System Architecture

Creativity

Thomas H. Speller, Jr.
Engineering Systems Division
Doctoral Candidate

January 12, 2007

ESD.34

Professor Ed Crawley

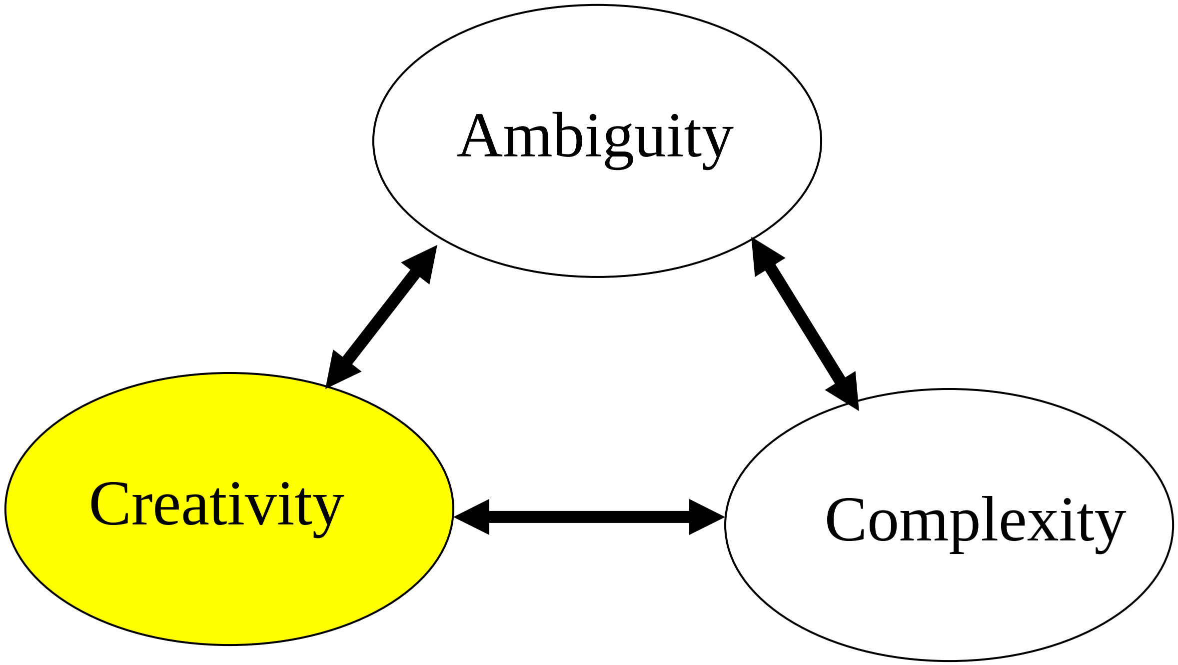
Today's Topics on Creativity

- Introduction
- Creativity
  - Nature
  - Design Rules and Combinatorics
  - Work of Vance and de Bono
- TRIZ theory
  - TRIZ, Value Engineering and the Semantic Web tool
- Radiant Thinking, Mind Mapping tool
- Appendix: Technological change: from its creation to economic growth and societal welfare

During the Lecture Think of a Creativity Principle

- “Tag Line Version”
- Descriptive version
- Prescriptive version
- Text which explains the principle, how it would apply to your enterprise
- Citation if appropriate
- You can expand this creativity principle as one of your end of fall term principles!

The Responsibility of the System Architect



Transform System Architecture from an Art into a Science

- SDM Design Challenge I
- Use of abstraction vs. real life instantiation
  - The Traditional University of Chicago and Harvard School of Business Approaches
    - Normative
    - Case study
- Normative System Architecting
  - the Best System Architecture
- Stimulate theses on System Architecture

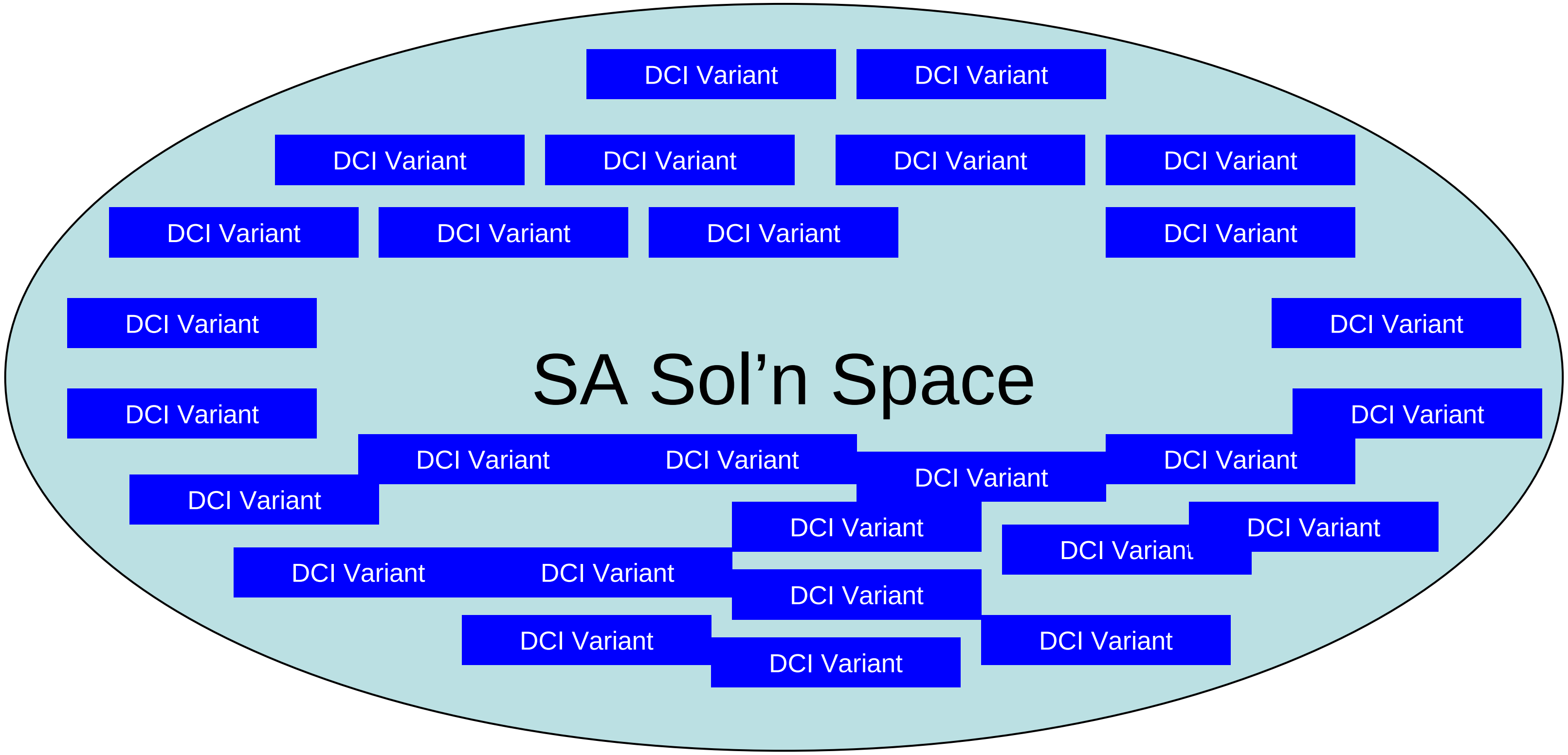
DCI Variations of System Architecture all satisfying a given specification

Photos of project vehicles removed due to copyright restrictions.

System Architecture Solution Space within the Creative Space

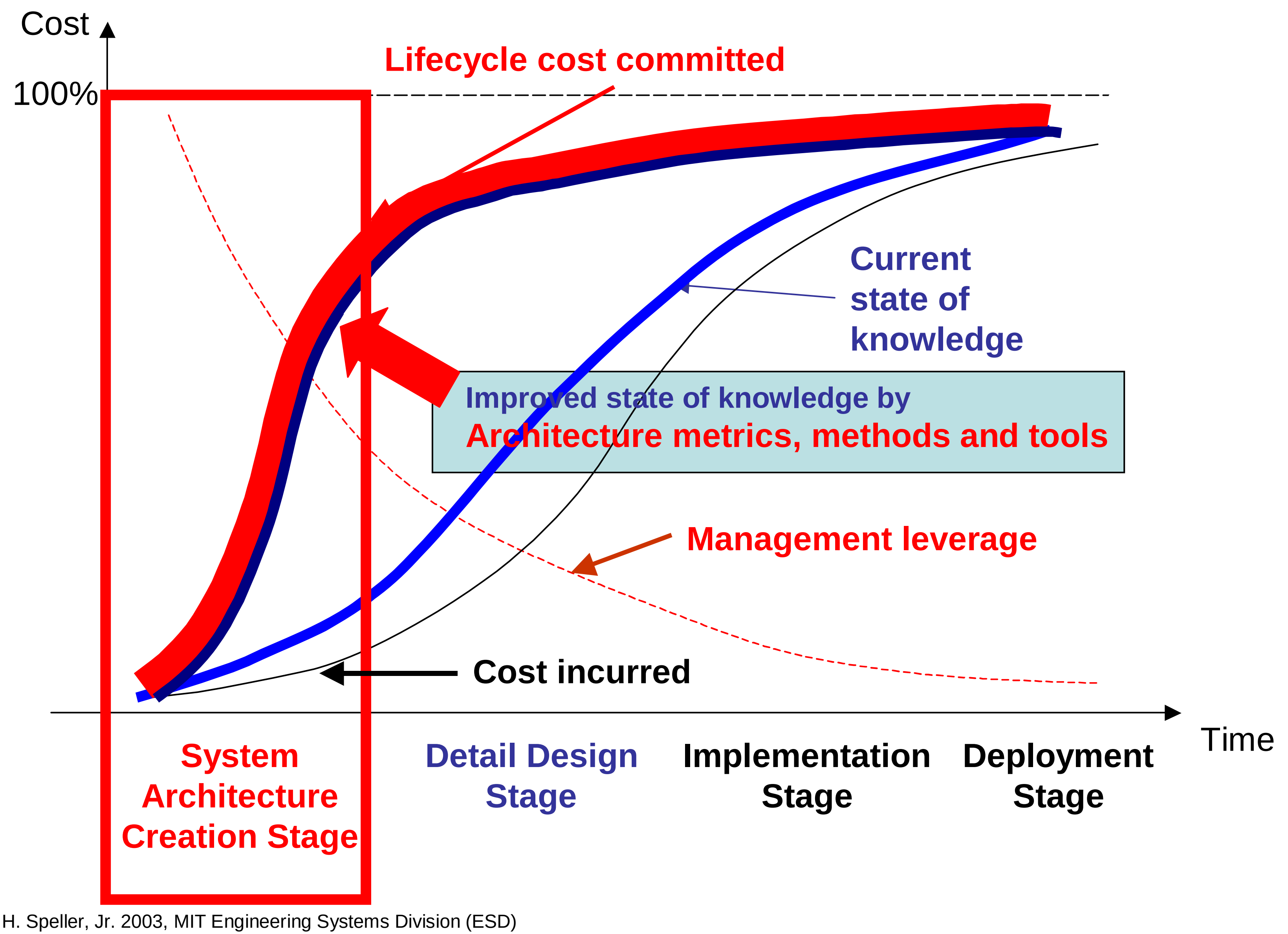
Definition

SA Sol'n Space:= SA's satisfying the given specification



The Problem Statement of System Architecting

Vital Importance of System Architecture



Accelerating rate of increase in the Stocks of Knowledge and Technology

- Worldwide Publication of Scientific Articles

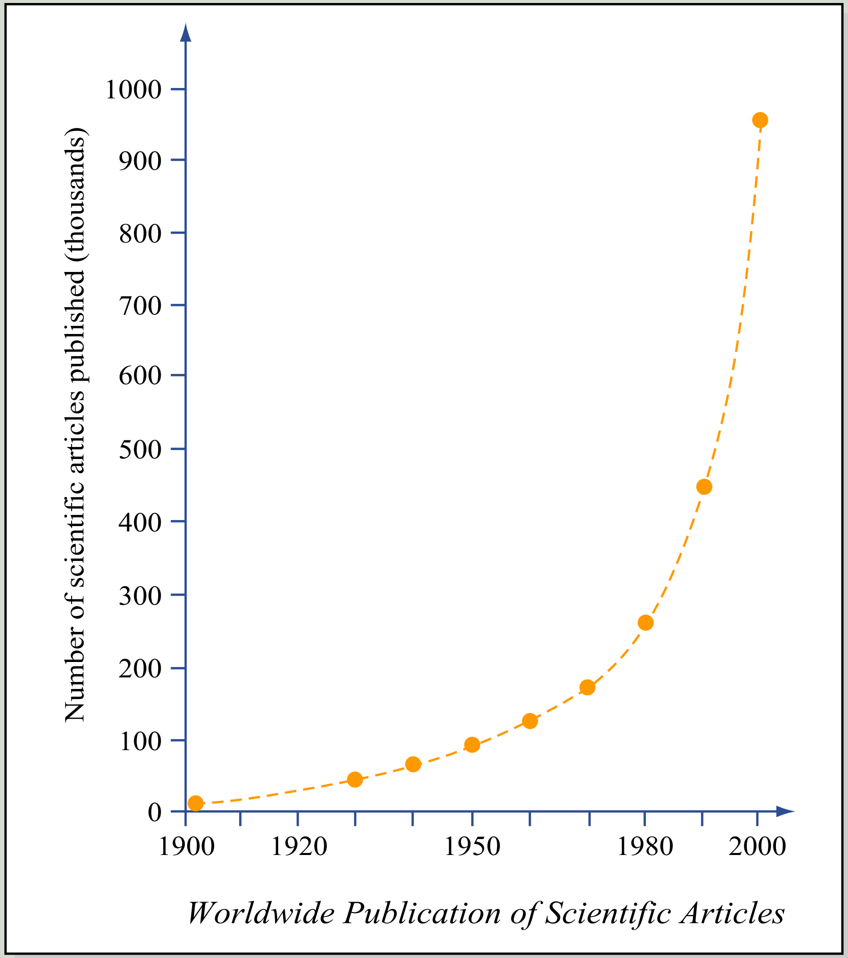


Figure by MIT OCW.

Stephen Hawking, The Universe in a Nutshell, 2001, pp. 156-168.

Accelerating rate of increase in the Stocks of Knowledge and Technology

- The Development of Complexity since the formation of the Earth

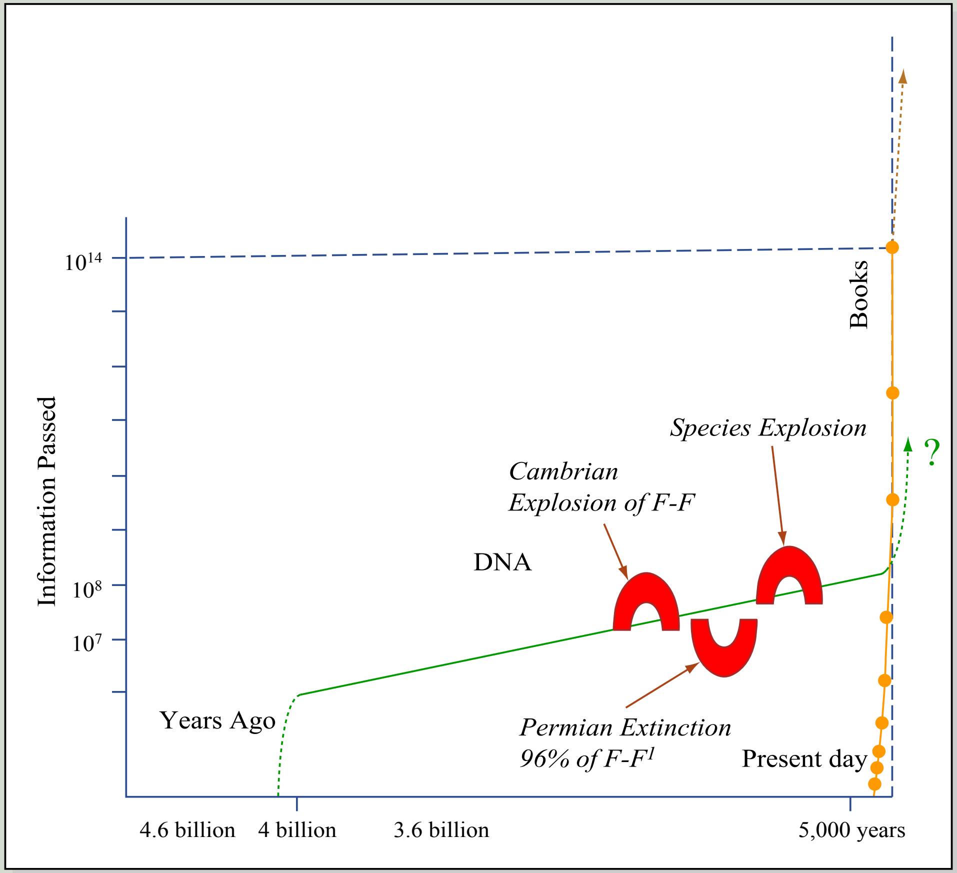


Figure by MIT OCW.

[1]S. Kauffman, The Origins of Order; Self-organization and Selection in Evolution, Oxford University Press, Inc., New York, New York, 1993, P. 76.

The Internet as a Knowledgebase Set

Figure removed due to copyright restrictions.

Current Art of System Architecture is in part limited by

- Too few concept alternatives considered
  - Limited time and budget
- Dominance of paradigms, subjective personalities, political positions and financial influencers (*The Structure of Scientific Revolutions,* Thomas Kuhn, 1970*)*
  - Individuals
  - Teams
  - Enterprises
- Insufficient interaction of concept design and selection with stakeholders to elicit their true wants
- Compulsion to <u>do</u> rather than <u>think</u>, <u>create alternatives</u>, <u>evaluate</u> and <u>rank</u> alternatives, <u>iterate</u> system architectures with stakeholders

Recommend reading <u>Skunk Works</u> by Ben Rich

- The monkey pole and bananas study (Drs. Gary Hamel and C. K. Prahalad, *Competing for the Future*, 1994)

Lack of Rigorous Algorithm to Select a Best Architecture

- Given SA alternatives, how do you compare them to determine which is better than another?
- How do you determine best trades to improve a SA?

Basic SA Creative Steps

- Problem Statement
- Functional decomposition
  - Conflict resolution
- Morphogenesis (form → function mapping)
  - Conflict resolution
- Aggregation (combining of F-F)
  - Conflict resolution
- Stakeholder selection from ranked choices and acceptance
  - Conflict resolution
- Launch SA

What is your creative process at each step?

The SA creative steps

Problem Statement → Specification of function, constraints, requirements and "ilities"

Functional Decomposition

Morphogenesis (function → form mapping)

Form

Function

Concept

Aggregation (combining of F-F)

Evaluation with respect to fitness function and "ilities"

Stakeholder SA selection from ranked choices and acceptance

Launch SA

Conflict resolution may follow each of the above steps. Iteration accompanies all these steps

What is your creative process at each step?

The System Architect's Dilemma

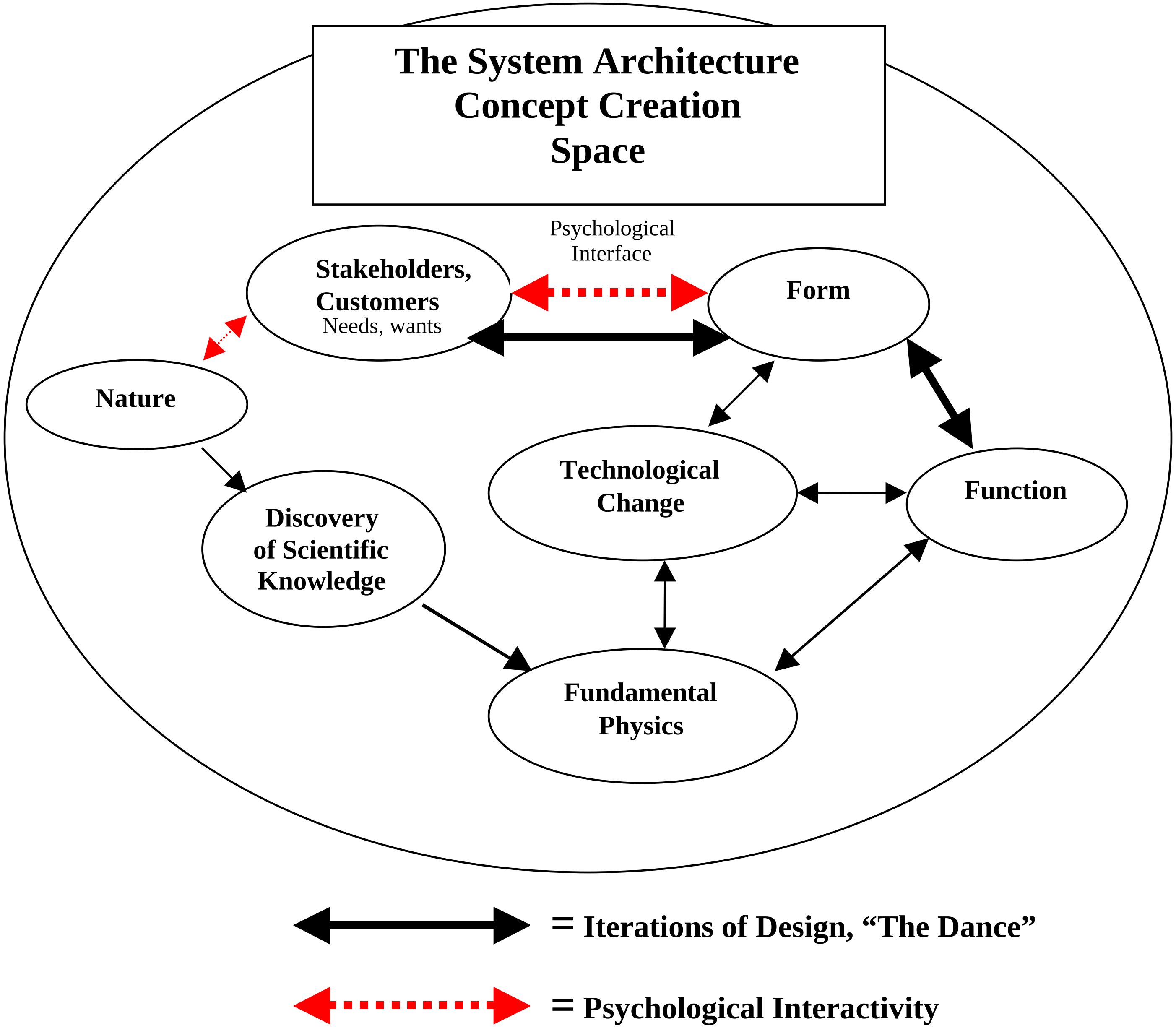
However,

- Currently impossible to consider all or even a small fraction of the possible creative and/or solution space[1]
- Designers individually and as teams tend to have paradigms
- Companies have paradigms

A result of the above factors limits the rate of new product success as well as "innovator's dilemma[2]" type threats to enterprise sustainability.

[1]H. A. Simon, Quarterly Journal of Economics 69 (1955) 99.
[2]Clayton Christensen, *The Innovator's Dilemma, 1997*

http://www.amazon.com/exec/obidos/tg/detail/-/0875845851/002-4197440-0554456?v=glance



The System Architecture
Concept Creation
Space
Psychological
Interface
Stakeholders,
Customers
Needs, wants
Form
Nature
Technological
Change
Function
Discovery
of Scientific
Knowledge
Fundamental
Physics
= Iterations of Design, "The Dance"
= Psychological Interactivity

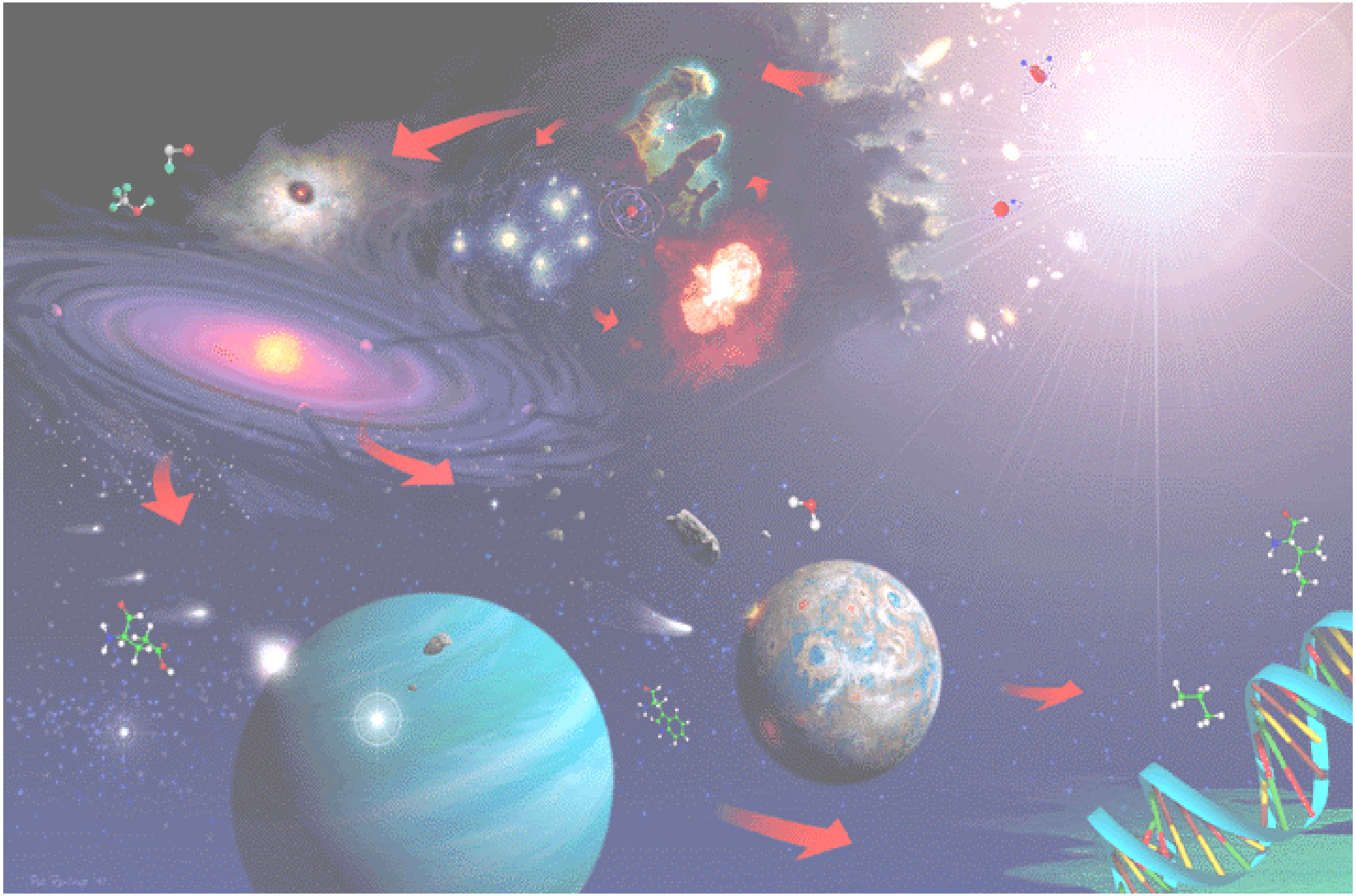
Today's Topics on Creativity

- Introduction
- Creativity
  - Nature
  - Design Rules and Combinatorics
  - Work of Vance and de Bono
- TRIZ theory
  - TRIZ, Value Engineering and the Semantic Web tool
- Radiant Thinking, Mind Mapping tool
- Appendix: Technological change: from its creation to economic growth and societal welfare

Nature's Processes as Creativity Models

- Darwinian processes
  - Evolutionary Computation
    - Genomic codes → Phenotypes
      - Genetic Algorithms (John Holland 1973),
    - genetic strategy,
    - evolutionary programming,
    - genetic programming
- Self-generative processes (my thesis research)
  - Obeys physical laws (rules)
  - Combinatorics
  - Algebraic and logical

Nature's Self-generative Creative Processes



Image; Courtesy of NASA/JPL/Caltech

The Tree of Life

Figures removed due to copyright restrictions.
Phylogeny of Archaea, Bacteria, and Eucarya.

Adapted from Baldauf (Science 2003) and Dawkins (2004)

Evolutionary Computation Approaches

characterized by as having in common:

1. a given and usually random population of points (potential solutions) in the search for a solution to the fitness function
2. direct “fitness” information instead of function derivatives
3. evolutionary processes using probabilistic rather than deterministic transition rules; starting with an initial population and the operators of selection, crossover, and mutation
4. evolution of solutions with parallel search for a solution to the fitness function
5. selection based on survival of the fittest

Evolutionary Computing:= {GA, EP, ES, GP}

Evolutionary Computing approach

1. EC consists of at least 4 sub-approaches
   - Genetic Algorithms (GA)
   - Evolutionary programming (EP)
   - Evolutionary strategies (ES)
   - Genetic programming (GP)
2. A convergence is occurring among {GA, EP, ES, GP}
3. A top-down process of adaptive behavior, such as ranking by a fitness function
4. Physics is not embedded in the EC process
5. Nature works by local information for selection and errors in ranking; the more fit may at times not reproduce and less fit can reproduce (causes diversity)

Evolutionary Computation Approaches

characterized by as having in common:

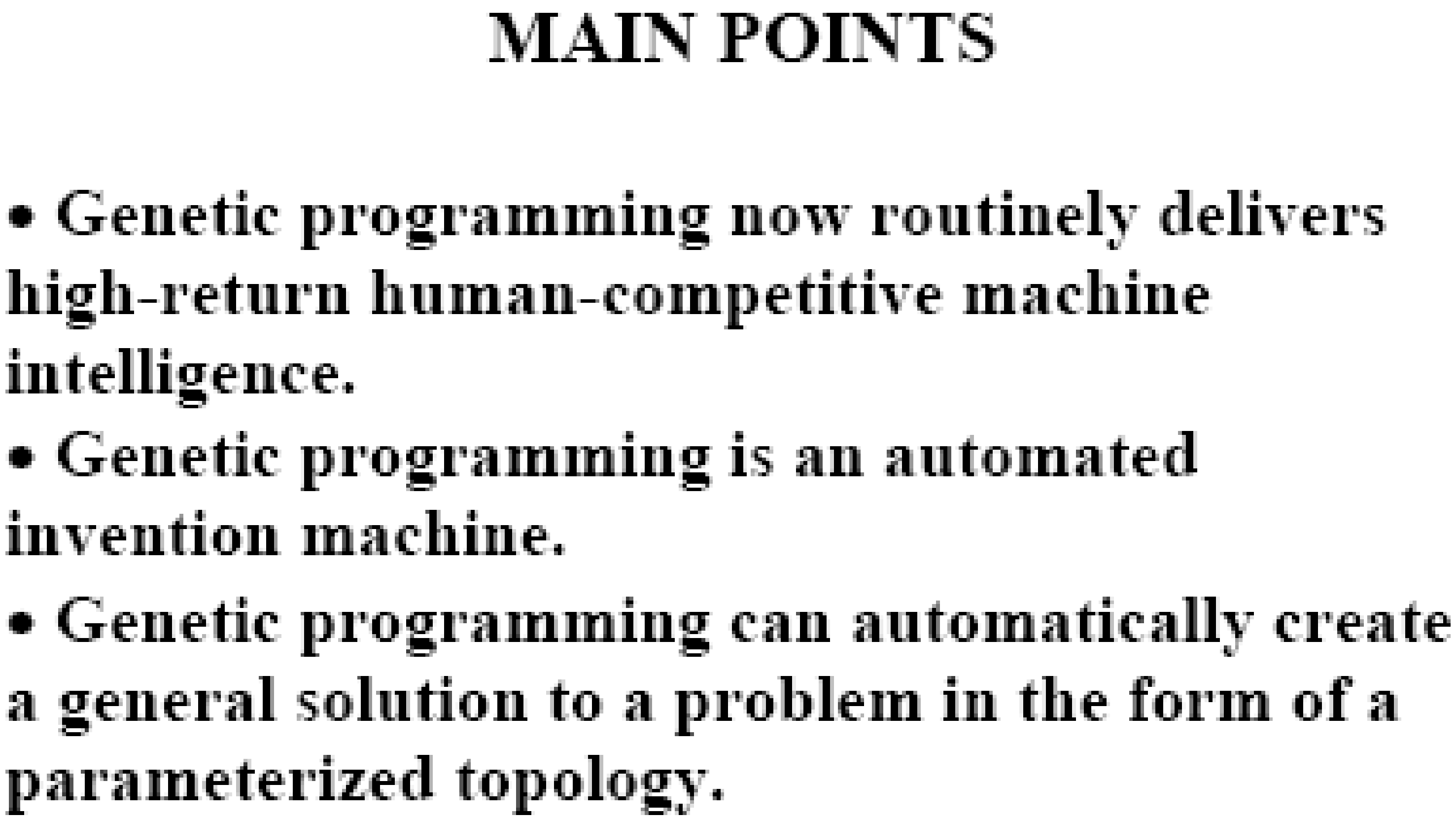
1. A population of points (potential solutions) in the search
2. Direct “fitness” information instead of function derivatives
3. Involve evolutionary processes using probabilistic rather than deterministic transition rules; use population, selection, crossover, mutation allowing a parallel search
4. Evolve solutions
5. Utilize some selection based on survival of the fittest

EC’s are a probabilistic and combinatoric set of methods that operate without regard to embedded physics, discards possibly superior fits by not exploring the full combinatoric space; are computationally bounded arbitrarily – halting is user defined.

“pure” Function Creativity

- Lambda calculus (A. Church 1941)
  - Investigation of computable functions
- Evolutionary Computation as functions applied for example to:
  - mPolymerase reading DNA (W. Fontana ~1992)
  - Genetic Programming (GP) (J. Koza ~1992)
    - Based on GA

Example of Evolutionary Computing: Genetic Programming



Courtesy of John Koza. Used with permission.

Genetic and Evolutionary Conference (2005).

GP Applications

CROSS-DOMAIN FEATURES OF RUNS OF GENETIC PROGRAMMING USED TO EVOLVE DESIGNS FOR ANALOG CIRCUITS, OPTICAL LENS SYSTEMS, CONTROLLERS, ANTENNAS, MECHANICAL SYSTEMS, AND QUANTUM COMPUTING CIRCUITS

- optical lens systems (Al-Sakran, Koza, and Jones, 2005; Koza, Al-Sakran, and Jones, 2005),
- analog electrical circuits (Koza, Bennett, Andre, and Keane 1996; Koza, Bennett, Andre, and Keane 1999),
- antennas (Lohn, Hornby, and Linden 2004; Comisky, Yu, and Koza 2000),
- controllers (Koza, Keane, Streeter, Mydlowec, Yu, and Lanza 2003; Keane, Koza, Streeter 2005),
- mechanical systems (Lipson 2004), and
- quantum computing circuits (Spector 2004)

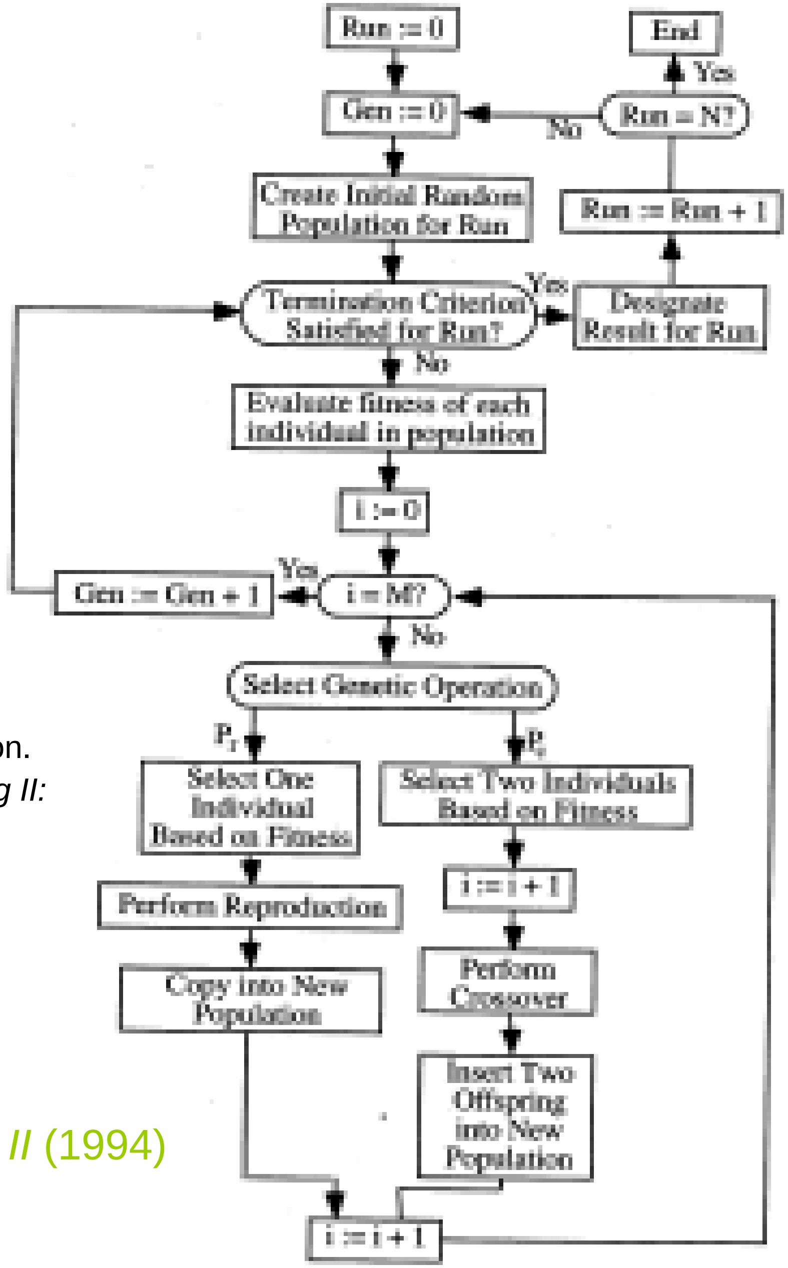
CROSS-DOMAIN FEATURES

- Native representations are sufficient when working with genetic programming
- Genetic programming breeds simulatability
- Genetic programming starts small
- Genetic programming frequently exploits a simulator's built-in assumption of reasonableness
- Genetic programming engineers around existing patents and creates novel designs more frequently than it creates infringing solutions

Courtesy of John Koza. Used with permission.

Genetic and Evolutionary Conference (2005)

Flowchart for Genetic Programming



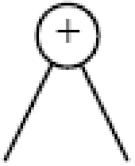
Courtesy of MIT Press. Used with permission.
Source: Koza, John R. *Genetic Programming II: Automatic Discovery of Reusable Programs.* Cambridge, MA: MIT Press, 1994.

John Koza, *Genetic Programming II* (1994)

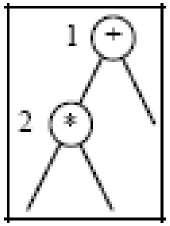
EXAMPLE OF RANDOM CREATION OF A PROGRAM TREE

- Terminal set T = {A, B, C}
- Function set F = {+, −, *, %, IFLTE}

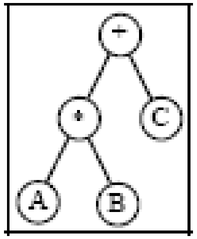
BEGIN WITH TWO-ARGUMENT +



CONTINUE WITH TWO-ARGUMENT *



FINISH WITH TERMINALS A, B, AND C

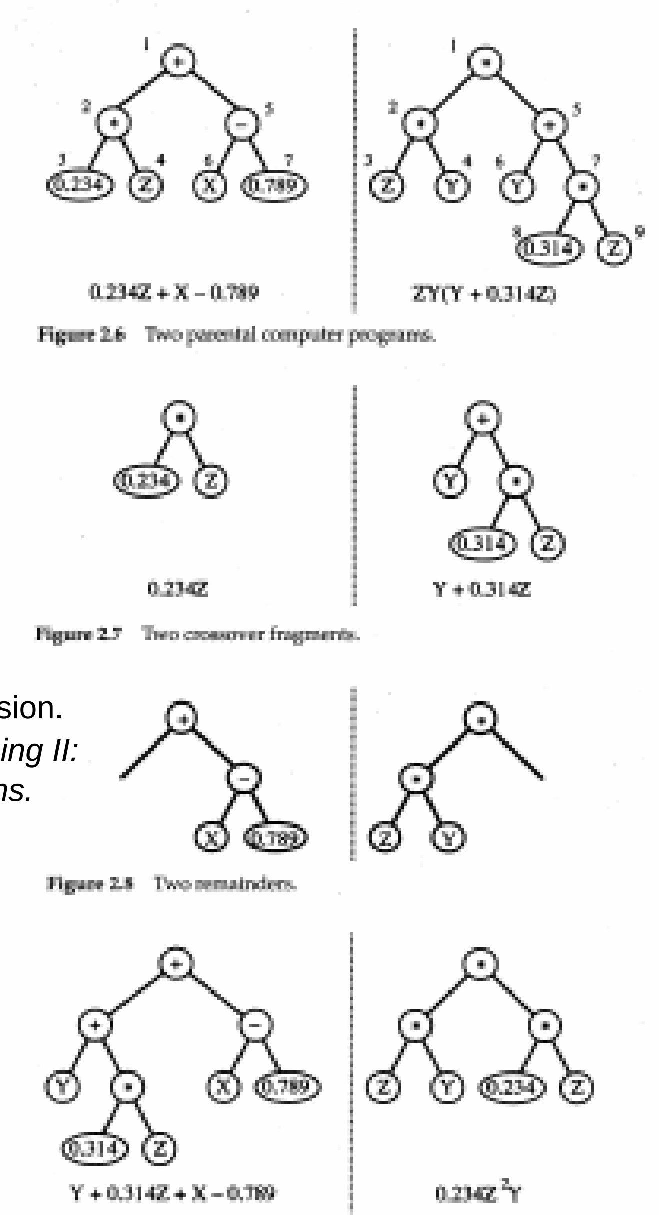


- The result is a syntactically valid executable program (provided the set of functions is closed)

Courtesy of John Koza.
Used with permission.

Genetic and Evolutionary Conference (2005)

Two Offspring Programs using Crossover



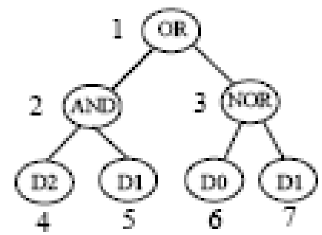
Courtesy of MIT Press. Used with permission.
Source: Koza, John R. *Genetic Programming II: Automatic Discovery of Reusable Programs.* Cambridge, MA: MIT Press, 1994.

John Koza, *Genetic Programming II* (1994)

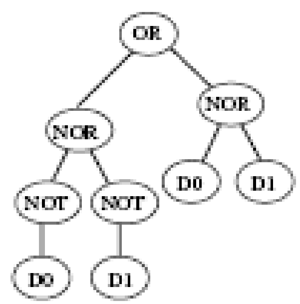
MUTATION OPERATION

- Select parent probabilistically based on fitness
- Pick point from 1 to NUMBER-OF-POINTS
- Delete subtree at the picked point
- Grow new subtree at the mutation point in same way as generated trees for initial random population (generation 0)
- The result is a syntactically valid executable program

ONE PARENTAL PROGRAM



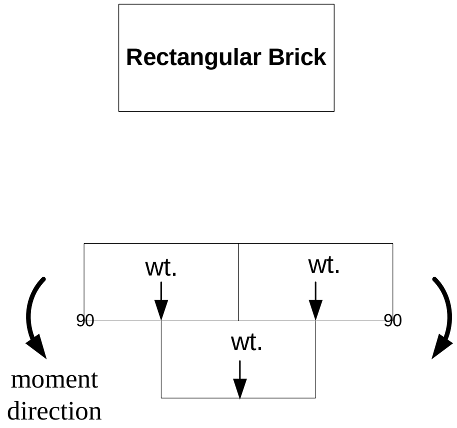
OFFSPRING PRODUCED BY MUTATION

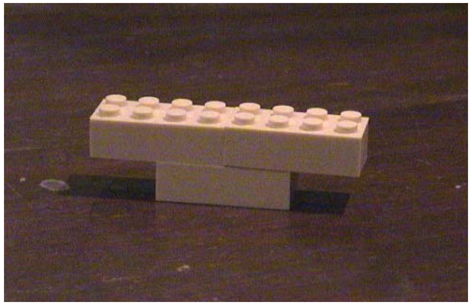


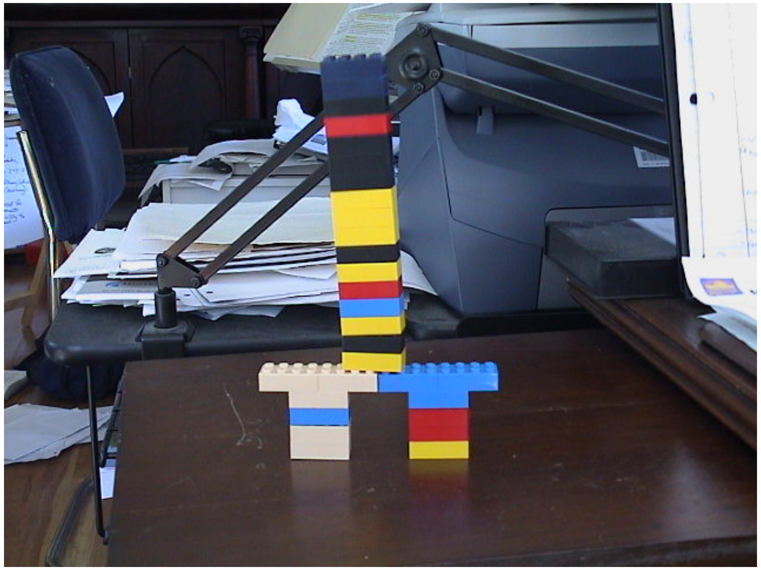
Courtesy of John Koza.
Used with permission.

Genetic and Evolutionary Conference (2005)

Assembly Sequence for the MIT T or column architecture



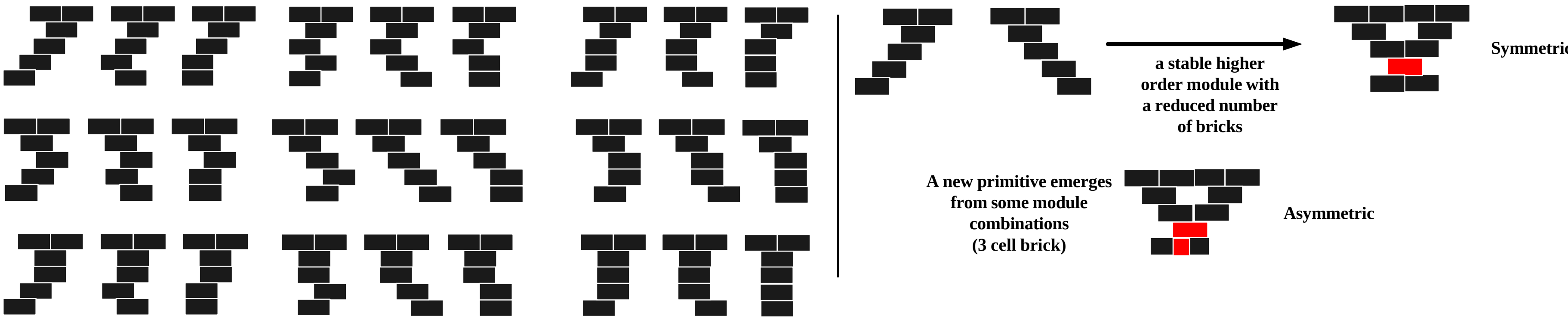




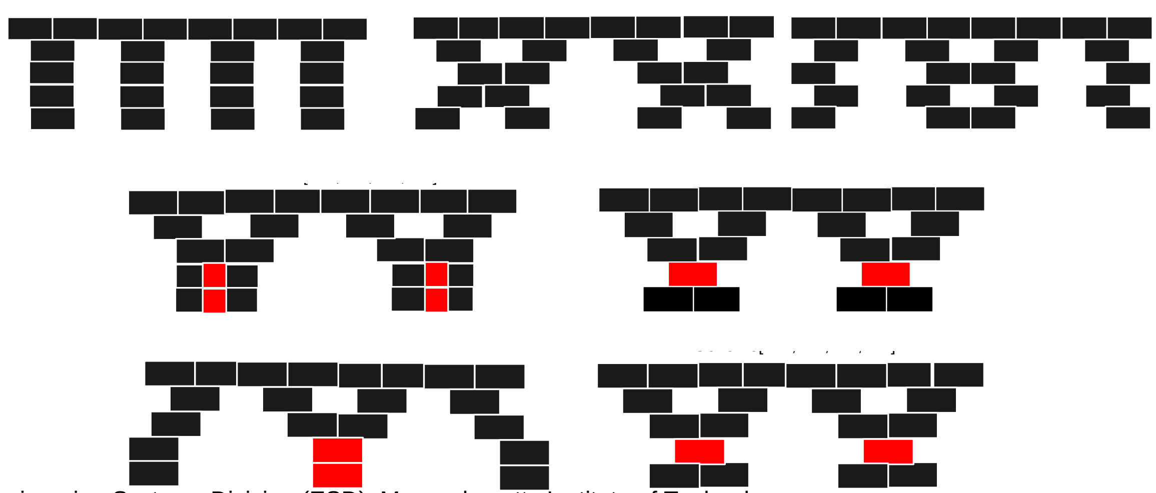
- The basic assembly module , developed from an initial primitive,
- T module derived from the physics that the least energy structure requires only 50% brick support to maintain structural equilibrium.
- The T acting as a column
- **How many ways can you sequentially assemble the T ?**

- Twenty-seven column modules → 729 Modules

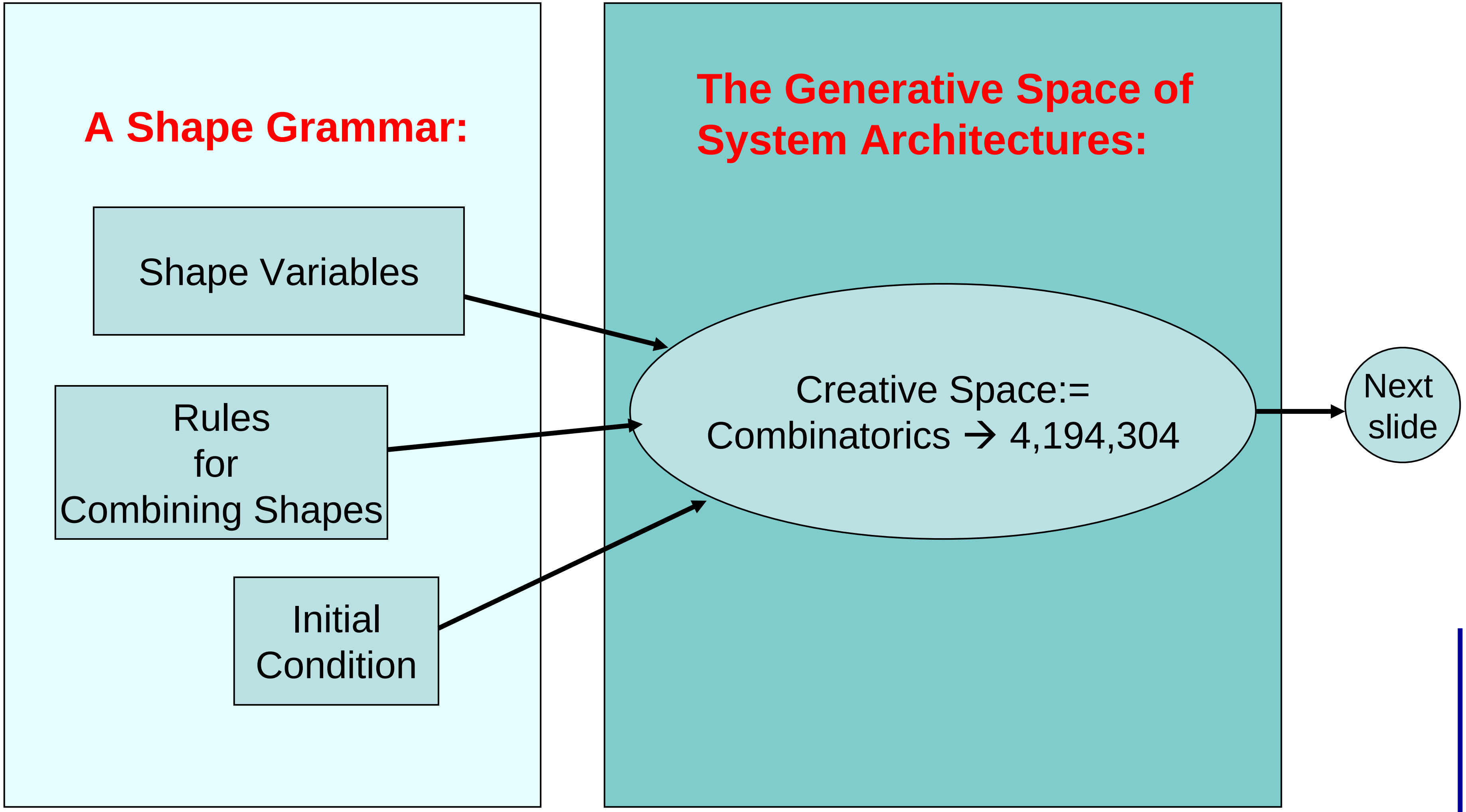
Emergence of diversity, new components, greater stability



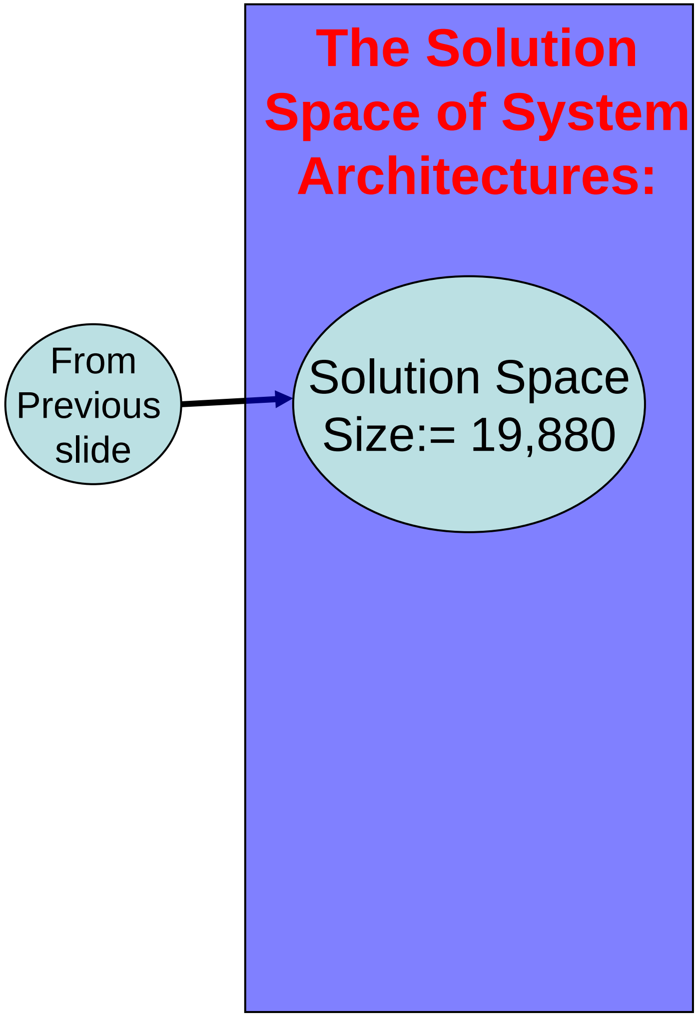
- These columns can be combined into 729 higher order modules and concatenated by repetition and reflection to create 1458 bridges



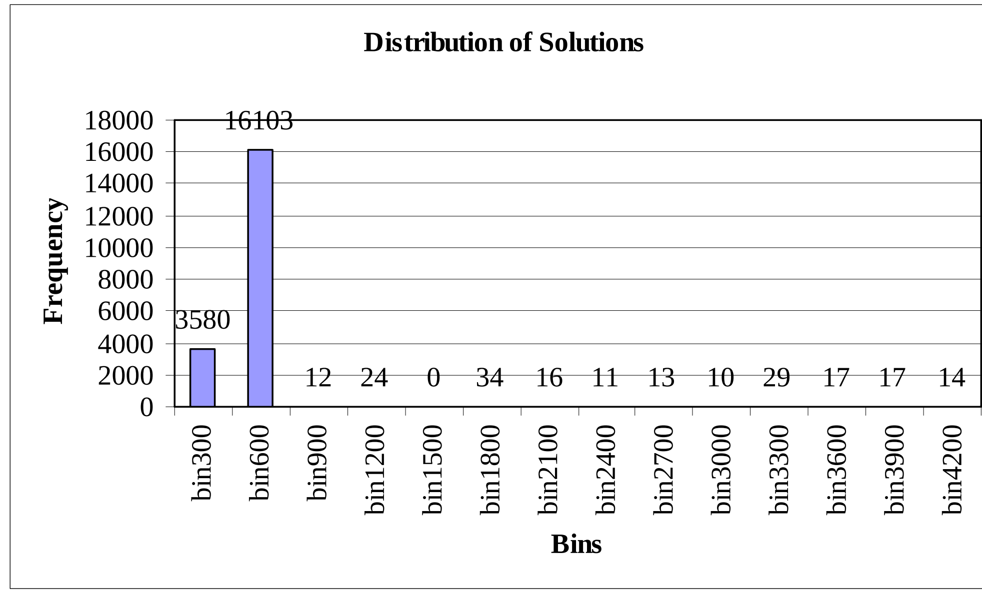
An Experiment on Circuit or Route SA Generation to Satisfy a Specification



Solution Space of System Architectures

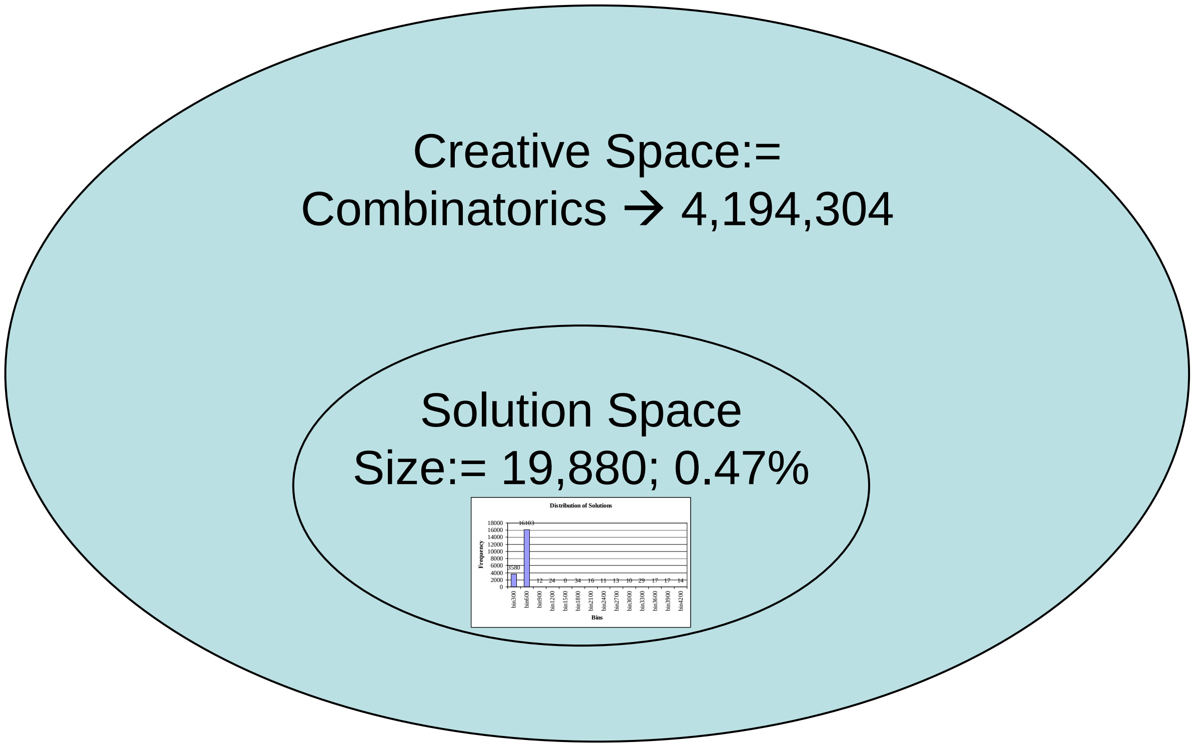


Distribution of Solutions in the Creative Space

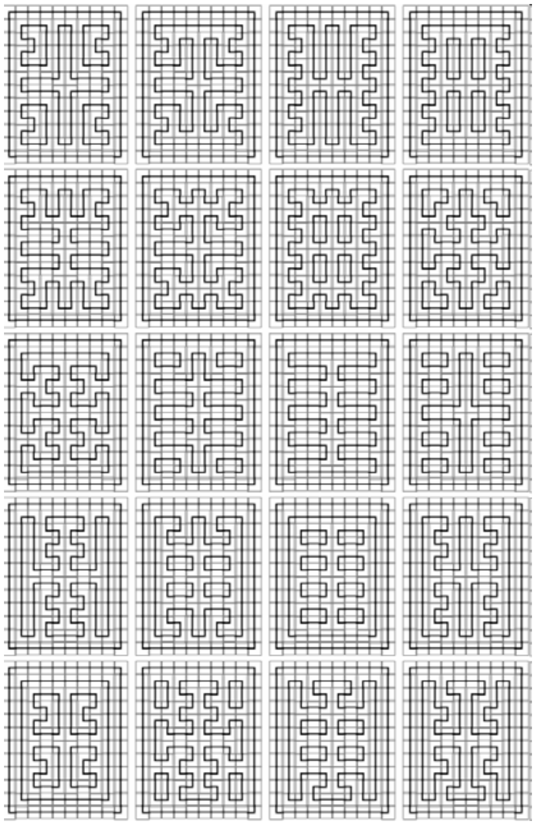


But, of these 19,880 solutions only 20 are unique;
99% of solutions are in bin300 & bin600
But, only 60% of the unique solutions are found in bin300 & bin600
The other 40% of unique solutions are nearly uniformly distributed over the space

Problem:
How do we find all the unique system architecture solutions efficiently?



The 20 Unique Solutions Satisfying the Specification



Shape Grammar-Cellular Automata Methodology[1] (SGCA)

- The given specification
  - Start in the upper right hand corner and traverse every cell once without crossing lines. Exit the lower left hand corner. Discontinuities are allowed.
- Shape Grammar
  - Shape variables:= {Primitives,Modules,Markers}
    - 6 Shape variables
    - 8 Shape markers:= {spatial,orientation,transformation}
    - 128 rules:= the formal simple relationships of Form-Function
  - Initial condition, configuration
- The generating machine is a two-dimensional cellular automaton
  - Transcribe the shapes into symbols, then
  - Compute generatively the system architectures by computing with the symbolically represented shapes
- Translate back to shape and provide graphical visual output

[1]Described in T. Speller, D. Whitney, E. Crawley, "Use of shape grammar to derive cellular automata rule patterns," Accepted for publication in *Complex Systems Journal,* 2007.

SGCA to generate and two alternatives to find the 20 unique solutions

- 20 unique solutions found by enumeration
  - Time: ~40 hours at 2.66GHz processing speed and program architecture
- 20 unique solutions found by evolutionary computation using 100 initial pop., 50% selection after fitness test, 50% cross-over, 4.5% mutation, halt at 20 unique sol'ns
  - Solutions found within 5 generations
  - Time: mean = .7 min. at 2.66GHz processing speed and program architecture
  - EC is 0.029% of the time than was necessary for enumeration
- Caveat:
  - enumeration deterministically finds all solutions (or the global, best solution(s))
  - evolutionary computation is probabilistic and there is no guarantee of finding all solutions (or the global, best solution)

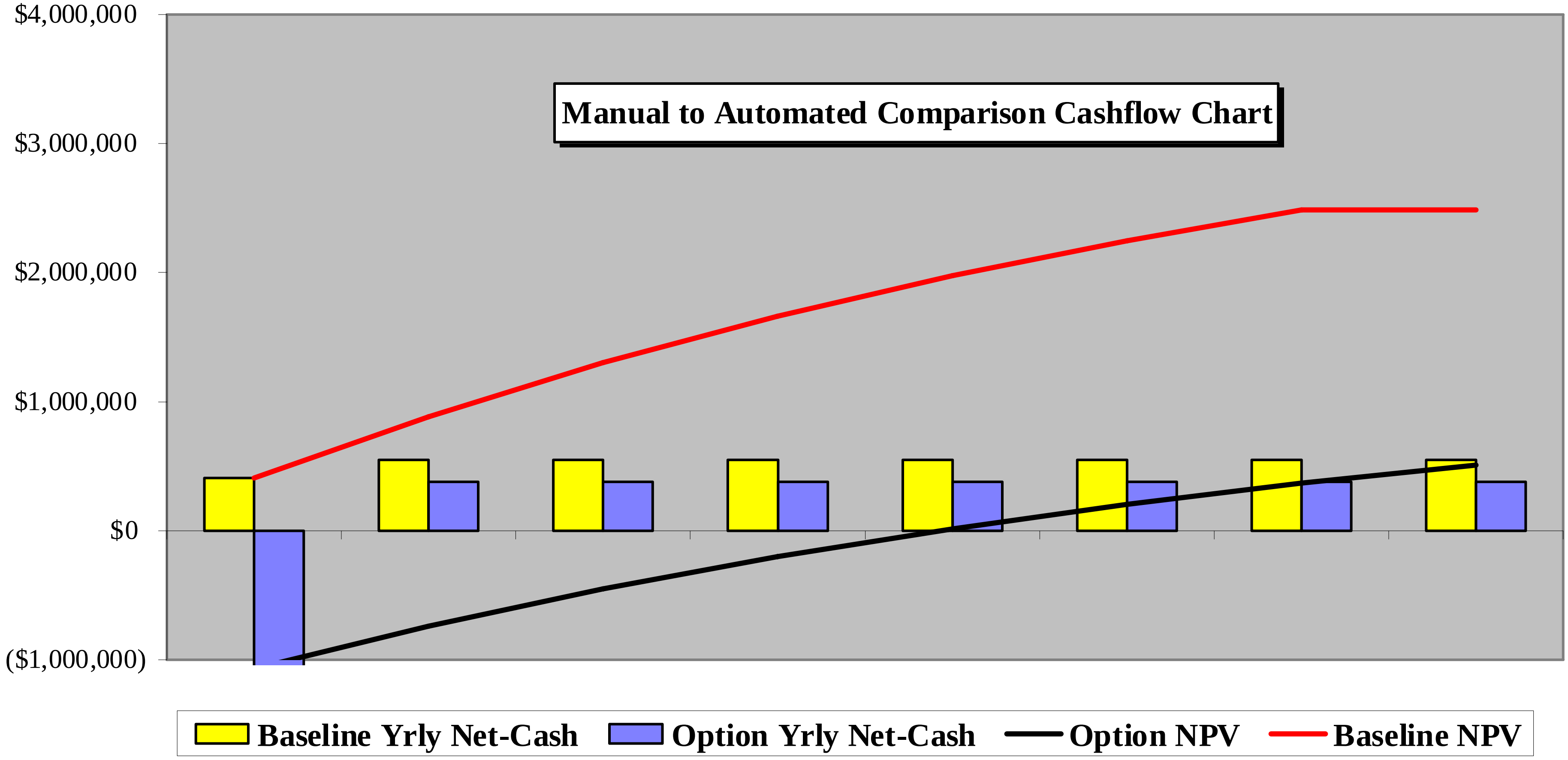
Stapler Example – Technical Description

Several slides removed due to copyright restrictions.
Stapler schematic and assembly diagrams.

Whitney, D., *Mechanical Assemblies*. 2004, New York: Oxford University Press.

Type 1 Assembly

Economic Comparison



Arising out of observations from my doctoral study is a definition of creativity

- Creativity is based on both emergence and combinatorics of form-function,

  but
- Emergence is a special type of combination that creates a phase change composition greater than the sum of the inputs
- Hence, creativity is a combinatoric process
- Upshot: the creative process can be made into a <u>formal</u> process

Today's Topics on Creativity

- Introduction
- Creativity
  - Nature
  - Design Rules and Combinatorics
  - Work of Vance and de Bono
- TRIZ theory
  - TRIZ, Value Engineering and the Semantic Web tool
- Radiant Thinking, Mind Mapping tool
- Appendix: Technological change: from its creation to economic growth and societal welfare

Neural Networks & Cognitive Processes

Images removed due to copyright restrictions.
Photo of tree branches; diagram of neural network; Mind Mapping diagram.

Creativity in Concept Development

- The Creative Urge: Human ability to overcome barriers and surpass limitations by utilizing imaginative schemes
- Most of the success of an architecture depends on the concept - a point of high leverage
- New concepts arrive from creative processes
- Creativity can be:
  - Group (Team) or individual
  - formal or informal

Adapted from mini lecture by Prof. E. Crawley

THOUGHTS

- Creativity is the making of the new, or the remaking of the old, in a new way[1]
- Inspired people create[1] Motivated people create[2]
- We're not all Mozart, but we teach people music. We're not all da Vinci, but…[2]
- European culture: analyze the present & fix; Asian culture: accept the present & improve[2]

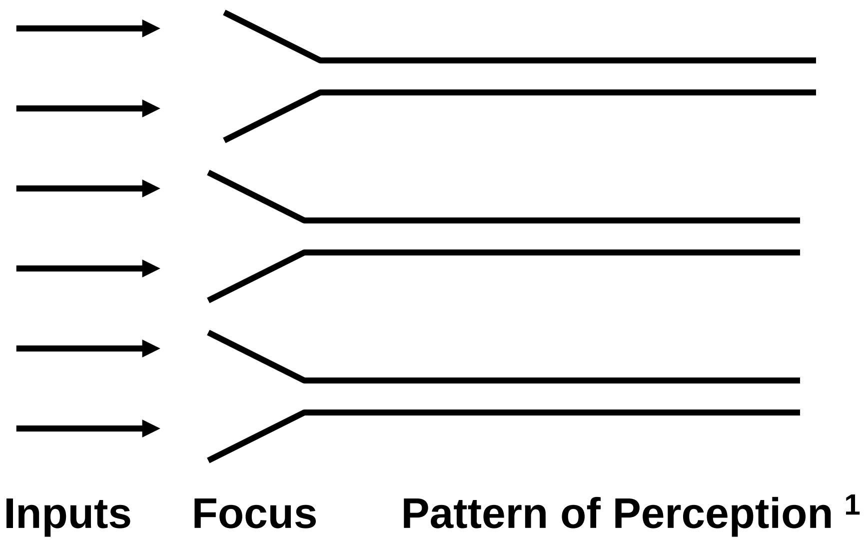
1 Think Out of the Box by Mike Vance, Diane Deacon

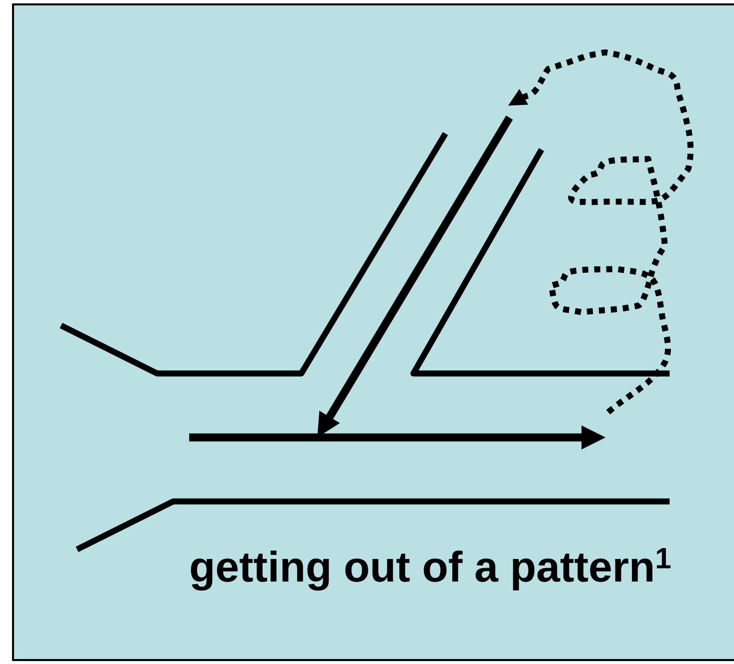
2 Serious Creativity, Lateral Thinking: Creativity Step-By-Step by Edward De Bono

Example of unstructured creativity: Serendipity

- Serendipity is defined by OED as: “The faculty of making happy and unexpected discoveries by accident. Also, the fact or an instance of such a discovery.”
- Error and trial
  - Mutation in nature
- Mistakes that are recognized as a solution, ex.
  - Telephone
  - Microwave
  - Post-it

HOW THE MIND WORKS





"Unless we purposely connect unconnected ideas, we may never reveal valuable relationships." Vance

[1] de Bono

APPROACHES TO CREATIVITY

- Group Dynamics

  pow-wows[1], Structured brainstorming, mind-boggling[1], workouts[1], six hats[2]

- Structured Processes

  TRIZ, category note taking[1], mind mapping

- Personal / Group Stimulants

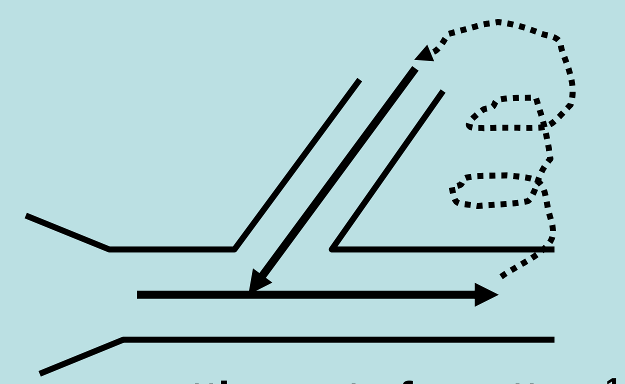
  5 senses[1], challenges[2], provocations[2], random inputs[2], "motion"[2]

- Models

  Edison, F.L. Wright, Fuller, Welch ...

  any Nobel Prize winner

1 Vance 2 de Bono



getting out of a pattern[1]

SIX (Thinking/roles) HATS

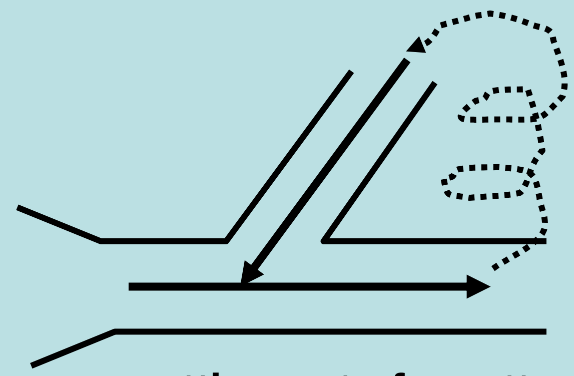
De Bono

- White Hat - examine information, neutral
- Red Hat - feelings, intuition, hunches, emotions
- Black Hat - “caution,” critical judgments, avoid mistakes
- Yellow Hat - optimism, feasibility, benefit
- Green Hat - creative thinking, new ideas, hypothesis
- Blue Hat - process control; agenda, next step, commentary of thinking

PAUSE, FOCUS, CHALLENGE de Bono

- Just stop & pause - what could be?
- Focus on improvement, task, opportunity, re-phrase the focus
- Challenge --
  - Continuity
  - Validity of concept
  - Dominating concept
  - Assumptions
  - Boundaries
  - "Essential" factors
  - "Avoidance" factors
  - Either/Or propositions

Why is it done this way?
Why does it have to be done this way?
Are there other ways of doing it?

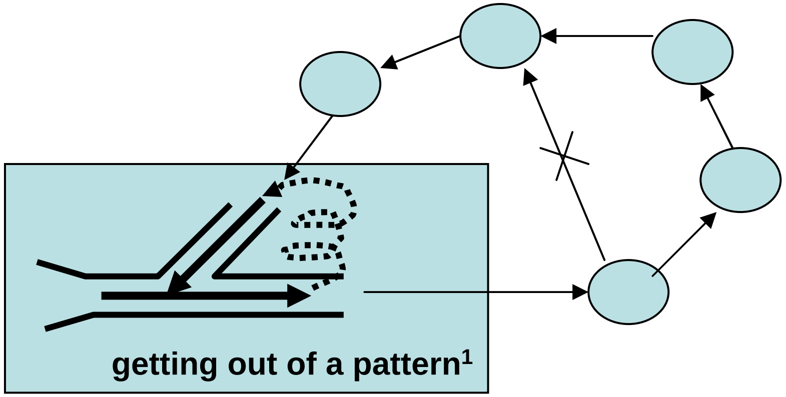


getting out of a pattern[1]

MOVEMENT

- Movement is the opposite of judgment when presented an unworkable idea - learn from it & move on – exploration without judgment

  (more active than "suspended judgment")
- Must have a general willingness to "move," plus techniques
  - Extract a principle
  - Focus on the difference
  - Moment to moment
  - Positive aspects
  - Circumstances with direct value
- Don't look back - keep trying to move ahead



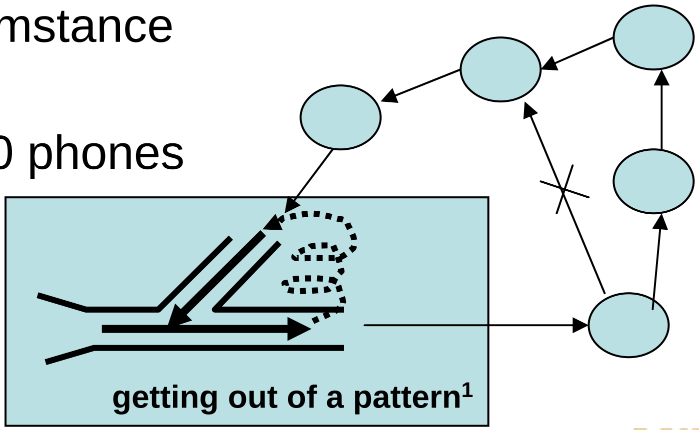
MOVEMENT STIMULATED BY RANDOM INPUT & PROVOCATION

- Random input - pick a word at random -- let the brain connect to idea at hand
- PO -- Provocative Operation (actually say PO)
  - Accidental provocation
  - Deliberate
    - Negate things we take for granted
      - Driver drives car PO – Car drives itself → Automated highway
      - Light varies in velocity PO – Velocity of light is constant → Relativity and → the Energy and Matter are one and the same, $E=MC^2$
      - Time is constant PO - Space-time is curved → General Relativity
      - This statement is false PO – This statement is true → Godel Incompleteness Theorem
    - Reversal of SUBJECT/OBJECT or circumstance

      I have diet -- PO diet has me
    - Exaggeration -- PO every house has 100 phones
    - Distortion of relationships or time series

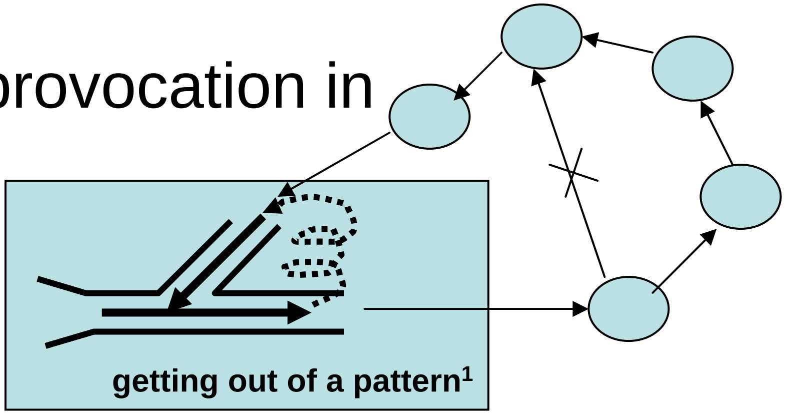
      PO -- students examine each other
    - Wishful thinking

      PO -- shoplifters identify themselves



MOVEMENT TECHNIQUES

- Extract a principle: look for new media
  PO -- bring back town crier
  → can't turn town crier off
- Focus of the difference: postage stamps
  PO -- make long and thin
  → more room for address
- Moment to moment -- imagine provocation in effect
  PO -- planes land upside down
  → pilots have better view



- Positive aspects
  PO -- cars should have their engines on their roof
  → better ease of access
- In what circumstances is there direct value?
  PO -- drinking glasses have round bottoms
  → bars sell more drinks

Scenarios as a creative practice

Scenarios are another creative practice increasingly being used by companies and gaining favor from the success of Shell Oil with their future scenarios process (see references below). We intend our system architectures to not just satisfy current requirements but also be easily evolvable over time. One way of working towards evolvability is by designing a product family. The family can be developed and produced all at one time, or to mitigate risk may be started with one product entry followed by variants of the product over time. This controlled release of new family members manages risk, can spread out resource requirements and reduces the need for a high upfront capital investment. Therefore, another 'ideal' design attribute is the system architecture's ability to evolve. How do we plan for the evolution of a product? One strategy is to make the current first release version of the SA so flexible that it can be evolved quickly and at minimal cost. However, typically in order to have notions of how to design in the flexibility we must know the future. Scenarios, which derives its name from scriptwriting in plays, is one means of thinking through possible futures. Your creative team can think of many different possible futures affecting your enterprise and its products. How do you know that you have explored all possible futures? You don't know, but large fluctuations in the "environment," or disturbances, must be considered, not just the different normal, possible paths out into the future. This is a 'what if' analysis. One must consider the seemingly improbable disturbances that can, and as we have seen in 9/11 and the recent tsunami, do occur. The scenarios can be done by experts, managers, futurists, your SA team, yourself and … . Think about how your product may be misused or used for purposes not originally intended. In this case you should consider testing it for robustness in this outside usage or 'protecting' the product from being misused. Once these scenarios are developed and discussed, their impact must be discounted to the present in your SA design as much as you judge is appropriate. In this viewpoint one might consider the selected present SA to be the result of all plausibly considered futures. By modularity and platforming along with other SA strategies, you can achieve flexibility and its close relative, extensibility, with minimal or modest added cost to your present SA design.

A good book by which to start being convinced of the usefulness of scenarios is http://www.amazon.com/exec/obidos/tg/detail/-/1578518202/qid=1105579205/sr=1-32/ref=sr_1_32/002-5285815-5183236?v=glance&s=books *The Living Company* by Arie de Geus.

Also, here are two websites to review on scenarios

http://www.shell.com/home/content/aboutshell-en/our_strategy/shell_global_scenarios/dir_global_scenarios_07112006.html (Accessed 21 August 2007). and http://www.sric-bi.com/consulting/ScenarioPlan.shtml .

Today's Topics on Creativity

- Introduction
- Creativity
  - Nature
  - Design Rules and Combinatorics
  - Work of Vance and de Bono
- TRIZ theory
  - TRIZ, Value Engineering and the Semantic Web tool
- Radiant Thinking, Mind Mapping tool
- Appendix: Technological change: from its creation to economic growth and societal welfare

TRIZ: A Theory of Inventive Problem Solving

Many thanks to Dan Frey, Don Clausing and Victor Fey for materials and advice

See http://www.triz-journal.com/

Axiomatic Design, Prof. Nam Suh, MIT

- Independence of Functions
- Minimum information

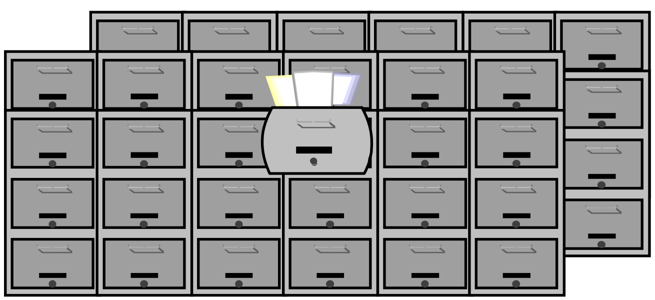
- A recommended course and reading:
  http://www.amazon.com/gp/product/0195134664/ref=nosim/002-5285815-5183236?camp=2025&dev-t=D26XECQVNV6NDQ&link%5Fcode=xm2&n=283155

Other Creativity JAM's

- Example: Pugh Controlled Convergence methodology
  - "Attack the Negatives" to form better, more robust hybrid concepts
- Another creativity enhancing tool
  - Osborn-Parnes Creative Problem Solving (CPS)

The Basic Concepts of TRIZ

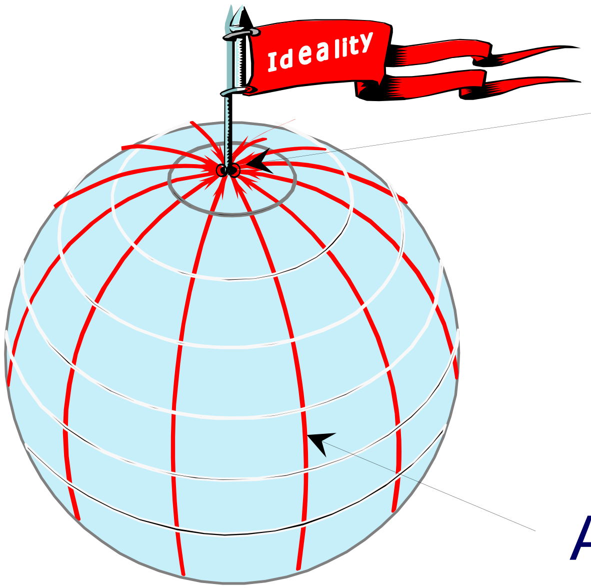
- Invention / innovation can be made more productive through a "scientific" approach (based on study of patents)
- Inventions evolve toward ideality
- Evolution is accomplished by resolution of technical conflicts



>200,000 Patents → ~1,250 typical system conflicts → ~40 techniques for overcoming system conflicts

Degree of Ideality

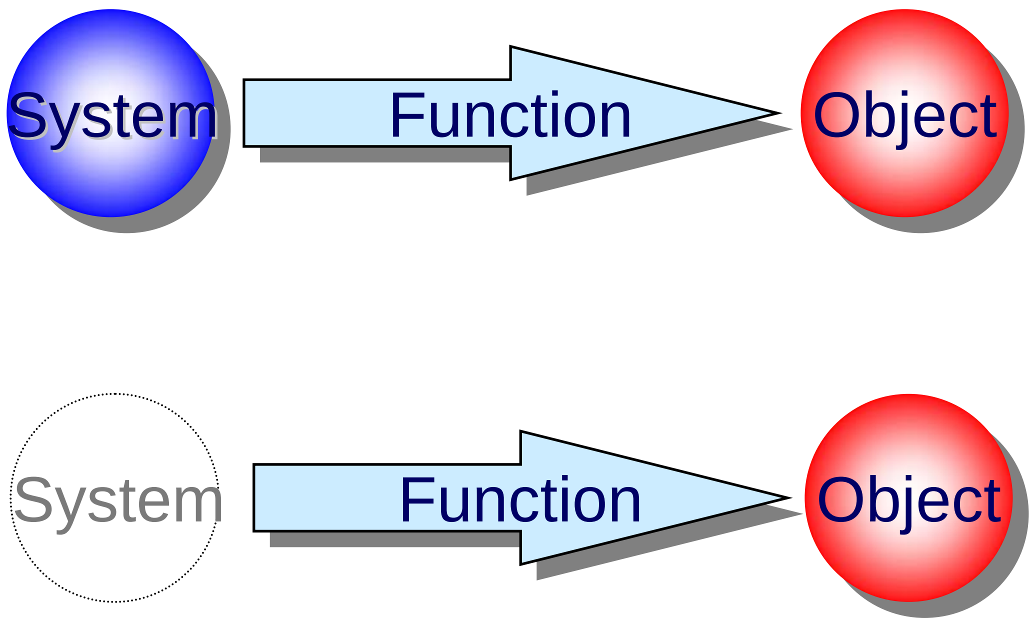
- Degree of ideality = $\dfrac{\text{Functionality}}{\text{Cost or Side Effects}}$



Ideality Pole

All inventions move toward ideality

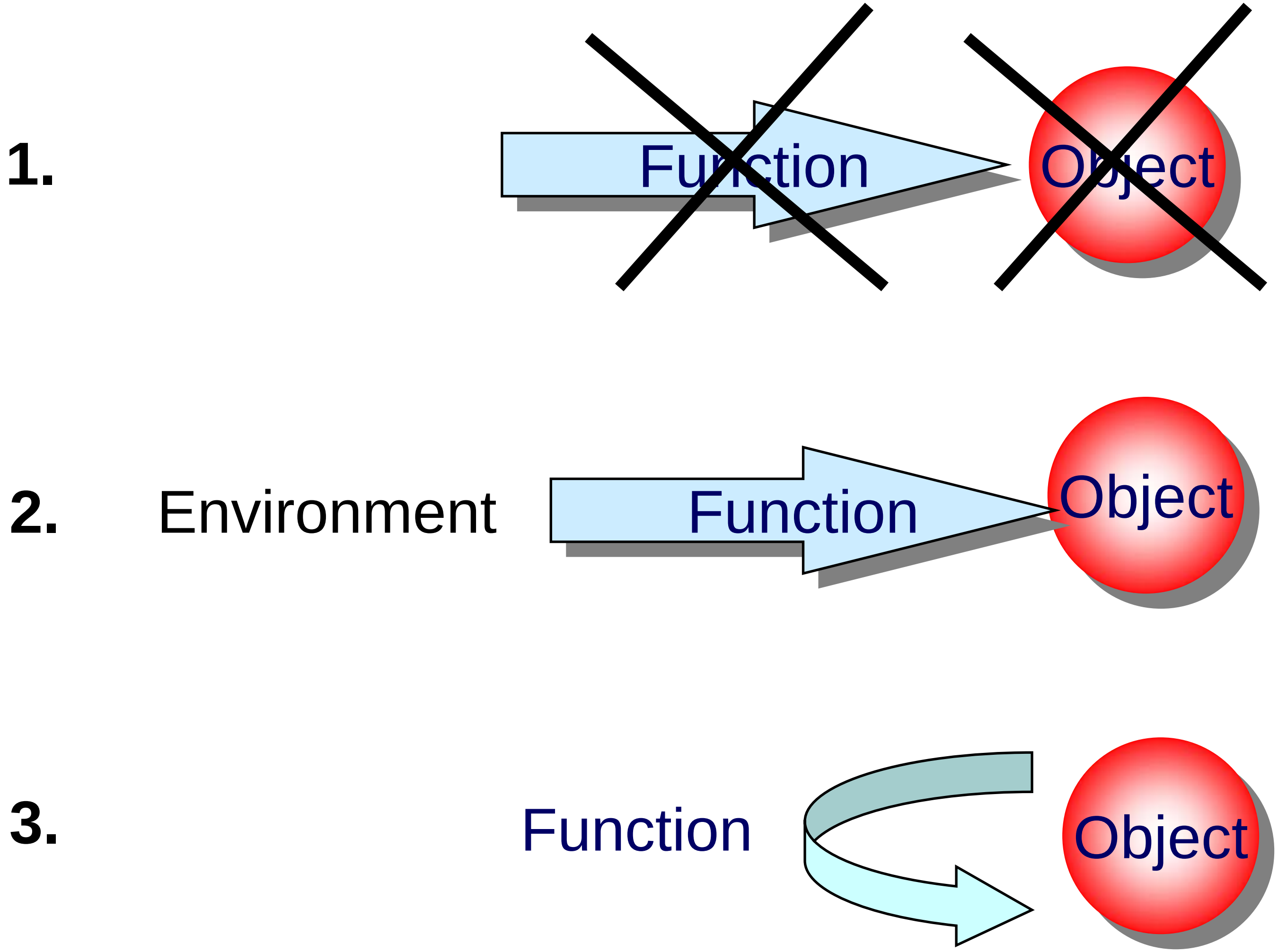
Ideal Technological System



An ideal system does not exist as a physical entity, but its function is fully performed

http://www.trizgroup.com/

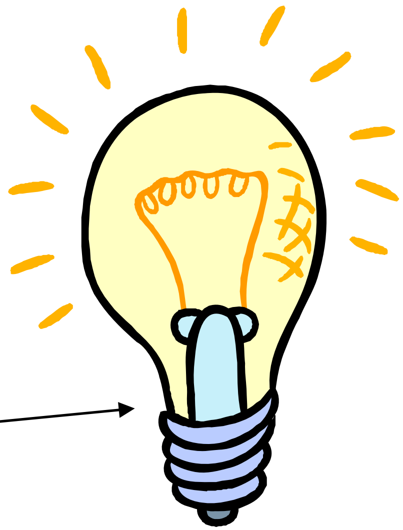
Ways to Achieve The Ideal



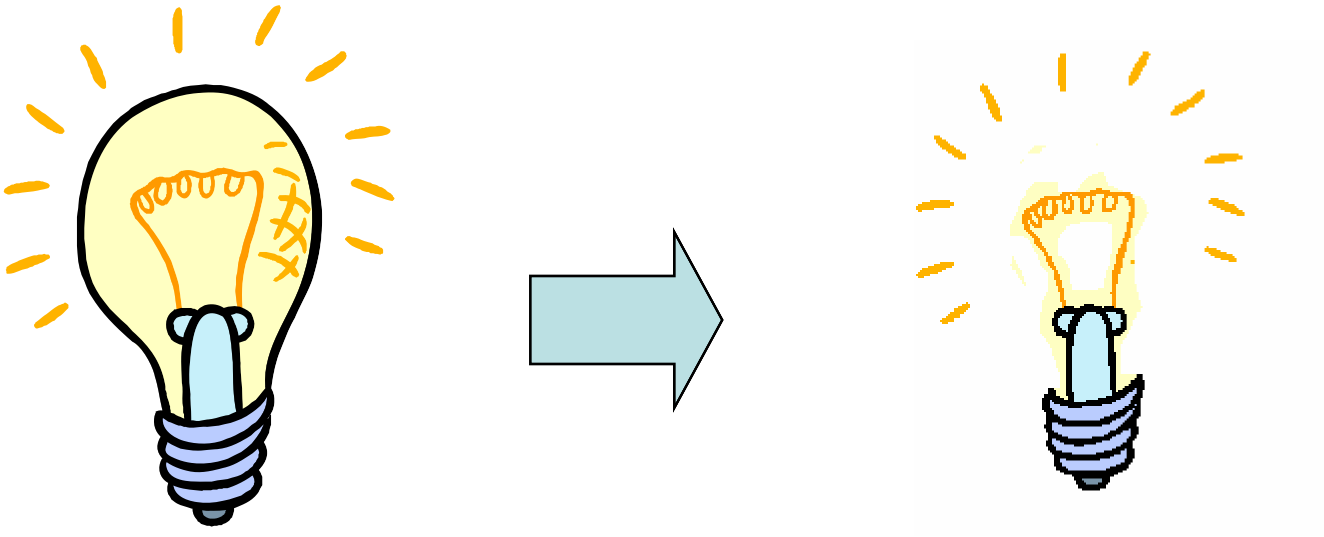
Ideality Example

The Russians launched an unmanned Lunar Probe to the moon's surface with the intention to transmit TV pictures to the Earth. A projector using a light bulb was designed to illuminate the lunar surface ahead of the vehicle. However, existing light bulbs would not survive the impact of landing on the Moon surface.

The most durable bulbs were ones used in tanks, but even those bulbs would crack at the joint between the glass and the screw base during tests.

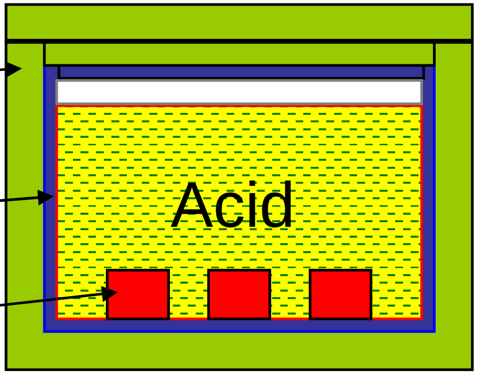


Solution

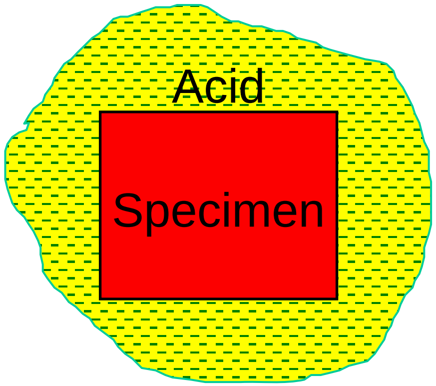


Ideality Example

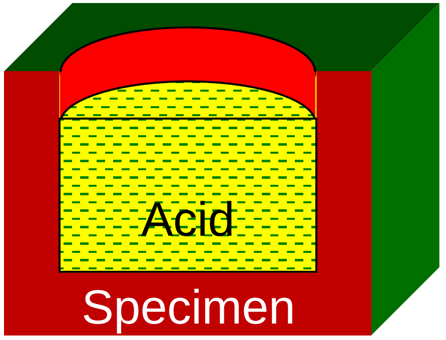
To study the effects of acids on metal alloys, specimens are placed into a hermetically sealed chamber filled with acid. The acid reacts not only with the specimen but also the walls, which necessitates a glass-coating to protect the walls. The glass coating cracks and has to be reapplied repeatedly for some tests.



Solution



Transition to an Ideal Solution: Chamber is absent



Ideal Solution: Specimen-Chamber

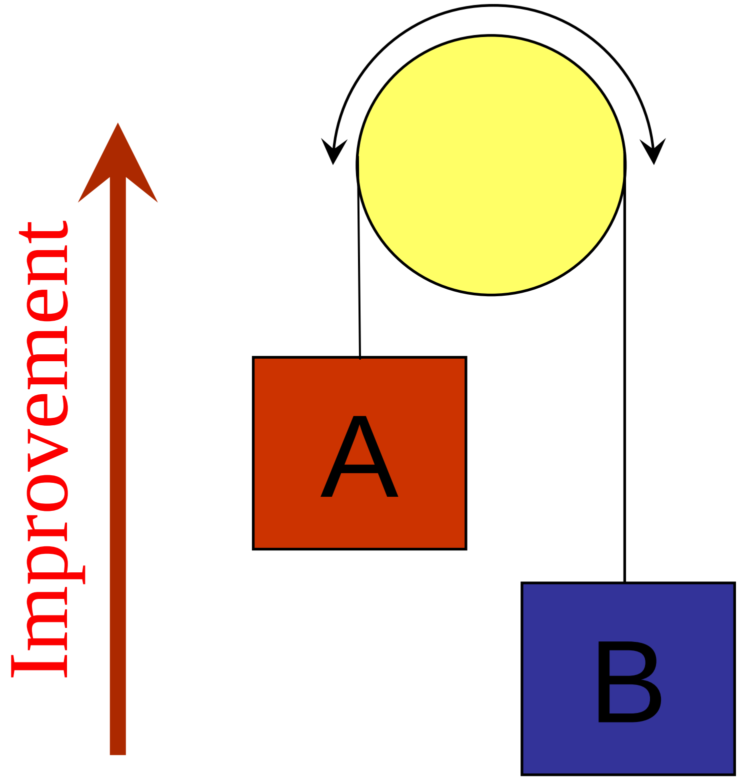
Ideality: Practice

- A delivery company is shipping meat by aircraft, but the refrigeration system is heavy and is reducing the useful payload.

- A Swiss company manufactures confections. Their process for cracking the shells occasionally breaks the nut meats inside the shell and they would prefer to keep the nut meat intact.

- Household TV reception drives a design of large antennas, but homeowners don't like the appearance.

System Conflict

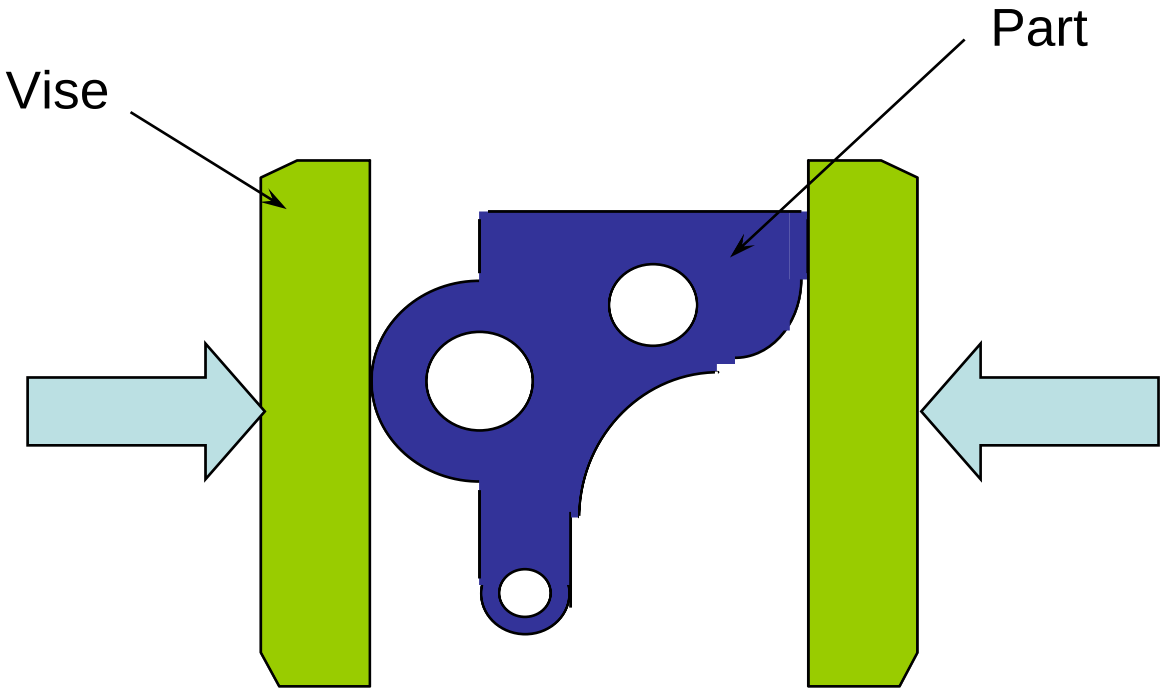


A situation when useful changes in one part of a system cause deterioration of another part is called a SYSTEM CONFLICT

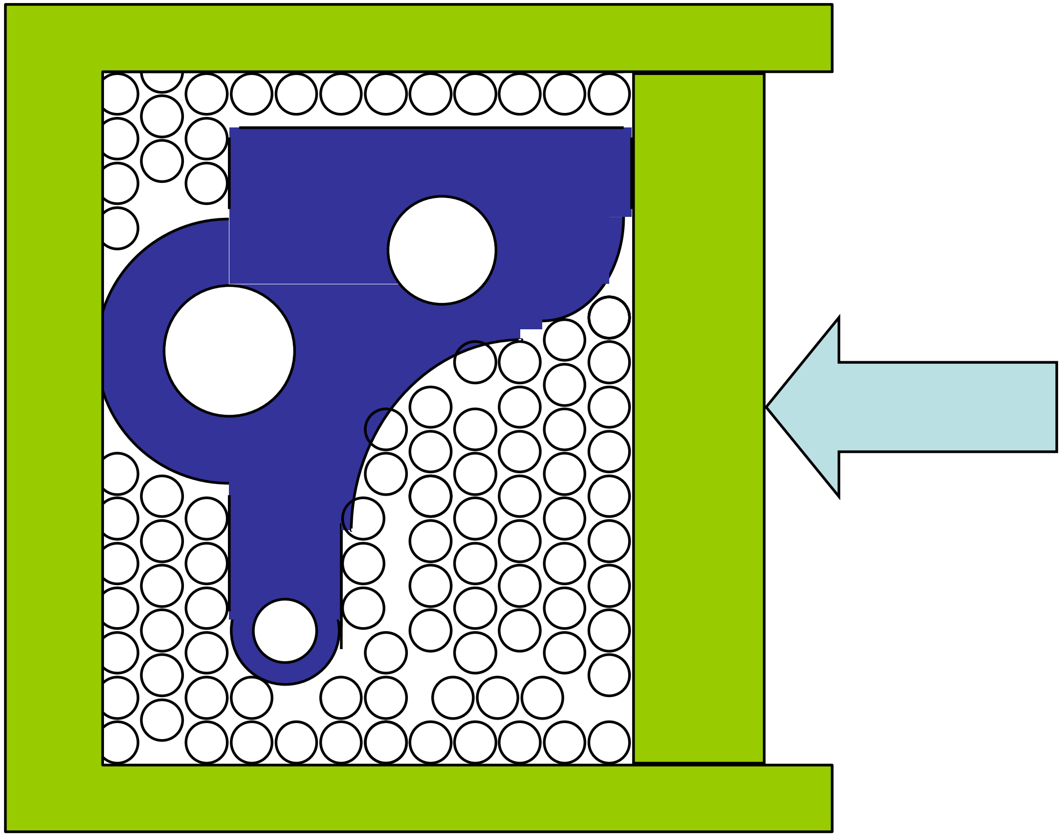
Basic Ways to Resolve Physical Contradictions

1. Separate the requirements in time
2. Separate the requirements in space
3. Separate requirements between system and its parts

Example Problem



Solution



Example Problem

- Armor plate has two requirements
  - Be hard, to shatter projectile
  - Be tough, so armor is not shattered
- Physical conflict
  - Hard and tough are in conflict
  - Hard steel is brittle, and tough steel is not hard enough to defeat projectile.

Solution

- Make the front half of the armor plate hard, and the back half tough
  - US Patent 3,475,812
  - Huge improvement in ballistics
- Basic invention principle: separate the requirements in space

- Current project in MIT's soldier nanotechnologies
  - Nanomaterial that is normally flexible unless it is impacted at which time it in microseconds changes state to become hard and stiff
- Basic invention principle: separate the requirements in time

40 Inventive Principles

1. Segmentation – divide an object into parts
2. Extraction – extract only the needed property
3. Local quality – transition from homogeneous to heterogeneous structure

…

7. Nesting
13. Inversion [same as provocation]
18. Mechanical vibration
29. Pneumatic construction

…

40. Composite materials

Table of Contradictions

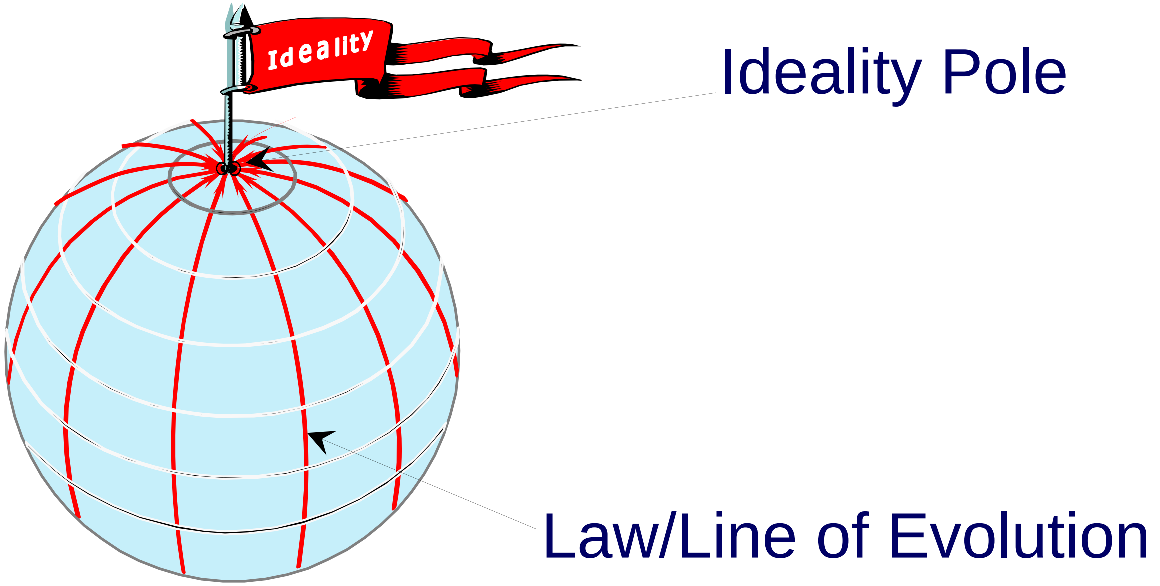
Image removed due to copyright restrictions.

Laws of Technological System Evolution

A Law of Technological System Evolution represents significant, stable, and repeatable interactions between elements of the systems, and between the systems and their environment in the process of their historical development.

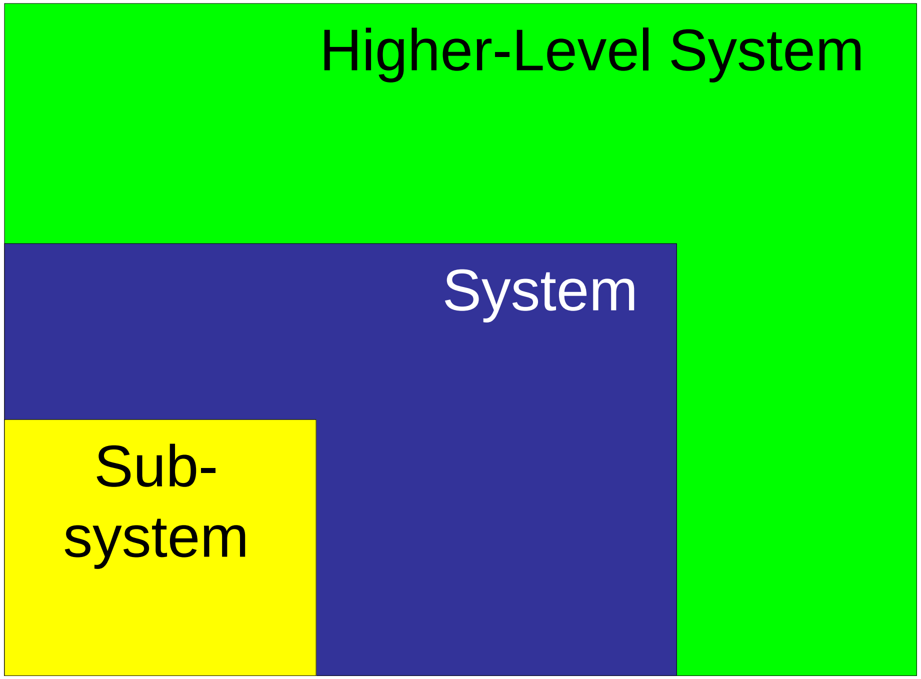
Primary Law of Evolution

- The Law of Increasing Degree of Ideality is the central law of evolution of technology.
- Other Laws of Evolution are mechanisms for increasing the Degree of Ideality.



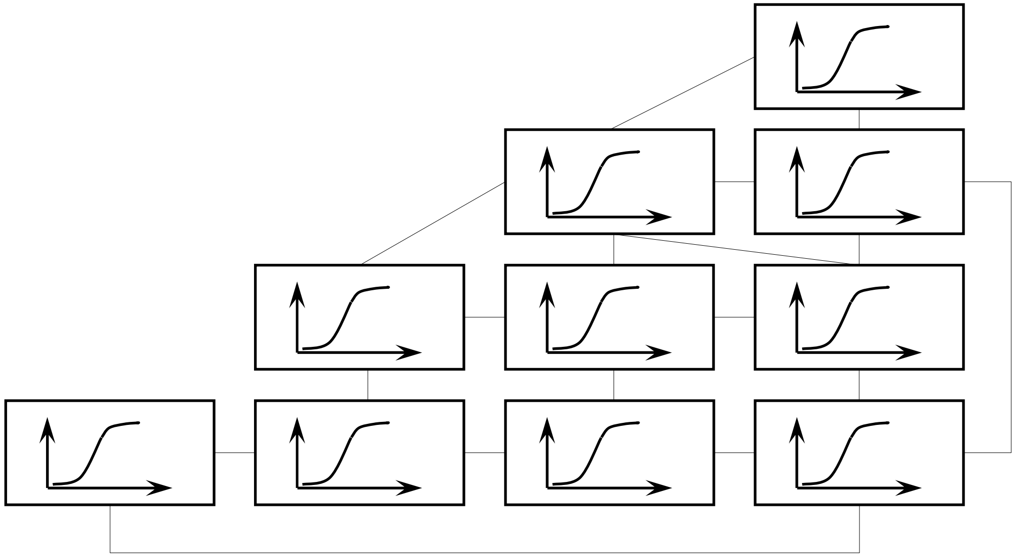
Domination of Higher-Level Systems

- Evolutionary trends of a system determine directions of evolution of its components.
- Also, a problem at the system's level may be a symptom of a problem at the higher hierarchical level.



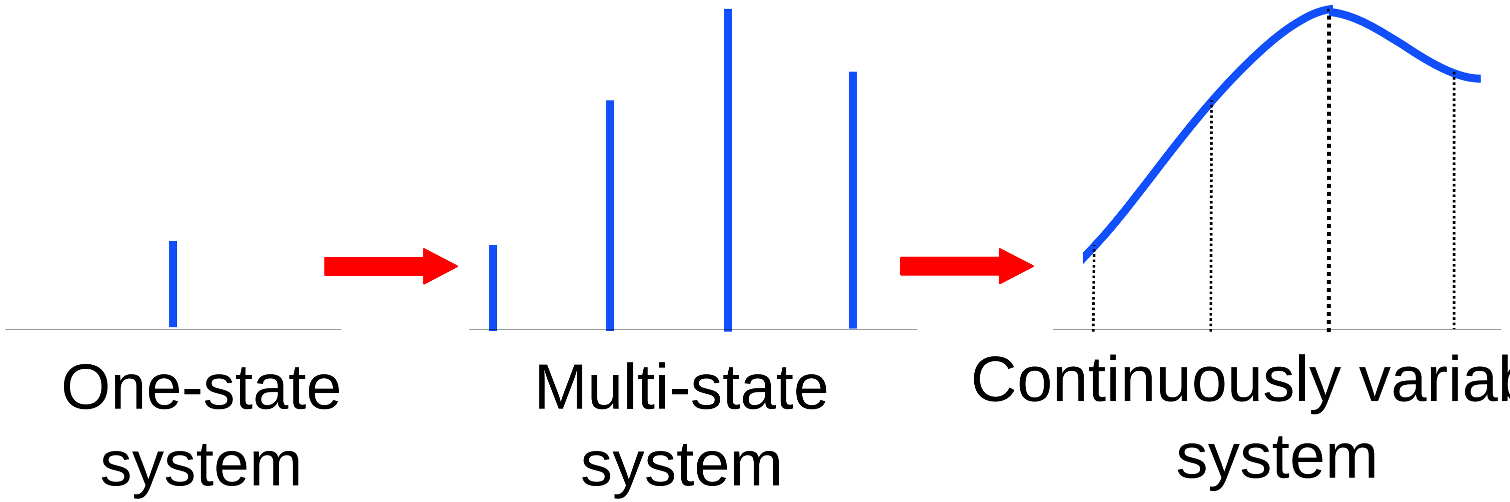
Non-Uniform Evolution of Subsystem

- Various sub-systems evolve along their own S-curves at non-uniform rates.
- This causes development of System Conflicts, which offer opportunities of innovation



Increasing Flexibility

- Technological systems evolve toward more flexible structures capable of adaptation to changing environmental conditions (varying performance regimes) and multi-functionality



Evolution of a Bicycle

1813 No pedals

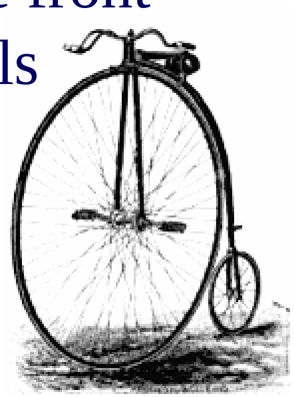
Walking Machine

1840 Pedals added

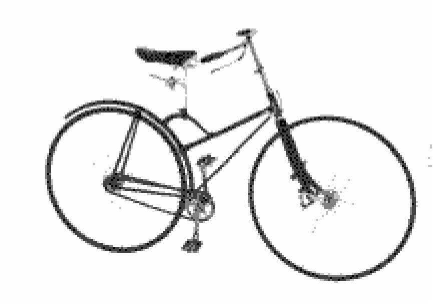
Velocipede or Boneshaker

made entirely of wood, then later with metal tires

1845 Brakes appear Large front wheels



1884 Chain transmission



1890 Pneumatic tires

1897 The over-running clutch

Today's Topics on Creativity

- Introduction
- Creativity
  - Nature
  - Design Rules and Combinatorics
  - Work of Vance and de Bono
- TRIZ theory
  - TRIZ, Value Engineering and Semantic Web tool
- Radiant Thinking, Mind Mapping tool
- Appendix: Technological change: from its creation to economic growth and societal welfare

The Invention Machine

Computational adaptation of TRIZ, Value Engineering and the Semantic Web

Thanks to Invention Machine and

Dr. Mikhail Verbitsky for materials and consultation

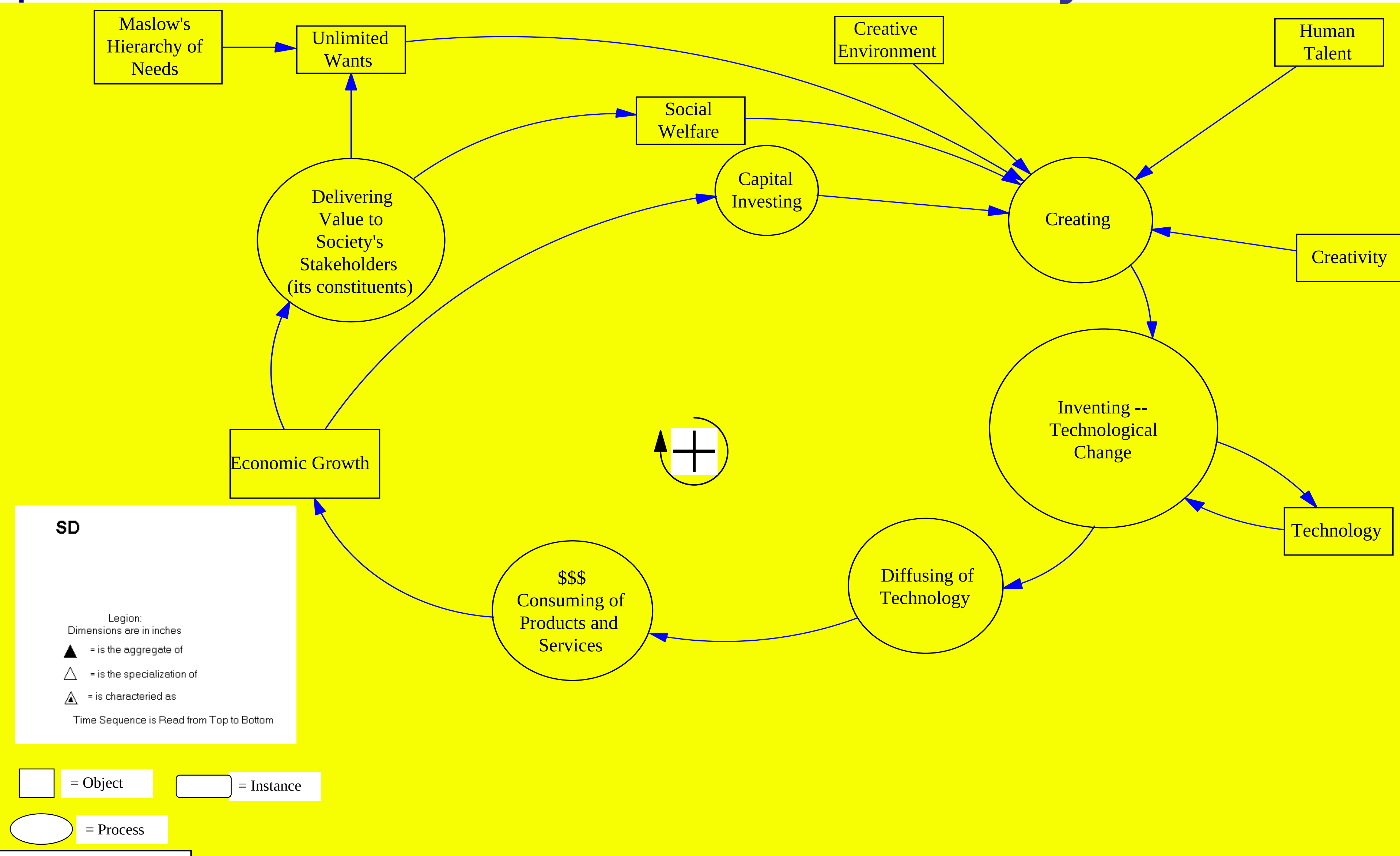
<see separate PowerPoint file>

Today's Topics on Creativity

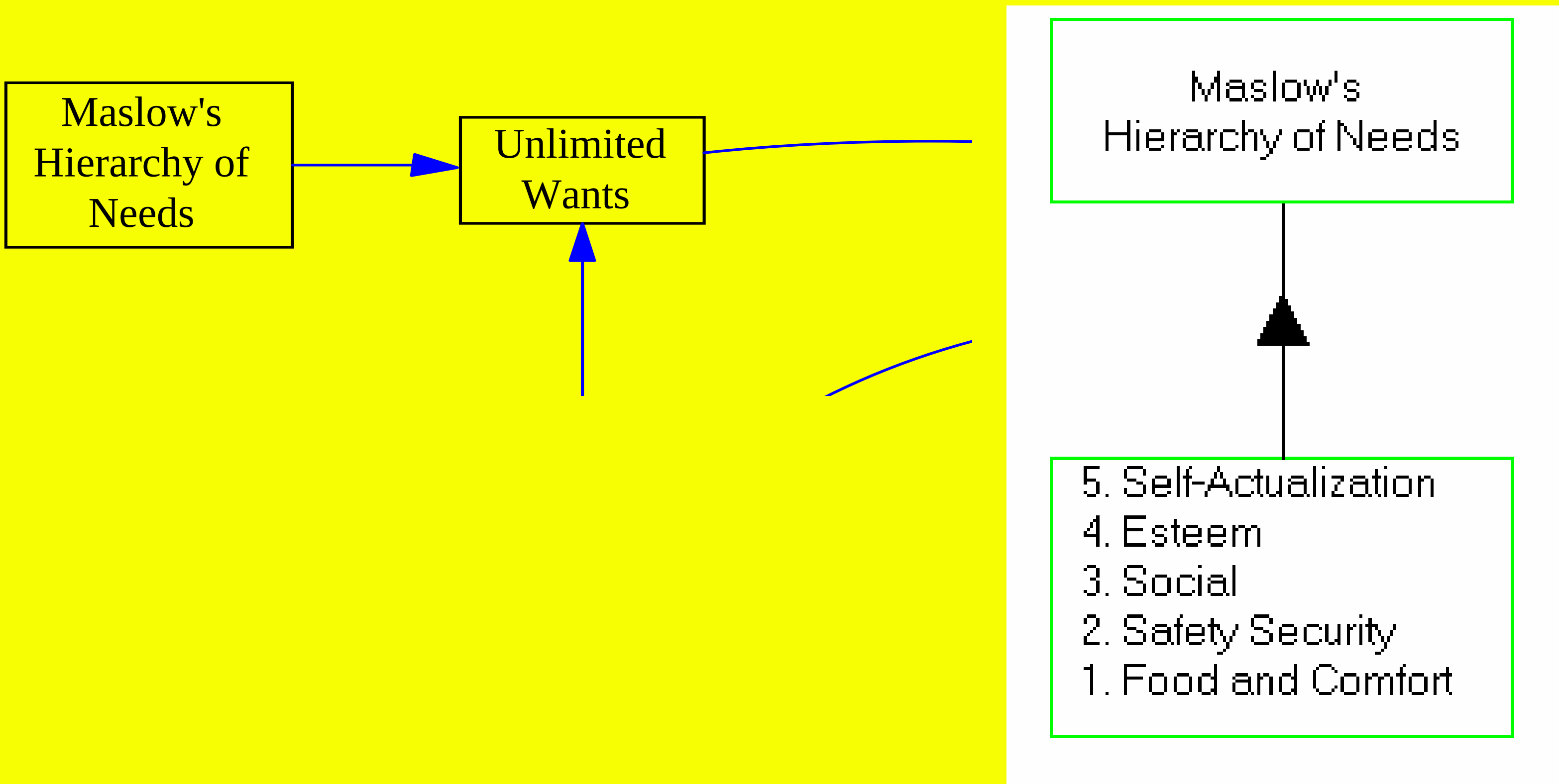
- Introduction
- Creativity
  - Nature
  - Design Rules and Combinatorics
  - Work of Vance and de Bono
- TRIZ theory
  - TRIZ, Value Engineering and the Semantic Web tool
- Radiant Thinking, Mind Mapping tool
- Appendix: Technological change: from its creation to economic growth and societal welfare

Holistic View of Creativity

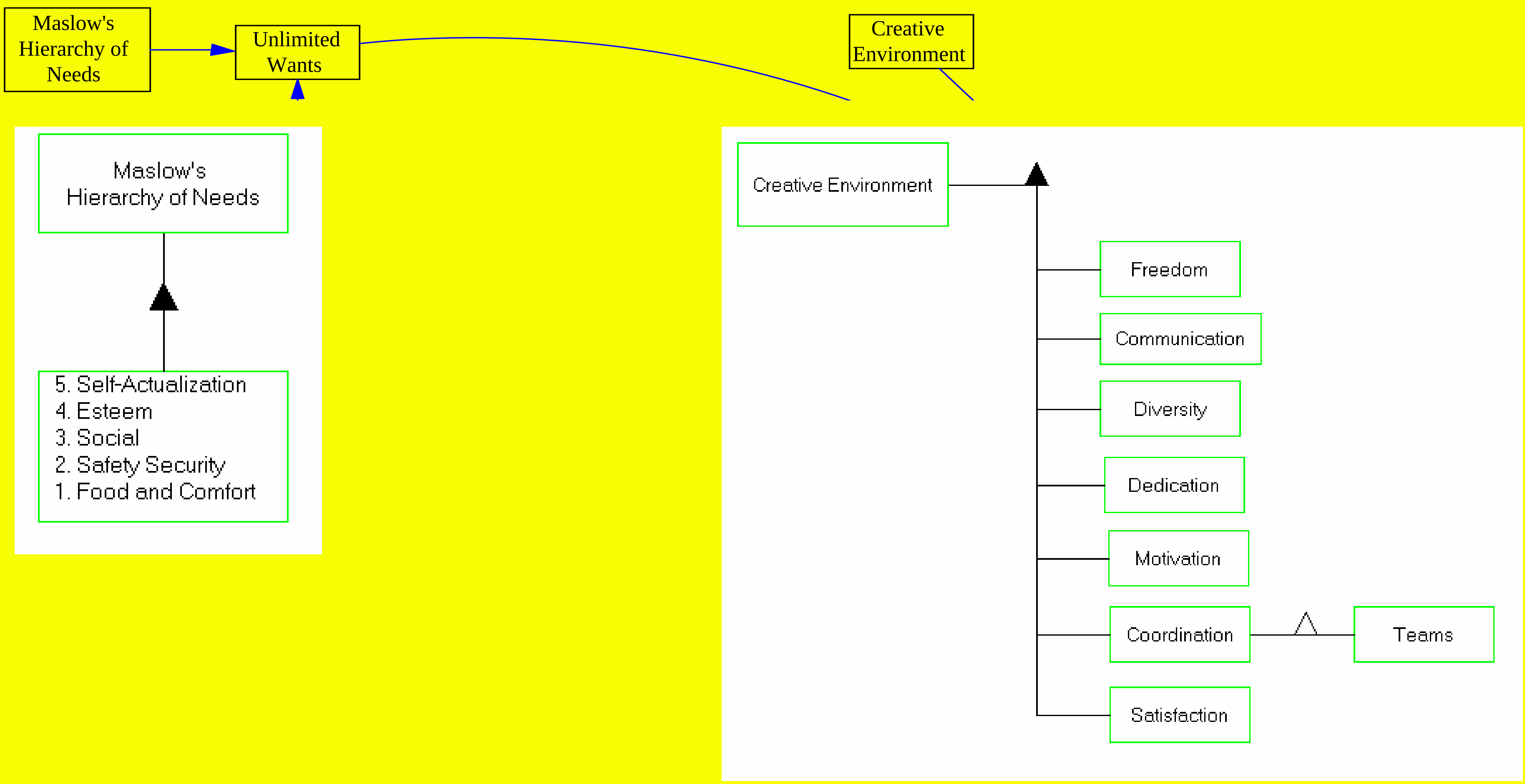
Economic System Dynamics
Holistic View of Creativity



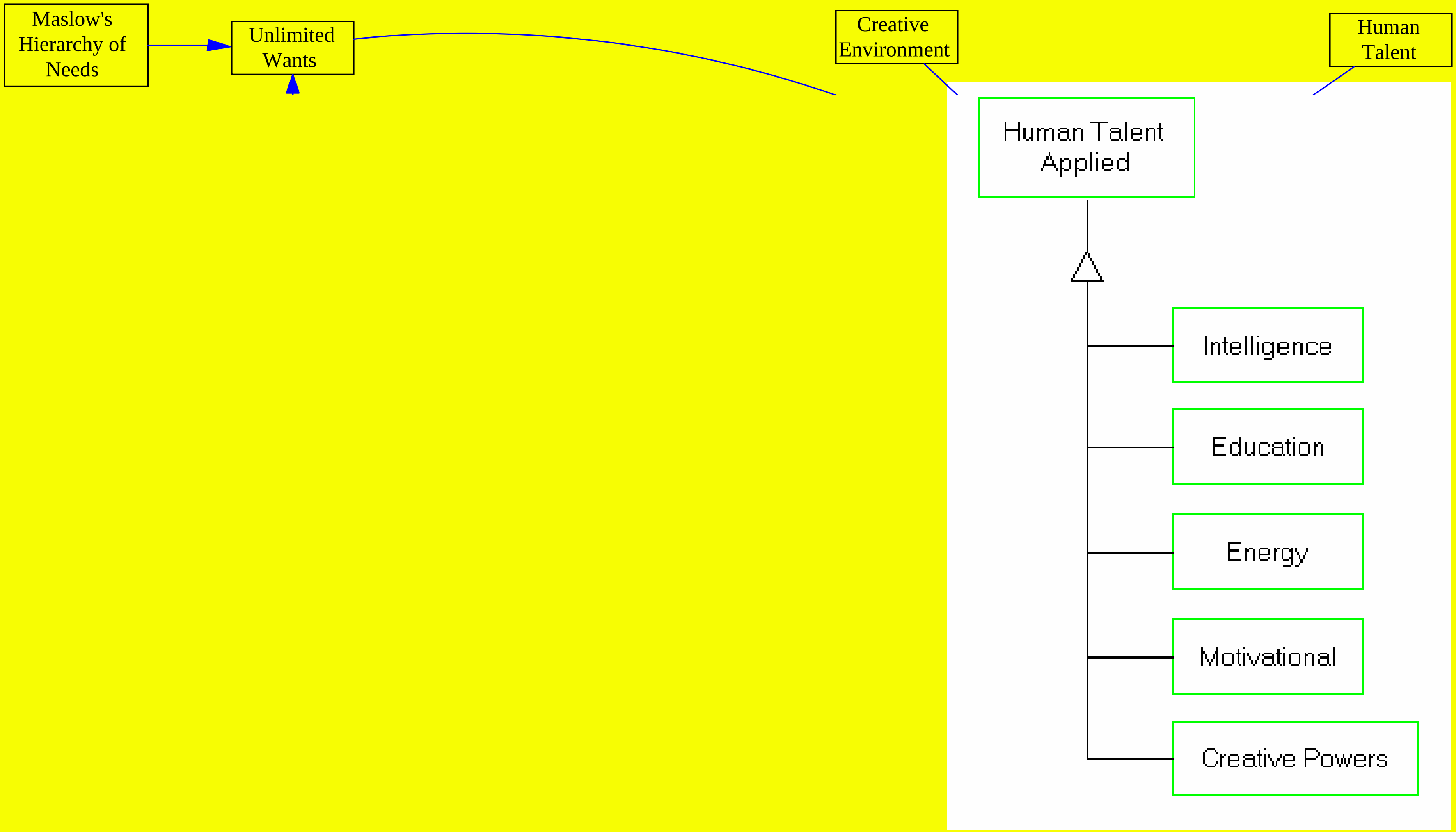
Maslow’s Hierarchy of Needs



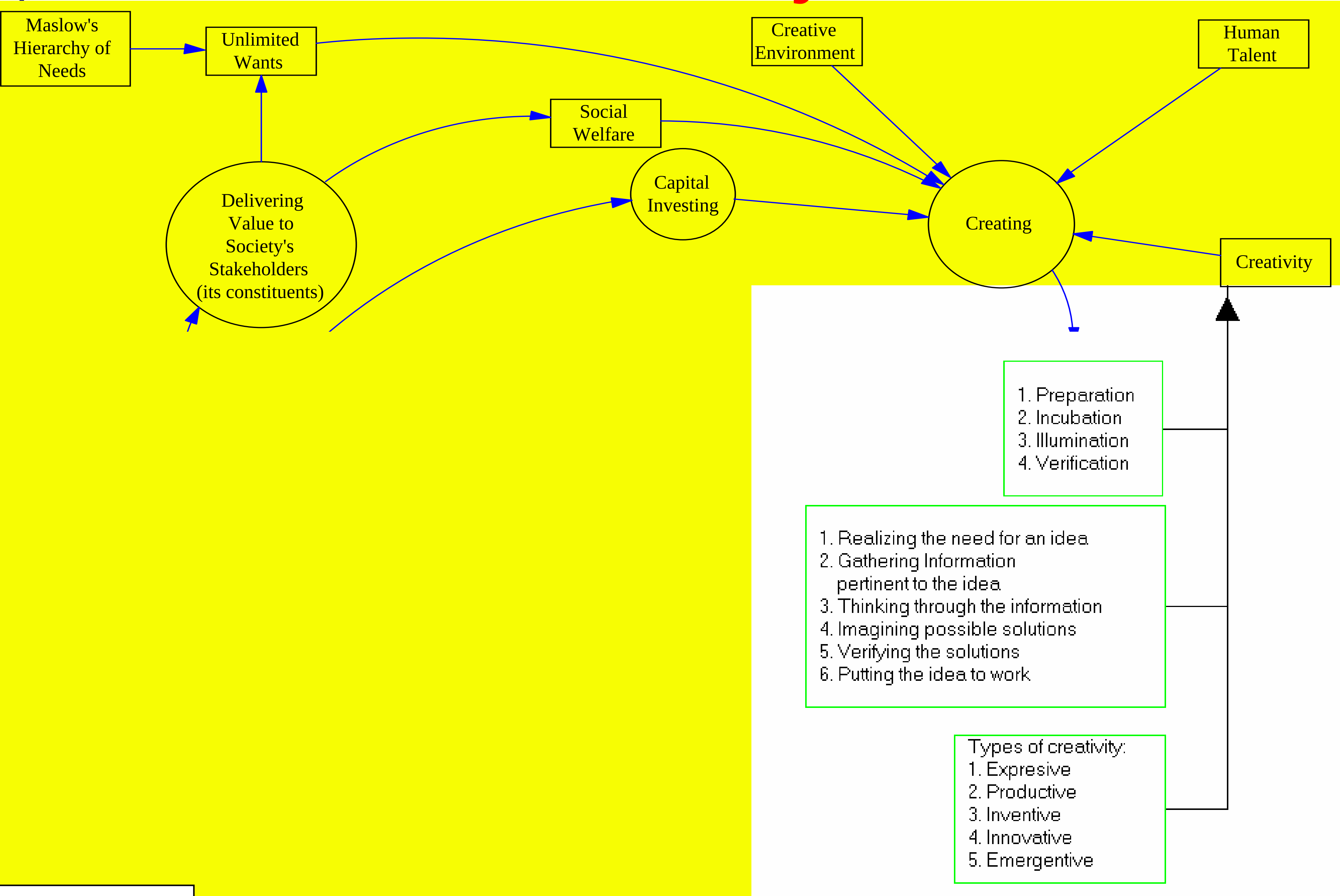
Creative Environment



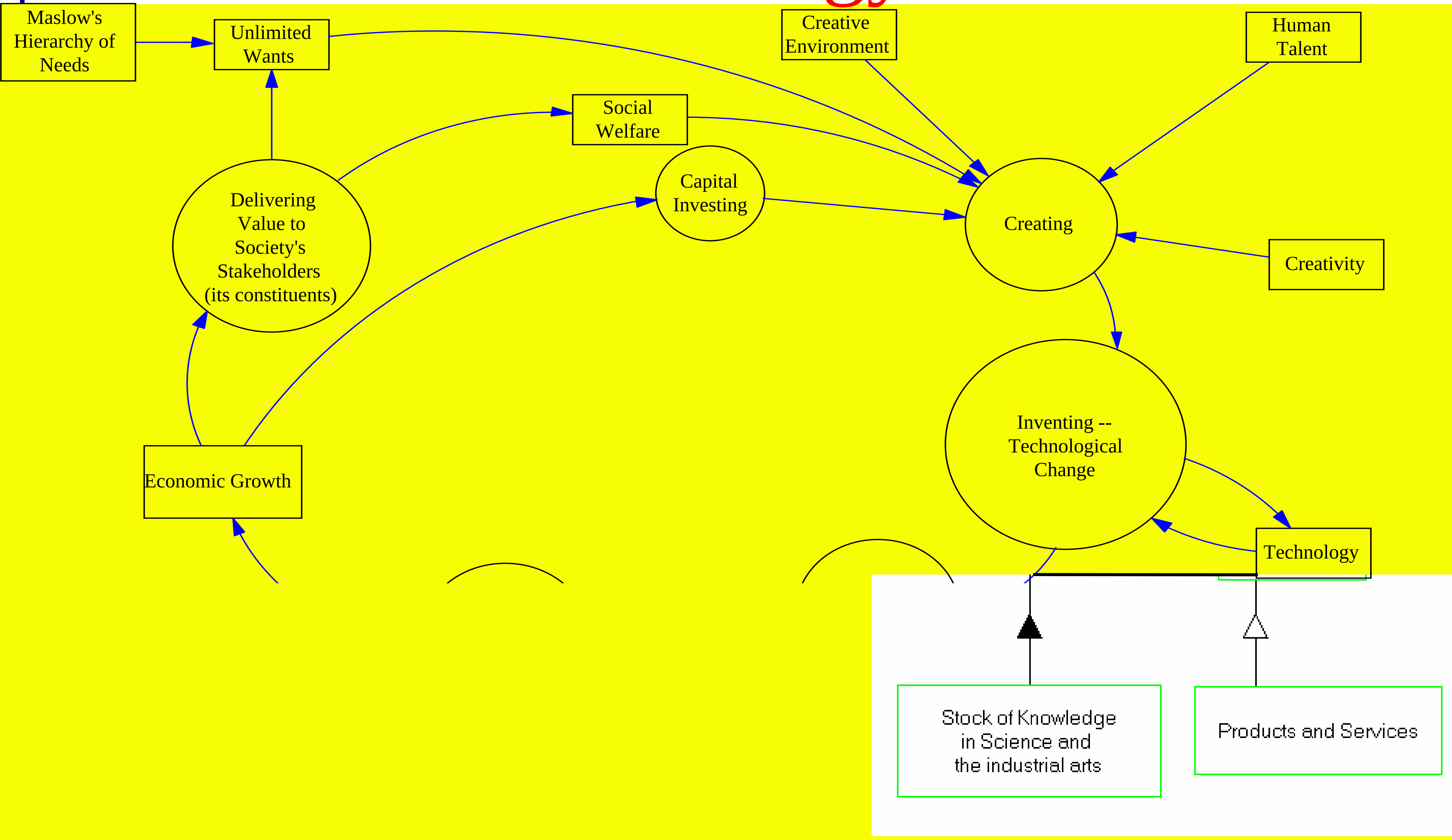
Human Talent Applied



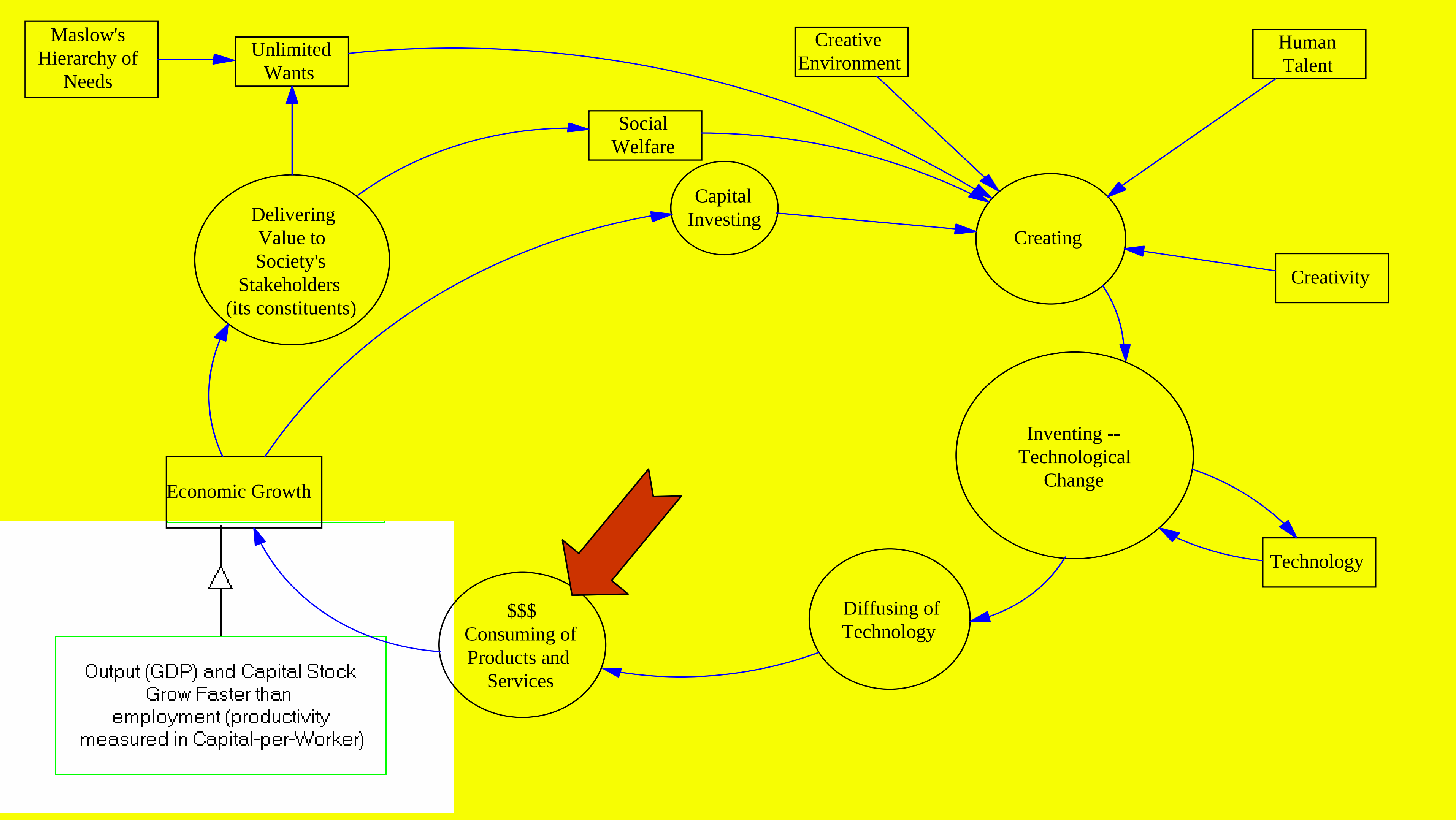
Creativity


Maslow's Hierarchy of Needs
Unlimited Wants
Creative Environment
Human Talent
Social Welfare
Capital Investing
Delivering Value to Society's Stakeholders (its constituents)
Creating
Creativity
1. Preparation
2. Incubation
3. Illumination
4. Verification
1. Realizing the need for an idea
2. Gathering Information pertinent to the idea
3. Thinking through the information
4. Imagining possible solutions
5. Verifying the solutions
6. Putting the idea to work
Types of creativity:
1. Expresive
2. Productive
3. Inventive
4. Innovative
5. Emergentive

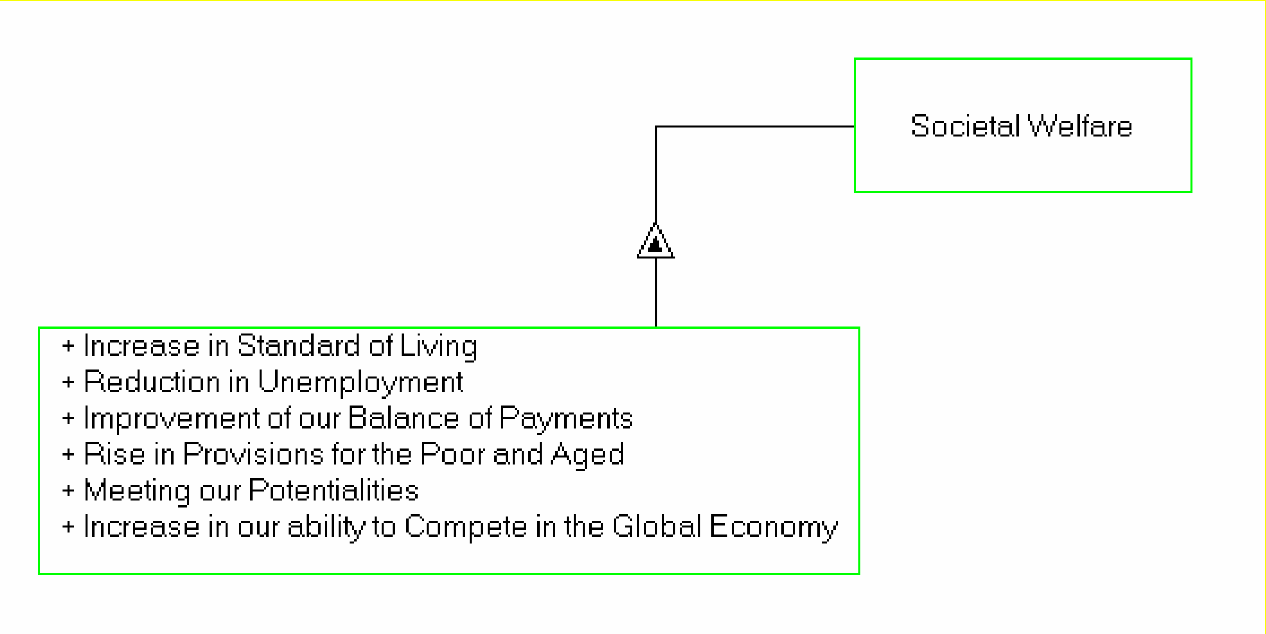
Technology


Maslow's Hierarchy of Needs
Unlimited Wants
Creative Environment
Human Talent
Social Welfare
Capital Investing
Creating
Delivering Value to Society's Stakeholders (its constituents)
Creativity
Inventing -- Technological Change
Economic Growth
Technology
Stock of Knowledge in Science and the industrial arts
Products and Services

Economic Growth



Social Welfare



Combining System Dynamics notation with Object Process Methodology

Stock ≡ Object

Flow ≡ Process