅する．

もう一つの操作の型も単純なものである．「変更モード」で，我々は“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”に介入して，既存のモデルやオブジェクトを無視して自由にオブジェクトの「持ち込み」「持ち去り」「取替え」ができる．ただ一つの制約は「変更モード」での介入後，スムーズに「利用モード」に移行できなければならないということだけである．

これは時間粒度によって二つの型に分けられる．

オブジェクト操作の型 RM：人は“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”におけ

る全てのオブジェクトまたはこのモデル全体を，既存のモデル，オブジェクトの制約を無視して「持ち込み」「持ち去り」「取替え」ができる.

オブジェクト操作の型 **RE**：人は“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”の連鎖を，既存のモデル，オブジェクトの制約を無視して「持ち込み」「持ち去り」「取替え」ができる.

2.3.3 まとめ

こうして現在の“オブジェクト1- プロセスオブジェクト- オブジェクト2 モデル”を対象にした場合の人によるオブジェクト操作の型と実世界におけるオブジェクト変換の原理を得た．これを表1, 表2 にまとめる.

表 1.　人によるオブジェクト操作の型

人によるオブジェクト操作の型		操作対象	何による変換	アウトプット	モード
AM	“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”におけるオブジェクト 1，オブジェクト 2 を直接操作．（オブジェクト 1 とオブジェクト 2 はシステムオブジェクトに限定）	システムオブジェクト	人によるシステムオブジェクトの直接操作	システムオブジェクト	変更モード
AO	人は外部から“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”における全てのオブジェクトを他のオブジェクトを媒介にして操作	システムオブジェクト，プロセスオブジェクト	人によるオブジェクトを介したオブジェクトの操作	システムオブジェクト，プロセスオブジェクト	変更モード
RM	“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”における全てのオブジェクト，このモデル全体を，既存の制約を無視して「持ち込み」「持ち去り」「取替え」	システムオブジェクト，プロセスオブジェクト “オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”	まるごと「持ち込み」「持ち去り」「取替え」	システムオブジェクト，プロセスオブジェクト “オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”	変更モード
RE	“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”の連鎖を，既存の制約を無視して「持ち込み」「持ち去り」「取替え」	“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル” の連鎖	まるごと「持ち込み」「持ち去り」「取替え」	“オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”の連鎖	変更モード

表 2.　オブジェクト変換の原理

オブジェクト変換の原理		インプット	何による変換	結果	モード
原理 **U**	オブジェクト 1 とプロセスオブジェクトがオブジェクト 2 を変化させる	システムオブジェクト，プロセスオブジェクト	モデル＊	システムオブジェクト	変更モード，利用モード
原理 **P**	オブジェクト 1 とオブジェクト 2 がプロセスオブジェクトを変化させる	システムオブジェクト	モデル＊	プロセスオブジェクト	変更モード
原理 **D**	全体属性とそのオブジェクトの構造の変更が，新しいオブジェクトの生成や自身の消滅をもたらす	オブジェクトの全体属性とそのオブジェクトの構造	弁証法の法則による自律変化	システムオブジェクト，プロセスオブジェクト	変更モード，利用モード

モデル＊: “オブジェクト 1- プロセスオブジェクト- オブジェクト 2 モデル”

人のオブジェクト操作のみで目的が達成できる場合もある．通常は**人によるオブジェクト操作が，オブジェクト変換を引き起こし，全体の結果が得られる**．これを次の図4 と表3 にまとめる．**オブジェクト変更実現が，**プロセスオブジェクト削除，プロセスオブジェクト生成，オブジェクト(属性)変更という**オブジェクト変更の論理型** (3.1 項参照) **毎に，オブジェクトの操作,変更を手段として行える**ことが分かる．これらにより“オブジェクト1- プロセスオブジェクト- オブジェクト2 モデル”を対象にした場合の**あらゆる操作可能な変更の型が網羅された.** 表3 が[2]表－５に換わるものである．**これには当該オブジェクトに影響を与える過去の変更は含まれないことに注意しなければならない.**

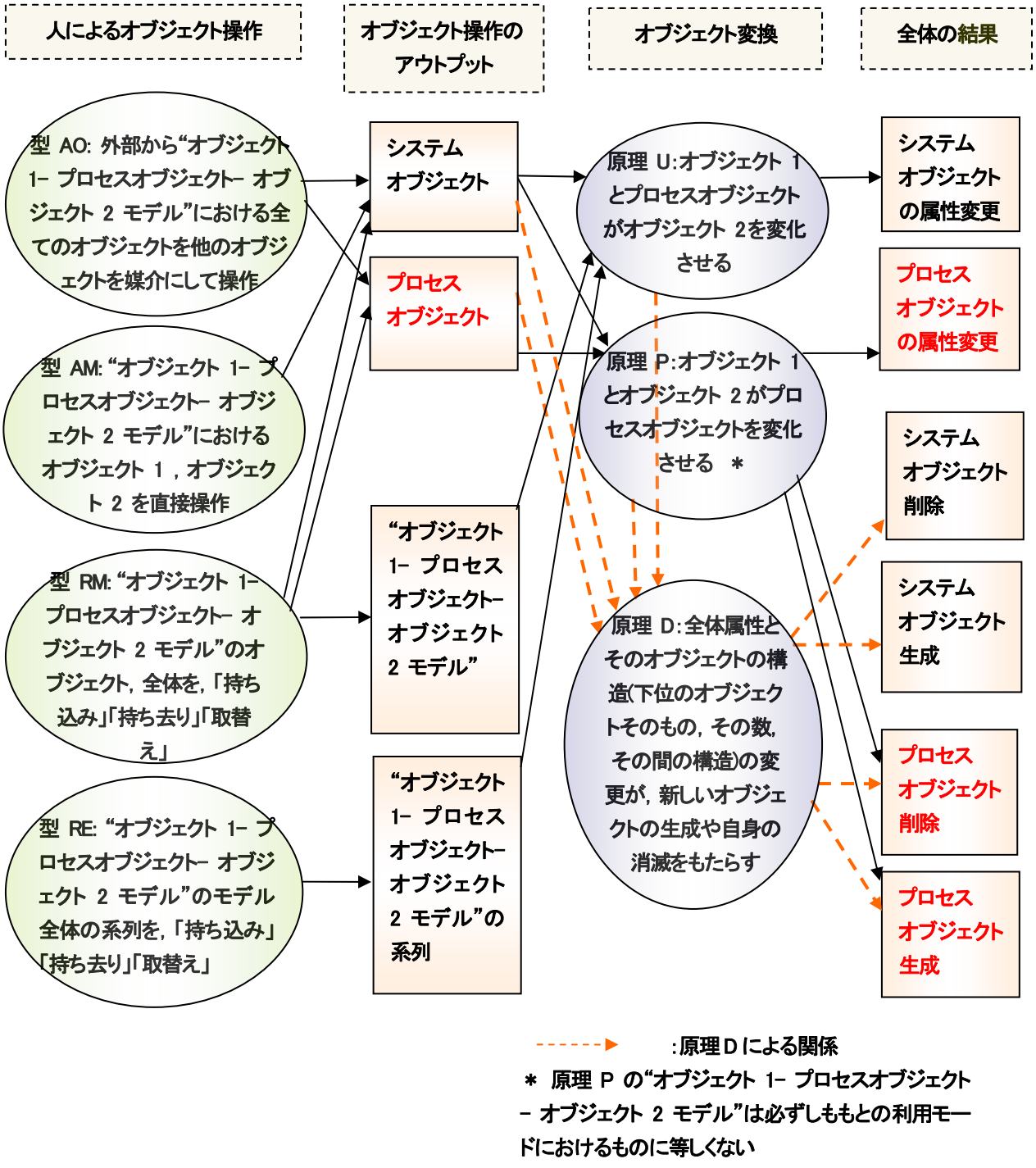

図 4. オブジェクト操作, 変更（人によるオブジェクト操作とそれに続くオブジェクト変換）の構造

表 3.　オブジェクト操作, 変更（人によるオブジェクト操作とそれに続くオブジェクト変換）

粒度	操作型	操作されるオブジェクト	変換原理＊	全体の結果					
				プロセスオブジェクト削除	プロセスオブジェクト生成	プロセスオブジェクト変更	システムオブジェクト削除	システムオブジェクト生成	システムオブジェクト変更
外部から	AO	オブジェクト 1, オブジェクト 2	U						○
			U-D				○	○	
			P	○	○	○			
			P-D	○	○				
			D				○	○	
		プロセスオブジェクト	U						○
			U-D				○	○	
			D	○	○				
モデル＊＊	AM	オブジェクト 1, オブジェクト 2	U						○
			U-D				○	○	
			P	○	○	○			
			P-D	○	○				
			D				○	○	
モデル＊＊	RM	オブジェクト 1, オブジェクト 2	U						○
			U-D				○	○	
			P	○	○	○			
			P-D	○	○				
			D				○	○	
		プロセスオブジェクト	U						○
			U-D				○	○	
			D	○	○				
		モデル全体	U						○
モデル連鎖	RE	モデル連鎖	U						○

変換原理＊：単独の原理と原理 U と原理 P がそれぞれ原理 D に作用する場合（表中の矢印で作用を示す）
モデル＊＊：“オブジェクト1- プロセスオブジェクト- オブジェクト2 モデル”

3. 差異, 差異構造, オブジェクト操作

[2]31 項「差異, 差異構造, オブジェクト操作の概要」と 321 項「差異解消の型」を見直す. [2]41 項「オブジェクト操作表」については23項に述べたとおりである.

3.1　オブジェクト変更の三つの論理型と差異解消の三つの目的型

特定の価値の把握を前提にした 望ましい**機能**, すなわち**目的**と現実の機能の差異が問題となる. 一般の**差異**は, 一つのオブジェクトの異なった時点間または二つのオブジェクト間の, 属性間の差異, ある属性とその属性の不在との差異（これはオブジェクトの属性そのもの）, あるオブジェクトとその不在との差異（これはオブジェクトそのもの）である. 差異解消は, 差異のある二項の片方から他方への変更であるから, その内容は, 属性変更, オブジェクト生成, オブジェクト消滅という三つの論理型のいずれかである（今, 属性の生成,消滅はとりあえず扱わない）. このうち機能に寄与する型に限定すると, **属性変更, プロセスオブジェクト生成, プロセスオブジェクト消滅という三つのオブジェクト変更の論理型**が得られる.

ここで, **差異の発生とその解消が起こる状況**を検討する. 第一は, **既存システムにおいては, これらの三つのオブジェクト変更の論理型は全て生起し得るが, 新システム生成の場合は, プロセスオブジェクト生成のみ生起する**ということである.

第二に, 差異解消をシステムの内から外の順に列挙していくと, 次の三種類があることが分かる.

1. 現実システムの内部から差異が生じ内部で解消し発展が自律的に起こる場合.

これは現実システムそれ自体が自律的に運動していく場合で, 差異が生じると同時に発展が行われ差異が

解消していく．生命の自律的発展は，現実システムの内部に発展の原動力が内蔵されている典型である．植物の種子にはその後の生育内容が内蔵されている．
2. 現実システムとその外部との関係である利用，運用から差異が生じ外部からこの差異解消の動きが起こる場合．

これは 21. 利用，運用に際して現実システムの不具合から生ずる場合，22 現実システムにない新しい機能が追加され対処する場合に分けられる．
3. 外部から強制的にシステムが作られ新しい機能が追加される場合．差異生成も解消も外部から行われる．

以上の分類は，発展のきっかけは網羅しているが，実際の人の行動を網羅していない．したがって，下記のように区分しなおす．

1. 現実システムの内部から差異が生じる場合は，自律運動には介入しないこととして，ここでは扱わない．

21. 利用, 運用に際して現実システムの不具合から生ずる行動を独立して扱い，**現実システムの問題を解決**する場合とする. 22. 現実システムにない新しい機能が追加され対処する場合と 3. 外部から強制的にシステムが作られる場合を一まとめにして**新しい機能を生成**する場合とする．このふたつは，**利用者または運用者に**必要不可欠な機能であり，差異の内容が明確である．

これらから抜け落ちる**直接には製作者にとっての**改良を，**理想化**としてまとめてとらえる．これは背反でない二つの面がある．一つは，機能追加，問題解決とは別の分野において，現実システムが持っている法則性をシステム内部の論理として，現実システムの改善に利用することである．これは TRIZ で「理想化」，技術進化のトレンド，トリミング等として扱われていた．

もう一つは，新しい機能の追加が済んでも，問題解決が済んでも残る，それらをより良く改善する作業である，極論すれば，問題を解決する場合，新しく機能を作る場合は，その解の出来具合はどうでもよいのである．これらについて「良い」解を求めるには，一つのオブジェクトの変更だけをする粒度の場合でさえ理想化が必要となる．‘Hierarchical TRIZ Algorithms’ (階層化 TRIZ アルゴリズム, HTA) ではこの意味の理想化を p. D3 で扱っている[6]．集中がよいか分散がよいか，CPU の性能はどの程度のものがいいかの検討等通常の意味のいわゆるシステム設計は殆どこれに属するといってよいであろう．これも何かを理想化しようとしているのである．

この**理想化**の範囲は今まで TRIZ の文脈で考えられているより広く，問題解決と新しく機能を作る場合以外の全てと考えなければならない．

こうして**新機能生成，問題解決，理想化という三つの差異解消の目的型**が得られた**.**

3.2　差異解消の構造

最も密度が細かい場合，すなわち一つのオブジェクトの変更だけをする場合から考えをすすめよう．

新しいプロセスオブジェクトを生成する場合は，新しい機能を追加する場合である．

現実システムの問題を解決する場合と**システムの理想化をする場合**のそれぞれには，プロセスオブジェクトを除去する場合とオブジェクトの属性を変更する場合がある．したがって**プロセスオブジェクトを除去する場合とオブジェクトの属性を変更する場合**には，それぞれ現実システムの問題を解決する場合とシステムの理想化をする場合の二つの場合がある．

人の行動ステップは，次のいずれかの役割を果たす．
1. 知見を利用しデータを変換して，データを作る，2. 知見を利用しデータを変換して，知見を得る，3. 後のステップのどれを実行するか決める．

差異発見は，3. 差異発見があれば次のステップに行き，なければ何もしない．

差異認識は，1. 望ましい機能と現実というデータを変換して，属性変更，プロセスオブジェクト生成，プロセスオブジェクト消滅という三つのうちのいずれかの**オブジェクト変更の論理型**の具体化として**目的データ**を得，3. 次の手段検討ステップに行く．差異認識のアウトプットは，**設定目的**と言っていい．

手段検討は，2. 新機能生成，問題解決，理想化という三つの**差異解消の目的型**に依存する知見を利用し，必要な事前手段の知見，オブジェクト操作の知見を得，3. 次のオブジェクト操作のステップに行く．**事前手段の実現が必要な場合はその目的が設定目的に置き換わる**．例えば，問題解決の手段検討においては，TRIZ[4]のように問題の原因の解決を事前手段とするか，ASIT[7]のように問題のない状態の直接実現か決定する必要がある．前者の場合原因解消の目的が設定目的に置き換わる．

オブジェクト操作は，2. オブジェクト操作の知見に基づき，1. オブジェクトの目的を達成するような，オブジェクトを操作する手段というデータを得る．これは，2.3 項で示したように，**図4，表3「オブジェクト操作,変更」**

を逆に使い，この図，表のアウトプットが目的=解になるようなオブジェクトの操作,変更をインプットとして求める．これはプロセスオブジェクト削除，プロセスオブジェクト生成，オブジェクト属性変更というオブジェクト変更の論理型毎に行われる．

手段検討，オブジェクト操作を合わせ，設計である．設計とは，実現する目的が決まった後，その実現のためにオブジェクトの内容を決定する運動である．

このように目的設定，オブジェクト操作は，属性変更，プロセスオブジェクト生成，プロセスオブジェクト消滅という三つのオブジェクト変更の論理型毎に行われる．オブジェクト操作を形式的に行うために，そのために必要な目的設定も合わせ本稿のオブジェクト把握と，オブジェクト変更の論理型による形式的な処理が必要だった．これに対し，手段検討は三つの差異解消の目的型という内容に依存する知見によって行われる．

以上の最も密度が細かい一つのオブジェクトの変更だけをする場合の差異解消の全体構造図を図5に示す．

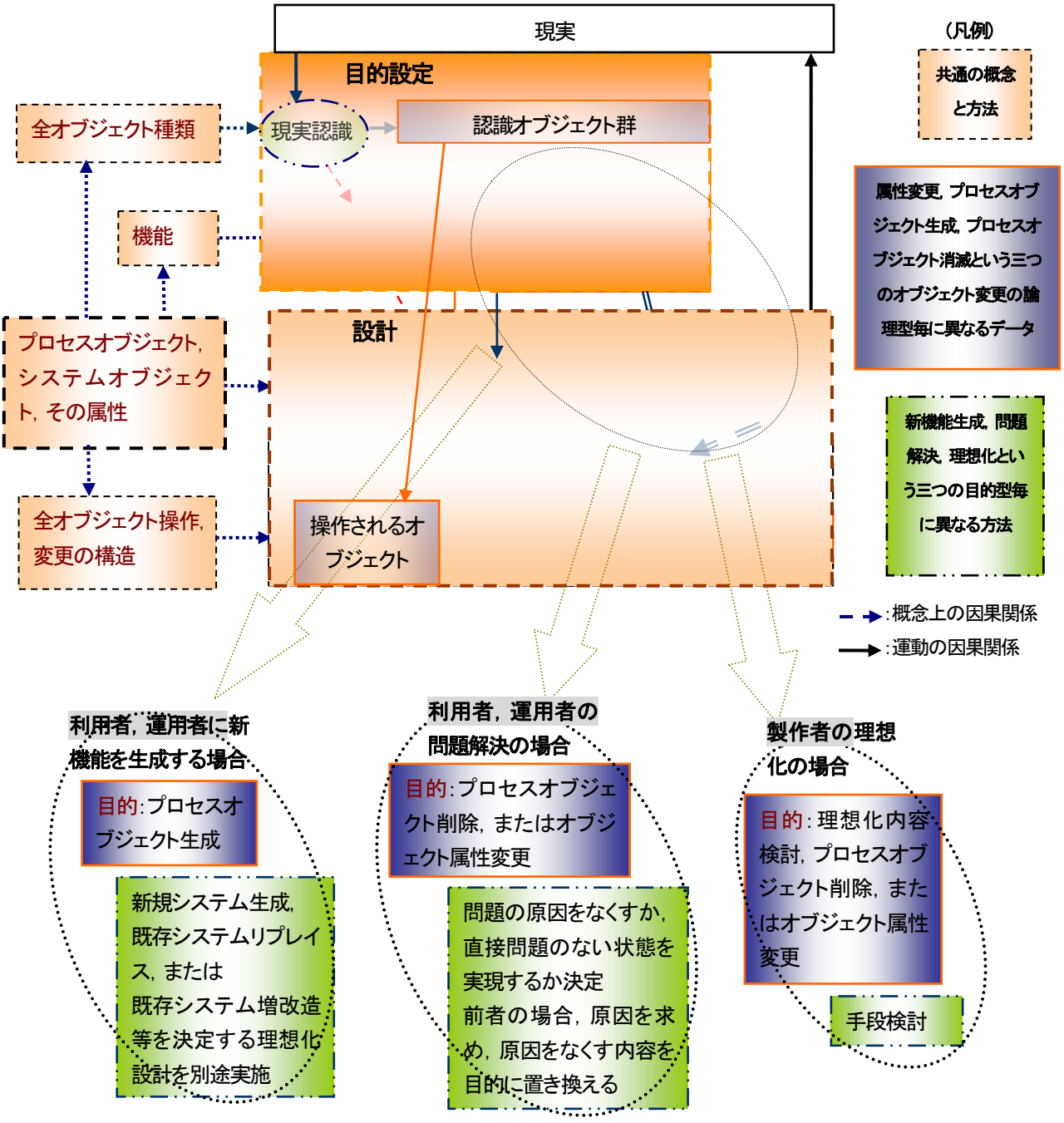

図 5.　一つのオブジェクト変更が目的の場合の差異解消の全体構造図

[2]321 項「差異解消の型」の分類を「差異解消の目的型」として次のように見直す.

[現実システムの**問題を解決する**型]

既存システムの変更の場合だけ存在する.

1. 問題の原因を除去することによる問題解決：例, TRIZ[4], USIT[5], HTA) [6] F－I 章
2. 直接目的を実現する問題解決：例, (通常の)ASIT[7]

[**新しく機能を作る**型]

既存システムの変更の場合と新しくシステムを作る場合の二つがある．新規システム生成，既存システムリプレイス，または既存システム増改造等のいずれが良いかの決定は理想化設計に属する.

3. システムの増改造
4. システムのリプレース
5. 通常の新設計：例, HTA[6] A－E 章
6. 外部から強制的にシステムが作られ新しい機能が追加される場合：例, ASIT の設計プロダクト[8]

[**理想化**の型]

既存システムの変更の場合だけに該当するように見える．しかし例えば新しくシステムを作り新機能の実現をする場合，「原理 P」によるシステムオブジェクト操作の対象と方法選択は理想化の範疇である．以下の二つは背反でない.

7. システムの法則利用による理想化：例, TRIZ
8. 新機能生成, 問題解決後の理想化：例, TRIZ , HTA[6] B－E 章

粒度，密度を大きくしていくと，

・複数のオブジェクト変更の場合

・新しいプロセスオブジェクトの生成，プロセスオブジェクトの除去，オブジェクトの属性変更が混在する場合

・新しい機能の生成，問題解決が混在する場合

が出てくる．理想化はこれらに必ず伴う.

4. 結論

前稿[1][2]と本稿で，全オブジェクトをプロセスオブジェクトを含めて理解し，そのオブジェクトの形式的操作方法を明らかにしたことにより，新機能生成，問題解決，理想化という個々の差異解消におけるオブジェクトの目的設定，オブジェクト操作が，統一した形式のインプット,アウトプットによって行える．これは理想化の一環である.

[2]6 項.「“階層化 TRIZ アルゴリズム”について」で「機能を理想化する場合，問題を解決する場合，矛盾を解決する場合に応じたオブジェクト操作の型はできる限り統一した扱いをすべきである」と書いた．本稿がこの統一した扱いの根拠である.

また制度分野への適用も技術分野の適用論理のアナロジーでない展開が可能になる.

今後の大きな課題として，技術，制度を問わず「理想化」の見直しが残っている．新機能生成と問題解決の分野の理想化は，目的決定が明確，容易でないだけでなく，その設計方法も明らかになっていない.

本稿は，TRIZ の「形式」に関する論文で（これは本稿が TRIZ の論考であることを認めていただいた上での話であるが），従来の TRIZ の貴重な成果の「内容」についてのものではないことを念のため述べておく.

参考文献

[1] 高原 :「オブジェクトの再把握とそのTRIZ, USIT, ASIT への適用」, 第一回TRIZシンポジウム, 2005.09.

[2] 高原 :「機能とプロセスオブジェクト概念を基礎にした差異解消方法」, 第二回 TRIZ シンポジウム, 2006.09.

[3] Takahara Toshio: ‘How People Interact with Objects using TRIZ and ASIT’, The TRIZ journal, Aug.2003.

[4] http://www.triz-journal.com/

[5] 中川:「やさしい USIT による創造的問題解決の方法」, 2005.03.
http://www.osaka-gu.ac.jp/php/nakagawa/TRIZ/jpapers/2005Papers/NakaTRIZUSITSeminar0502/NakaTII050225-USIT.pdf

[6] Larry Ball: ‘Hierarchical TRIZ Algorithms’, The TRIZ journal, 2005.05-2006. 日本語訳, “階層化TRIZアルゴリズム”, 高原, 中川訳, 2006-2007.
http://www.osaka-gu.ac.jp/php/nakagawa/TRIZ/jlectures/2006Lec/BallHTA0601/BallHTA-0.htm

[7] Roni Horowitz: ‘ASIT’s Five Thinking Tools with Examples’, The TRIZ journal, Sept.2001.

[8] Roni Horowitz: ‘Using ASIT to develop new products’, The TRIZ journal, Nov.2001.

[9] 高原ホームページ:
http://www.geocities.jp/takahara_t_ieice/