Introduction to Problem Solving

Kameron Ricks

First Edition, 2012

ISBN 978-81-323-1994-8

Published by:
Learning Press
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter- 1

Characteristics of Difficult Problems

Polytely

Polytely (from Greek roots meaning 'many goals') can be described as frequently, complex problem-solving situations characterized by the presence of not one, but several goals, endings.

Modern societies face an increasing incidence of various complex problems. In other words, the defining characteristics of our complex problems are a large number of variables (complexity) that interact in a nonlinear fashion (connectivity), changing over time (dynamic and time-dependent), and to achieve multiple goals (polytely).

Complexity

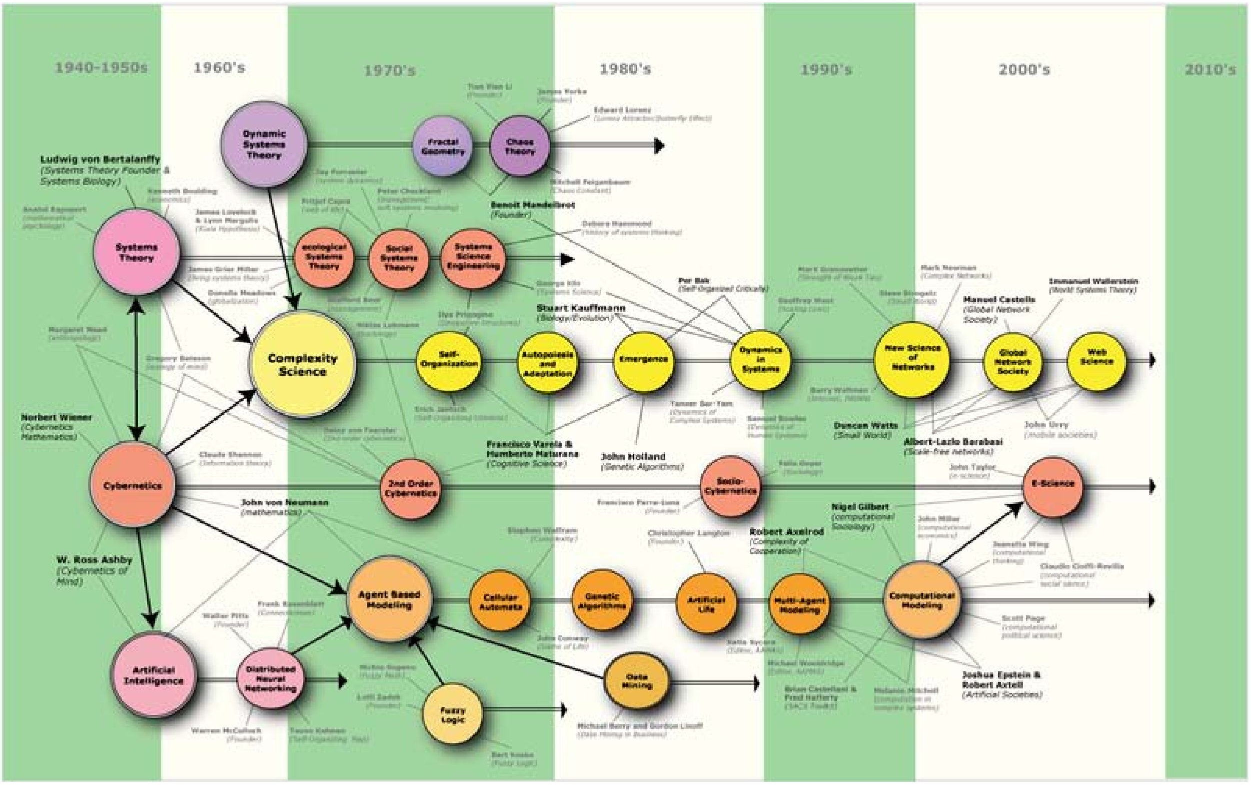
In general usage, **complexity** tends to be used to characterize something with many parts in intricate arrangement. The study of these complex linkages is the main goal of network theory and network science. In science there are at this time a number of approaches to characterizing complexity, many of which are reflected in this article. In a business context, complexity management is the methodology to minimize value-destroying complexity and efficiently control value-adding complexity in a cross-functional approach.

Definitions are often tied to the concept of a "system"—a set of parts or elements which have relationships among them differentiated from relationships with other elements outside the relational regime. Many definitions tend to postulate or assume that complexity expresses a condition of numerous elements in a system and numerous forms of relationships among the elements. At the same time, what is complex and what is simple is relative and changes with time.

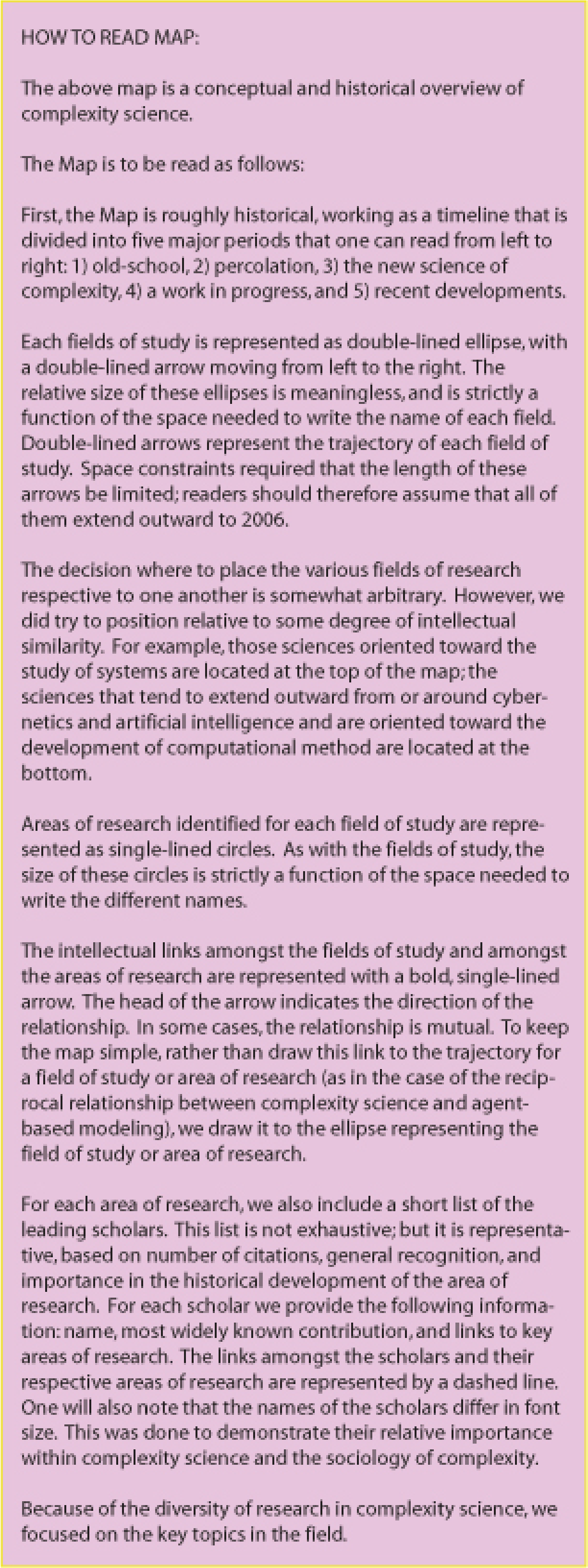
Some definitions key on the question of the probability of encountering a given condition of a system once characteristics of the system are specified. Warren Weaver has posited that the complexity of a particular system is the degree of difficulty in predicting the properties of the system if the properties of the system's parts are given. In Weaver's view, complexity comes in two forms: disorganized complexity, and organized complexity. Weaver's paper has influenced contemporary thinking about complexity.

The approaches which embody concepts of systems, multiple elements, multiple relational regimes, and state spaces might be summarized as implying that complexity arises from the number of distinguishable relational regimes (and their associated state spaces) in a defined system.

Some definitions relate to the algorithmic basis for the expression of a complex phenomenon or model or mathematical expression, as is later set out herein.



Map of Complexity Science. The web version of this map provides internet links to many of the leading scholars and areas of research in complexity science.



HOW TO READ MAP:

The above map is a conceptual and historical overview of complexity science.

The Map is to be read as follows:

First, the Map is roughly historical, working as a timeline that is divided into five major periods that one can read from left to right: 1) old-school, 2) percolation, 3) the new science of complexity, 4) a work in progress, and 5) recent developments.

Each fields of study is represented as double-lined ellipse, with a double-lined arrow moving from left to the right. The relative size of these ellipses is meaningless, and is strictly a function of the space needed to write the name of each field. Double-lined arrows represent the trajectory of each field of study. Space constraints required that the length of these arrows be limited; readers should therefore assume that all of them extend outward to 2006.

The decision where to place the various fields of research respective to one another is somewhat arbitrary. However, we did try to position relative to some degree of intellectual similarity. For example, those sciences oriented toward the study of systems are located at the top of the map; the sciences that tend to extend outward from or around cybernetics and artificial intelligence and are oriented toward the development of computational method are located at the bottom.

Areas of research identified for each field of study are represented as single-lined circles. As with the fields of study, the size of these circles is strictly a function of the space needed to write the different names.

The intellectual links amongst the fields of study and amongst the areas of research are represented with a bold, single-lined arrow. The head of the arrow indicates the direction of the relationship. In some cases, the relationship is mutual. To keep the map simple, rather than draw this link to the trajectory for a field of study or area of research (as in the case of the reciprocal relationship between complexity science and agent-based modeling), we draw it to the ellipse representing the field of study or area of research.

For each area of research, we also include a short list of the leading scholars. This list is not exhaustive; but it is representative, based on number of citations, general recognition, and importance in the historical development of the area of research. For each scholar we provide the following information: name, most widely known contribution, and links to key areas of research. The links amongst the scholars and their respective areas of research are represented by a dashed line. One will also note that the names of the scholars differ in font size. This was done to demonstrate their relative importance within complexity science and the sociology of complexity.

Because of the diversity of research in complexity science, we focused on the key topics in the field.

Map legend

Disorganized complexity vs. organized complexity

One of the problems in addressing complexity issues has been distinguishing conceptually between the large number of variances in relationships extant in random collections, and the sometimes large, but smaller, number of relationships between elements in systems where constraints (related to correlation of otherwise independent elements) simultaneously reduce the variations from element independence and create distinguishable regimes of more-uniform, or correlated, relationships, or interactions.

Weaver perceived and addressed this problem, in at least a preliminary way, in drawing a distinction between "disorganized complexity" and "organized complexity".

In Weaver's view, disorganized complexity results from the particular system having a very large number of parts, say millions of parts, or many more. Though the interactions of the parts in a "disorganized complexity" situation can be seen as largely random, the properties of the system as a whole can be understood by using probability and statistical methods.

A prime example of disorganized complexity is a gas in a container, with the gas molecules as the parts. Some would suggest that a system of disorganized complexity may be compared, for example, with the (relative) simplicity of the planetary orbits—the latter can be known by applying Newton's laws of motion, though this example involved highly correlated events.

Organized complexity, in Weaver's view, resides in nothing else than the non-random, or correlated, interaction between the parts. These correlated relationships create a differentiated structure which can, as a system, interact with other systems. The coordinated system manifests properties not carried by, or dictated by, individual parts. The organized aspect of this form of complexity vis a vis other systems than the subject system can be said to "emerge," without any "guiding hand".

The number of parts does not have to be very large for a particular system to have emergent properties. A system of organized complexity may be understood in its properties (behavior among the properties) through modeling and simulation, particularly modeling and simulation with computers. An example of organized complexity is a city neighborhood as a living mechanism, with the neighborhood people among the system's parts.

Sources and factors of complexity

The source of disorganized complexity is the large number of parts in the system of interest, and the lack of correlation between elements in the system.

There is no consensus at present on general rules regarding the sources of organized complexity, though the lack of randomness implies correlations between elements. Consistent with prior statements here, the number of parts (and types of parts) in the system and the number of relations between the parts would have to be non-trivial—however, there is no general rule to separate "trivial" from "non-trivial".

Complexity of an object or system is a relative property. For instance, for many functions (problems), such a computational complexity as time of computation is smaller when multitape Turing machines are used than when Turing machines with one tape are used. Random Access Machines allow one to even more decrease time complexity (Greenlaw and Hoover 1998: 226), while inductive Turing machines can decrease even the complexity class of a function, language or set (Burgin 2005). This shows that tools of activity can be an important factor of complexity.

Specific meanings of complexity

In several scientific fields, "complexity" has a specific meaning:

- In computational complexity theory, the amounts of resources required for the execution of algorithms is studied. The most popular types of computational complexity are the time complexity of a problem equal to the number of steps that it takes to solve an instance of the problem as a function of the size of the input (usually measured in bits), using the most efficient algorithm, and the space complexity of a problem equal to the volume of the memory used by the algorithm (e.g., cells of the tape) that it takes to solve an instance of the problem as a function of the size of the input (usually measured in bits), using the most efficient algorithm. This allows to classify computational problems by complexity class (such as P, NP ...). An axiomatic approach to computational complexity was developed by Manuel Blum. It allows one to deduce many properties of concrete computational complexity measures, such as time complexity or space complexity, from properties of axiomatically defined measures.

- In algorithmic information theory, the *Kolmogorov complexity* (also called *descriptive complexity*, *algorithmic complexity* or *algorithmic entropy*) of a string is the length of the shortest binary program which outputs that string. Different kinds of Kolmogorov complexity are studied: the uniform complexity, prefix complexity, monotone complexity, time-bounded Kolmogorov complexity, and space-bounded Kolmogorov complexity. An axiomatic approach to Kolmogorov complexity based on Blum axioms (Blum 1967) was introduced by Mark Burgin in the paper presented for publication by Andrey Kolmogorov (Burgin 1982). The axiomatic approach encompasses other approaches to Kolmogorov complexity. It is possible to treat different kinds of Kolmogorov complexity as particular cases of axiomatically defined generalized Kolmogorov complexity. Instead, of proving similar theorems, such as the basic invariance theorem, for each particular measure, it is possible to easily deduce all such results from one corresponding theorem proved in the axiomatic setting. This is a general advantage of the axiomatic approach in mathematics. The axiomatic approach to Kolmogorov complexity was further developed in the book (Burgin 2005) and applied to software metrics (Burgin and Debnath, 2003; Debnath and Burgin, 2003).

- In information processing, complexity is a measure of the total number of properties transmitted by an object and detected by an observer. Such a collection of properties is often referred to as a state.

- In business, complexity describes the variances and their consequences in various fields such as product portfolio, technologies, markets and market segments, locations, manufacturing network, customer portfolio, IT systems, organization, processes etc.

- In physical systems, complexity is a measure of the probability of the state vector of the system. This should not be confused with entropy; it is a distinct mathematical measure, one in which two distinct states are never conflated and considered equal, as is done for the notion of entropy statistical mechanics.

- In mathematics, Krohn-Rhodes complexity is an important topic in the study of finite semigroups and automata.

- In software engineering, programming complexity is a measure of the interactions of the various elements of the software. This differs from the computational complexity described above in that it is a measure of the design of the software.

There are different specific forms of complexity:

- In the sense of how complicated a problem is from the perspective of the person trying to solve it, limits of complexity are measured using a term from cognitive psychology, namely the hrair limit.

- Complex adaptive system denotes systems which have some or all of the following attributes
  - The number of parts (and types of parts) in the system and the number of relations between the parts is non-trivial – however, there is no general rule to separate "trivial" from "non-trivial";
  - The system has memory or includes feedback;
  - The system can adapt itself according to its history or feedback;
  - The relations between the system and its environment are non-trivial or non-linear;
  - The system can be influenced by, or can adapt itself to, its environment; and
  - The system is highly sensitive to initial conditions.

Study of complexity

Complexity has always been a part of our environment, and therefore many scientific fields have dealt with complex systems and phenomena. Indeed, some would say that only what is somehow complex—what displays variation without being random—is worthy of interest.

The use of the term complex is often confused with the term complicated. In today's systems, this is the difference between myriad connecting "stovepipes" and effective "integrated" solutions. This means that complex is the opposite of independent, while complicated is the opposite of simple.

While this has led some fields to come up with specific definitions of complexity, there is a more recent movement to regroup observations from different fields to study complexity in itself, whether it appears in anthills, human brains, or stock markets. One such interndisciplinary group of fields is relational order theories.

Complexity topics

Complex behaviour

The behaviour of a complex system is often said to be due to emergence and self-organization. Chaos theory has investigated the sensitivity of systems to variations in initial conditions as one cause of complex behaviour.

Complex mechanisms

Recent developments around artificial life, evolutionary computation and genetic algorithms have led to an increasing emphasis on complexity and complex adaptive systems.

Complex simulations

In social science, the study on the emergence of macro-properties from the micro-properties, also known as macro-micro view in sociology. The topic is commonly recognized as social complexity that is often related to the use of computer simulation in social science, i.e.: computational sociology.

Complex systems

Systems theory has long been concerned with the study of complex systems (In recent times, *complexity theory* and *complex systems* have also been used as names of the field). These systems can be biological, economic, technological, etc. Recently, complexity is a natural domain of interest of the real world socio-cognitive systems and emerging systemics research. Complex systems tend to be high-dimensional, non-linear and hard to model. In specific circumstances they may exhibit low dimensional behaviour.

Complexity in data

In information theory, algorithmic information theory is concerned with the complexity of strings of data.

Complex strings are harder to compress. While intuition tells us that this may depend on the codec used to compress a string (a codec could be theoretically created in any arbitrary language, including one in which the very small command "X" could cause the computer to output a very complicated string like "18995316"), any two Turing-complete languages can be implemented in each other, meaning that the length of two encodings in different languages will vary by at most the length of the "translation" language—which will end up being negligible for sufficiently large data strings.

These algorithmic measures of complexity tend to assign high values to random noise. However, those studying complex systems would not consider randomness as complexity.

Information entropy is also sometimes used in information theory as indicative of complexity.

Applications of complexity

Computational complexity theory is the study of the complexity of problems—that is, the difficulty of solving them. Problems can be classified by complexity class according to the time it takes for an algorithm—usually a computer program—to solve them as a function of the problem size. Some problems are difficult to solve, while others are easy. For example, some difficult problems need algorithms that take an exponential amount of time in terms of the size of the problem to solve. Take the travelling salesman problem, for example. It can be solved in time $O(n^2 2^n)$ (where n is the size of the network to visit—let's say the number of cities the travelling salesman must visit exactly once). As the size of the network of cities grows, the time needed to find the route grows (more than) exponentially.

Even though a problem may be computationally solvable in principle, in actual practice it may not be that simple. These problems might require large amounts of time or an inordinate amount of space. Computational complexity may be approached from many different aspects. Computational complexity can be investigated on the basis of time, memory or other resources used to solve the problem. Time and space are two of the most important and popular considerations when problems of complexity are analyzed.

There exist a certain class of problems that although they are solvable in principle they require so much time or space that it is not practical to attempt to solve them. These problems are called intractable.

There is another form of complexity called hierarchical complexity. It is orthogonal to the forms of complexity discussed so far, which are called horizontal complexity

Recursively enumerable set

In computability theory, traditionally called recursion theory, a set S of natural numbers is called **recursively enumerable**, **computably enumerable**, **semidecidable**, **provable** or **Turing-recognizable** if:

- There is an algorithm such that the set of input numbers for which the algorithm halts is exactly the set of numbers in S.

Or, equivalently,

- There is an algorithm that enumerates the members of S. That means that its output is simply a list of the members of S: s_1, s_2, s_3, ... If necessary, this algorithm may run forever.

The first condition suggests why the term *semidecidable* is sometimes used; the second suggests why *computably enumerable* is used. The abbreviations **r.e.** and **c.e.** are often used, even in print, instead of the full phrase.

In computational complexity theory, the complexity class containing all recursively enumerable sets is RE. In recursion theory, the lattice of r.e. sets under inclusion is denoted $\mathcal{E}$.

Formal definition

A set *S* of natural numbers is called **recursively enumerable** if there is a partial recursive function (synonymously, a partial computable function) whose domain is exactly *S*, meaning that the function is defined if and only if its input is a member of *S*.

The definition can be extended to an arbitrary countable set *A* by using Gödel numbers to represent elements of the set and declaring a subset of *A* to be recursively enumerable if the set of corresponding Gödel numbers is recursively enumerable.

Equivalent formulations

The following are all equivalent properties of a set *S* of natural numbers:

Semidecidability:

- The set *S* is recursively enumerable. That is, *S* is the domain (co-range) of a partial recursive function.
- There is a partial recursive function *f* such that:

$$f(x) = \begin{cases} 0 & \text{if } x \in S \\ \text{undefined/does not halt} & \text{if } x \notin S \end{cases}$$

Enumerability:

- The set *S* is the range of a partial recursive function.
- The set *S* is the range of a total recursive function or empty. If *S* is infinite, the function can be chosen to be injective.
- The set *S* is the range of a primitive recursive function or empty. Even if *S* is infinite, repetition of values may be necessary in this case.

Diophantine:

- There is a polynomial *p* with integer coefficients and variables *x*, *a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*, *i* ranging over the natural numbers such that

$$x \in S \Leftrightarrow \exists a, b, c, d, e, f, g, h, i \ (p(x, a, b, c, d, e, f, g, h, i) = 0).$$

- There is a polynomial from the integers to the integers such that the set *S* contains exactly the non-negative numbers in its range.

The equivalence of semidecidability and enumerability can be obtained by the technique of dovetailing.

The Diophantine characterizations of a recursively enumerable set, while not as straightforward or intuitive as the first definitions, were found by Yuri Matiyasevich as part of the negative solution to Hilbert's Tenth Problem. Diophantine sets predate recursion theory and are therefore historically the first way to describe these sets (although this equivalence was only remarked more than three decades after the introduction of recursively enumerable sets). The number of bound variables in the above definition of the Diophantine set is the best known so far; it might be that a lower number can be used to define all diophantine sets.

Examples

- Every recursive set is recursively enumerable, but it is not true that every recursively enumerable set is recursive.
- A recursively enumerable language is a recursively enumerable subset of a formal language.
- The set of all provable sentences in an effectively presented axiomatic system is a recursively enumerable set.
- Matiyasevich's theorem states that every recursively enumerable set is a Diophantine set (the converse is trivially true).
- The simple sets are recursively enumerable but not recursive.
- The creative sets are recursively enumerable but not recursive.
- Any productive set is **not** recursively enumerable.
- Given a Gödel numbering φ of the computable functions, the set $\{\langle i,x\rangle \mid \phi_i(x)\downarrow\}$ (where $\langle i,x\rangle$is the Cantor pairing function and $\phi_i(x)\downarrow$indicates $\varphi_i(x)$ is defined) is recursively enumerable. This set encodes the halting problem as it describes the input parameters for which each Turing machine halts.
- Given a Gödel numbering φ of the computable functions, the set $\{\langle x,y,z\rangle \mid \phi_x(y)=z\}$is recursively enumerable. This set encodes the problem of deciding a function value.
- Given a partial function *f* from the natural numbers into the natural numbers, *f* is a partial recursive function if and only if the graph of *f*, that is, the set of all pairs $\langle x,f(x)\rangle$such that *f(x)* is defined, is recursively enumerable.

Properties

If *A* and *B* are recursively enumerable sets then *A* ∩ *B*, *A* ∪ *B* and *A* × *B* (with the ordered pair of natural numbers mapped to a single natural number with the Cantor pairing function) are recursively enumerable sets. The preimage of a recursively enumerable set under a partial recursive function is a recursively enumerable set.

A set is recursively enumerable if and only if it is at level $\Sigma_1^0$of the arithmetical hierarchy.

A set *T* is called **co-recursively enumerable** or **co-r.e.** if its complement $\mathbb{N} \setminus T$is recursively enumerable. Equivalently, a set is co-r.e. if and only if it is at level $\Pi_1^0$of the arithmetical hierarchy.

A set *A* is recursive (synonym: computable) if and only if both *A* and the complement of *A* are recursively enumerable. A set is recursive if and only if it is either the range of an increasing total recursive function or finite.

Some pairs of recursively enumerable sets are effectively separable and some are not.

Remarks

According to the Church-Turing thesis, any effectively calculable function is calculable by a Turing machine, and thus a set *S* is recursively enumerable if and only if there is some algorithm which yields an enumeration of *S*. This cannot be taken as a formal definition, however, because the Church-Turing thesis is an informal conjecture rather than a formal axiom.

The definition of a recursively enumerable set as the *domain* of a partial function, rather than the *range* of a total recursive function, is common in contemporary texts. This choice is motivated by the fact that in generalized recursion theories, such as α-recursion theory, the definition corresponding to domains has been found to be more natural. Other texts use the definition in terms of enumerations, which is equivalent for recursively enumerable sets.

Chapter- 2

Problem-Solving Techniques

Abstraction

Abstraction is a conceptual process by which higher, more conceptual concepts are derived from the usage and classification of literal (ie. "real" or "concrete") concepts. An "abstraction" (noun) is a concept that acts as super-categorical noun for all subordinate concepts, and connects any related concepts as a *group, field,* or *category*.

Abstractions may be formed by reducing the information content of a concept or an observable phenomenon, typically to retain only information which is relevant for a particular purpose. For example, abstracting a leather soccer ball to the more general idea of a ball retains only the information on general ball attributes and behavior, eliminating the characteristics of that particular ball.

Origins

The first symbols of abstract thinking in humans can be traced to fossils dating between 50,000 and 100,000 years ago in Africa.

Thought process

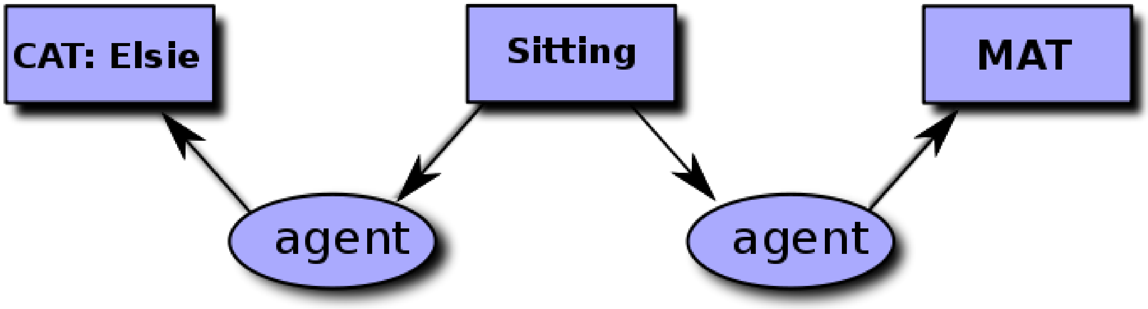
In philosophical terminology, *abstraction* is the thought process wherein ideas are distanced from objects.

Abstraction uses a strategy of simplification, wherein formerly concrete details are left ambiguous, vague, or undefined; thus effective communication about things in the abstract requires an intuitive or common experience between the communicator and the communication recipient. This is true for all verbal/abstract communication.

Cat on Mat *(picture 1)*

For example, many different things can be red. Likewise, many things sit on surfaces (as in *picture 1*, to the right). The property of *redness* and the relation *sitting-on* are therefore abstractions of those objects. Specifically, the conceptual diagram *graph 1* identifies only three boxes, two ellipses, and four arrows (and their six labels), whereas the *picture 1* shows much more pictorial detail, with the scores of implied relationships as implicit in the picture rather than with the nine explicit details in the graph.

Graph 1 details some explicit relationships between the objects of the diagram. For example the arrow between the *agent* and *CAT:Elsie* depicts an example of an *is-a* relationship, as does the arrow between the *location* and the *MAT*. The arrows between the gerund *SITTING* and the nouns *agent* and *location* express the diagram's basic relationship; *"agent is SITTING on location"*; *Elsie* is an instance of *CAT*.



Conceptual graph for A Cat sitting on the Mat *(graph 1)*

Although the description *sitting-on* (graph 1) is more abstract than the graphic image of a cat sitting on a mat (picture 1), the delineation of abstract things from concrete things is somewhat ambiguous; this ambiguity or vagueness is characteristic of abstraction. Thus something as simple as a newspaper might be specified to six levels, as in Douglas Hofstadter's illustration of that ambiguity, with a progression from abstract to concrete in *Gödel, Escher, Bach* (1979):

(1) a publication

(2) a newspaper

(3) *The San Francisco Chronicle*

(4) the May 18 edition of the *Chronicle*

(5) my copy of the May 18 edition of the *Chronicle*

(6) my copy of the May 18 edition of the *Chronicle* as it was when I first picked it up (as contrasted with my copy as it was a few days later: in my fireplace, burning)

An abstraction can thus encapsulate each of these levels of detail with no loss of generality. But perhaps a detective or philosopher/scientist/engineer might seek to learn about some thing, at progressively deeper levels of detail, to solve a crime or a puzzle.

Referents

Abstractions sometimes have ambiguous referents; for example, "happiness" (when used as an abstraction) can refer to as many things as there are people and events or states of being which make them happy. Likewise, "architecture" refers not only to the design of safe, functional buildings, but also to elements of creation and innovation which aim at elegant solutions to construction problems, to the use of space, and to the attempt to evoke an emotional response in the builders, owners, viewers and users of the building.

Instantiation

Things that do not exist at any particular place and time are often considered abstract. By contrast, instances, or members, of such an abstract thing might exist in many different places and times. Those abstract things are then said to be *multiply instantiated*, in the sense of *picture 1*, *picture 2*, etc., shown above.

It is not sufficient, however, to define *abstract* ideas as those that can be instantiated and to define *abstraction* as the movement in the opposite direction to instantiation. Doing so would make the concepts "cat" and "telephone" abstract ideas since despite their varying appearances, a particular cat or a particular telephone is an instance of the concept "cat" or the concept "telephone". Although the concepts "cat" and "telephone" are *abstractions*, they are not *abstract* in the sense of the objects in *graph 1* above.

We might look at other graphs, in a progression from *cat* to *mammal* to *animal*, and see that *animal* is more abstract than *mammal*; but on the other hand *mammal* is a harder idea to express, certainly in relation to *marsupial* or *monotreme*.

Physicality

A physical object (a possible referent of a concept or word) is considered *concrete* (not abstract) if it is a *particular individual* that occupies a particular place and time.

Abstract things are sometimes defined as those things that do not exist in reality or exist only as sensory experiences, like the color red. That definition, however, suffers from the difficulty of deciding which things are real (i.e. which things exist in reality). For example, it is difficult to agree to whether concepts like *God*, *the number three*, and *goodness* are real, abstract, or both.

An approach to resolving such difficulty is to use *predicates* as a general term for whether things are variously real, abstract, concrete, or of a particular property (e.g. *good*). Questions about the properties of things are then propositions about predicates, which propositions remain to be evaluated by the investigator. In the *graph 1* above, the graphical relationships like the arrows joining boxes and ellipses might denote predicates. Different levels of abstraction might be denoted by a progression of arrows joining boxes or ellipses in multiple rows, where the arrows point from one row to another, in a series of other graphs, say graph 2, etc.

Abstraction used in philosophy

Abstraction in philosophy is the process (or, to some, the alleged process) in concept-formation of recognizing some set of common features in individuals, and on that basis forming a concept of that feature. The notion of abstraction is important to understanding some philosophical controversies surrounding empiricism and the problem of universals. It has also recently become popular in formal logic under predicate abstraction. Another philosophical tool for discussion of abstraction is thought space.

Ontological status

The way that physical objects, like rocks and trees, have being differs from the way that properties of abstract concepts or relations have being, for example the way the concrete, particular, individuals pictured in *picture 1* exist differs from the way the concepts illustrated in *graph 1* exist. That difference accounts for the ontological usefulness of the word "abstract". The word applies to properties and relations to mark the fact that, if they exist, they do not exist in space or time, but that instances of them can exist, potentially in many different places and times.

Perhaps confusingly, some philosophies refer to *tropes* (instances of properties) as *abstract particulars*. E.g., the particular redness of a particular apple is an *abstract particular*. Akin to qualia and sumbebekos.

In linguistics

Reification, also called *hypostatization*, might be considered a formal fallacy whenever an abstract concept, such as "society" or "technology" is treated as if it were a concrete object. In linguistics this is called *metonymy*, in which abstract concepts are referred to using the same sorts of nouns that signify concrete objects. Metonymy is an aspect of the English language and of other languages. It can blur the distinction between abstract and concrete things:

1805: Horatio Nelson (*Battle of Trafalgar*) - "England expects that every man will do his duty"

Compression

An abstraction can be seen as a process of mapping multiple different pieces of constituent data to a single piece of abstract data based on similarities in the constituent data, for example many different physical cats map to the abstraction "CAT". This conceptual scheme emphasizes the inherent equality of both constituent and abstract data, thus avoiding problems arising from the distinction between "abstract" and "concrete". In this sense the process of abstraction entails the identification of similarities between objects and the process of associating these objects with an abstraction (which is itself an object).

For example, *picture 1* above illustrates the concrete relationship "Cat sits on Mat".

Chains of abstractions can therefore be constructed moving from neural impulses arising from sensory perception to basic abstractions such as color or shape to experiential abstractions such as a specific cat to semantic abstractions such as the "idea" of a CAT to classes of objects such as "mammals" and even categories such as "object" as opposed to "action".

For example, *graph 1* above expresses the abstraction "agent sits on location".

This conceptual scheme entails no specific hierarchical taxonomy (such as the one mentioned involving cats and mammals), only a progressive exclusion of detail.

The neurology of abstraction

A recent meta-analysis suggests that the verbal system has greater engagement for abstract concepts when the perceptual system is more engaged for processing of concrete concepts. This is because abstract concepts elicit greater brain activity in the inferior frontal gyrus and middle temporal gyrus compared to concrete concepts when concrete concepts elicit greater activity in the posterior cingulate, precuneus, fusiform gyrus, and parahippocampal gyrus.

Other research into the human brain suggests that the left and right hemispheres differ in their handling of abstraction. For example, one meta-analysis reviewing human brain lesions has shown a left hemisphere bias during tool usage.

Abstraction in art

Typically, *abstraction* is used in the arts as a synonym for abstract art in general. Strictly speaking, it refers to art unconcerned with the literal depiction of things from the visible world—it can, however, refer to an object or image which has been distilled from the real world, or indeed, another work of art. Artwork that reshapes the natural world for expressive purposes is called abstract; that which derives from, but does not imitate a recognizable subject is called nonobjective abstraction. In the 20th century the trend toward abstraction coincided with advances in science, technology, and changes in urban life, eventually reflecting an interest in psychoanalytic theory. Later still, abstraction was manifest in more purely formal terms, such as color, freed from objective context, and a reduction of form to basic geometric designs.

In music, the term *abstraction* can be used to describe improvisatory approaches to interpretation, and may sometimes indicate abandonment of tonality. Atonal music has no key signature, and is characterized the exploration of internal numeric relationships.

Abstraction in psychology

Carl Jung's definition of abstraction broadened its scope beyond the thinking process to include exactly four mutually exclusive, opposing complementary psychological functions: sensation, intuition, feeling, and thinking. Together they form a structural totality of the differentiating abstraction process. Abstraction operates in one of these opposing functions when it excludes the simultaneous influence of the other functions and other irrelevancies, such as emotion. Abstraction requires selective use of this structural split of abilities in the psyche. The opposite of abstraction is concretism. *Abstraction* is one of Jung's 57 definitions in Chapter XI of *Psychological Types*.

There is an abstract *thinking*, just as there is abstract *feeling*, *sensation* and *intuition*. Abstract thinking singles out the rational, logical qualities ... Abstract feeling does the same with ... its feeling-values. ... I put abstract feelings on the same level as abstract thoughts. ... Abstract sensation would be aesthetic as opposed to sensuous *sensation* and abstract intuition would be symbolic as opposed to fantastic *intuition*.

Abstraction in computer science

Computer scientists use abstraction to understand and solve problems and communicate their solutions with the computer in some particular computer language. Abstraction allows program designers to separate categories and concepts from instances of implementation, so that they do not depend on software or hardware.

Abstraction in mathematics

Abstraction in mathematics is the process of extracting the underlying essence of a mathematical concept, removing any dependence on real world objects with which it might originally have been connected, and generalizing it so that it has wider applications or matching among other abstract descriptions of equivalent phenomena.

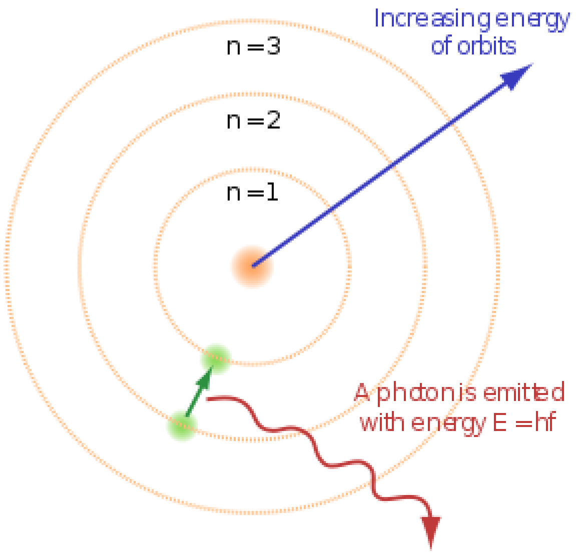
The advantages of abstraction in mathematics are:

- it reveals deep connections between different areas of mathematics
- known results in one area can suggest conjectures in a related area
- techniques and methods from one area can be applied to prove results in a related area

The main disadvantage of abstraction is that highly abstract concepts are more difficult to learn, and require a degree of mathematical maturity and experience before they can be assimilated.

Analogy

Analogy (from Greek "ἀναλογία" – *analogia*, "proportion") is a cognitive process of transferring information or meaning from a particular subject (the analogue or source) to another particular subject (the target), and a linguistic expression corresponding to such a process. In a narrower sense, analogy is an inference or an argument from one particular to another particular, as opposed to deduction, induction, and abduction, where at least one of the premises or the conclusion is general. The word *analogy* can also refer to the relation between the source and the target themselves, which is often, though not necessarily, a similarity, as in the biological notion of analogy.


Increasing energy of orbits
n = 3
n = 2
n = 1
A photon is emitted with energy E = hf

Niels Bohr's model of the atom, though inaccurate, made an analogy between the atom and the solar system.

Analogy plays a significant role in problem solving, decision making, perception, memory, creativity, emotion, explanation and communication. It lies behind basic tasks such as the identification of places, objects and people, for example, in face perception and facial recognition systems. It has been argued that analogy is "the core of cognition". Specific analogical language comprises exemplification, comparisons, metaphors, similes, allegories, and parables, but *not* metonymy. Phrases like *and so on*, *and the like*, *as if*, and the very word *like* also rely on an analogical understanding by the receiver of a message including them. Analogy is important not only in ordinary language and common sense (where proverbs and idioms give many examples of its application) but also in science, philosophy and the humanities. The concepts of association, comparison, correspondence, mathematical and morphological homology, homomorphism, iconicity, isomorphism, metaphor, resemblance, and similarity are closely related to analogy. In cognitive linguistics, the notion of conceptual metaphor may be equivalent to that of analogy.

Analogy has been studied and discussed since classical antiquity by philosophers, scientists and lawyers. The last few decades have shown a renewed interest in analogy, most notable in cognitive science.

Usage of the terms *source* and *target*

With respect to the terms *source* and *target* there are two distinct traditions of usage:

- The logical and mathematical tradition speaks of an *arrow*, *homomorphism*, *mapping*, or *morphism* from what is typically the more complex *domain* or *source* to what is typically the less complex *codomain* or *target*, using all of these words in the sense of mathematical category theory.

- The tradition that appears to be more common in cognitive psychology, literary theory, and specializations within philosophy outside of logic, speaks of a mapping from what is typically the more familiar area of experience, the *source*, to what is typically the more problematic area of experience, the *target*.

Models and theories

Identity of relation

In ancient Greek the word *αναλογια* (*analogia*) originally meant proportionality, in the mathematical sense, and it was indeed sometimes translated to Latin as *proportio*. From there analogy was understood as **identity of relation** between any two ordered pairs, whether of mathematical nature or not. Kant's *Critique of Judgment* held to this notion. Kant argued that there can be exactly the same relation between two completely different objects. The same notion of analogy was used in the US-based SAT tests, that included "analogy questions" in the form "A is to B as C is to *what*?" For example, "Hand is to palm as foot is to ____?" These questions were usually given in the Aristotelian format:

HAND : PALM : : FOOT : ____

While most competent English speakers will immediately give the right answer to the analogy question (*sole*), it is more difficult to identify and describe the exact relation that holds both between *hand* and *palm*, and between *foot* and *sole*. This relation is not apparent in some lexical definitions of *palm* and *sole*, where the former is defined as *the inner surface of the hand*, and the latter as *the underside of the foot*. Analogy and abstraction are different cognitive processes, and analogy is often an easier one.

Recently a computer algorithm has achieved human-level performance on multiple-choice analogy questions from the SAT test. The algorithm measures the similarity of relations between pairs of words (e.g., the similarity between the pairs HAND:PALM and FOOT:SOLE) by statistical analysis of a large collection of text. It answers SAT questions by selecting the choice with the highest relational similarity.

Shared abstraction

In several cultures, the sun is the source of an analogy to God

Greek philosophers such as Plato and Aristotle actually used a wider notion of analogy. They saw analogy as a **shared abstraction**. Analogous objects did not share necessarily a relation, but also an idea, a pattern, a regularity, an attribute, an effect or a function. These authors also accepted that comparisons, metaphors and "images" (allegories) could be used as arguments, and sometimes they called them *analogies*. Analogies should also make those abstractions easier to understand and give confidence to the ones using them.

The Middle Ages saw an increased use and theorization of analogy. Roman lawyers had already used analogical reasoning and the Greek word *analogia*. Medieval lawyers distinguished *analogia legis* and *analogia iuris* (see below). In Islamic logic, analogical reasoning was used for the process of Qiyas in Islamic sharia law and fiqh jurisprudence. In Christian theology, analogical arguments were accepted in order to explain the attributes of God. Aquinas made a distinction between *equivocal*, *univocal* and *analogical* terms, the latter being those like *healthy* that have different but related meanings. Not only a person can be "healthy", but also the food that is good for health. Thomas Cajetan wrote an influential treatise on analogy. In all of these cases, the wide Platonic and Aristotelian notion of analogy was preserved. James Francis Ross in *Portraying Analogy* (1982), the first substantive examination of the topic since Cajetan's *De Nominum Analogia*, demonstrated that analogy is a systematic and universal feature of natural languages, with identifiable and law-like characteristics which explain how the meanings of words in a sentence are interdependent.

Special case of induction

On the contrary, Ibn Taymiyya, Francis Bacon and later John Stuart Mill argued that analogy is simply **a special case of induction**. In their view analogy is an inductive inference from common known attributes to another probable common attribute, which is known only about the source of the analogy, in the following form:

Premises

a is C, D, E, F, G

b is C, D, E, F

Conclusion

b is probably G.

Alternative conclusion

every C, D, E, F is probably G.

This view does not accept analogy as an autonomous mode of thought or inference, reducing it to induction. However, autonomous analogical arguments are still useful in science, philosophy and the humanities (see below), which makes this reduction philosophically uninteresting. Moreover, induction tries to achieve general conclusions, while analogy looks for particular ones.

Hidden deduction

The opposite move could also be tried, **reducing analogy to deduction**. It is argued that every analogical argument is partially superfluous and can be rendered as a deduction stating as a premise a (previously hidden) universal proposition which applied both to the source and the target. In this view, instead of an argument with the form:

Premises

a is analogous to *b*.

b is F.

Conclusion

a is plausibly F.

We should have:

Hidden universal premise

all Gs are plausibly Fs.

Hidden singular premise

a is G.

Conclusion

a is plausibly F.

This would mean that premises referring the source and the analogical relation are themselves superfluous. However, it is not always possible to find a plausibly true universal premise to replace the analogical premises. And analogy is not only an argument, but also a distinct cognitive process.

Shared structure

According to Shelley (2003), the study of the coelacanth drew heavily on analogies from other fish

Contemporary cognitive scientists use a wide notion of analogy, extensionally close to that of Plato and Aristotle, but framed by Gentner's (1983) **structure mapping theory**. The same idea of mapping between source and target is used by conceptual metaphor and conceptual blending theorists. Structure mapping theory concerns both psychology and computer science. According to this view, analogy depends on the mapping or alignment of the elements of source and target. The mapping takes place not only between objects, but also between relations of objects and between relations of relations. The whole mapping yields the assignment of a predicate or a relation to the target. Structure mapping theory has been applied and has found considerable confirmation in psychology. It has had reasonable success in computer science and artificial intelligence (see below). Some studies extended the approach to specific subjects, such as metaphor and similarity.

Keith Holyoak and Paul Thagard (1997) developed their **multiconstraint theory** within structure mapping theory. They defend that the "coherence" of an analogy depends on structural consistency, semantic similarity and purpose. Structural consistency is maximal when the analogy is an isomorphism, although lower levels are admitted. Similarity demands that the mapping connects similar elements and relations of source and target, at any level of abstraction. It is maximal when there are identical relations and when connected elements have many identical attributes. An analogy achieves its purpose insofar as it helps solve the problem at hand. The multiconstraint theory faces some difficulties when there are multiple sources, but these can be overcome. Hummel and Holyoak (2005) recast the multiconstraint theory within a neural network architecture. A problem for the multiconstraint theory arises from its concept of similarity, which, in this respect, is not obviously different from analogy itself. Computer applications demand that there are some *identical* attributes or relations at some level of abstraction. Human analogy does not, or at least not apparently.

Mark T. Keane and Brayshaw (1988) developed their *Incremental Analogy Machine* (IAM) to include working memory constraints as well as structural, semantic and pragmatic constraints, so that a subset of the base analog is selected and mapping from base to target occurs in a serial manner. Empirical evidence shows that human analogical mapping performance is influenced by information presentation order.

High-level perception

Douglas Hofstadter and his team challenged the shared structure theory and mostly its applications in computer science. They argue that there is no line between perception, including high-level perception, and analogical thought. In fact, analogy occurs not only after, but also before and at the same time as high-level perception. In high-level perception, humans make representations by selecting relevant information from low-level stimuli. Perception is necessary for analogy, but analogy is also necessary for high-level perception. Chalmers et al. conclude that analogy ***is*** high-level perception. Forbus et al. (1998) claim that this is only a metaphor. It has been argued (Morrison and Dietrich 1995) that Hofstadter's and Gentner's groups do not defend opposite views, but are instead dealing with different aspects of analogy.

Analogy and Complexity

Antoine Cornuéjols has presented analogy as a *principle of economy* and *computational complexity*.

Reasoning by analogy is a process of, from a given pair *(x,f(x))*, extrapolating the function *f*. In the standard modeling, analogical reasoning involves two "objects": the *source* and the *target*. The target is supposed to be incomplete and in need for a complete description using the source. The target has an existing part S_t and a missing part R_t. We assume that we can isolate a situation of the source S_s, which corresponds to a situation of target S_t, and the result of the source R_s, which correspond to the result of the target R_t. With B_s, the relation between S_s and R_s, we want B_t, the relation between S_t and R_t.

If the source and target are completely known:

Using Kolmogorov complexity *K(x)*, defined as the size of the smallest description of *x* and Solomonoff's approach to induction, Rissanen (89), Wallace & Boulton (68) proposed the principle of Minimum description length. This principle leads to minimize the complexity *K(target| Source)* of producing the target from the source.

This is unattractive in Artificial Intelligence, as it requires a computation over abstract Turing machines. Suppose that M_s and M_t are local theories of the source and the target, available to the observer. The best analogy between a source case a and target case is the analogy that minimizes:

$K(M_s) + K(S_s|M_s) + K(B_s|M_s) + K(M_t|M_s) + K(S_t|M_t) + K(B_t|M_t)$ (1).

If the target is completely unknown:

All models and descriptions M_s, M_t, B_s, S_s, and S_t leading to the minimization of:

$K(M_s) + K(S_s|M_s) + K(B_s|M_s) + K(M_t|M_s) + K(S_t|M_t)$ (2)

are also those who allow to obtain the relationship B_t, and thus the most satisfactory R_t for formula (1).

The analogical hypothesis, which solves an analogy between a source case and a target case, has two parts:

- Analogy, like induction, is a *principle of economy*. The best analogy between two cases is the one which minimizes the amount of information necessary for the derivation of the source from the target (1). Its most fundamental measure is the computational complexity theory.
- When solving or completing a target case with a source case, the parameters which minimize (2) are postulated to minimize (1), and thus, produce the best response.

However, a *cognitive agent* may simply reduce the amount of information necessary for the interpretation of the source and the target, without taking into account the cost of data replication. So, it may prefer to the minimization of (2) the minimization of the following simplified formula:

$K(M_s) + K(B_s|M_s) + K(M_t|M_s)$ (3).

Applications and types

In language

Rhetoric

- An analogy can be a spoken or textual comparison between two words (or sets of words) to highlight some form of semantic similarity between them. Such analogies can be used to strengthen political and philosophical arguments, even when the semantic similarity is weak or non-existent (if crafted carefully for the audience). Analogies are sometimes used to persuade those that cannot detect the flawed or non-existent arguments.

Linguistics

- An analogy can be the linguistic process that reduces word forms perceived as irregular by remaking them in the shape of more common forms that are governed by rules. For example, the English verb *help* once had the preterite *holp* and the past participle *holpen*. These obsolete forms have been discarded and replaced by *helped* by the power of analogy (or by widened application of the productive Verb-*ed* rule.) This is called *leveling.* However, irregular forms can sometimes be created by analogy; one example is the American English past tense form of *dive*: *dove*, formed on analogy with words such as *drive*: *drove*.
- Neologisms can also be formed by analogy with existing words. A good example is *software*, formed by analogy with *hardware*; other analogous neologisms such as *firmware* and *vaporware* have followed. Another example is the humorous term *underwhelm*, formed by analogy with *overwhelm*.
- Analogy is often presented as an alternative mechanism to generative *rules* for explaining productive formation of structures such as words. Others argue that in fact they are the same mechanism, that rules are analogies that have become entrenched as standard parts of the linguistic system, whereas clearer cases of analogy have simply not (yet) done so (e.g. Langacker 1987.445–447). This view has obvious resonances with the current views of analogy in cognitive science which are discussed above.

In science

Analogues are often used in theoretical and applied sciences in the form of models or simulations which can be considered as strong analogies. Other much weaker analogies assist in understanding and describing functional behaviours of similar systems. For instance, an analogy commonly used in electronics textbooks compares electrical circuits to hydraulics. Another example is the analog ear based on electrical, electronic or mechanical devices.

Mathematics

Some types of analogies can have a precise mathematical formulation through the concept of isomorphism. In detail, this means that given two mathematical structures of the same type, an analogy between them can be thought of as a bijection between them which preserves some or all of the relevant structure. For example, $\mathbb{R}^2$and $\mathbb{C}$are isomorphic as vector spaces, but the complex numbers, $\mathbb{C}$, have more structure than $\mathbb{R}^2$does – $\mathbb{C}$is a field as well as a vector space.

Category theory takes the idea of mathematical analogy much further with the concept of functors. Given two categories C and D a functor F from C to D can be thought of as an analogy between C and D, because F has to map objects of C to objects of D and arrows of C to arrows of D in such a way that the compositional structure of the two categories is preserved. This is similar to the structure mapping theory of analogy of Dedre Gentner, in that it formalizes the idea of analogy as a function which satisfies certain conditions.

Anatomy

In anatomy, two anatomical structures are considered to be *analogous* when they serve similar functions but are not evolutionarily related, such as the legs of vertebrates and the legs of insects. Analogous structures are the result of convergent evolution and should be contrasted with homologous structures.

Engineering

Often a physical prototype is built to model and represent some other physical object. For example, wind tunnels are used to test scale models of wings and aircraft, which act as an analog to full-size wings and aircraft.

For example, the MONIAC (an analog computer) used the flow of water in its pipes as an analog to the flow of money in an economy.

In normative matters

Morality

Analogical reasoning plays a very important part in morality. This may be in part because morality is supposed to be impartial and fair. If it is wrong to do something in a situation A, and situation B is analogous to A in all relevant features, then it is also wrong to perform that action in situation B. Moral particularism accepts analogical moral reasoning, rejecting both deduction and induction, since only the former can do without moral principles.

Law

In law, analogy is used to resolve issues on which there is no previous authority. A distinction has to be made between analogous reasoning from written law and analogy to precedent case law.

Analogies from codes and statutes

In civil law systems, where the preeminent source of law is legal codes and statutes, a lacuna (a gap) arises when a specific issue is not explicitly dealt with in written law. Judges will try to identify a provision whose purpose applies to the case at hand. That process can reach a high degree of sophistication, as judges sometimes not only look at a specific provision to fill lacunae (gaps), but at several provisions (from which an underlying purpose can be inferred) or at general principles of the law to identify the legislator's value judgement from which the analogy is drawn. Besides the not very frequent filling of lacunae, analogy is very commonly used between different provisions in order to achieve substantial coherence. Analogy from previous judicial decisions is also common, although these decisions are not binding authorities.

Analogies from precedent case law

By contrast, in common law systems, where precedent cases are the primary source of law, analogies to codes and statutes are rare (since those are not seen as a coherent system, but as incursions into the common law). Analogies are thus usually drawn from precedent cases: The judge finds that the facts of another case are similar to the one at hand to an extent that the analogous application of the rule established in the previous case is justified.

Brainstorming

Brainstorming is a group creativity technique designed to generate a large number of ideas for the solution of a problem. In 1953 the method was popularized by Alex Faickney Osborn in a book called *Applied Imagination.* Osborn proposed that groups could double their creative output with brainstorming.

Although brainstorming has become a popular group technique, when applied in a traditional group setting, researchers have not found evidence of its effectiveness for enhancing either quantity or quality of ideas generated. Because of such problems as distraction, social loafing, evaluation apprehension, and production blocking, conventional brainstorming groups are little more effective than other types of groups, and they are actually less effective than individuals working independently. In the Encyclopedia of Creativity, Tudor Rickards, in his entry on brainstorming, summarizes its controversies and indicates the dangers of conflating productivity in group work with quantity of ideas.

Although traditional brainstorming does not increase the productivity of groups (as measured by the number of ideas generated), it may still provide benefits, such as boosting morale, enhancing work enjoyment, and improving team work. Thus, numerous attempts have been made to improve brainstorming or use more effective variations of the basic technique.

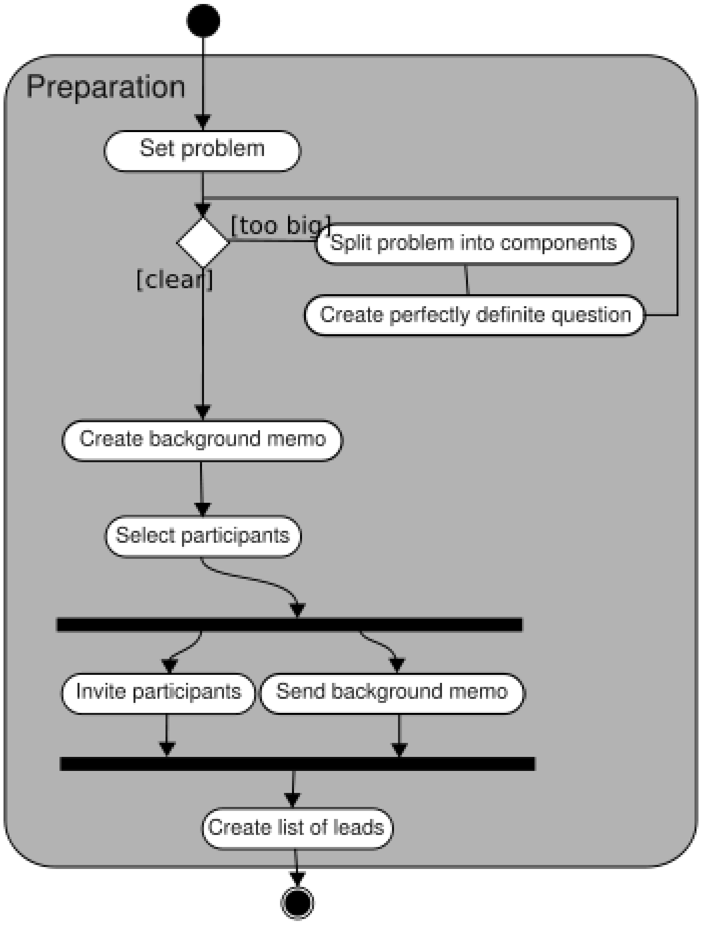
Professor Olivier Toubia of Columbia University has conducted extensive research in the field of idea generation and has concluded that incentives are extremely valuable within the brainstorming context.

From these attempts to improve brainstorming, electronic brainstorming stands out. Mainly through anonymization and parallelization of input, electronic brainstorming enforces the ground rules of effective brainstorming and thereby eliminates most of the deleterious or inhibitive effects of group work. The positive effects of electronic brainstorming become more pronounced with group size.

Ground Rules

There are four basic rules in brainstorming. These are intended to reduce social inhibitions among group members, stimulate idea generation, and increase overall creativity of the group.

1. **Focus on quantity**: This rule is a means of enhancing divergent production, aiming to facilitate problem solving through the maxim *quantity breeds quality*. The assumption is that the greater the number of ideas generated, the greater the chance of producing a radical and effective solution.
2. **Withhold criticism**: In brainstorming, criticism of ideas generated should be put 'on hold'. Instead, participants should focus on extending or adding to ideas, reserving criticism for a later 'critical stage' of the process. By suspending judgment, participants will feel free to generate unusual ideas.
3. **Welcome unusual ideas**: To get a good and long list of ideas, unusual ideas are welcomed. They can be generated by looking from new perspectives and suspending assumptions. These new ways of thinking may provide better solutions.
4. **Combine and improve ideas**: Good ideas may be combined to form a single better good idea, as suggested by the slogan "1+1=3". It is believed to stimulate the building of ideas by a process of association.



Process of preparing for a brainstorming session

Method

Set the problem

Before a brainstorming session, it is critical to define the problem. The problem must be clear, not too big, and captured in a specific question such as "What service for mobile phones is not available now, but needed?". If the problem is too big, the facilitator should break it into smaller components, each with its own question.

Create a background memo

The background memo is the invitation and informational letter for the participants, containing the session name, problem, time, date, and place. The problem is described in the form of a question, and some example ideas are given. The memo is sent to the participants well in advance, so that they can think about the problem beforehand.

Select participants

The facilitator composes the brainstorming panel, consisting of the participants and an idea collector. A group of 10 or fewer members is generally more productive. Many variations are possible but the following composition is suggested.

- Several core members of the project who have proved themselves.
- Several guests from outside the project, with affinity to the problem.
- One idea collector who records the suggested ideas.

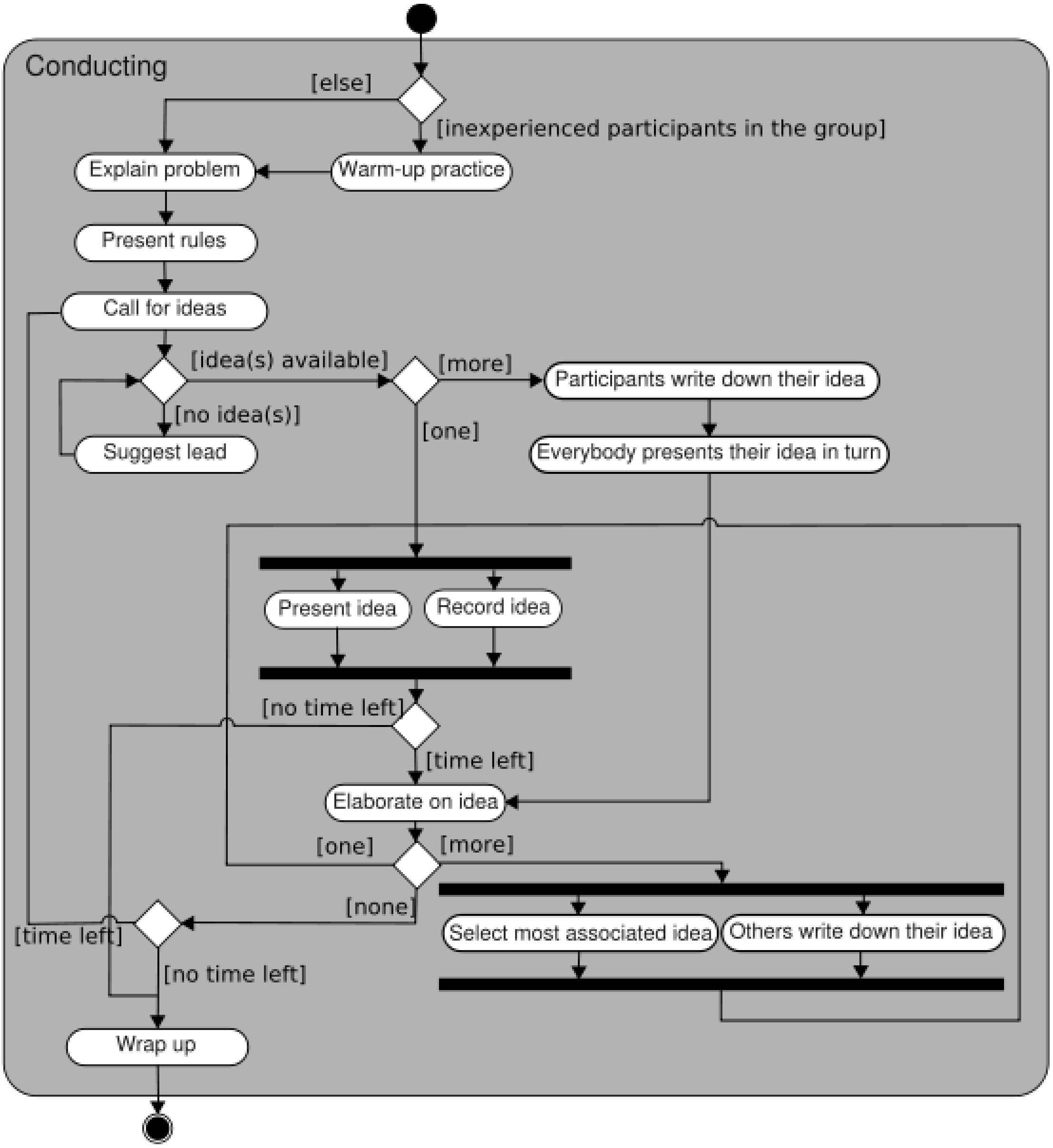
Create a list of lead questions

During the brainstorm session the creativity may decrease. At this moment, the facilitator should stimulate creativity by suggesting a lead question to answer, such as *Can we combine these ideas?* or *How about looking from another perspective?*. It is best to prepare a list of such leads before the session begins.

Session conduct

The facilitator leads the brainstorming session and ensures that ground rules are followed. The steps in a typical session are:

1. A warm-up session, to expose novice participants to the criticism-free environment. A simple problem is brainstormed, for example *What should be the CEO's retirement present?* or *What can be improved in Microsoft Windows?*.
2. The facilitator presents the problem and gives a further explanation if needed.
3. The facilitator asks the brainstorming group for their ideas.
4. If no ideas are forthcoming, the facilitator suggests a lead to encourage creativity.
5. All participants present their ideas, and the idea collector records them.
6. To ensure clarity, participants may elaborate on their ideas.
7. When time is up, the facilitator organizes the ideas based on the topic goal and encourages discussion.
8. Ideas are categorized.
9. The whole list is reviewed to ensure that everyone understands the ideas.
10. Duplicate ideas and obviously infeasible solutions are removed.
11. The facilitator thanks <shubham> and gives each a token of appreciation.



Process of conducting a brainstorming session

The process

- Participants who have ideas but were unable to present them are encouraged to write down the ideas and present them later.
- The idea collector should number the ideas, so that the chairperson can use the number to encourage an idea generation goal, for example: *We have 44 ideas now, let's get it to 50!*.
- The idea collector should repeat the idea in the words he or she has written verbatim, to confirm that it expresses the meaning intended by the originator.
- When many participants are having ideas, the one with the most associated idea should have priority. This is to encourage elaboration on previous ideas.

- During a brainstorming session, managers and other superiors may be discouraged from attending, since it may inhibit and reduce the effect of the four basic rules, especially the generation of unusual ideas.

Evaluation

Brainstorming is not just about generating ideas for others to evaluate and select. Usually the group itself will, in its final stage, evaluate the ideas and select one as the solution to the problem proposed to the group.

- The solution should not require resources or skills the members of the group do not have or cannot acquire.
- If acquiring additional resources or skills is necessary, that needs to be the first part of the solution.
- There must be a way to measure progress and success.
- The steps to carry out the solution must be clear to all, and amenable to being assigned to the members so that each will have an important role.
- There must be a common decision making process to enable a coordinated effort to proceed, and to reassign tasks as the project unfolds.
- There should be evaluations at milestones to decide whether the group is on track toward a final solution.
- There should be incentives to participation so that participants maintain their efforts.

Variations

Nominal group technique

The nominal group technique is a type of brainstorming that encourages all participants to have an equal say in the process. It is also used to generate a ranked list of ideas.

Participants are asked to write their ideas anonymously. Then the moderator collects the ideas and each is voted on by the group. The vote can be as simple as a show of hands in favor of a given idea. This process is called distillation.

After distillation, the top ranked ideas may be sent back to the group or to subgroups for further brainstorming. For example, one group may work on the color required in a product. Another group may work on the size, and so forth. Each group will come back to the whole group for ranking the listed ideas. Sometimes ideas that were previously dropped may be brought forward again once the group has re-evaluated the ideas.

It is important that the facilitator be trained in this process before attempting to facilitate this technique. The group should be primed and encouraged to embrace the process. Like all team efforts, it may take a few practice sessions to train the team in the method before tackling the important ideas.

Group passing technique

Each person in a circular group writes down one idea, and then passes the piece of paper to the next person in a clockwise direction, who adds some thoughts. This continues until everybody gets his or her original piece of paper back. By this time, it is likely that the group will have extensively elaborated on each idea.

The group may also create an "Idea Book" and post a distribution list or routing slip to the front of the book. On the first page is a description of the problem. The first person to receive the book lists his or her ideas and then routes the book to the next person on the distribution list. The second person can log new ideas or add to the ideas of the previous person. This continues until the distribution list is exhausted. A follow-up "read out" meeting is then held to discuss the ideas logged in the book. This technique takes longer, but it allows individuals time to think deeply about the problem.

Team idea mapping method

This method of brainstorming works by the method of association. It may improve collaboration and increase the quantity of ideas, and is designed so that all attendees participate and no ideas are rejected.

The process begins with a well-defined topic. Each participant brainstorms individually, then all the ideas are merged onto one large idea map. During this consolidation phase, participants may discover a common understanding of the issues as they share the meanings behind their ideas. During this sharing, new ideas may arise by the association, and they are added to the map as well. Once all the ideas are captured, the group can prioritize and/or take action.

Electronic brainstorming

Electronic brainstorming is a computerized version of the manual brainstorming technique. It is typically supported by an electronic meeting system (EMS) but simpler forms can also be done via email and may be browser based, or use peer-to-peer software.

With an electronic meeting system, participants share a list of ideas over the Internet. Ideas are entered independently. Contributions become immediately visible to all and are typically anonymized to encourage openness and reduce personal prejudice. Modern EMS also support asynchronous brainstorming sessions over extended periods of time as well as typical follow-up activities in the creative-problem-solving process such as categorization of ideas, elimination of duplicates, assessment and discussion of prioritized or controversial ideas.

Electronic brainstorming eliminates many of the problems of standard brainstorming, such as production blocking and evaluation apprehension. An additional advantage of this method is that all ideas can be archived electronically in their original form, and then retrieved later for further thought and discussion. Electronic brainstorming also enables much larger groups to brainstorm on a topic than would normally be productive in a traditional brainstorming session.

Some web based brainstorming techniques allow contributors to post their comments anonymously through the use of avatars. This technique also allows users to log on over an extended time period, typically one or two weeks, to allow participants some "soak time" before posting their ideas and feedback. This technique has been used particularly in the field of new product development, but can be applied in any number of areas where collecting and evaluating ideas would be useful.

Directed brainstorming

Directed brainstorming is a variation of electronic brainstorming (described above). It can be done manually or with computers. Directed brainstorming works when the solution space (that is, the criteria for evaluating a good idea) is known prior to the session. If known, that criteria can be used to intentionally constrain the ideation process.

In directed brainstorming, each participant is given one sheet of paper (or electronic form) and told the brainstorming question. They are asked to produce one response and stop, then all of the papers (or forms) are randomly swapped among the participants. The participants are asked to look at the idea they received and to create a new idea that improves on that idea based on the initial criteria. The forms are then swapped again and respondents are asked to improve upon the ideas, and the process is repeated for three or more rounds.

In the laboratory, directed brainstorming has been found to almost triple the productivity of groups over electronic brainstorming.

Individual brainstorming

"Individual Brainstorming" is the use of brainstorming on a solitary basis. It typically includes such techniques as free writing, free speaking, word association, and drawing a mind map, which is a visual note taking technique in which people diagram their thoughts. Individual brainstorming is a useful method in creative writing and has been shown to be superior to traditional group brainstorming.

Question Brainstorming

This process involves brainstorming the *questions*, rather than trying to come up with immediate answers and short term solutions. This technique stimulates creativity and promotes everyone's participation because no one has to come up with answers. The answers to the questions form the framework for constructing future action plans. Once the list of questions is set, it may be necessary to prioritize them to reach to the best solution in an orderly way. Another of the problems for brainstorming can be to find the best evaluation methods for a problem.

Brainstorming all the questions has also been called *questorming*.

Conclusion

Brainstorming is a popular method of group interaction in both educational and business settings. Although it does not provide a measurable advantage in creative output, conventional brainstorming is an enjoyable exercise that is typically well received by participants. Electronic brainstorming effectively overcomes barriers inherent in group work like production blocking mainly through anonymization and parallelization of contributions. Other variations of brainstorming that do not require an electronic system may also prove superior to the original technique. How well these methods work, and whether or not they should be classified as brainstorming, are questions that require further research.

Controversy over term

Some governmental organisations (The Welsh Development Agency and the Department of Enterprise, Trade and Investment in Belfast) have reached the conclusion that the term 'brainstorming' is offensive to people with epilepsy and have suggested the alternative "thought-showers". However, research by the National Society for Epilepsy found of those affected by epilepsy questioned, 93% considered the term inoffensive. A specific comment states that changes need not be made since that could promote an undesirable image of epileptics being easily offended.

Statistical hypothesis testing

A **statistical hypothesis test** is a method of making decisions using experimental data. In statistics, a result is called **statistically significant** if it is unlikely to have occurred by chance. The phrase "*test of significance*" was coined by Ronald Fisher: "Critical tests of this kind may be called tests of significance, and when such tests are available we may discover whether a second sample is or is not significantly different from the first."

Hypothesis testing is sometimes called **confirmatory data analysis**, in contrast to exploratory data analysis. In frequency probability, these decisions are almost always made using null-hypothesis tests (i.e., tests that answer the question *Assuming that the null hypothesis is true, what is the probability of observing a value for the test statistic that is at least as extreme as the value that was actually observed?*) One use of hypothesis testing is deciding whether experimental results contain enough information to cast doubt on conventional wisdom.

Statistical hypothesis testing is a key technique of frequentist statistical inference, and is widely used, but also much criticized. While controversial, the Bayesian approach to hypothesis testing is to base rejection of the hypothesis on the posterior probability. Other approaches to reaching a decision based on data are available via decision theory and optimal decisions.

The *critical region* of a hypothesis test is the set of all outcomes which, if they occur, will lead us to decide that there is a difference. That is, cause the null hypothesis to be rejected in favor of the alternative hypothesis. The critical region is usually denoted by C.

Examples

The following examples should solidify these ideas.

Example 1 - Court Room Trial

A statistical test procedure is comparable to a trial; a defendant is considered innocent as long as his guilt is not proven. The prosecutor tries to prove the guilt of the defendant. Only when there is enough charging evidence the defendant is condemned.

In the start of the procedure, there are two hypotheses H_0: "the defendant is innocent", and H_1: "the defendant is guilty". The first one is called *null hypothesis*, and is for the time being accepted. The second one is called *alternative (hypothesis)*. It is the hypothesis one tries to prove.

The hypothesis of innocence is only rejected when an error is very unlikely, because one doesn't want to condemn an innocent defendant. Such an error is called *error of the first kind* (i.e. the condemnation of an innocent person), and the occurrence of this error is controlled to be seldom. As a consequence of this asymmetric behaviour, the *error of the second kind* (setting free a guilty person), is often rather large.

	Null Hypothesis (H_0) is true **He truly is innocent**	**Alternative Hypothesis (H_1) is true** **He truly is guilty**
Accept Null Hypothesis	Right decision	Wrong decision Type II Error
Reject Null Hypothesis	Wrong decision Type I Error	Right decision

Example 2 - Clairvoyant Card Game

A person (the subject) is tested for clairvoyance. He is shown the reverse of a randomly chosen play card 25 times and asked which suit it belongs to. The number of hits, or correct answers, is called X.

As we try to find evidence of his clairvoyance, for the time being the null hypothesis is that the person is not clairvoyant. The alternative is, of course: the person is (more or less) clairvoyant.

If the null hypothesis is valid, the only thing the test person can do is guess. For every card, the probability (relative frequency) of guessing correctly is 1/4. If the alternative is valid, the test subject will predict the suit correctly with probability greater than 1/4. We will call the probability of guessing correctly p. The hypotheses, then, are:

- null hypothesis : $H_0 : p = \frac{1}{4}$ (just guessing)

and

- alternative hypothesis :$H_1 : p > \frac{1}{4}$ (true clairvoyant).

When the test subject correctly predicts all 25 cards, we will consider him clairvoyant, and reject the null hypothesis. Thus also with 24 or 23 hits. With only 5 or 6 hits, on the other hand, there is no cause to consider him so. But what about 12 hits, or 17 hits? What is the critical number, *c*, of hits, at which point we consider the subject to be clairvoyant? How do we determine the critical value *c*? It is obvious that with the choice *c*=25 (i.e. we only accept clairvoyance when all cards are predicted correctly) we're more critical than with *c*=10. In the first case almost no test subjects will be recognized to be clairvoyant, in the second case, some number more will pass the test. In practice, one decides how critical one will be. That is, one decides how often one accepts an error of the first kind - a false positive, or Type I error. With *c* = 25 the probability of such an error is:

$$P(\text{reject } H_0 | H_0 \text{ is valid}) = P(X \geq 25 | p = \tfrac{1}{4}) = \left(\tfrac{1}{4}\right)^{25} \approx 10^{-15},$$

and hence, very small. The probability of a false positive is the probability of randomly guessing correctly all 25 times.

Being less critical, with *c*=10, gives:

$$P(\text{reject } H_0 | H_0 \text{ is valid}) = P(X \geq 10 | p = \tfrac{1}{4}) \approx 0.07.$$

Thus, *c*=10 yields a much greater probability of false positive.

Before the test is actually performed, the desired probability of a Type I error is determined. Typically, values in the range of 1% to 5% are selected. Depending on this desired Type 1 error rate, the critical value *c* is calculated. For example, if we select an error rate of 1%, *c* is calculated thus:

$$P(\text{reject } H_0 | H_0 \text{ is valid}) = P(X \geq c | p = \tfrac{1}{4}) \leq 0.01.$$

From all the numbers c, with this property, we choose the smallest, in order to minimize the probability of a Type II error, a false negative. For the above example, we select: *c* = 12.

But what if the subject did not guess any cards at all? Having zero correct answers is clearly an oddity too. The probability of guessing incorrectly once is equal to p'=(1-p)=3/4. Using the same approach we can calculate that probability of randomly calling all 25 cards wrong is:

$$P(\text{reject } H_0 | H_0 \text{ is valid}) = P(X \geq 25 | p' = \tfrac{3}{4}) = \left(\tfrac{3}{4}\right)^{25} \approx 0.00075.$$

This is highly unlikely (less than 1 in a 1000 chance). While the subject can't guess the cards correctly, dismissing H_0 in favour of H_1 would be an error. In fact, the result would suggest a trait on the subject's part of avoiding calling the correct card. A test of this could be formulated: for a selected 1% error rate the subject would have to answer correctly at least twice, for us to believe that card calling is based purely on guessing.

Example 3 - Radioactive Suitcase

As an example, consider determining whether a suitcase contains some radioactive material. Placed under a Geiger counter, it produces 10 counts per minute. The null hypothesis is that no radioactive material is in the suitcase and that all measured counts are due to ambient radioactivity typical of the surrounding air and harmless objects. We can then calculate how likely it is that we would observe 10 counts per minute if the null hypothesis were true. If the null hypothesis predicts (say) on average 9 counts per minute and a standard deviation of 1 count per minute, then we say that the suitcase is compatible with the null hypothesis (this does not guarantee that there is no radioactive material, just that we don't have enough evidence to suggest there is). On the other hand, if the null hypothesis predicts 3 counts per minute and a standard deviation of 1 count per minute, then the suitcase is not compatible with the null hypothesis, and there are likely other factors responsible to produce the measurements.

The test described here is more fully the null-hypothesis statistical significance test. The null hypothesis represents what we would believe by default, before seeing any evidence. Statistical significance is a possible finding of the test, declared when the observed sample is unlikely to have occurred by chance if the null hypothesis were true. The name of the test describes its formulation and its possible outcome. One characteristic of the test is its crisp decision: to reject or not reject the null hypothesis. A calculated value is compared to a threshold, which is determined from the tolerable risk of error.

Again, the designer of a statistical test wants to maximize the good probabilities and minimize the bad probabilities.

Example 4 - Lady Tasting Tea

The following example is summarized from Fisher, and is known as the *Lady tasting tea* example. Fisher thoroughly explained his method in a proposed experiment to test a Lady's claimed ability to determine the means of tea preparation by taste. The article is less than 10 pages in length and is notable for its simplicity and completeness regarding terminology, calculations and design of the experiment. The example is loosely based on an event in Fisher's life. The Lady proved him wrong.

1. The null hypothesis was that the Lady had no such ability.
2. The test statistic was a simple count of the number of successes in 8 trials.
3. The distribution associated with the null hypothesis was the binomial distribution familiar from coin flipping experiments.
4. The critical region was the single case of 8 successes in 8 trials based on a conventional probability criterion (< 5%).

5. Fisher asserted that no alternative hypothesis was (ever) required.

If and only if the 8 trials produced 8 successes was Fisher willing to reject the null hypothesis – effectively acknowledging the Lady's ability with > 98% confidence (but without quantifying her ability). Fisher later discussed the benefits of more trials and repeated tests.

The Testing Process

Hypothesis testing is defined by the following general procedure:

1. The first step in any hypothesis testing is to state the relevant **null and alternative hypotheses** to be tested. This is important as mis-stating the hypotheses will muddy the rest of the process.
2. The second step is to consider the statistical assumptions being made about the sample in doing the test; for example, assumptions about the statistical independence or about the form of the distributions of the observations. This is equally important as invalid assumptions will mean that the results of the test are invalid.
3. Decide which test is appropriate, and stating the relevant **test statistic** *T*.
4. Derive the distribution of the test statistic under the null hypothesis from the assumptions. In standard cases this will be a well-known result. For example the test statistics may follow a Student's t distribution or a normal distribution.
5. The distribution of the test statistic partitions the possible values of *T* into those for which the null-hypothesis is rejected, the so called critical region, and those for which it is not.
6. Compute from the observations the observed value t_{obs} of the test statistic *T*.
7. Decide to either **fail to reject** the null hypothesis or **reject** it in favor of the alternative. The decision rule is to reject the null hypothesis H_0 if the observed value t_{obs} is in the critical region, and to accept or "fail to reject" the hypothesis otherwise.

It is important to note the philosophical difference between accepting the null hypothesis and simply failing to reject it. The "fail to reject" terminology highlights the fact that the null hypothesis is assumed to be true from the start of the test; if there is a lack of evidence against it, it simply continues to be assumed true. The phrase "accept the null hypothesis" may suggest it has been proved simply because it has not been disproved, a logical fallacy known as the argument from ignorance. Unless a test with particularly high power is used, the idea of "accepting" the null hypothesis may be dangerous. Nonetheless the terminology is prevalent throughout statistics, where its meaning is well understood.

Definition of terms

The following definitions are mainly based on the exposition in the book by Lehmann and Romano:

Simple hypothesis

Any hypothesis which specifies the population distribution completely.

Composite hypothesis

Any hypothesis which does *not* specify the population distribution completely.

Statistical test

A decision function that takes its values in the set of hypotheses.

Region of acceptance

The set of values for which we fail to reject the null hypothesis.

Region of rejection / Critical region

The set of values of the test statistic for which the null hypothesis is rejected.

Power of a test ($1 - \beta$)

The test's probability of correctly rejecting the null hypothesis. The complement of the false negative rate, β.

Size / Significance level of a test (α)

For simple hypotheses, this is the test's probability of *incorrectly* rejecting the null hypothesis. The false positive rate. For composite hypotheses this is the upper bound of the probability of rejecting the null hypothesis over all cases covered by the null hypothesis.

Most powerful test

For a given *size* or *significance level*, the test with the greatest power.

Uniformly most powerful test (UMP)

A test with the greatest *power* for all values of the parameter being tested.

Consistent test

When considering the properties of a test as the sample size grows, a test is said to be consistent if, for a fixed size of test, the power against any fixed alternative approaches 1 in the limit.

Unbiased test

For a specific alternative hypothesis, a test is said to be **unbiased** when the probability of rejecting the null hypothesis is not less than the significance level when the alternative is true *and* is less than or equal to the significance level when the null hypothesis is true.

Conservative test

A test is conservative if, when constructed for a given nominal significance level, the true probability of *incorrectly* rejecting the null hypothesis is never greater than the nominal level.

Uniformly most powerful unbiased (UMPU)

A test which is UMP in the set of all unbiased tests.

p-value

The probability, assuming the null hypothesis is true, of observing a result at least as extreme as the test statistic.

Interpretation

The direct interpretation is that if the p-value is less than the required significance level, then we say the null hypothesis is rejected at the given level of significance. Criticism on this interpretation can be found in the corresponding section.

Common test statistics

In the table below, the symbols used are defined at the bottom of the table. Many other tests can be found in other articles.

Name	Formula	Assumptions or notes
One-sample z-test	$z = \frac{\overline{x} - \mu_0}{(\sigma/\sqrt{n})}$	(Normal population **or** n > 30) **and** σ known. (z is the distance from the mean in relation to the standard deviation of the mean). For non-normal distributions it is possible to calculate a minimum proportion of a population that falls within k standard deviations for any k.
Two-sample z-test	$z = \frac{(\overline{x}_1 - \overline{x}_2) - d_0}{\sqrt{\frac{\sigma_1^2}{n1} + \frac{\sigma_2^2}{n2}}}$	Normal population **and** independent observations **and** σ_1 and σ_2 are known
Two-sample pooled t-test, equal variances*	$t = \frac{(\overline{x}_1 - \overline{x}_2) - d_0}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$ $s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2},$ $df = n_1 + n_2 - 2$	(Normal populations **or** $n_1 + n_2 > 40$) **and** independent observations **and** $\sigma_1 = \sigma_2$ **and** σ_1 and σ_2 unknown

Two-sample unpooled t-test, unequal variances*	$t = \frac{(\overline{x}_1 - \overline{x}_2) - d_0}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}},$ $df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1-1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2-1}}$	(Normal populations **or** $n_1 + n_2 > 40$) **and** independent observations **and** $\sigma_1 \neq \sigma_2$ **and** σ_1 and σ_2 unknown
One-proportion z-test	$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$	$n \cdot p_0 > 10$ **and** $n\ (1 - p_0) > 10$ **and** it is a SRS (Simple Random Sample), see notes.
Two-proportion z-test, pooled for $d_0 = 0$	$z = \frac{(\hat{p}_1 - \hat{p}_2) - d_0}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_1} + \frac{1}{n_2})}}$ $\hat{p} = \frac{x_1 + x_2}{n_1 + n_2}$	$n_1\ p_1 > 5$ **and** $n_1(1 - p_1) > 5$ **and** $n_2\ p_2 > 5$ **and** $n_2(1 - p_2) > 5$ **and** independent observations, see notes.
Two-proportion z-test, unpooled for $\mid d_0 \mid > 0$	$z = \frac{(\hat{p}_1 - \hat{p}_2) - d_0}{\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}}$	$n_1\ p_1 > 5$ **and** $n_1(1 - p_1) > 5$ **and** $n_2\ p_2 > 5$ **and** $n_2(1 - p_2) > 5$ **and** independent observations, see notes.
One-sample chi-square test	$\chi^2 = \frac{(n-1)s^2}{\sigma_0^2}$	One of the following • All expected counts are at least 5 • All expected counts are > 1 and no more that 20% of expected counts are less than 5
*Two-sample F test for equality of variances	$F = \frac{s_1^2}{s_2^2}$	Arrange so $s_1^2 > s_2^2$ and reject H_0 for $F > F(\alpha / 2, n_1 - 1, n_2 - 1)$

In general, the subscript 0 indicates a value taken from the null hypothesis, H_0, which should be used as much as possible in constructing its test statistic. ... *Definitions of other symbols:*

- α, the probability of Type I error (rejecting a null hypothesis when it is in fact true)
- n = sample size

- s = sample standard deviation
- s^2 = sample variance
- s_1 = sample 1 standard deviation

- $\hat{p} = x/n$ = sample proportion, unless specified otherwise
- p_0 = hypothesized population proportion

- n_1 = sample 1 size
- n_2 = sample 2 size
- $\overline{x}$= sample mean
- μ_0 = hypothesized population mean
- μ_1 = population 1 mean
- μ_2 = population 2 mean
- σ = population standard deviation
- σ^2 = population variance
- s_2 = sample 2 standard deviation
- t = t statistic
- df = degrees of freedom
- $\overline{d}$= sample mean of differences
- d_0 = hypothesized population mean difference
- s_d = standard deviation of differences
- p_1 = proportion 1
- p_2 = proportion 2
- d_p = hypothesized difference in proportion
- min{n_1,n_2} = minimum of n_1 and n_2
- $x_1 = n_1p_1$
- $x_2 = n_2p_2$
- χ^2 = Chi-squared statistic
- F = F statistic

Origins

Hypothesis testing is largely the product of Ronald Fisher, Jerzy Neyman, Karl Pearson and (son) Egon Pearson. Fisher was an agricultural statistician who emphasized rigorous experimental design and methods to extract a result from few samples assuming Gaussian distributions. Neyman (who teamed with the younger Pearson) emphasized mathematical rigor and methods to obtain more results from many samples and a wider range of distributions. Modern hypothesis testing is an (extended) hybrid of the Fisher vs Neyman/Pearson formulation, methods and terminology developed in the early 20th century.

Importance

Statistical hypothesis testing plays an important role in the whole of statistics and in statistical inference. For example, Lehmann (1992) in a review of the fundamental paper by Neyman and Pearson (1933) says: "Nevertheless, despite their shortcomings, the new paradigm formulated in the 1933 paper, and the many developments carried out within its framework continue to play a central role in both the theory and practice of statistics and can be expected to do so in the foreseeable future".

Potential misuse

One of the more common problems in significance testing is the tendency for multiple comparisons to yield spurious significant differences even where the null hypothesis is true. For instance, in a study of twenty comparisons, using an α-level of 5%, one comparison will likely yield a significant result despite the null hypothesis being true. In these cases p-values are adjusted in order to control either the familywise error rate or the false discovery rate.

Yet another common pitfall often happens when a researcher writes the qualified statement "we found no statistically significant difference," which is then misquoted by others as "they found that there was no difference." Actually, statistics cannot be used to prove that there is exactly zero difference between two populations. Failing to find evidence that there is a difference does not constitute evidence that there is no difference. This principle is sometimes described by the maxim "Absence of evidence is not evidence of absence."

According to J. Scott Armstrong, attempts to educate researchers on how to avoid pitfalls of using statistical significance have had little success. In the papers "Significance Tests Harm Progress in Forecasting," and "Statistical Significance Tests are Unnecessary Even When Properly Done," Armstrong makes the case that even when done properly, statistical significance tests are of no value. A number of attempts failed to find empirical evidence supporting the use of significance tests. Tests of statistical significance are harmful to the development of scientific knowledge because they distract researchers from the use of proper methods. Armstrong suggests authors should avoid tests of statistical significance; instead, they should report on effect sizes, confidence intervals, replications/extensions, and meta-analyses.

Criticism

Significance and practical importance

A common misconception is that a statistically significant result is always of practical significance, or demonstrates a large effect in the population. Unfortunately, this problem is commonly encountered in scientific writing. Given a sufficiently large sample, extremely small and non-notable differences can be found to be statistically significant, and statistical significance says nothing about the practical significance of a difference.

Use of the statistical significance test has been called seriously flawed and unscientific by authors Deirdre McCloskey and Stephen Ziliak. They point out that "insignificance" does not mean unimportant, and propose that the scientific community should abandon usage of the test altogether, as it can cause false hypotheses to be accepted and true hypotheses to be rejected.

Some statisticians have commented that pure "significance testing" has what is actually a rather strange goal of detecting the existence of a "real" difference between two populations. In practice a difference can almost always be found given a large enough sample. The typically more relevant goal of science is a determination of causal effect size. The amount and nature of the difference, in other words, is what should be studied. Many researchers also feel that hypothesis testing is something of a misnomer. In practice a single statistical test in a single study never "proves" anything.

An additional problem is that frequentist analyses of p-values are considered by some to overstate "statistical significance".

Meta-criticism

The criticism here is of the application, or of the interpretation, rather than of the method. Attacks and defenses of the null-hypothesis significance test are collected in Harlow *et al.*.

The original purposes of Fisher's formulation, as a tool for the experimenter, was to plan the experiment and to easily assess the information content of the small sample. There is little criticism, Bayesian in nature, of the formulation in its original context.

In other contexts, complaints focus on flawed interpretations of the results and over-dependence/emphasis on one test.

Numerous attacks on the formulation have failed to supplant it as a criterion for publication in scholarly journals. The most persistent attacks originated from the field of Psychology. After review, the American Psychological Association did not explicitly deprecate the use of null-hypothesis significance testing, but adopted enhanced publication guidelines which implicitly reduced the relative importance of such testing.

The International Committee of Medical Journal Editors recognizes an obligation to publish negative (not statistically significant) studies under some circumstances.

The applicability of the null-hypothesis testing to the publication of observational (as contrasted to experimental) studies is doubtful.

Philosophical criticism

Philosophical criticism to hypothesis testing includes consideration of borderline cases. Any process that produces a crisp decision from uncertainty is subject to claims of unfairness near the decision threshold. (Consider close election results.) The premature death of a laboratory rat during testing can impact doctoral theses and academic tenure decisions.

"... surely, God loves the .06 nearly as much as the .05"

The statistical significance required for publication has no mathematical basis, but is based on long tradition.

"It is usual and convenient for experimenters to take 5% as a standard level of significance, in the sense that they are prepared to ignore all results which fail to reach this standard, and, by this means, to eliminate from further discussion the greater part of the fluctuations which chance causes have introduced into their experimental results."

Ambivalence attacks all forms of decision making. A mathematical decision-making process is attractive because it is objective and transparent. It is repulsive because it allows authority to avoid taking personal responsibility for decisions.

Pedagogic criticism

Pedagogic criticism of the null-hypothesis testing includes the counter-intuitive formulation, the terminology and confusion about the interpretation of results.

"Despite the stranglehold that hypothesis testing has on experimental psychology, I find it difficult to imagine a less insightful means of transiting from data to conclusions."

Students find it difficult to understand the formulation of statistical null-hypothesis testing. In rhetoric, examples often support an argument, but a mathematical proof "is a logical argument, not an empirical one". A single counterexample results in the rejection of a conjecture. Karl Popper defined science by its vulnerability to disproof by data. Null-hypothesis testing shares the mathematical and scientific perspective rather than the more familiar rhetorical one. Students expect hypothesis testing to be a statistical tool for illumination of the research hypothesis by the sample; it is not. The test asks indirectly whether the sample can illuminate the research hypothesis.

Students also find the terminology confusing. While Fisher disagreed with Neyman and Pearson about the theory of testing, their terminologies have been blended. The blend is not seamless or standardized. While this article teaches a pure Fisher formulation, even it mentions Neyman and Pearson terminology (Type II error and the alternative hypothesis). The typical introductory statistics text is less consistent. The Sage Dictionary of Statistics would not agree with the title of this article, which it would call null-hypothesis testing. "...there is no alternate hypothesis in Fisher's scheme: Indeed, he violently opposed its inclusion by Neyman and Pearson." In discussing test results, "significance" often has two distinct meanings in the same sentence; One is a probability, the other is a subject-matter measurement (such as currency). The significance (meaning) of (statistical) significance is significant (important).

There is widespread and fundamental disagreement on the interpretation of test results.

"A little thought reveals a fact widely understood among statisticians: The null hypothesis, taken literally (and that's the only way you can take it in formal hypothesis testing), is almost always false in the real world.... If it is false, even to a tiny degree, it must be the case that a large enough sample will produce a significant result and lead to its rejection. So if the null hypothesis is always false, what's the big deal about rejecting it?" (The above criticism only applies to point hypothesis tests. If one were testing, for example, whether a parameter is greater than zero, it would not apply.)

"How has the virtually barren technique of hypothesis testing come to assume such importance in the process by which we arrive at our conclusions from our data?"

Null-hypothesis testing just answers the question of "how well the findings fit the possibility that chance factors alone might be responsible."

Null-hypothesis significance testing does not determine the truth or falsity of claims. It determines whether confidence in a claim based solely on a sample-based estimate exceeds a threshold. It is a research quality assurance test, widely used as one requirement for publication of experimental research with statistical results. It is uniformly agreed that statistical significance is not the only consideration in assessing the importance of research results. Rejecting the null hypothesis is not a sufficient condition for publication.

"Statistical significance does not necessarily imply practical significance!"

Practical criticism

Practical criticism of hypothesis testing includes the sobering observation that published test results are often contradicted. Mathematical models support the conjecture that most published medical research test results are flawed. Null-hypothesis testing has not achieved the goal of a low error probability in medical journals.

Straw man

Hypothesis testing is controversial when the alternative hypothesis is suspected to be true at the outset of the experiment, making the null hypothesis the reverse of what the experimenter actually believes; it is put forward as a straw man only to allow the data to contradict it. Many statisticians have pointed out that rejecting the null hypothesis says nothing or very little about the likelihood that the null is true. Under traditional null hypothesis testing, the null is rejected when the conditional probability P(Data as or more extreme than observed | Null) is very small, say 0.05. However, some say researchers are really interested in the probability P(Null | Data as actually observed) which cannot be inferred from a p-value: some like to present these as inverses of each other but the events "Data as or more extreme than observed" and "Data as actually observed" are very different. In some cases, P(Null | Data) approaches 1 while P(Data as or more extreme than observed | Null) approaches 0, in other words, we can reject the null when it's virtually certain to be true. For this and other reasons, Gerd Gigerenzer has called null hypothesis testing "mindless statistics" while Jacob Cohen described it as a ritual conducted to convince ourselves that we have the evidence needed to confirm our theories.

Bayesian criticism

Bayesian statisticians reject classical null hypothesis testing, since it violates the Likelihood principle and is thus incoherent and leads to sub-optimal decision-making. The Jeffreys–Lindley paradox illustrates this. Along with many frequentist statisticians, Bayesians prefer to provide an estimate, along with a confidence interval, (although Bayesian confidence intervals are different from classical ones). Some Bayesians (James Berger in particular) have developed Bayesian hypothesis testing methods, though these are not accepted by all Bayesians (notably, Andrew Gelman). Given a prior probability distribution for one or more parameters, sample evidence can be used to generate an updated posterior distribution. In this framework, but *not* in the null hypothesis testing framework, it is meaningful to make statements of the general form "the probability that the true value of the parameter is greater than 0 is *p*". According to Bayes' theorem,

$$P(\text{Null}|\text{Data}) = P(\text{Data}|\text{Null})\frac{P(\text{Null})}{P(\text{Data})},$$

thus P(Null | Data) may approach 1 while P(Data | Null) approaches 0 only when P(Null)/P(Data) approaches infinity, i.e. (for instance) when the a priori probability of the null hypothesis, P(Null), is also approaching 1, while P(Data) approaches 0: then P(Data | Null) is low because the data are extremely unlikely, but the Null hypothesis is extremely likely to be true.

Publication bias

In 2002, a group of psychologists launched a new journal dedicated to experimental studies in psychology which support the null hypothesis. The *Journal of Articles in Support of the Null Hypothesis* (JASNH) was founded to address a scientific publishing bias against such articles. According to the editors,

> "other journals and reviewers have exhibited a bias against articles that did not reject the null hypothesis. We plan to change that by offering an outlet for experiments that do not reach the traditional significance levels ($p < 0.05$). Thus, reducing the file drawer problem, and reducing the bias in psychological literature. Without such a resource researchers could be wasting their time examining empirical questions that have already been examined. We collect these articles and provide them to the scientific community free of cost."

The "File Drawer problem" is a problem that exists due to the fact that academics tend not to publish results that indicate the null hypothesis could not be rejected. This does not mean that the relationship they were looking for did not exist, but it means they couldn't prove it. Even though these papers can often be interesting, they tend to end up unpublished, in "file drawers."

Ioannidis has inventoried factors that should alert readers to the risks of publication bias.

Improvements

Jones and Tukey suggested a modest improvement in the original null-hypothesis formulation to formalize handling of one-tail tests. They conclude that, in the "Lady Tasting Tea" example, Fisher ignored the 8-failure case (equally improbable as the 8-success case) in the example test involving tea, which altered the claimed significance by a factor of 2.

Lateral thinking

Lateral thinking refers to solving problems through an indirect and creative approach. Lateral thinking is about reasoning that is not immediately obvious and about ideas that may not be obtainable by using only traditional step-by-step logic.

Methods

Critical thinking is primarily concerned with judging the true value of statements and seeking errors. Lateral thinking is more concerned with the movement value of statements and ideas. A person would use lateral thinking when they want to move from one known idea to creating new ideas. Edward de Bono defines four types of thinking tools:

- Idea generating tools that are designed to break current thinking patterns—routine patterns, the status quo
- Focus tools that are designed to broaden where to search for new ideas
- Harvest tools that are designed to ensure more value is received from idea generating output
- Treatment tools that are designed to consider real-world constraints, resources, and support

Random Entry Idea Generating Tool: Choose an object at random, or a noun from a dictionary, and associate that with the area you are thinking about.

For example imagine you are thinking about how to improve a web site. Choosing an object at random from an office you might see a fax machine. A fax machine transmits images over the phone to paper. Fax machines are becoming rare. People send faxes directly to phone numbers. Perhaps this could be a new way to embed the web site's content in emails and other sites.

Provocation Idea Generating Tool: choose to use any of the provocation techniques—wishful thinking, exaggeration, reversal, escape, or arising. Create a list of provocations and then use the most outlandish ones to move your thinking forward to new ideas.

Challenge Idea Generating Tool: A tool which is designed to ask the question "Why?" in a non-threatening way: why something exists, why it is done the way it is. The result is a very clear understanding of "Why?" which naturally leads to fresh new ideas. The goal is to be able to challenge anything at all, not just items which are problems.

For example you could challenge the handles on coffee cups. The reason for the handle seems to be that the cup is often too hot to hold directly. Perhaps coffee cups could be made with insulated finger grips, or there could be separate coffee cup holders similar to beer holders.

Concept Fan Idea Generating Tool: Ideas carry out concepts. This tool systematically expands the range and number of concepts in order to end up with a very broad range of ideas to consider.

Disproving: Based on the idea that the majority is always wrong (Henrik Ibsen, Galbraith), take anything that is obvious and generally accepted as "goes without saying", question it, take an opposite view, and try to convincingly disprove it.

The other focus, harvesting and treatment tools deal with the output of the generated ideas and the ways to use them.

Lateral thinking and problem solving

Problem Solving: When something creates a problem, the performance or the status quo of the situation drops. Problem solving deals with finding out what caused the problem and then figuring out ways to fix the problem. The objective is to get the situation to where it should be.

For example, a production line has an established run rate of 1000 items per hour. Suddenly, the run rate drops to 800 items per hour. Ideas as to why this happened and solutions to repair the production line must be thought of.

Creative Problem Solving: Using creativity, one must solve a problem in an indirect and unconventional manner.

For example, if a production line produced 1000 books per hour, creative problem solving could find ways to produce more books per hour, use the production line, or reduce the cost to run the production line.

Creative Problem Identification: Many of the greatest non-technological innovations are identified while realizing an improved process or design in everyday objects and tasks either by accidental chance or by studying and documenting real world experience.

Means-ends analysis

Means-Ends Analysis (MEA) is a technique used in Artificial Intelligence for controlling search in problem solving computer programs.

It is also a technique used at least since the 1950s as a creativity tool, most frequently mentioned in engineering books on design methods. Means-Ends Analysis is also a way to clarify one's thoughts when embarking on a mathematical proof.

Problem-solving as search

An important aspect of intelligent behavior as studied in AI is *goal-based* problem solving, a framework in which the solution of a problem can be described by finding a sequence of *actions* that lead to a desirable goal. A goal-seeking system is supposed to be connected to its outside environment by sensory channels through which it receives information about the environment and motor channels through which it acts on the environment. (The term "afferent" is used to describe "inward" sensory flows, and "efferent" is used to describe "outward" motor commands.) In addition, the system has some means of storing in a *memory* information about the *state* of the environment (afferent information) and information about actions (efferent information). Ability to attain goals depends on building up associations, simple or complex, between particular changes in states and particular actions that will bring these changes about. Search is the process of discovery and assembly of sequences of actions that will lead from a given state to a desired state. While this strategy may be appropriate for machine learning and problem solving, it is not always suggested for humans (e.g. cognitive load theory and its implications).

How MEA works

The MEA technique is a strategy to control search in problem-solving. Given a current state and a goal state, an action is chosen which will reduce the *difference* between the two. The action is performed on the current state to produce a new state, and the process is recursively applied to this new state and the goal state.

Note that, in order for MEA to be effective, the goal-seeking system must have a means of associating to any kind of detectable difference those actions that are relevant to reducing that difference. It must also have means for detecting the progress it is making (the changes in the differences between the actual and the desired state), as some attempted sequences of actions may fail and, hence, some alternate sequences may be tried.

When knowledge is available concerning the importance of differences, the most important difference is selected first to further improve the average performance of MEA over other brute-force search strategies. However, even without the ordering of differences according to importance, MEA improves over other search heuristics (again in the average case) by focusing the problem solving on the actual differences between the current state and that of the goal.

Some AI systems using MEA

The MEA technique as a problem-solving strategy was first introduced in 1963 by Allen Newell and Herbert Simon in their computer problem-solving program General Problem Solver (GPS). In that implementation, the correspondence between differences and actions, also called *operators*, is provided a priori as knowledge in the system. (In GPS this knowledge was in the form of *table of connections*.)

When the action and side-effects of applying an operator are penetrable, the search may select the relevant operators by inspection of the operators and do without a table of connections. This latter case, of which the canonical example is STRIPS, an automated planning computer program, allows task-independent correlation of differences to the operators which reduce them.

Prodigy, a problem solver developed in a larger learning-assisted automated planning project started at Carnegie Mellon University by Jaime Carbonnell, Steven Minton and Craig Knoblock, is another system that used MEA.

Professor Morten Lind, at Technical University of Denmark has developed a tool called multilevel flow modeling MFM. It performs means-end based diagnostic reasoning for industrial control and automation systems.

Method of focal objects

The technique of **focal object** for problem solving involves synthesizing the seemingly non-matching characteristics of different objects into something new.

Another way to think of focal objects is as a memory cue: if you're trying to find all the different ways to use a brick, give yourself some random "objects" (situations, concepts, etc.) and see if you can find a use. Given "blender", for example, I would try to think of all the ways a brick could be used with a blender (as a lid?). Another concept for the brick game: find patterns in your solutions, and then break those patterns. If you keep finding ways to build things with bricks, think of ways to use bricks that don't involve construction. Pattern-breaking, combined with focal object cues, can lead to very divergent solutions. (Grind the brick up and use it as pigment?)

Morphological analysis (problem-solving)

Morphological analysis or **General Morphological Analysis** is a method developed by Fritz Zwicky (1967, 1969) for exploring all the possible solutions to a multi-dimensional, non-quantified problem complex.

As a problem-structuring and problem-solving technique, morphological analysis was designed for multi-dimensional, non-quantifiable problems where causal modeling and simulation do not function well or at all. Zwicky developed this approach to address seemingly non-reducible complexity. Using the technique of cross consistency assessment (CCA) (Ritchey, 1998), the system however does allow for reduction, not by reducing the number of variables involved, but by reducing the number of possible solutions through the elimination of the illogical solution combinations in a grid box. A detailed introduction to morphological modeling is given in Ritchey (2002, 2006).

Overview

Morphology comes from the classical Greek concept *morphé*, meaning shape or form. Morphological Analysis (MA) concerns the arrangement of objects and how they conform to create a whole or Gestalt. The objects in question can be a physical system (e.g. anatomy), a social system (e.g. an organisation) or a logical system (e.g. word forms or a system of ideas).

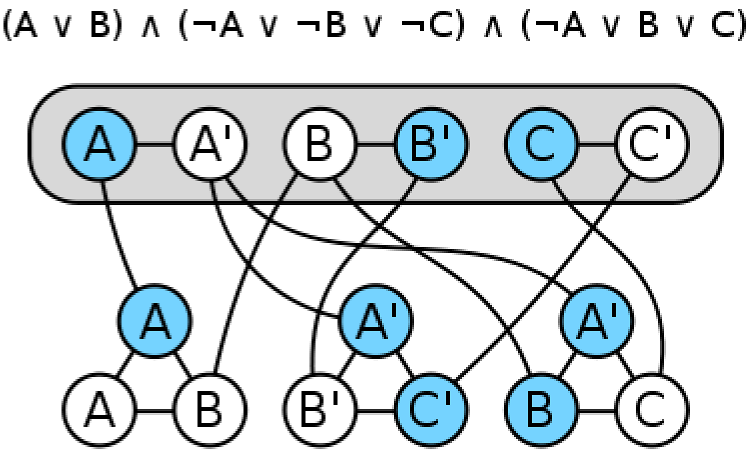
General morphology was developed by Fritz Zwicky, the Swiss astrophysicist based at the California Institute of Technology. Zwicky applied MA inter alia to astronomical studies and the development of jet and rocket propulsion systems.

Illustration of the need for Morphological Analysis

Consider a complex real world problem, like those of marketing or making policies for a nation, where there are many governing factors, and most of them cannot be expressed as numerical time series data, as one would like to have for building mathematical models. The conventional approach here would be to break the system down into parts, isolate the vital parts (dropping the trivial components) for their contributions to the output and solve the simplified system for creating desired models or scenarios. The disadvantage of this approach is that real world scenarios do not behave rationally and more often than not a simplified model will break down when the contribution of trivial components becomes significant. Also significantly, the behaviour of many components will be governed by states of, and relations with other components, perhaps minor.

Morphological Analysis on the other hand, does not drop any of the components of the system itself, but works backwards from the output towards the system internals. Again, the interactions and relations get to play their parts in MP and their effects are accounted for in the analysis.

Reduction (complexity)



Example of a reduction from a boolean satisfiability problem to a vertex cover problem. Blue vertices form a vertex cover which corresponds to truth values.

In computability theory and computational complexity theory, a **reduction** is a transformation of one problem into another problem. Depending on the transformation used this can be used to define complexity classes on a set of problems.

Intuitively, problem A is reducible to problem B if solutions to B exist and give solutions to A whenever A has solutions. Thus, solving A cannot be harder than solving B. We write $A \leq_m B$, usually with a subscript on the ≤ to indicate the type of reduction being used (m: mapping reduction, p: polynomial reduction).

Introduction

Often we find ourselves trying to solve a problem that is similar to a problem we've already solved. In these cases, often a quick way of solving the new problem is to transform each instance of the new problem into instances of the old problem, solve these using our existing solution, and then use these to obtain our final solution. This is perhaps the most obvious use of reductions.

Another, more subtle use is this: suppose we have a problem that we've proven is hard to solve, and we have a similar new problem. We might suspect that it, too, is hard to solve. We argue by contradiction: suppose the new problem is easy to solve. Then, if we can show that every instance of the old problem can be solved easily by transforming it into instances of the new problem and solving those, we have a contradiction. This establishes that the new problem is also hard.

A very simple example of a reduction is from *multiplication* to *squaring*. Suppose all we know how to do is to add, subtract, take squares, and divide by two. We can use this knowledge, combined with the following formula, to obtain the product of any two numbers:

$$a \times b = \frac{\left((a+b)^2 - a^2 - b^2\right)}{2}$$

We also have a reduction in the other direction; obviously, if we can multiply two numbers, we can square a number. This seems to imply that these two problems are equally hard. This kind of reduction corresponds to Turing reduction.

However, the reduction becomes much harder if we add the restriction that we can only use the squaring function one time, and only at the end. In this case, even if we're allowed to use all the basic arithmetic operations, including multiplication, no reduction exists in general, because we may have to compute an irrational number like $\sqrt{2}$from rational numbers. Going in the other direction, however, we can certainly square a number with just one multiplication, only at the end. Using this limited form of reduction, we have shown the unsurprising result that multiplication is harder in general than squaring. This corresponds to many-one reduction.

Definition

Given two subsets *A* and *B* of **N** and a set of functions *F* from **N** to **N** which is closed under composition, *A* is called **reducible** to *B* under *F* if

$$\exists f \in F \; . \; \forall x \in \mathbb{N} \; . \; x \in A \Leftrightarrow f(x) \in B$$

We write

$$A \leq_F B$$

Let *S* be a subset of **P**(**N**) and ≤ a reduction, then *S* is called **closed** under ≤ if

$$\forall s \in S \ . \ \forall A \in P(\mathbb{N}) \ . \ A \leq s \Rightarrow A \in S$$

A subset *A* of **N** is called **hard** for *S* if

$$\forall s \in S \ . \ s \leq A$$

A subset *A* of **N** is called **complete** for *S* if *A* is hard for *S* and *A* is in *S*.

Properties

A reduction is a preordering, that is a reflexive and transitive relation, on **P**(**N**)×**P**(**N**), where **P**(**N**) is the power set of the natural numbers.

Types and applications of reductions

As described in the example above, there are two main types of reductions used in computational complexity, the many-one reduction and the Turing reduction. Many-one reductions map *instances* of one problem to *instances* of another; Turing reductions *compute* the solution to one problem, assuming the other problem is easy to solve. A many-one reduction is weaker than a Turing reduction. Weaker reductions are more effective at separating problems, but they have less power, making reductions harder to design.

A problem is complete for a complexity class if every problem in the class reduces to that problem, and it is also in the class itself. In this sense the problem represents the class, since any solution to it can, in combination with the reductions, be used to solve every problem in the class.

However, in order to be useful, reductions must be *easy*. For example, it's quite possible to reduce a difficult-to-solve NP-complete problem like the boolean satisfiability problem to a trivial problem, like determining if a number equals zero, by having the reduction machine solve the problem in exponential time and output zero only if there is a solution. However, this does not achieve much, because even though we can solve the new problem, performing the reduction is just as hard as solving the old problem. Likewise, a reduction computing a noncomputable function can reduce an undecidable problem to a decidable one. As Michael Sipser points out in *Introduction to the Theory of Computation*: "The reduction must be easy, relative to the complexity of typical problems in the class [...] If the reduction itself were difficult to compute, an easy solution to the complete problem wouldn't necessarily yield an easy solution to the problems reducing to it."

Therefore, the appropriate notion of reduction depends on the complexity class being studied. When studying the complexity class NP and harder classes such as the polynomial hierarchy, polynomial-time reductions are used. When studying classes within P such as NC and NL, log-space reductions are used. Reductions are also used in computability theory to show whether problems are or are not solvable by machines at all; in this case, reductions are restricted only to computable functions.

In case of optimization (maximization or minimization) problems, we often think in terms of approximation-preserving reductions. Suppose we have two optimization problems such that instances of one problem can be mapped onto instances of the other, in a way that nearly-optimal solutions to instances of the latter problem can be transformed back to yield nearly-optimal solutions to the former. This way, if we have an optimization algorithm (or approximation algorithm) that finds near-optimal (or optimal) solutions to instances of problem B, and an efficient approximation-preserving reduction from problem A to problem B, by composition we obtain an optimization algorithm that yields near-optimal solutions to instances of problem A. Approximation-preserving reductions are often used to prove hardness of approximation results: if some optimization problem A is hard to approximate (under some complexity assumption) within a factor better than α for some α, and there is a β-approximation-preserving reduction from problem A to problem B, we can conclude that problem B is hard to approximate within factor α/β.

Examples

- To show that a decision problem P is undecidable we must find a reduction from a decision problem which is already known to be undecidable to P. That reduction function must be a computable function. In particular, we often show that a problem P is undecidable by showing that the halting problem reduces to P.
- The complexity classes P, NP and PSPACE are closed under polynomial-time reductions.
- The complexity classes L, NL, P, NP and PSPACE are closed under log-space reduction.

Detailed example

The following example shows how to use reduction from the halting problem to prove that a language is undecidable. Suppose *H*(*M*, *w*) is the problem of determining whether a given Turing machine *M* halts (by accepting or rejecting) on input string *w*. This language is known to be undecidable. Suppose *E*(*M*) is the problem of determining whether the language a given Turing machine *M* accepts is empty (in other words, whether *M* accepts any strings at all). We show that *E* is undecidable by a reduction from *H*.

To obtain a contradiction, suppose *R* is a decider for *E*. We will use this to produce a decider *S* for *H* (which we know does not exist). Given input *M* and *w* (a Turing machine and some input string), define *S*(*M*, *w*) with the following behavior: *S* creates a Turing machine *N* that accepts only if the input string to *N* is *w* and *M* halts on input *w*, and does not halt otherwise. The decider *S* can now evaluate *R*(*N*) to check whether the language accepted by *N* is empty. If *R* accepts *N*, then the language accepted by *N* is empty, so in particular *M* does not halt on input *w*, so *S* can reject. If *R* rejects *N*, then the language accepted by *N* is nonempty, so *M* does halt on input *w*, so *S* can accept. Thus, if we had a decider *R* for *E*, we would be able to produce a decider *S* for the halting problem *H*(*M*, *w*) for any machine *M* and input *w*. Since we know that such an *S* cannot exist, it follows that the language *E* is also undecidable.

Root cause analysis

Root cause analysis (**RCA**) is a class of problem solving methods aimed at identifying the root causes of problems or incidents. The practice of RCA is predicated on the belief that problems are best solved by attempting to correct or eliminate root causes, as opposed to merely addressing the immediately obvious symptoms. By directing corrective measures at root causes, it is hoped that the likelihood of problem recurrence will be minimized. However, it is recognized that complete prevention of recurrence by a single intervention is not always possible. Thus, RCA is often considered to be an iterative process, and is frequently viewed as a tool of continuous improvement.

RCA, initially is a reactive method of problem detection and solving. This means that the analysis is done *after* an incident has occurred. By gaining expertise in RCA it becomes a pro-active method. This means that RCA is able to *forecast* the possibility of an incident even *before* it could occur. While one follows the other, RCA is a completely separate process to Incident Management.

Root cause analysis is not a single, sharply defined methodology; there are many different tools, processes, and philosophies of RCA in existence. However, most of these can be classed into five, very-broadly defined "schools" that are named here by their basic fields of origin: safety-based, production-based, process-based, failure-based, and systems-based.

- Safety-based RCA descends from the fields of accident analysis and occupational safety and health.
- Production-based RCA has its origins in the field of quality control for industrial manufacturing.
- Process-based RCA is basically a follow-on to production-based RCA, but with a scope that has been expanded to include business processes.
- Failure-based RCA is rooted in the practice of failure analysis as employed in engineering and maintenance.
- Systems-based RCA has emerged as an amalgamation of the preceding schools, along with ideas taken from fields such as change management, risk management, and systems analysis.

Despite the seeming disparity in purpose and definition among the various schools of root cause analysis, there are some general principles that could be considered as universal. Similarly, it is possible to define a general process for performing RCA.

General principles of root cause analysis

1. The primary aim of RCA is to identify the root cause of a problem in order to create effective corrective actions that will prevent that problem from ever re-occurring, otherwise known as the '100 year fix'.
2. To be effective, RCA must be performed systematically as an investigation, with conclusions and the root cause backed up by documented evidence.
3. There is always one true root cause for any given problem, the difficult part is having the stamina to reach it.

4. To be effective the analysis must establish a sequence of events or timeline to understand the relationships between contributory factors, the root cause and the defined problem.
5. Root cause analysis can help to transform an old culture that reacts to problems into a new culture that solves problems before they escalate but more importantly; reduces the instances of problems occurring over time within the environment where the RCA process is operated.

General process for performing and documenting an RCA-based Corrective Action

Notice that RCA (in steps 3, 4 and 5) forms the most critical part of successful corrective action, because it directs the corrective action at the true root cause of the problem. The root cause is secondary to the goal of prevention, but without knowing the root cause, we cannot determine what an effective corrective action for the defined problem will be.

1. Define the problem.
2. Gather data/evidence.
3. Ask why and identify the true root cause associated with the defined problem.
4. Identify corrective action(s) that will prevent recurrence of the problem (your 100 year fix).
5. Identify effective solutions that prevent recurrence, are within your control, meet your goals and objectives and do not cause other problems.
6. Implement the recommendations.
7. Observe the recommended solutions to ensure effectiveness.
8. Variability Reduction methodology for problem solving and problem avoidance.

Root cause analysis techniques

- Barrier analysis - a technique often used in particularly in process industries. It is based on tracing energy flows, with a focus on barriers to those flows, to identify how and why the barriers did not prevent the energy flows from causing harm.
- Bayesian inference
- Causal factor tree analysis - a technique based on displaying causal factors in a tree-structure such that cause-effect dependencies are clearly identified.
- Change analysis - an investigation technique often used for problems or accidents. It is based on comparing a situation that does not exhibit the problem to one that does, in order to identify the changes or differences that might explain why the problem occurred.
- Current Reality Tree - A method developed by Eliahu M. Goldratt in his theory of constraints that guides an investigator to identify and relate all root causes using a cause-effect tree whose elements are bound by rules of logic (Categories of Legitimate Reservation). The CRT begins with a brief list of the undesirables things we see around us, and then guides us towards one or more root causes. This method is particularly powerful when the system is complex, there is no obvious link between the observed undesirable things, and a deep understanding of the root cause(s) is desired.
- Failure mode and effects analysis
- Fault tree analysis
- 5 Whys

- Ishikawa diagram, also known as the fishbone diagram or cause-and-effect diagram. The Ishikawa diagram is the preferred method for project managers for conducting RCA, mainly due to its simplicity, and the complexity of the rest of the methods.
- Pareto analysis
- RPR Problem Diagnosis - An ITIL-aligned method for diagnosing IT problems.

Common cause analysis (CCA) common modes analysis (CMA) are evolving engineering techniques for complex technical systems to determine if common root causes in hardware, software or highly integrated systems interaction may contribute to human error or improper operation of a system. Systems are analyzed for root causes and causal factors to determine probability of failure modes, fault modes, or common mode software faults due to escaped requirements. Also ensuring complete testing and verification are methods used for ensuring complex systems are designed with no common causes that cause severe hazards. Common cause analysis are sometimes required as part of the safety engineering tasks for theme parks, commercial/military aircraft, spacecraft, complex control systems, large electrical utility grids, nuclear power plants, automated industrial controls, medical devices or other safety safety-critical systems with complex functionality.

Basic elements of root cause

- Materials
  - Defective raw material
  - Wrong type for job
  - Lack of raw material
- Man Power
  - Inadequate capability
  - Lack of Knowledge
  - Lack of skill
  - Stress
  - Improper motivation
- Machine / Equipment
  - Incorrect tool selection
  - Poor maintenance or design
  - Poor equipment or tool placement
  - Defective equipment or tool
- Environment
  - Orderly workplace
  - Job design or layout of work
  - Surfaces poorly maintained
  - Physical demands of the task
  - Forces of nature
- Management
  - No or poor management involvement
  - Inattention to task
  - Task hazards not guarded properly
  - Other (horseplay, inattention....)
  - Stress demands

    - Lack of Process
- Methods
    - No or poor procedures
    - Practices are not the same as written procedures
    - Poor communication
- Management system
    - Training or education lacking
    - Poor employee involvement
    - Poor recognition of hazard
    - Previously identified hazards were not eliminated

Trial and error

Trial and error, or **trial by error** or **try an error**, is a general method of problem solving, fixing things, or for obtaining knowledge. "Learning doesn't happen from failure itself but rather from analyzing the failure, making a change, and then trying again."

In the field of computer science, the method is called **generate and test**. In elementary algebra, when solving equations, it is **"guess and check"**.

This approach can be seen as one of the two basic approaches to problem solving and is contrasted with an approach using insight and theory.

Methodology

This approach is more successful with simple problems and in games, and is often resorted to when no apparent rule applies. This does not mean that the approach need be careless, for an individual can be methodical in manipulating the variables in an attempt to sort through possibilities that may result in success. Nevertheless, this method is often used by people who have little knowledge in the problem area.

Simplest applications

Ashby (1960, section 11/5) offers three simple strategies for dealing with the same basic exercise-problem; and they have very different efficiencies: Suppose there are 1000 on/off switches which have to be set to a particular combination by random-based testing, each test to take one second. The strategies are:

- the perfectionist all-or-nothing method, with no attempt at holding partial successes. This would be expected to take more than 10^301 seconds, [i.e. 2^1000 seconds, or 3·5×(10^291) centuries!];

- a serial-test of switches, holding on to the partial successes (assuming that these are manifest) would take 500 seconds; while
- a parallel-but-individual testing of all switches simultaneously would take only one second.

Note the tacit assumption here that no intelligence or insight is brought to bear on the problem. However, the existence of different available strategies allows us to consider a separate ("superior") domain of processing — a *"meta-level"* above the mechanics of switch handling — where the various available strategies can be randomly chosen. Once again this is "trial and error", but of a different type. This leads us to:

Trial-and-error Hierarchies

Ashby's book develops this "meta-level" idea, and extends it into a whole recursive sequence of levels, successively above each other in a systematic hierarchy. On this basis he argues that human intelligence emerges from such organization: relying heavily on trial-and-error (at least initially at each new stage), but emerging with what we would call "intelligence" at the end of it all. Thus presumably the topmost level of the hierarchy (at any stage) will still depend on simple trial-and-error.

Traill (1978/2006) suggests that this Ashby-hierarchy probably coincides with Piaget's well-known theory of developmental stages. [This work also discusses Ashby's 1000-switch example; see §C1.2]. After all, it is part of Piagetian doctrine that children learn by first *actively doing* in a more-or-less random way, and then hopefully learn from the consequences — which all has a certain resemblance to Ashby's random "trial-and-error".

The basic strategy in many fields?

Traill (2008, *espec. Table "S" on p.31*) follows Jerne and Popper in seeing this strategy as probably underlying *all* knowledge-gathering systems — *at least in their initial phase*.

Four such systems are identified:

- Darwinian evolution which "educates" the DNA of the species!
- The brain of the individual (just discussed);
- The "brain" of society-as-such (including the publicly-held body of science); and
- The immune system.

An ambiguity: Can we have "intention" during a "trial"

In the Ashby-and-Cybernetics tradition, the word "trial" usually implies *random-or-arbitrary*, without any *deliberate* choice. However amongst non-cyberneticians, "trial" will often imply a deliberate subjective act by some adult human agent; (e.g. in a court-room, or laboratory). So that has sometimes led to confusion.

Of course the situation becomes even more confusing if one accepts Ashby's hierarchical explanation of intelligence, and its implied ability to be deliberate and to creatively design — all based ultimately on *non-deliberate* actions! The lesson here seems to be that one must simply be careful to clarify the meaning of one's own words, and indeed the words of others. [Incidentally it seems that *consciousness* is not an essential ingredient for *intelligence* as discussed above.]

Features

Trial and error has a number of features:

- solution-oriented: trial and error makes no attempt to discover *why* a solution works, merely that it *is* a solution.
- problem-specific: trial and error makes no attempt to generalise a solution to other problems.
- non-optimal: trial and error is generally an attempt to find *a* solution, not *all* solutions, and not the *best* solution.
- needs little knowledge: trials and error can proceed where there is little or no knowledge of the subject.

It is possible to use trial and error to find all solutions or the best solution, when a testably finite number of possible solutions exist. To find all solutions, one simply makes a note and continues, rather than ending the process, when a solution is found, until all solutions have been tried. To find the best solution, one finds all solutions by the method just described and then comparatively evaluates them based upon some predefined set of criteria, the existence of which is a condition for the possibility of finding a best solution. (Also, when only one solution can exist, as in assembling a jigsaw puzzle, then any solution found is the only solution and so is necessarily the best.)

Examples

Trial and error has traditionally been the main method of finding new drugs, such as antibiotics. Chemists simply try chemicals at random until they find one with the desired effect. In a more sophisticated version, chemists select a narrow range of chemicals it is thought may have some effect. (The latter case can be alternatively considered as a changing of the problem rather than of the solution strategy: instead of "What chemical will work well as an antibiotic?" the problem in the sophisticated approach is "Which, if any, of the chemicals in this narrow range will work well as an antibiotic?") The method is used widely in many disciplines, such as polymer technology to find new polymer types or families.

The scientific method can be regarded as containing an element of trial and error in its formulation and testing of hypotheses. Also compare genetic algorithms, simulated annealing and reinforcement learning - all varieties for search which apply the basic idea of trial and error.

Biological evolution is also a form of trial and error. Random mutations and sexual genetic variations can be viewed as trials and poor reproductive fitness, or lack of improved fitness, as the error. Thus after a long time 'knowledge' of well-adapted genomes accumulates simply by virtue of them being *able* to reproduce.

Bogosort, a conceptual sorting algorithm (that is extremely inefficient and impractical), can be viewed as a trial and error approach to sorting a list. However, typical simple examples of bogosort do not track which orders of the list have been tried and may try the same order any number of times, which violates one of the basic principles of trial and error. Trial and error is actually more efficient and practical than bogosort; unlike bogosort, it is guaranteed to halt in finite time on a finite list, and might even be a reasonable way to sort extremely short lists under some conditions.

Issues with trial and error

Trial and error is usually a last resort for a particular problem, as there are a number of problems with it. For one, trial and error is tedious and monotonous. Also, it is very time-consuming; chemical engineers must sift through millions of various potential chemicals before they find one that works. There is also an element of risk, in that if a certain attempt at a solution is extremely erroneous, it can produce disastrous results that may or may not be repairable. Fortunately, computers are best suited for trial and error; they do not succumb to the boredom that humans do, can test physical challenges in a virtual environment where they will not do harm, and can potentially do thousands of trial-and-error segments in the blink of an eye.

Chapter- 3

Problem-Solving Methodologies

Eight Disciplines Problem Solving

Eight Disciplines Problem Solving is a method used to approach and to resolve problems, typically employed by quality engineers or other professionals.

D1: **Use a Team**: Establish a team of people with product/process knowledge.

D2: **Define the Problem**: Specify the problem by identifying in quantifiable terms the who, what, where, when, why, how and how many (5W2H) for the problem.

D3: **Implement and verify Interim Actions**: Define and implement containment actions to isolate the problem from any customer.

D4: **Identify and Verify Root Causes**: Identify all potential causes that could explain why the problem occurred. Also identify why the problem has not been noticed at the time it occurred. All causes shall be verifed or proved, not determined by fuzzy brainstorming.

D5: **Choose and verify Permanent Corrective Actions (PCAs)**: Through pre-production programs quantitatively confirm that the selected corrective actions will resolve the problem for the customer.

D6: **Implement and validate PCAs**: Define and Implement the best corrective actions.

D7: **Prevent recurrence**: Modify the management systems, operation systems, practices and procedures to prevent recurrence of this and all similar problems.

D8: **Congratulate your Team**: Recognize the collective efforts of the team. The team needs to be formally thanked by the organization.

8D has become a standard in the Auto, Assembly and other industries that require a thorough structured problem solving process using a team approach.

History

People have been using structured problem solving methodologies since the Dark Ages. No single source can claim to be the fountain of modern problem solving techniques. Ford developed their method, while the military also developed and quantified their own process during World War II. Both of these methods revolve around the Eight Disciplines.

Ford's Perspective

The development of a Team Oriented Problem Solving strategy, based on the use of statistical methods of data analysis, was developed at Ford Motor Company. The executives of the Powertrain Organization (transmissions, chassis, engines) wanted a methodology where teams(design engineering, manufacturing engineering, and production) could work on recurring problems. In 1986, the assignment was given to develop a manual and a subsequent course that would achieve a new approach to solving tough engineering design and manufacturing problems. The manual for this methodology was documented and defined in "Team Oriented Problem Solving"(TOPS), first published in 1987. The manual and subsequent course material was piloted at World Headquarters in Dearborn, Michigan. Many changes and revisions were made based on feedback from the pilot sessions. This has been Ford's approach to problem solving ever since. It was never based on any military standard or other existing problem solving methodology. The material is extensive and the 8D titles are merely the chapter headings for each step in the process. Ford also refers to their current variant as G8D (Global 8D).

Military Usage

The US Government first standardized a process during the Second World War as Military Standard 1520 'Corrective Action and Disposition System for Nonconforming Material'. This military standard focused on nonconforming material and the disposition of the material.

The 8D Problem Solving Process is used to identify, correct and eliminate recurring problems. The methodology is useful in product and process improvement. It establishes a permanent corrective action based on statistical analysis of the problem. It focuses on the origin of the problem by determining Root Causes.

This 'Determine a Root Cause' step is a part of the military usage of the 8D's but was not a reference in the development of the 8D problem solving methodology and is not referenced or included in the TOPS manual or course.

Usage

Many disciplines are typically involved in the "8D" process, all of which can be found in various textbooks and reference materials used by Quality Assurance professionals. For example, an "Is/Is Not" worksheet is a common tool employed at D2, and a "Fishbone Diagram" or "5 Why Analysis" are common tools employed at step D4.

In the late 1990s, Ford developed a revised version of the 8D process, that they call "Global 8D" (G8D) which is the current global standard for Ford and many other companies in the automotive supply chain. The major revisions to the process are as follows:

- Addition of a D0 (D-Zero) step as a gateway to the process. At D0, the team documents the symptoms that initiated the effort along with any Emergency Response Actions (ERAs) that were taken before formal initiation of the G8D. D0 also incorporates standard assessing questions meant to determine whether a full G8D is required. The assessing questions are meant to ensure that in a world of limited problem-solving resources, the efforts required for a full team-based problem-solving effort are limited to those problems that warrant these resources.

- Addition of Escape Point to D4 through D6. The idea here is to consider not only the Root cause of a problem, but equally importantly, what went wrong with the control system in allowing this problem to escape. Global 8D requires the team to identify and verify this Escape Point (defined as the earliest control point in the control system following the Root Cause that should have detected the problem but failed to do so) at D4. Then, through D5 and D6, the process requires the team to choose, verify, implement, and validate Permanent Corrective Actions to address the Escape Point.

Recently, the 8D process has been employed significantly outside the auto industry. As part of Lean initiatives and Continuous Improvement Processes it is employed extensively within Food Manufacturing, High Tech and Health Care industries.

GROW model

The **GROW model** (or process) is a technique for problem solving or goal setting. It was developed in the UK by Graham Alexander, Alan Fine and Sir John Whitmore and used extensively in the corporate coaching market in the late 1980s and 1990s. The three are well known because of their significant contributions in the world of executive coaching. Max Landsberg, who worked as an adjunct consultant with the core development team also describes GROW in his book *The Tao of Coaching*.

GROW is very well known in the business arena but it also has many applications in everyday life. The particular value of GROW is that it provides an effective, structured methodology which both helps set goals effectively **and** is a problem solving process.

It can be used by anyone without special training. While there are many methodologies that can be used to address problems, the value of GROW is that it is easily understood, straightforward to apply and very thorough. In addition it is possible to apply it to a large variety of issues in a very effective way.

Stages of GROW

There are a number of different versions of the GROW model. This version presents one view of the stages but there are others. The 'O' in this version has two meanings.

G	Goal	This is the end point, where the client wants to be. The goal has to be defined in such a way that it is very clear to the client when they have achieved it.
R	Reality	This is how far the client is away from their goal. If the client were to look at all the steps they need to take in order to achieve the goal, the Reality would be the number of those steps they have completed so far.
O	Obstacles	There will be Obstacles stopping the client getting from where they are now to where they want to go. If there were no Obstacles the client would already have reached their goal.
	Options	Once Obstacles have been identified the client need to find ways of dealing with them if they are to make progress. These are the Options.
W	Way Forward	The Options then need to be converted into action steps which will take the client to their goal. These are the Way Forward.

As with many simple principles any user of GROW can apply a great deal of skill and knowledge at each stage but the basic process remains as written above. There are numerous questions which the coach could use at any point and part of the skill of the coach is to know which questions to use and how much detail to uncover.

This is a very simple example of using the GROW model to achieve a goal.

This example deals with weight loss. The clients wants: ‘To bring my weight down to 120 pounds in three months and keep it down’.

That is their Goal. The GROW approach would then be to establish the Reality by stating what their weight is now. The coach would then ask awareness questions to deepen understanding of what is happening when the client tries to lose weight, thus identifying the Obstacles. These questions could include:

When you have been able to lose weight – what made the difference?

What is the difference between the times you are able to keep weight off and the times when you put it on again?

What would have to change for you to be sure you could lose the weight and keep it off?

If the client genuinely answers these questions they will discover new information about what works and does not work for them in terms of weight loss, and create some potential for change. It then becomes possible to create some strategies or Options which get around the Obstacles. These could include looking at which diets or exercise regimes work best, or finding a specific type of support. Once the client knows the strategies that are likely to work they can establish a Way Forward which involves taking action steps. This is where they commit to what they will do in the short term to put the strategies into effect. For instance, one action might be asking a particular person for support, and another might be to buy a different selection of foods.

The same principles can be applied whatever goal or problem the client has. GROW can be used on technical problems, issues regarding processes, strategy questions, interpersonal issues and many more. Almost any situation where there is something to be achieved and there is an Obstacle can be tackled with GROW. The model can also be used by a group who are all working on the same problem or goal.

Here is an example of using GROW to coach someone with a business problem e.g. bad time management.

The model is described below with the output you should expect from each section.

Goal Stage:

The goal could be success criteria with a date by which they wanted to achieve it. So the coach and client would have to be clear on what they would be seeing and hearing that would tell them that the client had solved their issue in relation to time management.

Output: A clear goal in SMART format which both coach and client would know if it was achieved.

Reality Stage:

The Reality would be what was actually happening now in relation to time management. The key point about the Reality section is that it must be in the same terms as the Goal. So if part of the Goal was to create to do lists every day part of the Reality statement should be how often to do lists are created now.

Output: A clear statement of what is happening at the moment in the terms of the Goal.

Obstacles stage:

Obstacles will be found in one of four places:

1. The person themselves

2. Caused by other people

3. A lack of skills, knowledge or experience

4. Caused by the physical environment

The coach should check for Obstacles in all four areas unless it is absolutely obvious it is not needed. In most cases there will be Obstacles within the client themselves as there are usually things the client is avoiding which means they are not achieving their goal. But the other areas can be equally important.

The coach should ensure that she/he understands how each of the Obstacles is obstructing the client from their Goal. Very often the client will present beliefs about themselves or justifications of their position as Obstacles. For instance ‘I am too old to change’ or ‘I have no self discipline’. In these instances the coach should probe to discover what is actually happening that causes the client to not create or stick to plans rather than opinions or beliefs.

Output: A reasonably comprehensive list of individual Obstacles having considered all the four areas. (It is a matter of judgment to say how much investigation you should do at any stage of the model) Both coach and client should clearly understand how each of the Obstacles is stopping the client going straight from where they are to where they want to go.

Options stage:

Once Obstacles have been clearly separated out it become much easier to find ways to get around them. This is the purpose of the Options section. It is very important to take the Obstacles one at a time as clients tend to want to keep them muddled.

Options can often simply be created by asking the client how they could get around a particular Obstacle. Once the Obstacles are separated out clients find it much easier to deal with them. If the client needs further ideas the coach can ask questions to help the client think how to get around the Obstacle. Specifically:

1. Where they could get additional knowledge or resources.

2. Who might be prepared to help

3. If they have dealt with a similar issue in the past

4. If the Obstacle can be minimized

5. How someone who could deal with the Obstacle well would go about it.

All these ideas exist within the client but will not necessarily be considered unless the coach asks the client to look at them.

Output: There should be a list of Options which coach and client agree should get the client around all of the Obstacles.

Way Forward or Actions stage:

Once you have a list of Options it is usually fairly straightforward to convert them into actions which can be completed in one to four weeks. The coach should check the client is committed to carrying out the actions and has thought how they will deal with any adverse consequences that might occur.

Output: A list of action steps which the client is committed to and both coach and client are confident will deal with the Obstacles and move the client towards their Goal.

The GROW principle and the Inner Game

GROW was developed out of the Inner Game theory developed by Timothy Gallwey, Gallwey was a tennis coach who noticed that he could often see what a player was doing incorrectly but that simply telling them what they should be doing did not bring about lasting change.

This is often illustrated by the example of a player who does not keep their eye on the ball. Most coaches would give instructions such as: ‘Keep your eye on the ball’ to try and correct this. The problem with this sort of instruction is that a player will be able to follow it for a short while but be unable to keep it in the front of their minds in the long term. This means that progress was slow. The result was that coaches and players grew increasingly frustrated at the slowness of progress but no one had better system of coaching.

So one day, instead of giving an instruction, Gallwey asked the player to say `bounce' out loud when the ball bounced and `hit' out loud when they hit it

The result was that the players started to improve without a lot of effort because they were keeping their eye on the ball. But because of the way the instruction was given they did not have a voice in their heads saying ‘I must keep my eye on the ball.’ They were simply playing a simple game while they were playing tennis. Once Gallwey saw how play could be improved in this way he stopped giving instructions and started asking questions that would help the player discover for himself what worked and what needed to change. This was the birth of the Inner Game.

The basic methodology of GROW came out of Gallweys work with tennis players. For example the first stage in this process would be to set a target which the player wanted to achieve. For example if a player wanted to improve his first serve Gallwey would ask how many first serves out of ten they would like to get in. This was the target or goal.

The Reality would be defined by asking the player to serve 10 balls and seeing how many first serves went in.

Gallwey would then ask awareness raising questions such as "What do you notice you are doing differently when you the ball goes in or out?" This would enable the player to discover for themselves what they were changing about their mind and body when the serve when in or out.

They had then defined their Obstacles and Options. They therefor learnt for themselves what they had to change in order to meet their serving targets and they had a clear Way Forward.

From Gallwey's experience with tennis players it is possible to define a number of learning principles which can be applied to any learning situation whether sport based or not. For example:

1. In most learning situations the learner is rarely focused on what is happening during the process. But if they focus their attention on a relevant aspect of what is actually happening during the process, rather than what they 'should' be doing or trying to get it 'right' they will make progress much faster.

2 Learning happens best when the learner is focused on the present. If they are focused on the present they will not struggle to prove or remember something but rather make discoveries as they go along.

3 If the learner is trying to look good or using a lot of unfocused effort they will interfere with the learning process. The less interference with their learning, the faster they progress.

Coaches using the Inner Game soon realised they could apply the principles in other learning situations. GROW was developed as a structured framework to use the Inner Game principles to achieve goals. The originators saw that, just as in sport, many individuals were struggling to achieve goals because they were not learning from experience and were not aware of the knowledge within themselves that would help them.

How to Solve It

Four principles

How to Solve It suggests the following steps when solving a mathematical problem:

1. First, you have to *understand the problem*.
2. After understanding, then *make a plan*.
3. *Carry out the plan*.
4. *Look back* on your work. How could it be better?

If this technique fails, Pólya advises: "If you can't solve a problem, then there is an easier problem you can solve: find it." Or: "If you cannot solve the proposed problem, try to solve first some related problem. Could you imagine a more accessible related problem?"

First principle: Understand the problem

"Understand the problem" is often neglected as being obvious and is not even mentioned in many mathematics classes. Yet students are often stymied in their efforts to solve it, simply because they don't understand it fully, or even in part. In order to remedy this oversight, Pólya taught teachers how to prompt each student with appropriate questions, depending on the situation, such as:

- What are you asked to find or show?
- Can you restate the problem in your own words?
- Can you think of a picture or a diagram that might help you understand the problem?
- Is there enough information to enable you to find a solution?
- Do you understand all the words used in stating the problem?
- Do you need to ask a question to get the answer?

The teacher is to select the question with the appropriate level of difficulty for each student to ascertain if each student understands at their own level, moving up or down the list to prompt each student, until each one can respond with something constructive.

Second principle: Devise a plan

Pólya mentions that there are many reasonable ways to solve problems. The skill at choosing an appropriate strategy is best learned by solving many problems. You will find choosing a strategy increasingly easy. A partial list of strategies is included:

- Guess and check
- Make an orderly list
- Eliminate possibilities
- Use symmetry
- Consider special cases
- Use direct reasoning
- Solve an equation

Also suggested:

- Look for a pattern
- Draw a picture
- Solve a simpler problem
- Use a model
- Work backward
- Use a formula
- Be creative
- Use your head/noggin

Third principle: Carry out the plan

This step is usually easier than devising the plan. In general, all you need is care and patience, given that you have the necessary skills. Persist with the plan that you have chosen. If it continues not to work discard it and choose another. Don't be misled; this is how mathematics is done, even by professionals.

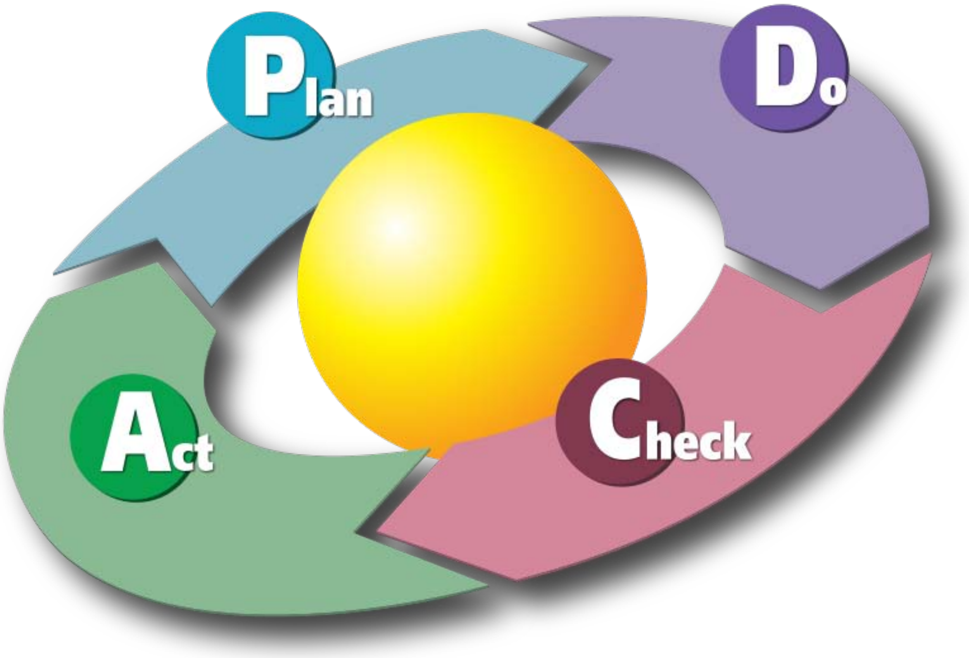
Fourth principle: Review/extend

Pólya mentions that much can be gained by taking the time to reflect and look back at what you have done, what worked and what didn't. Doing this will enable you to predict what strategy to use to solve future problems, if these relate to the original problem.

PDCA

PDCA (**plan-do-check-act**) is an iterative four-step problem-solving process typically used in business process improvement. It is also known as the **Deming circle**, **Shewhart cycle**, **Deming cycle**, **Deming wheel**, or **plan-do-study-act**.

Meaning



The PDCA cycle

PLAN

Establish the objectives and processes necessary to deliver results in accordance with the expected output. By making the expected output the focus, it differs from other techniques in that the completeness and accuracy of the specification is also part of the improvement.

DO

Implement the new processes. Often on a small scale if possible.

CHECK

Measure the new processes and compare the results against the expected results to ascertain any differences.

ACT

Analyze the differences to determine their cause. Each will be part of either one or more of the P-D-C-A steps. Determine where to apply changes that will include improvement. When a pass through these four steps does not result in the need to improve, refine the scope to which PDCA is applied until there is a plan that involves improvement.

About

PDCA was made popular by Dr. W. Edwards Deming, who is considered by many to be the father of modern quality control; however he always referred to it as the "Shewhart cycle". Later in Deming's career, he modified PDCA to "Plan, Do, Study, Act" (PDSA) so as to better describe his recommendations.

The concept of PDCA is based on the scientific method, as developed from the work of Francis Bacon (Novum Organum, 1620). The scientific method can be written as "hypothesis" - "experiment" - "evaluation" or plan, do, and check. Shewhart described manufacture under "control" - under statistical control - as a three step process of specification, production, and inspection. He also specifically related this to the scientific method of hypothesis, experiment, and evaluation. Shewhart says that the statistician "must help to change the demand [for goods] by showing...how to close up the tolerance range and to improve the quality of goods". Clearly, Shewhart intended the analyst to take action based on the conclusions of the evaluation. According to Deming, during his lectures in Japan in the early 1950s, the Japanese participants shortened the steps to the now traditional *plan, do, check, act*. Deming preferred *plan, do, study, act* because "study" has connotations in English closer to Shewhart's intent than "check".

A fundamental principle of the scientific method and PDSA is iteration - once a hypothesis is confirmed (or negated), executing the cycle again will extend the knowledge further. Repeating the PDSA cycle can bring us closer to the goal, usually a perfect operation and output.

In Six Sigma programs, the PDSA cycle is called "define, measure, analyze, improve, control" (DMAIC). The iterative nature of the cycle must be explicitly added to the DMAIC procedure.

PDSA should be repeatedly implemented in spirals of increasing knowledge of the system that converge on the ultimate goal, each cycle closer than the previous. One can envision an open coil spring, with each loop being one cycle of the scientific method - PDSA, and each complete cycle indicating an increase in our knowledge of the system under study. This approach is based on the belief that our knowledge and skills are limited, but improving. Especially at the start of a project, key information may not be known; the PDSA - scientific method - provides feedback to justify our guesses (hypotheses) and increase our knowledge. Rather than enter "analysis paralysis" to get it perfect the first time, it is better to be approximately right than exactly wrong. With the improved knowledge, we may choose to refine or alter the goal (ideal state). Certainly, the PDSA approach can bring us closer to whatever goal we choose.

Rate of change, that is, rate of improvement, is a key competitive factor in today's world. PDSA allows for major 'jumps' in performance ('breakthroughs' often desired in a Western approach), as well as Kaizen (frequent small improvements associated with an Eastern approach). In the United States a PDSA approach is usually associated with a sizable project involving numerous people's time, and thus managers want to see large 'breakthrough' improvements to justify the effort expended. However, the scientific method and PDSA apply to all sorts of projects and improvement activities.

The power of Deming's concept lies in its apparent simplicity. The concept of feedback in the scientific method, in the abstract sense, is today firmly rooted in education. While apparently easy to understand, it is often difficult to accomplish on an on-going basis due to the intellectual difficulty of judging one's proposals (hypotheses) on the basis of measured results. Many people have an emotional fear of being shown "wrong", even by objective measurements. To avoid such comparisons, we may instead cite complacency, distractions, loss of focus, lack of commitment, re-assigned priorities, lack of resources, etc.

RPR Problem Diagnosis

RPR is a problem diagnosis method specifically designed to determine the Root Cause of IT problems.

Overview

RPR (Rapid Problem Resolution) deals with failures, incorrect output and performance issues, and its particular strengths are in the diagnosis of ongoing & recurring grey problems. The method comprises:

- Core Process

- Supporting Techniques

The Core Process defines a step-by-step approach to problem diagnosis and has three phases:

- Discover
    - Gather & review existing information
    - Reach an agreed understanding
- Investigate
    - Create & execute a diagnostic data capture plan
    - Analyse the results & iterate if necessary
    - Identify Root Cause
- Fix
    - Translate diagnostic data
    - Determine & implement fix
    - Confirm Root Cause addressed

The Supporting Techniques detail how the objectives of the Core Process steps are achieved, and cite examples using tools and techniques that are available in every business.

Standards alignment

RPR has been fully aligned with ITIL v3 since RPR 2.01 was released in April 2008. RPR fits directly into the ITIL v3 Problem Management Process as a sub-process. Some organisations handle ongoing recurring problems within Incident Management, and RPR also fits into the ITIL v3 Incident Management Process as a sub-process.

COBIT also defines a Problem Management Process (DS10) with key activity of *Perform root cause analysis*. RPR is a superset of this step in that it defines a process that covers all of the activities needed to perform Problem investigation & diagnosis, including Root Cause identification.

Limitations

RPR has some limitations and considerations, including:

- RPR deals with a single symptom at a time
- RPR is not a forensic technique and so historical data alone is rarely sufficient
- The Investigate phase requires the user to experience the problem one more time

History

The method was originally developed by Advance7 in 1990 as the Rapid Problem Resolution Method, with the first fully documented version produced in 1995. Early versions included problem management guidance but this was removed over time as the method became more closely aligned to ITIL. RPR is now focused on Problem Diagnosis based on Root Cause

Identification. Due to the highly practical nature of the Supporting Techniques and the ever changing IT landscape, Advance7 continues to develop RPR to keep it relevant to current IT environments.

Until November 2007 Advance7 made the RPR material available to its employees only, although a limited number of other IT professionals had been trained in the use of the method. In late 2007 the company announced its intention to make RPR training and material more widely available.

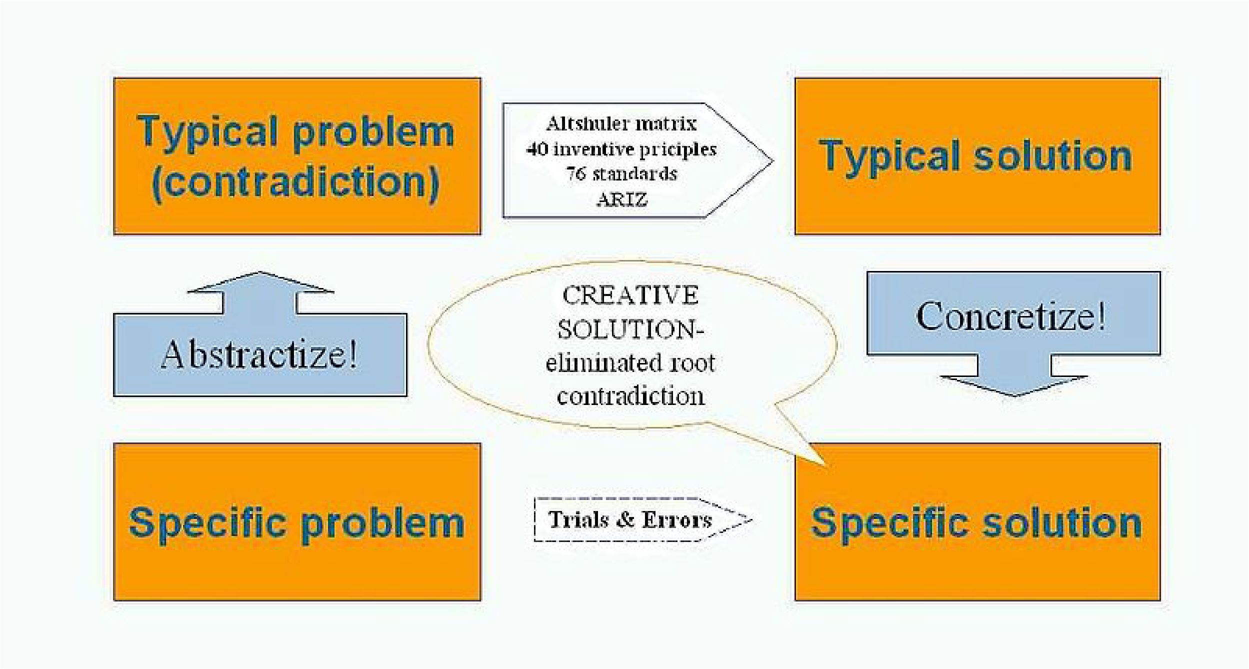
In March 2009 the TSO added a significant amount of RPR information to the ITIL Best Practice Live website within the areas dealing with Problem Management.

TRIZ

TRIZ Russian: Теория решения изобретательских задач (***T****eoriya* ***R****esheniya* ***I****zobretatelskikh* ***Z****adatch*)) is "a problem-solving, analysis and forecasting tool derived from the study of patterns of invention in the global patent literature". It was developed by Soviet engineer and researcher Genrich Altshuller and his colleagues, beginning in 1946. In English the name is typically rendered as "the theory of inventive problem solving", and occasionally goes by the English acronym "TIPS". The approach involves identifying generalisable problems and borrowing solutions from other fields. TRIZ practitioners aim to create an algorithmic approach to the invention of new systems, and the refinement of old systems.

TRIZ is variously described as a methodology, tool set, knowledge base, and model-based technology for generating new ideas and solutions for problem solving. It is intended for application in problem formulation, system analysis, failure analysis, and patterns of system evolution. Splits have occurred within TRIZ advocacy, and interpretation of its findings and applications are disputed.

History



TRIZ process for creative problem solving

The development of TRIZ began in 1946 with the mechanical engineer Genrich Altshuller studying patents on behalf of the Russian navy. Altshuller's job was to inspect invention proposals, help document them, and help others to invent. By 1969 he had reviewed about 40,000 patent abstracts in order to find out in what way the innovation had taken place. By examining a large database of inventions, Altshuller concluded that only one per cent of inventions were genuinely original; the rest represented the novel application of previous ideas. Altshuller argued that "An invention is the removal of technical contradictions". Along these lines, he said that to develop a method for inventing, one must scan a large number of inventions, identify the contradictions underlying them, and formulate the principle used by the inventor for their removal. Over the next years, he developed 40 Principles of Invention, several Laws of Technical Systems Evolution, the concepts of technical and physical contradictions that creative inventions resolve, the concept of Ideality of a system and numerous other theoretical and practical approaches.

Concerned at the state of the Soviet Union following World War II, in 1949 Altshuller and his colleague Rafael Shapiro wrote to Josef Stalin, encouraging the use of TRIZ to rebuild the economy. However, the letter also contained sharp criticism of the situation for innovation in the USSR at the time. Altshuller was arrested and sentenced to 25 years hard labour, although he was freed following Stalin's death in 1953. Altshuller and Shapiro published their first paper in 1956 in *Voprosy Psikohologii* (Questions of Psychology) entitled "On the psychology of inventive creation".

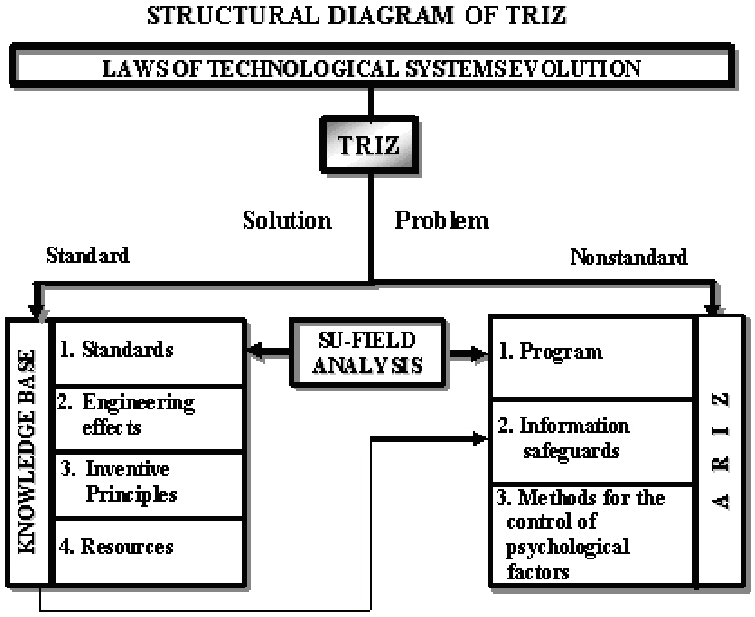
In 1971, after returning to Baku some years earlier, Altshuller founded the Azerbaijan Public Institute for Inventive Creation. This institute incubated the TRIZ movement, producing a generation of Altshuller proteges. Altshuller himself gave workshops across the Soviet Union. In 1989 the TRIZ Association was formed, with Altshuller chosen as President.

Following the end of the cold war, overseas interest in TRIZ increased. In 1996 the online English-language publication *The TRIZ journal* began.

Basic principles of TRIZ

The TRIZ process presents an algorithm for the analysis of problems in a technological system. The fundamental view is that almost all "inventions" are reiterations of previous discoveries already made in the same or other fields, and that problems can be reduced to contradictions between two elements. The goal of TRIZ analysis is to achieve a better solution than a mere trade-off between the two elements, and the belief is that the solution almost certainly already exists somewhere in the patent literature.

A problem is first defined in terms of the ideal solution. The problem is analysed into its basic, abstract constituents according to a list of 39 items (for example, the weight of a stationery object, the use of energy by a moving object, the ease of repair etc.), and reframed as a contradiction between two of these constituents. Using a contradiction matrix based upon large-scale analysis of patents, a series of suggested abstract solutions (for example "move from straight lines to curved", or "make the object porous") is offered, helping the analyst find creative solutions.



Structural diagram of TRIZ

ARIZ consists of a program (sequence of actions) for the exposure and solution of contradictions, i.e. the solution of problems. ARIZ includes: the program itself, information safeguards supplied by the knowledge base (shown by an arrow in fig.1.1), and methods for the control of psychological factors, which are a component part of the methods for developing a creative imagination. Furthermore, sections of ARIZ are predetermined for the selection of problems and the evaluation of the received solution.

Classification of a system of standard solutions for inventive problems, as well as the standards themselves, is built on the basis of *Su-Field Analysis* of technological systems. Su-Field Analysis is also a component part of the program ARIZ (shown by arrows in fig. 1).

The use of different elements of TRIZ for specific functions are shown in Table 1: "Functions and Structure of TRIZ." A system of laws for the development of technology, a system of standards for the solution of inventive problems, and Su-Field Analysis are used to forecast the development of technology, to search for and select problems, and to evaluate the received solution. For the development of a creative imagination, all elements of TRIZ can be used, although particular stress is given to methods for developing a creative imagination. The solution of inventive problems is realized with the help of laws for the development of technological systems, the knowledge base, Su-Field Analysis, ARIZ, and, in part, with the help of methods for the development of a creative imagination.

By means of TRIZ, both known and unknown types of problems could be solved. Known (standard) types of inventive problems are solved with the use of the knowledge base, and unknown (nonstandard) – with the use of ARIZ. As experience grows, solutions for a class of known types of problems increase and exhibit a structure.

At the present time, computer programs have been developed on the basis of TRIZ that try to provide intellectual assistance to engineers and inventors during the solution of technological problems. These programs also try to reveal and forecast emergency situations and undesirable occurrences.

Essentials

Basic terms

- Ideal Final Result (IFR) - the ultimate idealistic solution of a problem when the desired result is achieved by itself;
- Administrative Contradiction - contradiction between the needs and abilities;
- Technical Contradiction - an inverse dependence between parameters/characteristics of a machine or technology;
- Physical Contradiction - opposite/contradictory physical requirements to an object;
- Separation principle - a method of resolving physical contradictions by separating contradictory requirements;
- VePol or SuField - a minimal technical system consisting of two material objects (substances) and a "field". "Field" is the source of energy whereas one of the substances is "transmission" and the other one is the "tool";

- FePol - a sort of VePol where "substances" are ferromagnetic objects;
- Level of Invention;
- Standard - a standard inventive solution of a higher level;
- Law of Technical Systems Evolution;
- ARIZ - Algorithm of Inventive Problems Solving, which combines various specialized methods of TRIZ into one universal tool;

Identifying a problem: contradictions

Altshuller believed that inventive problems stem from contradictions (one of the basic TRIZ concepts) between two or more elements, such as, "If we want more acceleration, we need a larger engine; but that will increase the cost of the car," that is, more of something desirable also brings more of something less desirable, or less of something else also desirable.

These are called Technical Contradictions by Altshuller. He also defined so-called physical or inherent contradictions: More of one thing and less of the same thing may both be desired in the same system. For instance, a higher temperature may be needed to melt a compound more rapidly, but a lower temperature may be needed to achieve a homogeneous mixture.

An inventive situation might involve several such contradictions. The inventor typically *trades* one contradictory parameter for another; no special inventiveness is needed for that. Rather, the inventor would develop some creative approach for *resolving* the contradiction, such as inventing an engine that produces more acceleration without increasing the cost of the engine.

Inventive principles and the matrix of contradictions

Altshuller screened patents in order to find out what kind of contradictions were resolved or dissolved by the invention and the way this had been achieved. From this he developed a set of 40 inventive principles and later a Matrix of Contradictions. Rows of the matrix indicate the 39 system features that one typically wants to improve, such as speed, weight, accuracy of measurement and so on. Columns refer to typical undesired results. Each matrix cell points to principles that have been most frequently used in patents in order to resolve the contradiction.

For instance, Dolgashev mentions the following contradiction: Increasing accuracy of measurement of machined balls while avoiding the use of expensive microscopes and elaborate control equipment. The matrix cell in row "accuracy of measurement" and column "complexity of control" points to several principles, among them the Copying Principle, which states, "Use a simple and inexpensive optical copy with a suitable scale instead of an object that is complex, expensive, fragile or inconvenient to operate." From this general invention principle, the following idea might solve the problem: Taking a high-resolution image of the machined ball. A screen with a grid might provide the required measurement.

Laws of technical system evolution

Altshuller also studied the way technical systems have been developed and improved over time. From this, he discovered several trends (so called Laws of Technical Systems Evolution) that help engineers predict what the most likely improvements that can be made to a given product are. The most important of these laws involves the ideality of a system.

Substance-field analysis

One more technique that is frequently used by inventors involves the analysis of substances, fields and other resources that are currently not being used and that can be found within the system or nearby. TRIZ uses non-standard definitions for substances and fields. Altshuller developed methods to analyze resources; several of his invention principles involve the use of different substances and fields that help resolve contradictions and increase ideality of a technical system. For instance, videotext systems used television signals to transfer data, by taking advantage of the small time segments between TV frames in the signals.

Su-Field Analysis (structural substance-field analysis) produces a structural model of the initial technological system, exposes its characteristics, and with the help of special laws, transforms the model of the problem. Through this transformation the structure of the solution that eliminates the shortcomings of the initial problem is revealed. Su-Field Analysis is a special language of formulas with which it is possible to easily describe any technological system in terms of a specific (structural) model. A model produced in this manner is transformed according to special laws and regularities, thereby revealing the structural solution of the problem.

ARIZ - algorithm of inventive problems solving

ARIZ (Russian acronym of Алгоритм решения изобретательских задач - АРИЗ) - Algorithm of Inventive Problems Solving - is a list of about 85 step-by-step procedures to solve complicated invention problems, where other tools of TRIZ alone (Su-field analysis, 40 inventive principles, etc.) are not sufficient.

Various TRIZ software is based on this algorithm (or an improved one).

Starting with an updated matrix of contradictions, semantic analysis, subcategories of inventive principles and lists of scientific effects, some new interactive applications are other attempts to simplify the problem formulation phase and the transition from a generic problem to a whole set of specific solutions.

Use of TRIZ methods in industry

It has been reported that car companies Fordand Daimler-Chrysler, Johnson & Johnson, aeronautics companies Boeing, NASA, technology companies Hewlett Packard, Motorola, General Electric, Xerox, IBM, LG and Samsung, and Procter and Gamble and Kodak have used TRIZ methods in some projects.

Approaches which are modifications/derivatives of TRIZ

1. SIT (Systematic Inventive Thinking)
2. ASIT (Advanced Systematic Inventive Thinking)
3. USIT (Unified Systematic Inventive Thinking)
4. JUSIT (Japanese version of Unified Systematic Inventive Thinking)
5. Southbeach notation

Example applications

Problem solving is of crucial importance in engineering when products or processes fail, so corrective action can be taken to prevent further failures. Perhaps of more value, problem solving can be applied to a product or process prior to an actual fail event i.e. a potential problem can be predicted, analyzed and mitigation applied so the problem never actually occurs. Techniques like Failure Mode Effects Analysis can be used to proactively reduce the likelihood of problems occurring. Forensic engineering is an important technique of failure analysis which involves tracing product defects and flaws. Corrective action can then be taken to prevent further failures.