SYSTEM DESIGN

Dr. Gerrit Viljoen

Tutorial Outline

1. INTRODUCTION – Systems and Theory (Session 1 10h00-12h00)
   - Tutorial purpose and content
   - Systems theory and systems philosophy
   - Systems thinking
   - System engineering & design as part of systems theory
   - System design in the product lifecycle
   - Systems architecting
2. SYSTEM DESIGN - Theory (Session 2, 13h30-15h00)
   - System design synthesis process
   - Methods
   - Example Exercise background

Tutorial Outline (Cont.)

3. SYSTEM DESIGN – Example (Session 3, 15h30-17h00)
   - Example Exercise background
   - Breakout into groups for exercise
   - Feedback and discussion
   - Summary

1. Introduction

1.1 Course purpose and content

- Put system design in context
- Identify the role of a system designer on a programme
- Give an overview of the system acquisition process
- Teach system design processes, methods and evaluation techniques
- Work through examples

1.2. Systems philosophy and systems theory

- What is a system?
  - Complexes of elements can exhibit the following distinctions:
    - According to their number
    - According to their type
    - According to the relations of the elements

1. a. o o o o o   b. o o o o
2. a. o o o o o   b. o o o o ●
3. a. o—o—o—o—o   b. o—o
                      | |
                      o—o

1.2. Systems Theory

1.2.1 Systems Theory (Ludwig von Bertalanffy 1928, 1952, 1968)

- Fundamental concepts of the Machine Age (Descartes et al.)
  - Reductionism
  - Analysis
  - Mechanization
- Fundamental concepts of the Systems Age
  - Holism
  - Open vs. Closed systems
  - Hierarchies
  - Systems view of Nature
  - Systems view of ourselves (Mankind)

1.2. Systems theory

1.2.2 Systems Theory (Ludwig von Bertalanffy 1928, 1952, 1968)

– A dynamic system can often be described mathematically as follows:

  - If Q_i is the ith state that describes the p elements of the system we have:

$$\frac{dQ_1}{dt} = f_1(Q_1, Q_2, \ldots, Q_n)$$

$$\frac{dQ_2}{dt} = f_2(Q_1, Q_2, \ldots, Q_n) \quad (1)$$

$$\ldots$$

$$\frac{dQ_n}{dt} = f_n(Q_1, Q_2, \ldots, Q_n)$$

1.2. Systems theory

1.2.3 Systems Theory (Cont.)

- This system has equilibrium points which can be stable, or unstable. At the equilibrium point there is no change in the system states, so we have:

$$f_1 = f_2 = ... = f_n = 0$$

- We then have n equations for n variables that can be solved:

$$Q_1 = Q_1^*, \quad Q_2 = Q_2^*, \quad ..., \; Q_n = Q_n^*$$

- If we introduce a new variable which represents a perturbation around the equilibrium $Q_i = Q_i^* - Q_i'$, we can reformulate the system in (1) with respect to Q_i' and then do a Taylor expansion:

1.2. Systems theory

1.2.4 Systems Theory (Cont.)

$$\frac{dQ_1^{'}}{dt} = a_{11}Q_1^{'} + a_{12}Q_2^{'} + ... + a_{1n}Q_n^{'} + a_{111}Q_1^{'2} + a_{112}Q_1^{'}Q_2^{'} + a_{122}Q_2^{'2} + ...$$

$$\frac{dQ_2^{'}}{dt} = a_{21}Q_1^{'} + a_{22}Q_2^{'} + ... + a_{2n}Q_n^{'} + a_{211}Q_1^{'2} + a_{212}Q_1^{'}Q_2^{'} + a_{222}Q_2^{'2} + ...$$

...

$$\frac{dQ_n^{'}}{dt} = a_{n1}Q_1^{'} + a_{n2}Q_2^{'} + ... + a_{nn}Q_n^{'} + a_{n11}Q_1^{'2} + a_{n12}Q_1^{'}Q_2^{'} + a_{n22}Q_2^{'2} + ...$$

- A general solution of this system of equations is:

$$Q_1^{'} = G_{11}e^{\lambda_1 t} + G_{12}e^{\lambda_2 t} + ...G_{1n}e^{\lambda_n t} + G_{111}e^{2\lambda_1 t} + ...$$

$$Q_2^{'} = G_{21}e^{\lambda_1 t} + G_{22}e^{\lambda_2 t} + ...G_{2n}e^{\lambda_n t} + G_{211}e^{2\lambda_1 t} + ...$$

....

$$Q_n^{'} = G_{n1}e^{\lambda_1 t} + G_{n2}e^{\lambda_2 t} + ...G_{nn}e^{\lambda_n t} + G_{n11}e^{2\lambda_1 t} + ...$$

1.2. Systems theory

1.2.5 Systems Theory (Cont.)

- Where G are constants and λ the roots of the characteristic equation:

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22}-\lambda & \dots & a_{2n} \\ \dots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn}-\lambda \end{vmatrix} = 0$$

- Inspection of the roots allow a number of conclusions to be drawn about the system. If all the real parts are negative, the system is stable. If the roots are imaginary with negative real parts, the system is asymptotically stable. If there are any real roots that are positive, the system is unstable.
- These effects can be graphically described on the phase plane

1.2. Systems theory

1.2.6 Systems Theory Concepts

- **Wholeness**
- **Summativity**
- **Progressive segregation**
- **Centralisation**
- **Finality or Teleology**
- **Isomorphisms**

1.2. Systems theory

1.2.6 Systems Theory Concepts (cont.)

– **<u>Wholeness</u>**:

- Inspect the Taylor series expansion (2):

$$\frac{dQ_1'}{dt} = a_{11}Q_1' + a_{12}Q_2' + \ldots + a_{1n}Q_n' + a_{111}Q_1'^2 + a_{112}Q_1'Q_2' + a_{122}Q_2'^2 + \ldots$$

- We see that any change in some quantity Q_1 , is a function of the quantities of all the elements Q_1 to Q_n . On the other hand, a change in a certain Q_i causes a change in all the other elements and in the total system. The system therefore behaves as a ***whole***, the changes in every element depending on all the others.

1.2. Systems theory

1.2.7 Systems Theory Concepts (Cont.)

– **Summativity**:

- Let the coefficients of the variables Q_i ($j \neq i$) be zero:

$$\frac{dQ_1'}{dt} = a_{11}Q_1' + a_{111}Q_1'^2 + \dots$$

- A change in each element depends only on that element itself. Such behaviour is called physical summativity or independence and is true for those complexes that we may call “heaps”. It does not apply to those systems which are called *Gestalten* in German.
- It is used for the **Mechanization** of relative simple, independent parts of a system, e.g. the eye.
- We conclude that all systems are non-summative by nature.

1.2. Systems theory

1.2.8 Systems Theory Concepts (Cont.)

– **Progressive Segregation**

- The coefficient of the system can reduce as a function of time:

$$\lim_{t \to \infty} a_{ij} = 0$$

- The system passes from a state of wholeness to a state of independence of the elements.
- This system undergoes *progressive mechanization* which plays an important role in biology e.g. embryonic development and cell differentiation.

1.2. Systems theory

1.2.9 Systems Theory Concepts (Cont.)

– **Centralisation**

- Suppose the coefficients of one element, p_s are large in all equations while the coefficients of the other elements are small:

$$\frac{dQ_1}{dt} = a_{11}Q_1 + \ldots + a_{1s}Q_s + \ldots$$

$$\frac{dQ_s}{dt} = a_{s1}Q_s + \ldots$$

$$\frac{dQ_n}{dt} = a_{11}Q_n + \ldots + a_{n1}Q_n + \ldots$$

- The system is then *centred* around element $p_{s.}$
- If the coefficients a_{is} of p_s in some or all equations are large while the coefficients in the equation of p_s itself are small, a small change in p_s will cause considerable change in the total system. p_s is then called a *trigger.*

1.2. Systems theory

1.2.10 Systems Theory Concepts (Cont.)

- **Finality**
  - Systems of the type considered have three kinds of solution:
    - They asymptotically attain a stable stationary state
    - They may never attain such a state (which is impossible in real life systems)
    - Or there may be periodic oscillations:
  - It has been maintained that certain formulations in physics have an apparently finalising character. The systems seem to aim at an equilibrium to be reached in the future.
  - Types of finality
    - Static, to be useful for a specific purpose
    - Dynamic, meaning a directiveness of processes (man made machines)
    - Equifinality, the same final condition can be reached from different initial conditions through different ways

1.2. Systems theory

1.2.11 Systems Theory Concepts (Cont.)

- **Teleology**
  - True Finality is also called the *teleology* of certain systems, or the minimum principle of mechanics. Everywhere in physics we have the principle of a maximum effect with minimum effort. Teleology is the doctrine that there is evidence of purpose or design in the universe. This doctrine stands in opposition to the anthropological argument.

1.2. Systems theory

1.2.12 Systems Theory Concepts (Cont.)

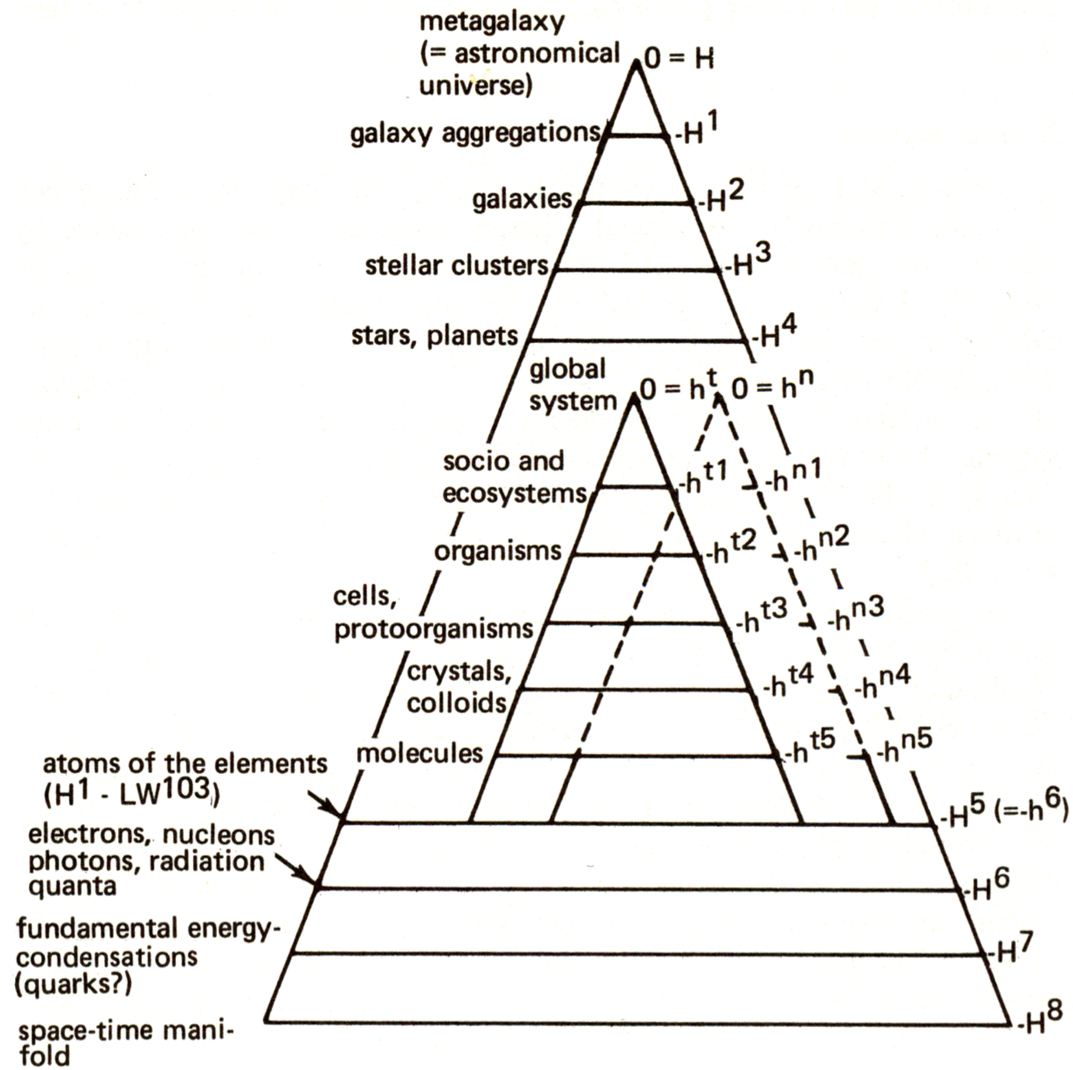
- **Isomorphism in Science**
  - Some system principles have application over a wide variety of sciences, called an Isomorph
  - Types of Isomorphism
    - Analogies (Scientifically worthless, but useful)
    - Homologies, The respective laws are identical
    - Explanation, the general functions f of eq 1 are replaced by specified functions applicable to the individual case.

1.2. Systems view of Nature

1.2.14

- Natural Systems

  Intra and inter-systemic hierarchies

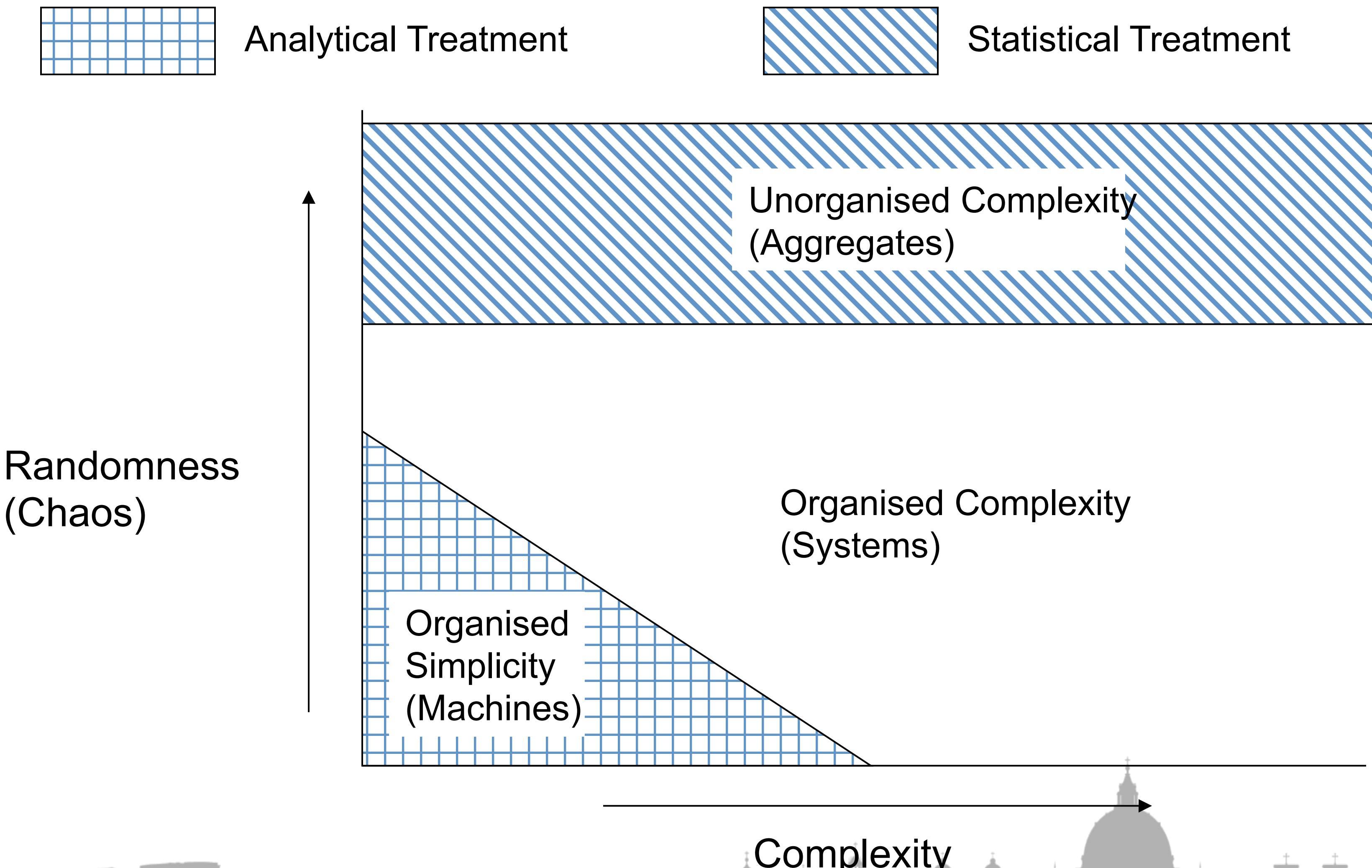


1. Introduction

1.3. Systems thinking

- Barry Richmond – The thinking in Systems Thinking
  - Seven essential skills
- Russel Ackoff – Lectures on Systems Thinking
  - Applying systems thinking to management
- Peter Senge – The fifth discipline
- Gerald M Weinberger – An Introduction to General Systems Thinking
  - Theory and research into systems thinking
- Ian Mitroff – Smart Thinking for Crazy Times
  - The art of solving the right problems

System Types vs. Thinking Methods



1.3 Systems Thinking

1.3.1 Systems Thinking

"You cannot solve problems created by the current paradigm of thought within the current paradigm of thought"

Albert Einstein

- Systems thinking (Plato) stands in opposition to Analytical thinking (Aristoteles)
- Analysis can determine how things work, but can never say why things work! For that we need systems thinking!
- Explanations lies outside the system. The product of explanation is called understanding, the problem of science and analysis is knowledge, not explanation

1.3 Systems Thinking

1.3.2 Systems Thinking: Synthesis vs Analysis

Analytical Thinking	Systems Thinking
What are the parts?	What is this a part of?
What are the properties and behaviors of the parts separately?	What is the behavior of the containing whole?
Aggregate the understanding of the parts to get an understanding of the whole	Disaggregate the understanding of the containing whole by identifying the role or function of what we want to explain in the containing whole

- A system as a whole is defined by its functions in its larger system. A system cannot be divided into independent parts. The functions lie in the interactions between the parts, e.g. the emergent behavior
- Example, and Architect design a house first, then puts rooms in it

1.3 Systems Thinking

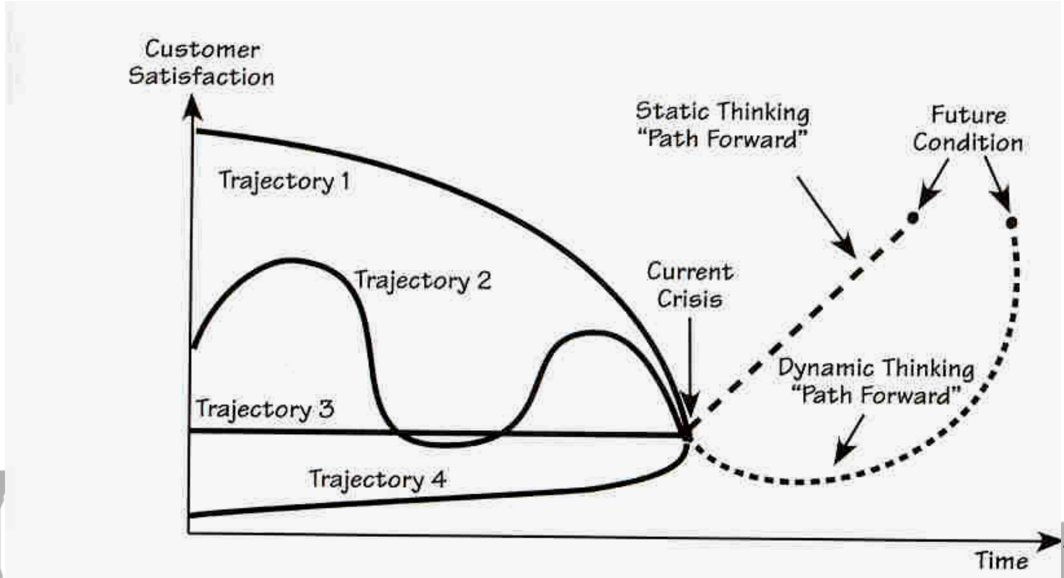
1.3.3 Seven Thinking Skills for Systems Thinking:

- Specifying the problem or issue, and setting boundaries:
  1. Dynamic Thinking
  2. System-as-cause thinking
  3. Forest thinking
- Construction of your model:
  4. Operational thinking
  5. Closed-loop thinking
  6. Quantitative thinking
- Testing your model:
  7. Scientific thinking

1.3 Systems Thinking

1.3.4 Dynamic Thinking:

- Dynamic thinking skills enable you to depict your issue or challenge as a set of patterns that unfold over time.
- Static thinking looks at the current condition only, e.g.
  - Customer satisfaction is in the pits, solution – increase customer satisfaction
  - It says nothing about how it got there, and it says nothing about the path that must be followed to get it right
- The best tool to aid dynamic thinking is the behavior over time graph:



1.3 Systems Thinking

1.3.5 System-as-cause thinking:

- System-as-cause thinking help you determine which underlying set of relationships are most relevant for improving the behaviour pattern of interest
- It encourages you to view the system itself as the cause of the behaviour it is exhibiting
- Hone this skill by reframing your perception of any behaviour that has been chalked up to “outside forces”. Instead, view the behaviour as a result of relationships involving variables that are under your system’s control. Determine which variables are outside, partially and completely under your system’s control.

1.3 Systems Thinking

1.3.6 Forest thinking:

- This type of thinking helps you finalize the breadth and depth that your hypothesis, or model, will have.
- It is the view from 10000m, and rising above the details that count, *"To see the forest for the trees"*
- The first skill is elevation, the ability to rise above the local space-time surroundings. This can be done by constantly questioning the boundaries of the system by asking how one can influence something outside of the system.
- The second skill is filtering, the ability to sift out all but the most essential detail. To hone this skill, look for similarities rather than differences in people, situations, and problems you encounter.
- This skill support out-of-the-box thinking needed to reveal higher leverage interventions

1.3 Systems Thinking

1.3.7 Operational thinking:

- Correlation based factors thinking vs. causal thinking
- Causal thinking, or operational thinking, asks what the processes are that causes an outcome. Correlation based thinking asks which factors, or drivers, influence the outcome. These two methods can give widely different answers.
- Operational thinking supports more effective communication, and it enables one to identify leverage points in the system for improving performance
- Whenever you are asked to make a list of success factors, or drivers, ask yourselves first, “what really causes this phenomenon?”
- An example is “benchmarkings”, which can lead to extremely wrong conclusions

1.3 Systems Thinking

1.3.8 Closed-loop thinking :

- This type of thinking looks for feedback relationships in your model
- An excellent example is “downsizing”. The straight-line thinking gives us that cutting staff would reduce company cost and hence increase profits. Unfortunately, as many firms discovered, there was a closed loop relation as the remaining workers were demoralized, some were overburdened and their productivity fell.
- Closed loop relationships often lead to unintended consequences
- To hone closed-loop thinking skills, just listen carefully whenever causality is at issue. Begin with the one way causal link that is being mentioned, and then simply close the loop
- For example, “advertising leads to kids smoking” can become: “money kid smokers pay for cigarettes underwrites the advertising that seduces their friends”

1.3 Systems Thinking

1.3.9 Quantitative thinking:

- Quantifying your problem leads to increased clarity, perspective and boosts the level of rigor in the thinking process
- Quantification is usually done through simulation, even of soft issues such as self-esteem.
- It is important to remember that not all system parameters can be accurately measured, e.g. cost-effectiveness. What is important is to get the model structure correct.
- The skill can be honed through working with computer simulation models

1.3 Systems Thinking

1.3.10 Scientific thinking:

- Scientific thinking uses scientific principles to to systematically build confidence that a system model is useful for developing insights into and how to improve performance
- It does not try to get a “best fit” model, but it is tested by ensuring that all the parameters and their relation in the system that can effect the outcome are included.
- Scientific thinking does not predict the future, but identify levers to create the future e.g. Jay Forester’s Market growth model:

1.3 Systems Thinking

1.3.11 Smart thinking-the art of solving the right problem:

Solving the wrong problem precisely:

1. Picking the wrong stakeholders
2. Selecting too narrow a set of options
3. Phrasing the problem incorrectly
4. Setting the boundaries/scope too narrowly
5. Failing to think systematically

How to solve the right problems:

1. Picking the right stakeholders
2. Expanding your options
3. Phrasing your problem correctly
4. Expand the boundaries of the problems
5. Managing the Paradoxes inherent in Problems

1. Introduction

1.4. System engineering and design as part of systems theory

- Ludwig von Bertalanffy
- Benjamin Blanchard and Wolter Fabrycky
- Sarah Sheard

1. Introduction

1.4.1 System engineering as part of systems theory

- Bertalanffy identified systems engineering as one of the advances in systems theory
- Blanchard defines systems engineering as follows:

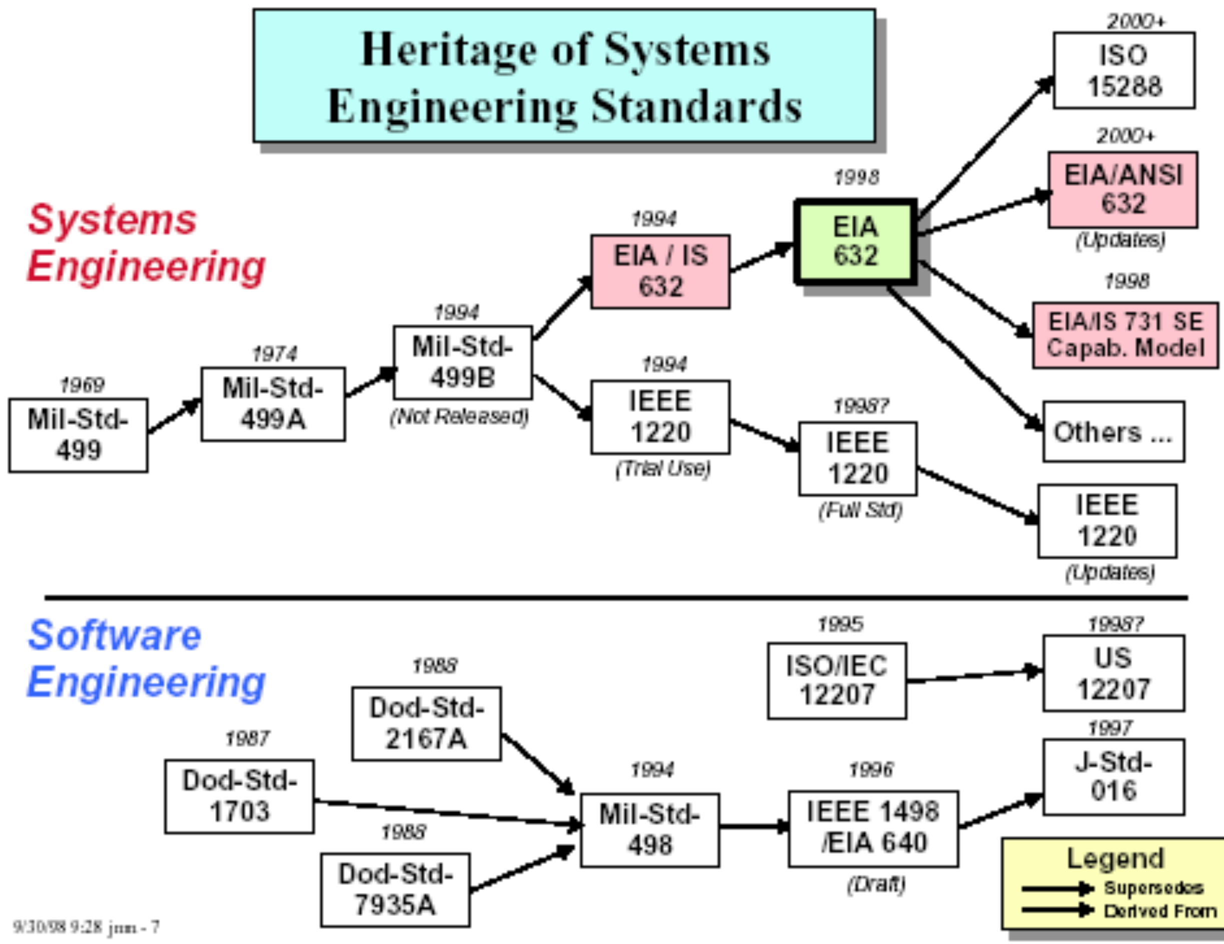
  " An interdisciplinary collaborative approach to derive, evolve, and verify a life-cycle balanced system solution which satisfies customer expectations and meets public acceptability"

- Systems engineering is good engineering with special areas of emphasis:
  - A top-down approach
  - A life-cycle orientation
  - A complete definition of system requirements
  - An interdisciplinary or team approach

1.4 System Engineering

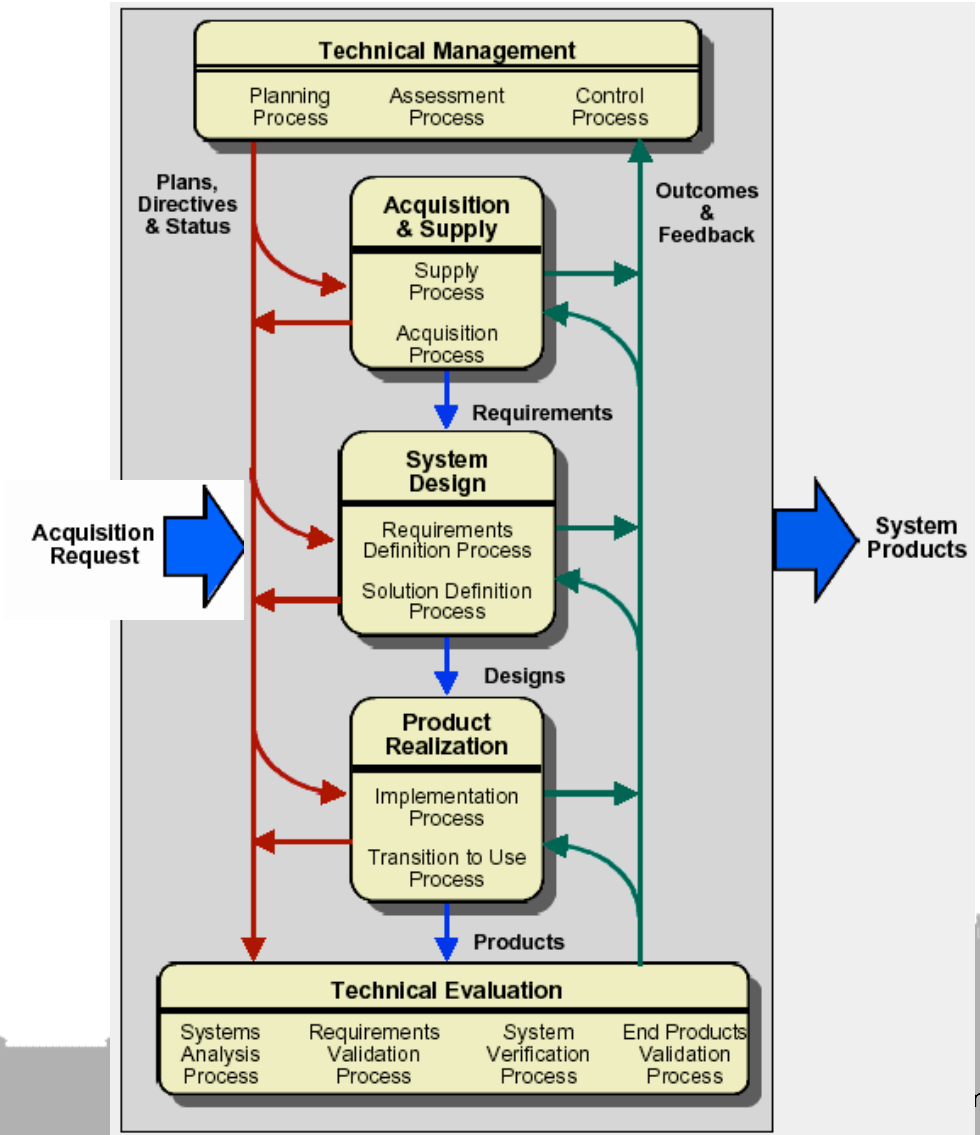
1.4.1 SYSTEM ENGINEERING PROCESSES

- Documented in many standards

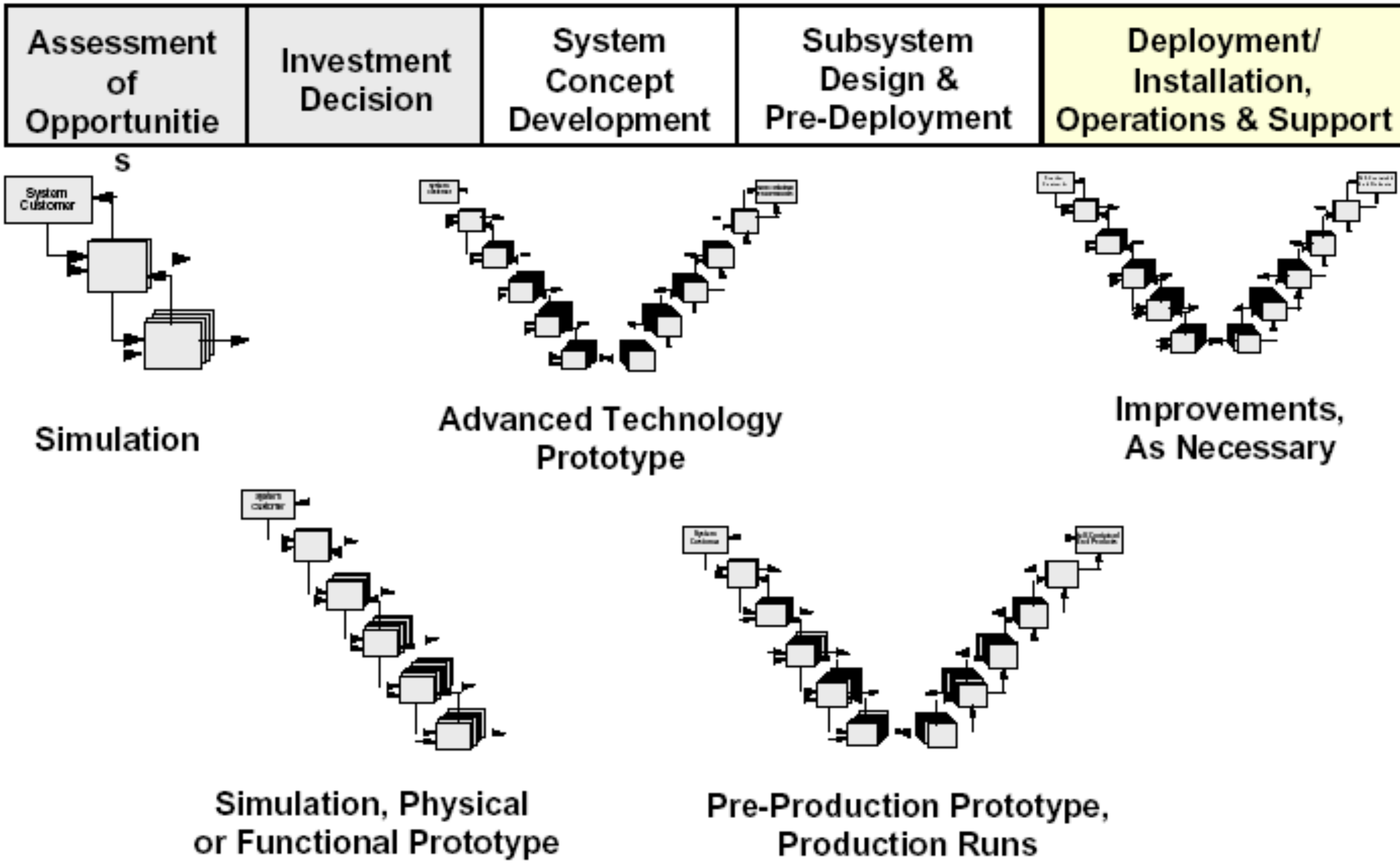


1.4 System Engineering Standards

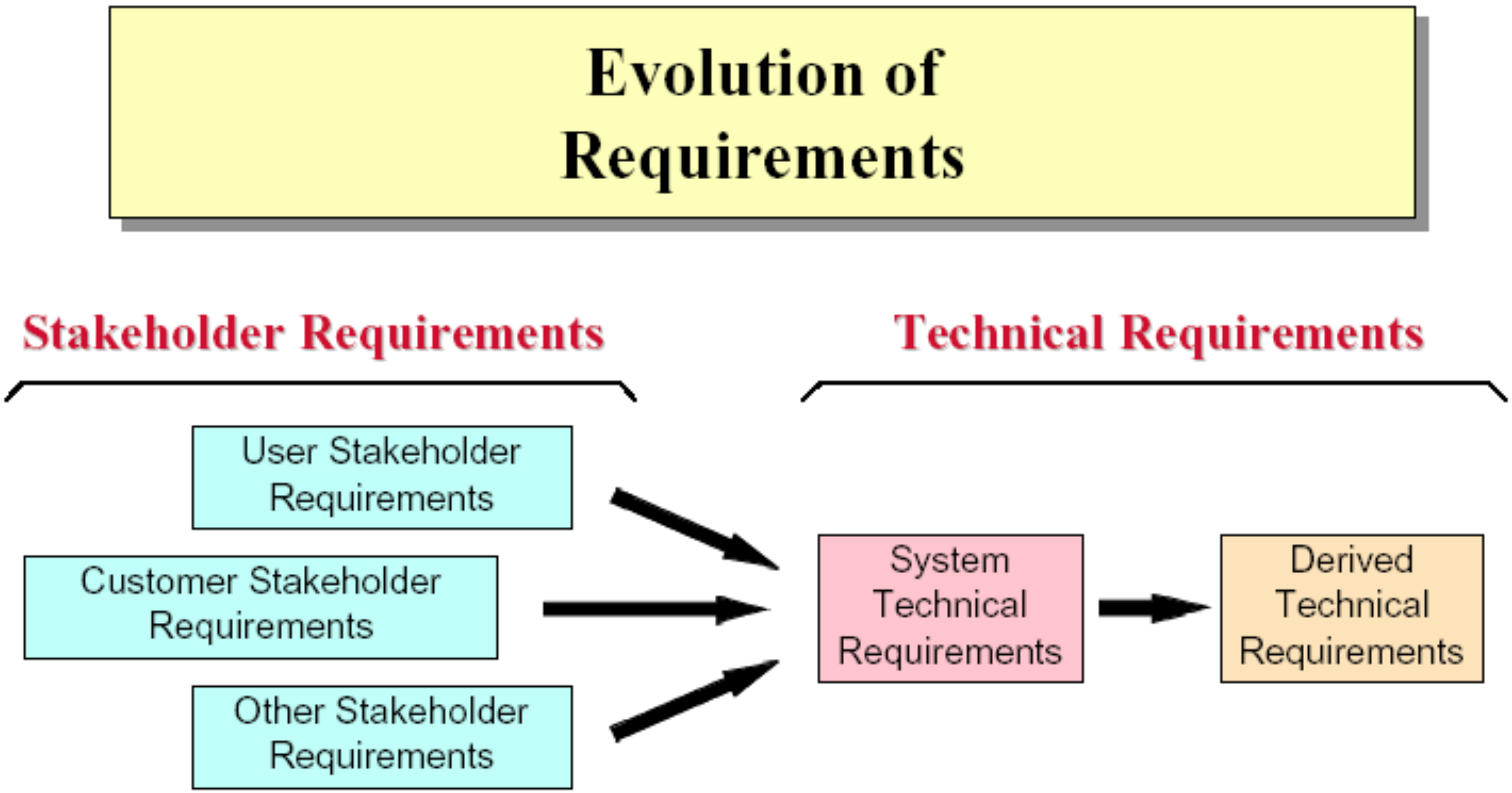
1.4.1 EIA 632 has 13 Processes and 32 requirements



1.4.2 Requirements analysis



1.4.3 Requirements analysis

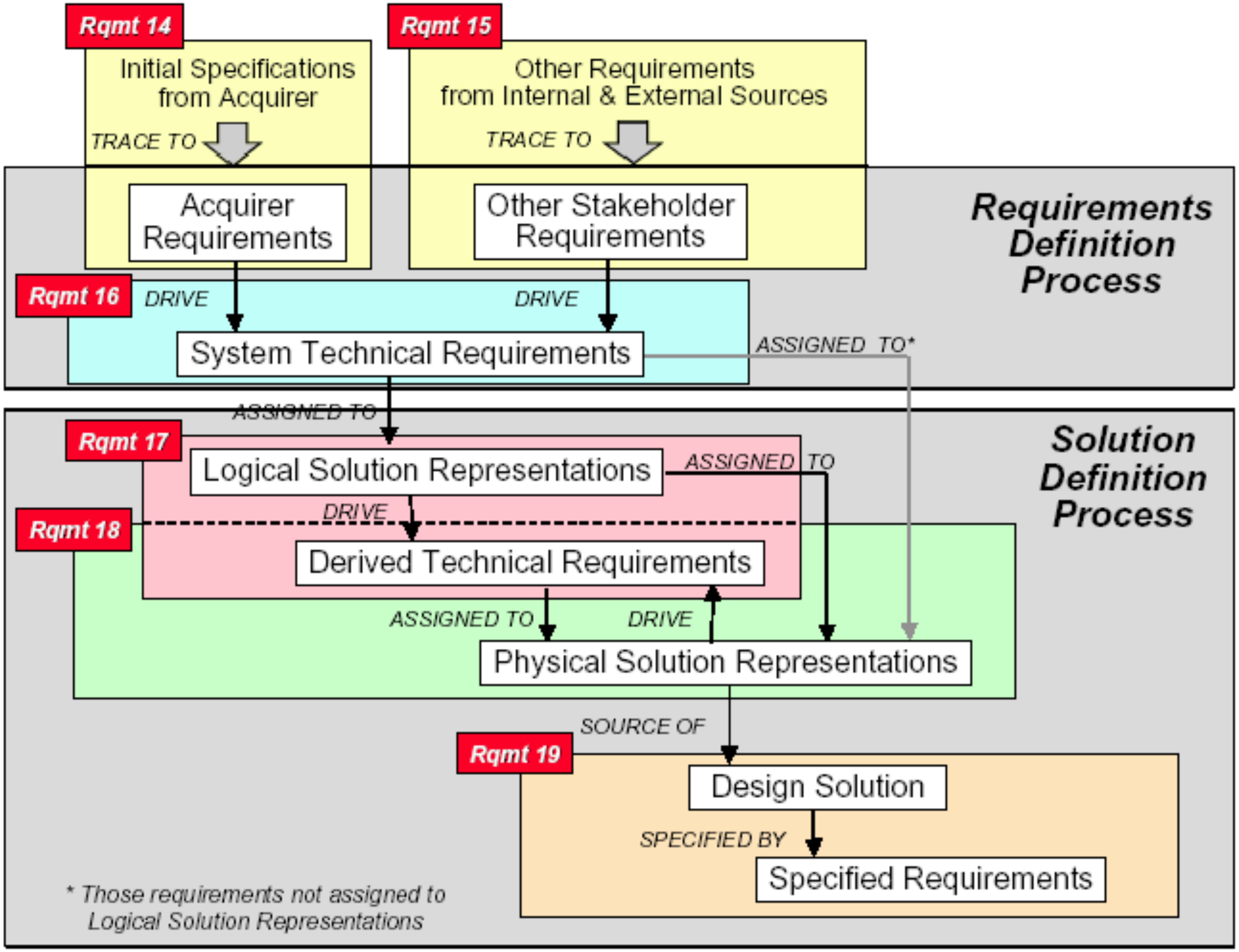


1.4.4 Requirements analysis

Types of Requirements

- **Functional Requirements**
  - » *What* an item is to accomplish
    - Behavior of an item
    - An effect produced
    - Action or service to be performed
- **Performance Requirements**
  - » *How well* an item is to accomplish a function
    - ... like how much, how often, how many, how few, . . .
- **Interface Requirements**
  - » *Conditions* of interaction between items
    - ... could be functional, physical, logical, . . .

1.4.5 EIA 632 Solution Process



1.4.6 Solution Process

Solution Process Requirements

17. Logical Solution Representation

    The developer **shall** define one or more validated sets of logical solution representations that conform with the technical requirements of the system

18. Physical Solution Representations

    The developer **shall** define a preferred set of physical solution representations that agrees with the assigned logical solution representations, derived technical requirements, and system technical requirements

19. Specified Requirements

    The developer **shall** specify requirements for the design solution

1.4.7 System Engineering Standards

ISO 15288 – Life Cycle Processes

Enterprise Processes
- Enterprise Management Process
- Investment Management Process
- System Life Cycle Management Process
- Resource Management Process

Agreement Processes
- Acquisition Process
- Supply Process

Project Processes
- Planning Process
- Assessment Process
- Control Process
- Decision Making Process
- Risk Management Process
- Configuration Management Process

Technical Processes
- Stakeholder Needs Definition Process
- Requirements Analysis Process
- Architectural Design Process
- Implementation Process
- Integration Process
- Verification Process
- Transition Process
- Validation Process
- Operation and Maintenance Process
- Disposal Process

1.4.8 ISO 15288 Architectural Design Process

Synthesize a solution that satisfies system requirements:

- Encapsulate and define areas of solution expressed as a set of separate problems of manageable, conceptual and ultimately realizable proportions.
- Identify and explore one or more implementation strategies at a level of detail consistent with the system's technical and commercial requirements and risks.
- Define an architectural design solution in terms of the requirements for the set of system elements from which the system is configured. The specified design requirements resulting from this process are the basis for verifying the realized system and for devising an assembly and verification strategy.
  - Define the architecture.
  - Analyze and evaluate the architecture.
  - Document and maintain the architecture.

1. 4 System engineering

1.4.9 System design as part of systems engineering

- Sarah Sheard identified these 12 roles of system engineering:

Role	Abbr.	Short Name
1	RO	Requirements Owner
2	SD	System Designer
3	SA	System Analyst
4	VV	Validation/Verification Engr.
5	LO	Logistics/Ops Engineer
6	G	Glue Among Subsystems
7	CI	Customer Interface
8	TM	Technical Manager
9	IM	Information Manager
10	PE	Process Engineer
11	CO	Coordinator
12	CA	Classified Ads SE

1. Introduction

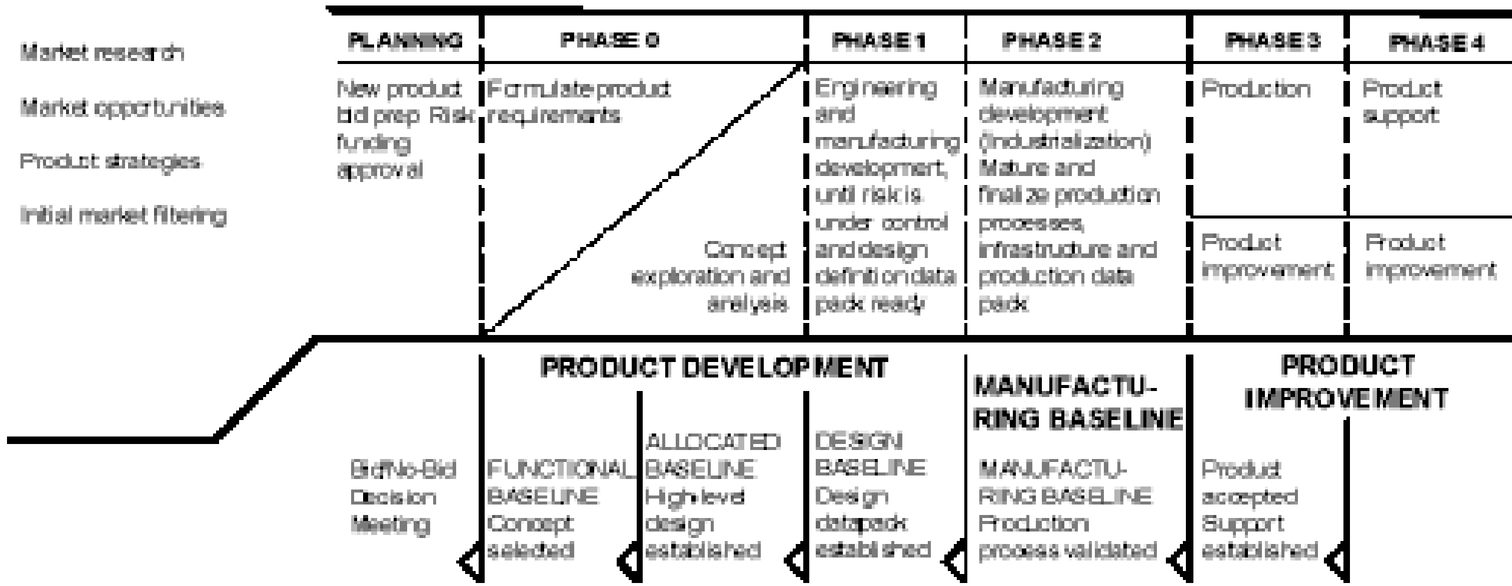
1.5 System design – overview

System Design Definition

- **System Design Process**: A process for converting stakeholder requirements into Design Solutions.
- The allocated requirements are the basis for the synthesis of the system solution (the system design). System engineering sees the whole design, down to the deepest levels, as the system design, whereas in this course we only look at systems design at system level, it seems to be similar to systems architecting

1.5 System Design - Overview

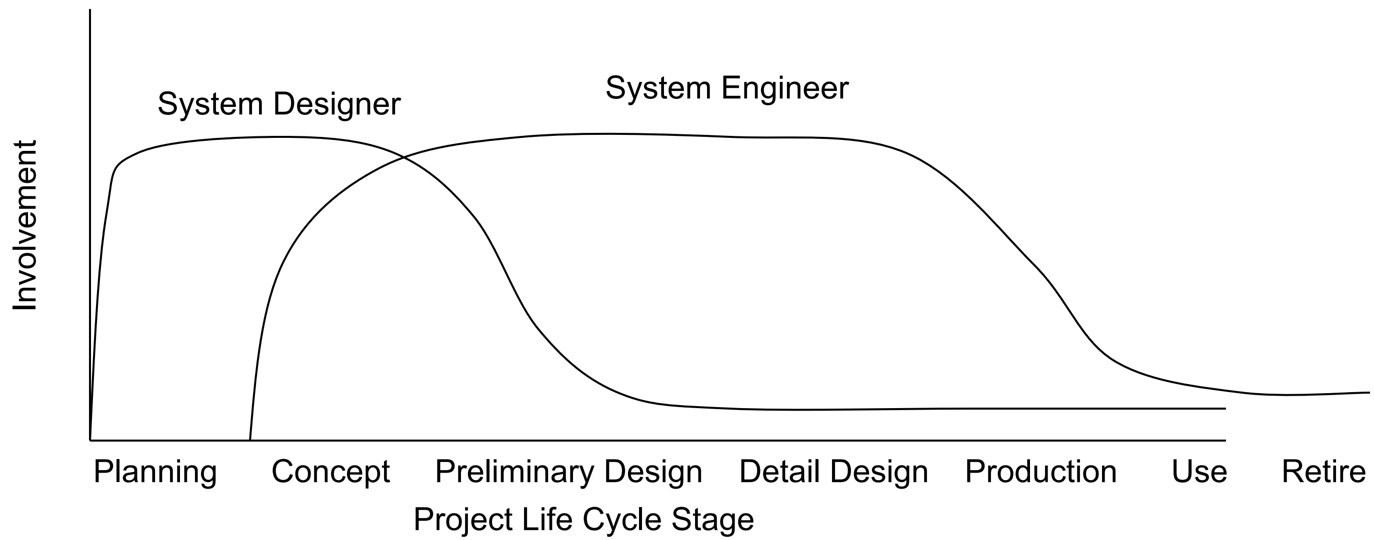
1.5.1 When does it occur in the product lifecycle?



- According to the standards, it should occur during the concept phase
- But on new products, the concept phase has not yet been contracted, so it actually occurs in the planning phase, or bid preparation phase

1.5 System Design - Overview

1.5.1 When does it occur in the product lifecycle (cont.)?



- The system design activity does reduce sharply after the concept phase, but it is never completely phased out

1.5 System Design - Overview

1.5.2 Who does system design?

- On new products
  - The marketer and client
  - The system designer (if one exists)
  - The programme manager and client
- Upgrades/modification to existing products
  - The system engineer
- The system designer rarely works alone, he gets his work done through a team of specialists. Two styles appear to be successful:
  - The benevolent dictator, who must be an extremely competent, and knowledgeable person
  - The democrat, who leads the team to get to the solution

1.5 System Design - Overview

1.5.3 Objectives of system design

- To create a competitive product system that
  - a) meets the market's performance requirements and
  - b) at a market driven price
- That is, to create system products that sell
- Remember!
  - The price of a product is determined by the market
  - The cost of the product is largely determined by the system design
  - The difference is the profit!

1.6 Systems Architecting

1.6.1 System Architecting (Eberhardt Rechtin)

- Systems Architecting = **Architect**ure + Engineer**ing**

- General definition: " An architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution", it's a model!

  IEEE STD 1471-2000

- Systems architecting is creating and building systems. It is both an art and a science – synthesis and analysis, induction and deduction, and conceptualisation and certification – using guidelines from its art and methods from its science.

1.6 Systems Architecting

1.6.1 System Architecting (Continued)

- As a process it is distinguished from standard system engineering in its greater use of heuristic reasoning, lesser use of analytics, closer ties to the client, and a particular concern with certification of readiness for use.

- The foundations of system architecting are a systems approach, a purpose orientation, a modelling methodology, ultra-quality, certification and insight.

- System architects need a variable depth of understanding over a wide range of disciplines. This can only be achieved with experience.

1.6 Systems Architecting

1.6.2 Heuristics

- Heuristics – using or obtained by exploration of possibilities rather than by following set rules (Collins)
- *The art in architecting lies not in the wisdom of the heuristics, but in the wisdom of knowing which heuristics apply, a priori, to the current project*
- The top 4 Heuristics:
  - Don’t assume that the original statement of the problem is necessarily the best, or even the right one
  - In partitioning, choose the elements so that they are as independent as possible; that is, elements with low external complexity and high internal complexity
  - Simplify. Simplify. Simplify
  - Build in and maintain options as long as possible in the design and implementation of complex systems. You will need them.

1.6 Systems Architecting

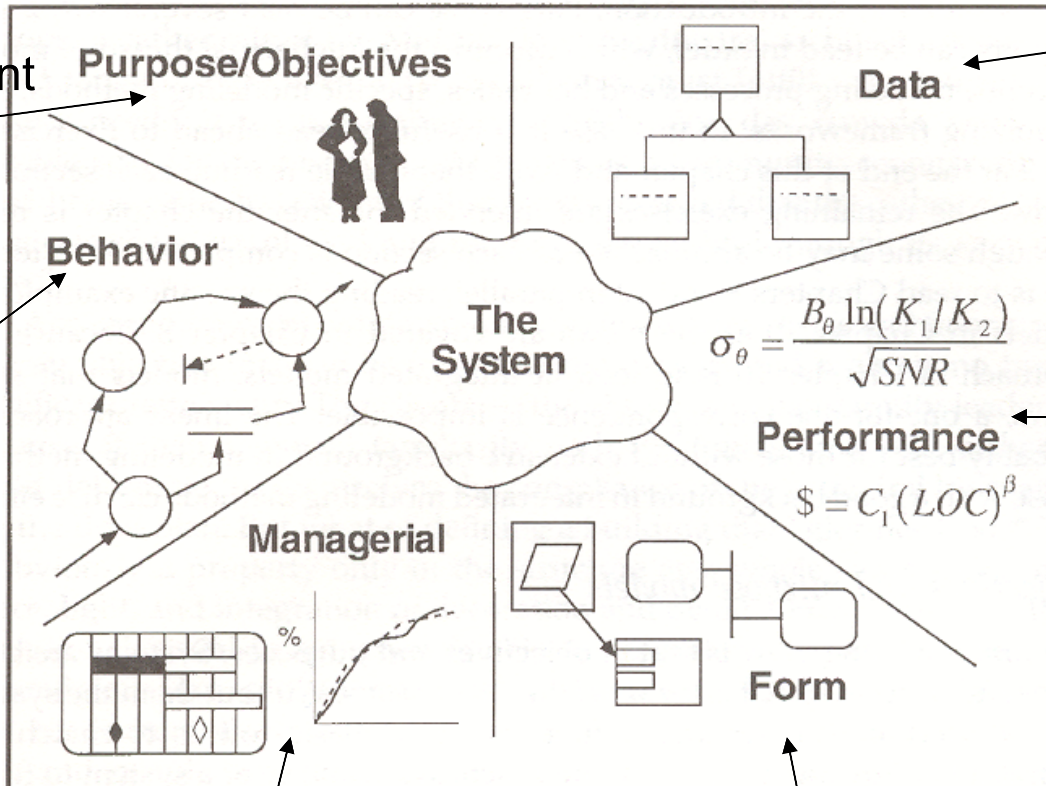
1.6.2 Heuristics (Continued)

- When is a rule an heuristic?
- For instance: *Look before you leap* vs. *He who hesitates is lost*.
- The rules to determine a heuristic are:
  - The heuristic must make sense in its original domain or context.
  - The general sense of the heuristic should apply beyond the original context
  - The heuristic should be easily rationalized
  - The opposite statement of the heuristic should be foolish, clearly not common sense
  - The heuristic should have stood the test of time
- Example Murphy's law: *If it can go wrong it will*.

1.6 Systems Architecting

1.6.3 Models

- Classification of models by views



What the client wants

The information retained in the system and its interrelationships

What the System does

How effective the System does it

The process of construction and management

What the system is

1.6 Systems Architecting

1.6.4 Model Example:

- DODAF (Department of Defence (USA) architectural framework) that consists of 26 products (views):
  - All views
    - AV-1 Overview and summary of information
    - AV-2 Integrated dictionary
  - Operational View
    - OV-1 High level operational concept graphic
    - OV-2 Operational node connectivity descriptions
    - OV-3 Operational information exchange matrix
    - OV-4 Organizational chart
    - OV-5 Operational activity model
    - OV-6 Operation activity state, sequence and timing descriptions
    - OV-7 Logical data model

1.6. Systems Architecting

1.6.4 DODAF (Continued…)

- Systems view
  - SV-1 Systems interface description
  - SV-2 Systems communications description
  - SV-3 Systems – systems matrix
  - SV-4 Systems functionality description
  - SV-5 Operational activity to systems function traceability matrix
  - SV-6 Systems data exchange matrix
  - SV-7 Systems performance parameters matrix
  - SV-8 Systems evolution description
  - SV-9 Systems technology forecast
  - SV 10 b,c Systems state transition and sequence / timing descriptions
  - SV 11 Systems physical schema
- Technical view
  - TV-1 Technical standards profile
  - TV-2 Technical standards forecast

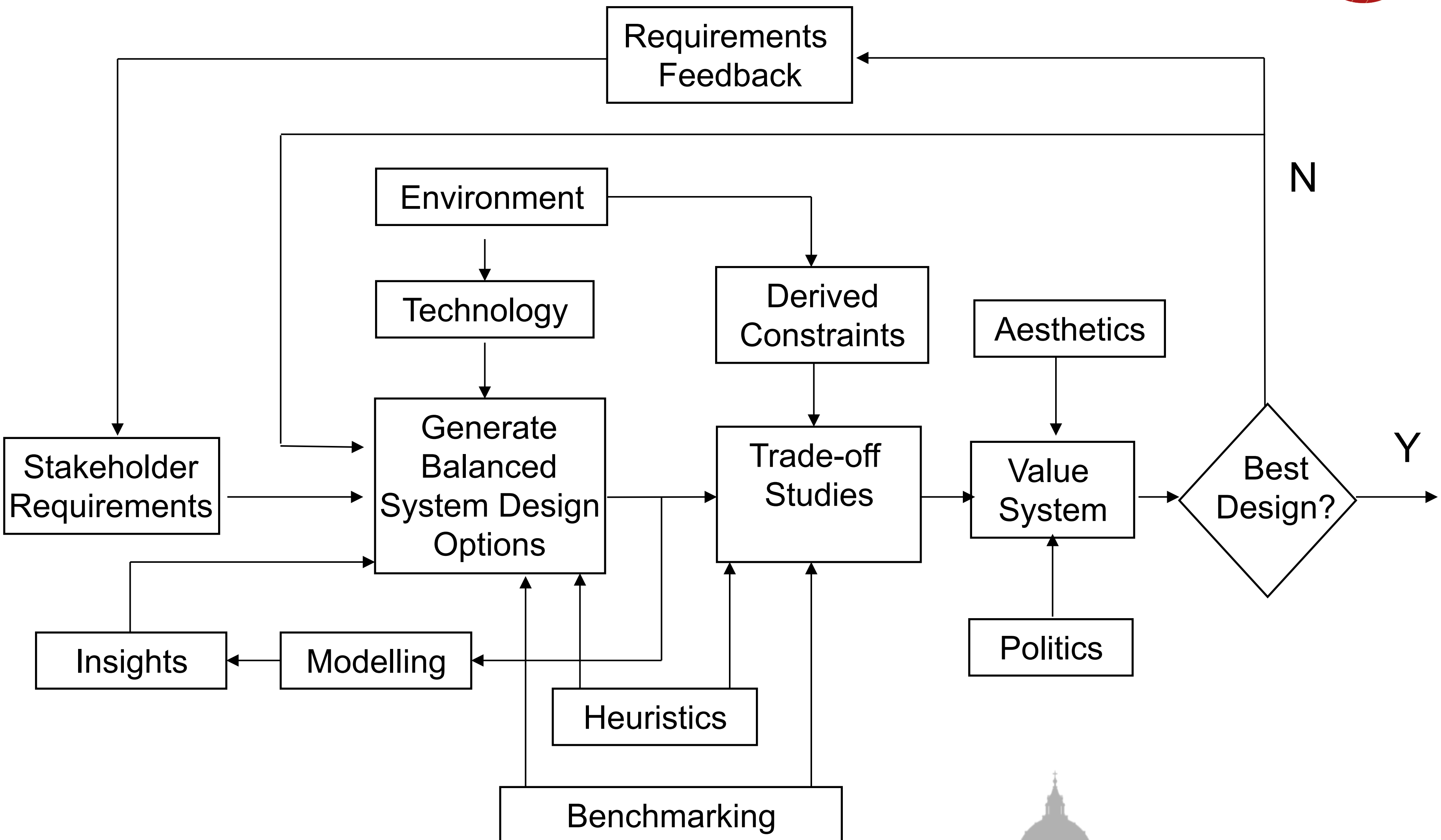
Tutorial Outline

1. INTRODUCTION – Systems and Theory (Session 1 10h00-12h00)
    - Tutorial purpose and content
    - Systems theory and systems philosophy
    - Systems thinking
    - System engineering & design as part of systems theory
        - EIA 632 – Requirements processes
        - EIA 632 – Solutions processes
    - System design in the product lifecycle
    - Systems architecting
2. SYSTEM DESIGN - Theory (Session 2, 13h30-15h00)
    - System design synthesis process
    - Methods

2.4 System Design Process

- The following processes have been identified that contributes to system design:
  - Stakeholders Requirements Identification
  - Generate balanced system design options
    - Benchmarking
    - Heuristics
    - Environment
    - Technology
    - Modelling and insights
  - Trade-off studies
    - Benchmarking
    - Heuristics
    - Value system
    - Uncommunicated values – politics and aesthetics

2.4 System Design Process



2.4 System Design Process

- Stakeholders Requirements Identification
  - The stakeholder's requirement identification process is well described in System Engineering texts and includes:
    - Requirements analysis
      - Mission analysis
      - Functional analysis
    - Quality function deployment (house of quality and design dependent parameters)
    - Etc.

2.4 System Design Process

- Generate balanced system design options
  - Benchmarking
    - Benchmarking is the startingpoint of the system design process, one needs to know how well existing products/systems meet the stakeholders requirements and determine the gap between what is currently on offer and what is required.
  - Heuristics
    - Heuristics are abstractions of experience, they are rules that have been learnt over time and can be applied in a general way
    - They can tell a designer what he should do, and also what he should **not** do.
  - Environment
    - The environment determines the scope of the system, and can be used to determine the boundaries of the system design

2.4 System Design Process

- Generate balanced system design options (continued)
  - Technology
    - Technology is the most powerful process that can be used by the system designer to address the requirements gap.
    - The technology either needs to be developed, adapted or may be available off the shelf. Development can only proceed once the technology used is mature.
    - Technology can become a crutch for system design, an elegant design, utilizing existing technologies in an innovative fashion may be a much better solution than the so-called state of the art solution.
  - Modelling and insights
    - It is important for the system designer to develop a model of what he is doing, this can be a conceptual framework (mental model), a physical model, a mathematical model.
    - By constructing a model, the interrelationships in the system and its environment is better understood, and this can lead to system design insights.

2.4 System Design Process

- Trade-off studies
  - Benchmarking
    - Compare design to existing designs, if not better, why use it?
  - Heuristics
    - Does the design break common sense design rules?
    - Can one describe the system design's logic, based on past experience?
  - Value system
    - How well does the design support the client's value system?
  - Uncommunicated values – politics and aesthetics
    - Politics always wins, don't use political incorrect technologies
    - Aesthetics is important!

2.4 System Design Process

Aesthetics Example:
Joint Strike Fighter

2.5. System Design Synthesis Methods

2.5.1 Requirement Analysis

- Operational Analysis
- Operations Research
- Functional Analysis
- UML-SYS

2.5.2 Synthesis methods

- Synthesis process block diagram
- Technology analysis – Rias van Wyk
- Theory of Constraints – Eliyahu Goldratt
- Genetic algorithms – Scientific American article

3.3 Evaluation methods

- User feedback
- Modeling and Simulation
- Decision making models
- Economic evaluation models

2.5.1 Requirement Methods

2.5.1.1 Operational Analysis

- Mission analysis and threat analysis
- Storyboards - “A day in the life of the operational system”
- Scenarios
- Use case data sheets

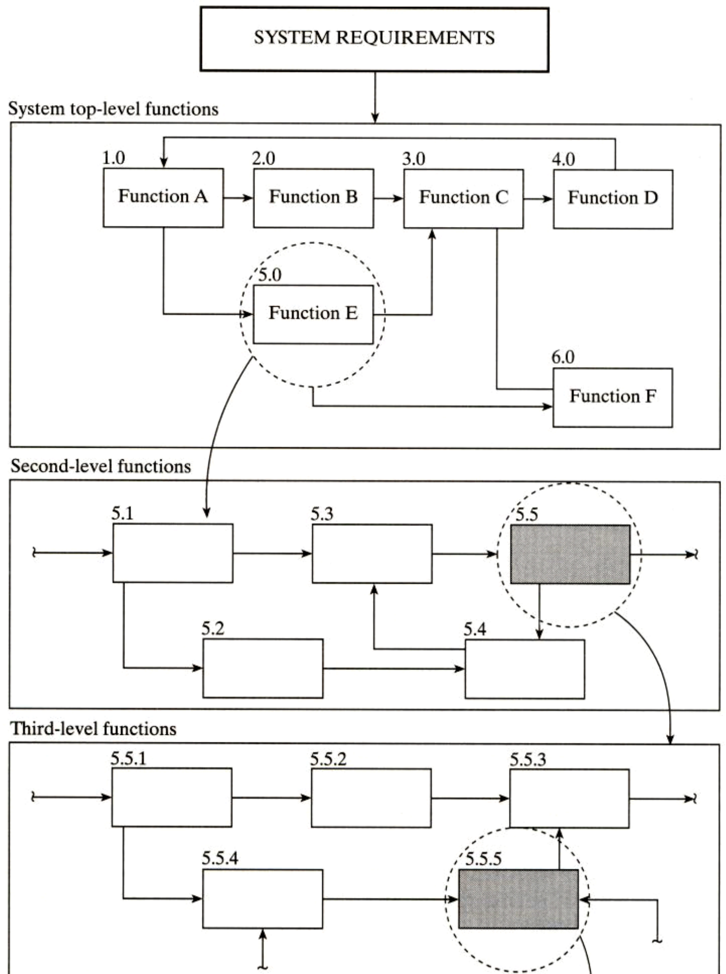
2.5.1. Requirement Methods

2.5.1.2 Operations Research

- Analysis
  - Weapons effort studies
  - Error budgets
  - Decision models
  - Etc.
- Monte-Carlo Modelling
  - One on one
  - Many on many
  - Realistic environmental and threat modelling

2.5.1. Requirement Methods

2.5.1.3 Functional Analysis and allocation



2.5.1. Requirement Methods

2.5.1.4 Object modelling and UML for Requirements

- Object modelling describes the system from a number of views:
  - Use Case view with use case diagrams
  - Object view with collaboration and statechart diagrams
  - Dynamic view with sequence diagrams
  - Function view with activity diagrams
  - See OMG Unified modelling language specification

- This is a more powerful tool than functional analysis?

- See:
  - Engineering complex systems *with Models and Objects by* David W. Oliver, Timothy P. Kelliher and James G. Keegan, Jr.
  - Telelogic whitepaper “Using UML2 to solve system engineering problems”
  - OMG Unified modelling language specification for systems – SYSML

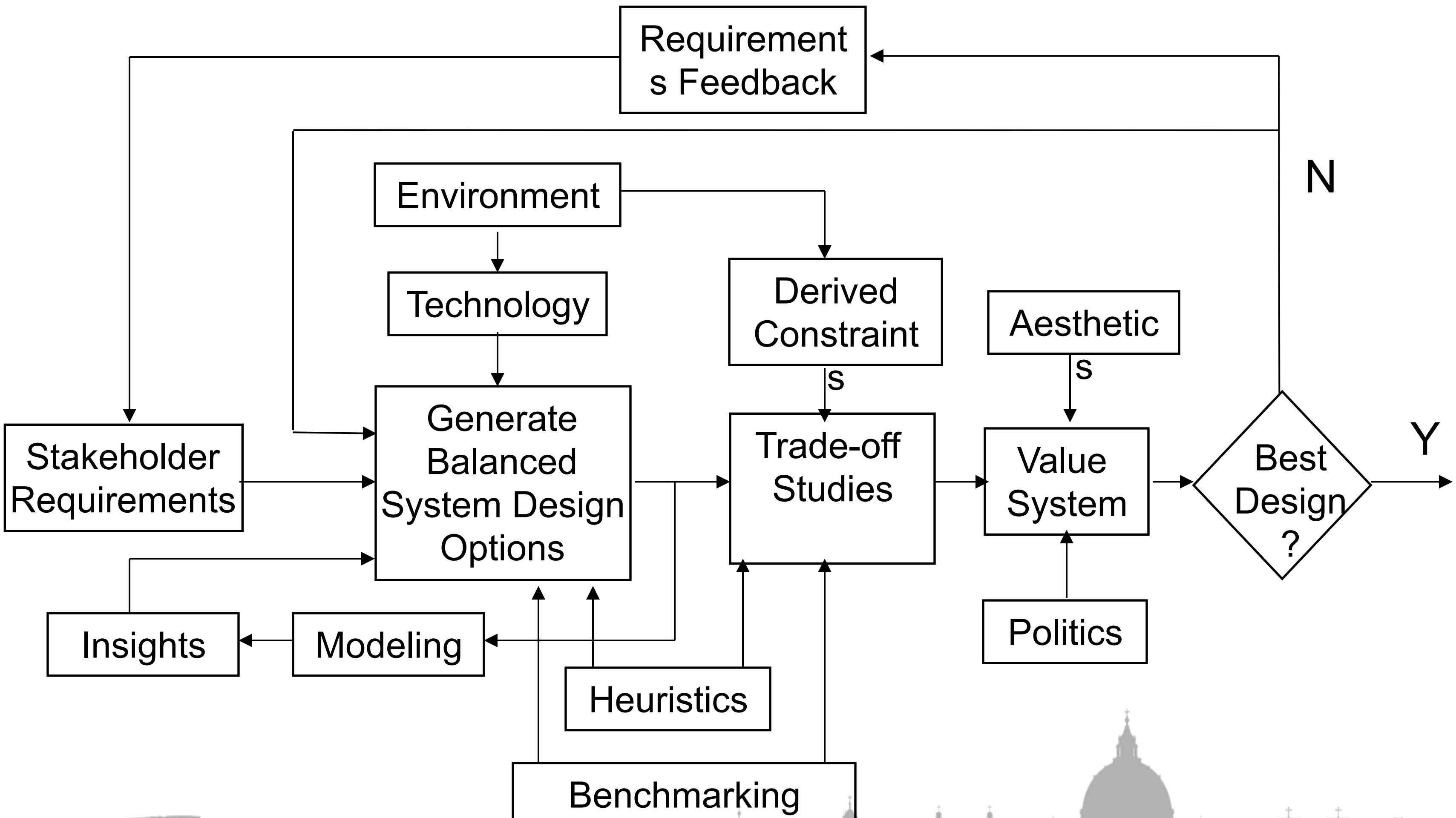
2.5 System Design Methods

2.5.2 Synthesis methods

- Synthesis process block diagram
- Technology analysis – Rias van Wyk
- Theory of Constraints – Eliyahu Goldratt
- Genetic algorithms – Scientific American article

2.5.2.1 System Synthesis

System Synthesis Process Diagram



2.5.2.2 Technology

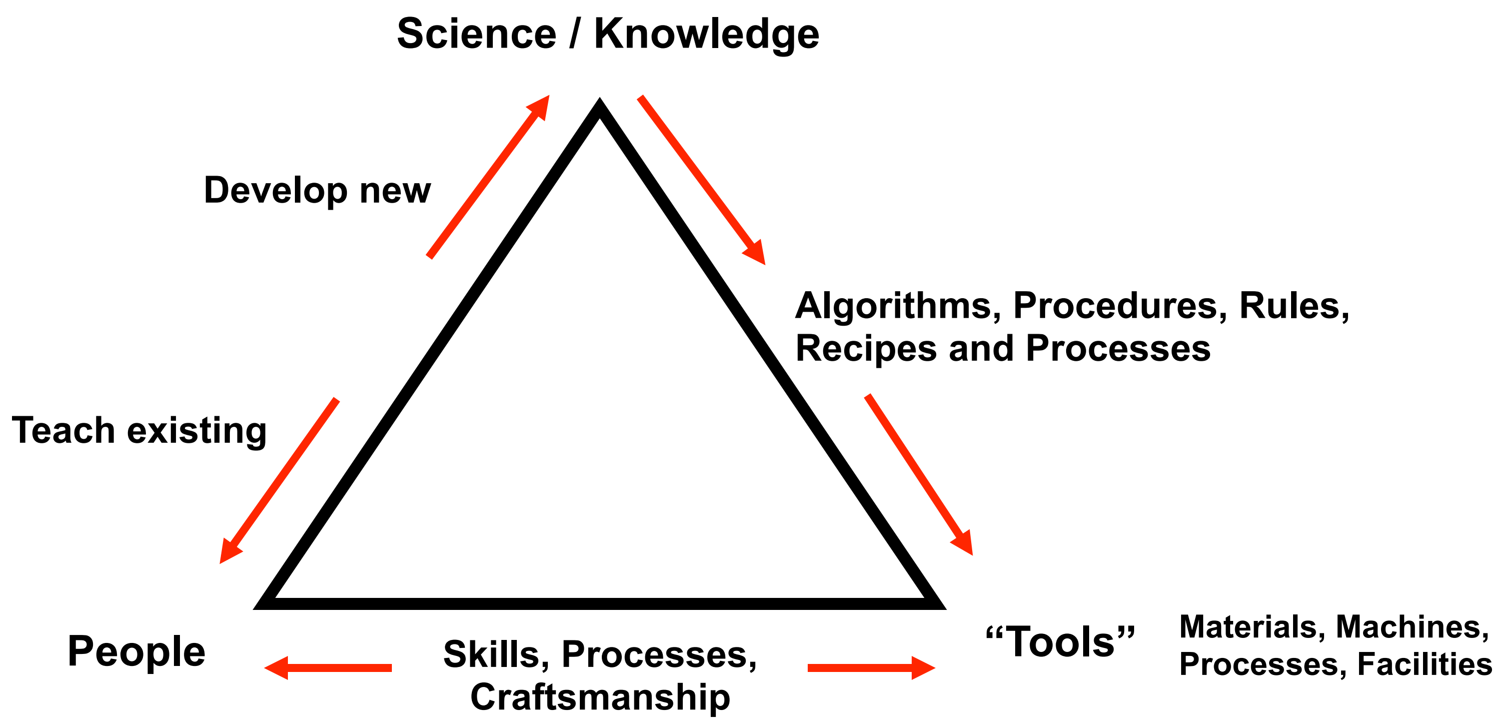
Technology

- What is technology?
  - Technology is a set of means created by people to facilitate human endeavour. In the briefest possible terms technology may be viewed as “created capability”
  - The emphasis is on “means”, this is the essence of technology, it is not an end in itself
  - “Created” – Technology is not natural, it is made by people
  - The size of the “set of means” can be limited or universal, depending on the focus
  - “Facilitate human endeavour”. This mean to enhance human performance or enable task beyond human capacity

2.5.2.2 Technology

Technology (definition cont.)

- Graphic



2.5.2.2 Technology

Technology Analysis

- Five tools are required to analyse technology:
  - A standard format for viewing and describing technologies
  - A classification of technologies
  - A cascade of trends describing technological change
  - A chart of technological breakthrough zones
  - A profile of social preferences with respect to technology

2.5.2.2 Technology

Technology Standard Format

- The description responds to six questions:
  - What does the technological entity do- what is its function?
  - How does it do it – what is the principle of operation?
  - How well does it do it – what is the level of performance?
  - What does the technological entity look like – what is its structure?
  - What is it made of – from what materials?
  - How big is it – what is its size?

2.5.2.2 Technology

Technology Classification

- The nine cell Functional Classification:

	Processor	Transporter	Store
Manipulators of Matter			
Manipulators of Energy			
Manipulators of Information			

2.5.2.2 Technology

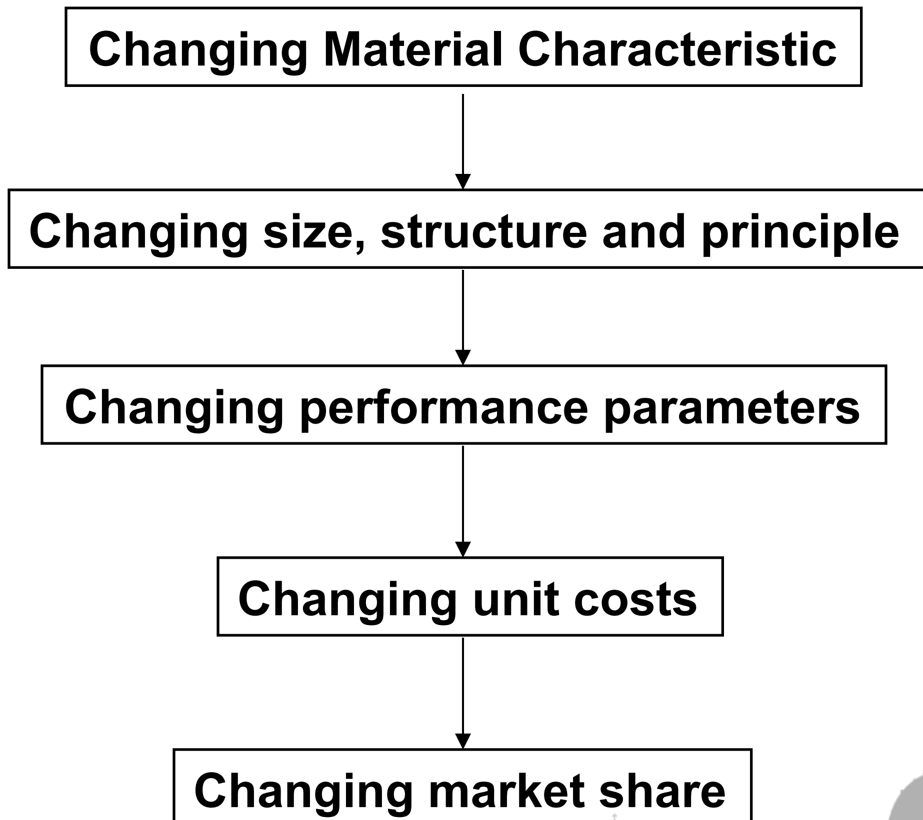
Technology Classification (Cont.)

- Use of the nine cell Functional Classification:
  - To Structure a technology audit and to classify core technologies in an organisation
  - To structure a technology scan, i.e. to provide a basis for reviewing emerging technologies in the global technologies environment
  - To study interactions between various technologies
  - To provide an overview of the portfolio of projects of a research organisation
  - To help brainstorm possible technology solutions

2.5.2.2 Technology

Tracking Technological Change

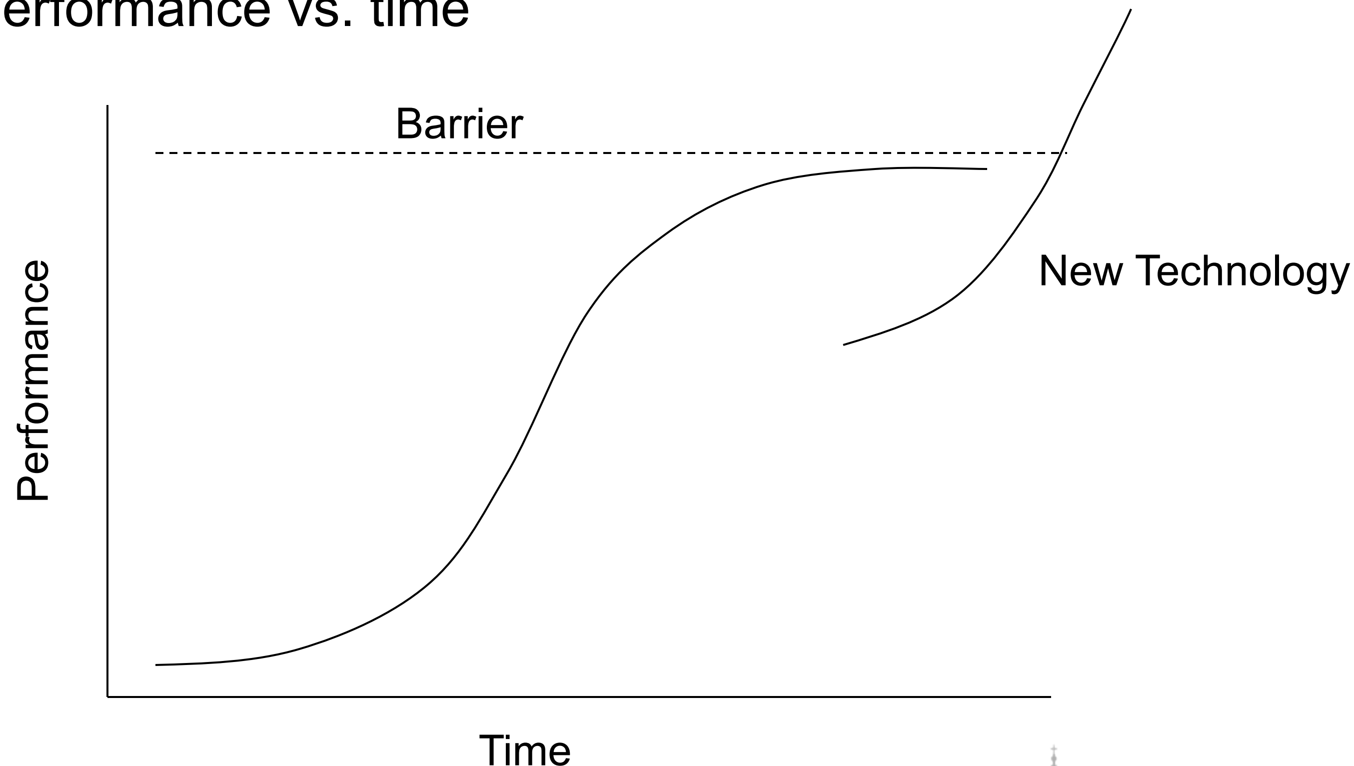
- The cascade approach to viewing technological change:



2.5.2.2 Technology

Tracking Technological Change (Cont.)

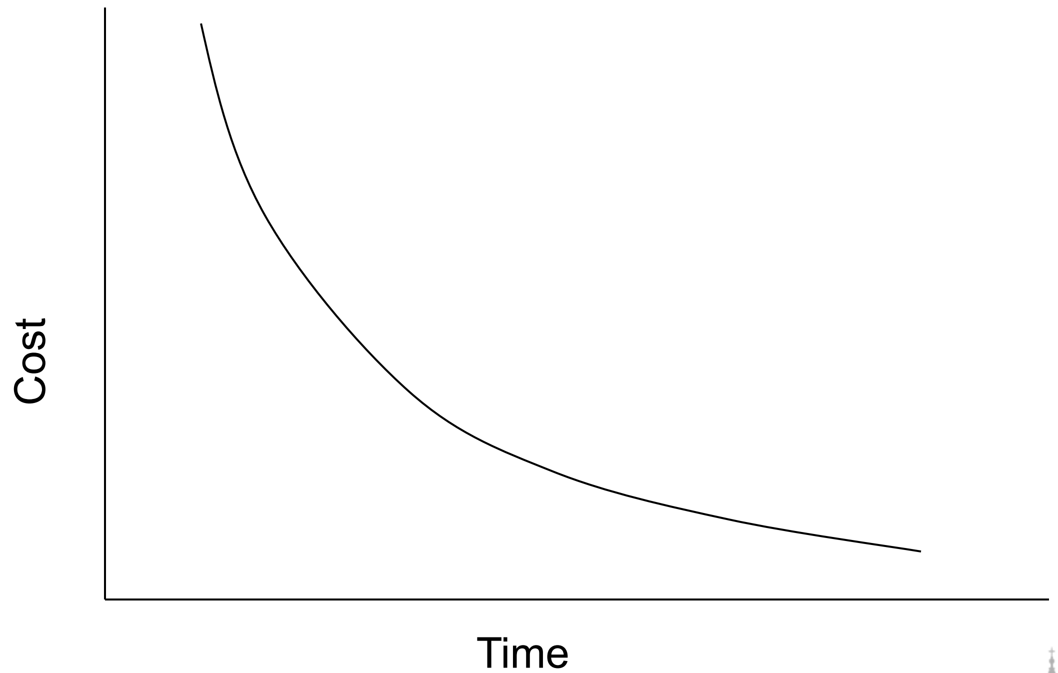
- Plotting Technological change
  - Performance vs. time



2.5.2.2 Technology

Tracking Technological Change (Cont.)

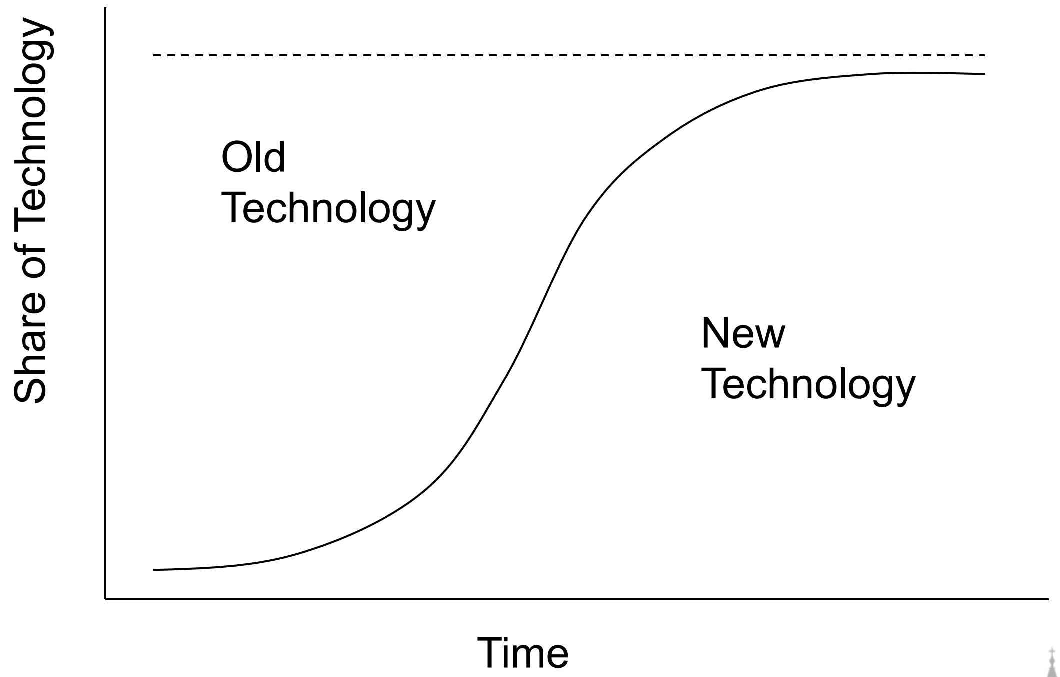
- Plotting Technological change
  - Cost Curve



2.5.2.2 Technology

Tracking Technological Change (Cont.)

- Plotting Technological change
  - Substitution Curve $f=q/(1-q)$



2.5.2.2 Technology

Identifying technology interactions

- Various technologies can come together to produce new complex technological systems far more advanced than its predecessors.
- Type of interactions
  - Contingent
    - The one technology is intimately dependent on the other
  - Supplementary
    - Changes in one technology affect the other in the same direction
  - Independent
    - There is no link
  - Competitive
    - The one technology competes with the other and could replace it

2.5.2.2 Technology

Social Preferences and technology

- Socio technological interaction
  - Allergy – rejection by society
  - Deviation – partial acceptance by society, with restructuring
  - Enforced penetration – by a powerful agent of change
  - Synergy – enthusiastically accepted by society
- Embedded social values
  - Safety
  - Health
  - Energistics – renewable is preferred
  - Ecology – the technology must contribute to sustainability
  - Entropics – minimal contribution to global entropy
  - Economics

2.5.2.3 System Design Method

Theory of Constraints (Eliyahu Goldratt)

- Applied to System Design
  - It gives a procedure that can be used to improve an existing system design
  - It provides a technique which promises to minimise compromises in the design of a system
  - It focuses on cause-effect relationships which is important for the system design

2.5.2.3 Theory of Constraints

Theory of Constraints

- The process starts by defining the goal of the system:
  1 Identify the system's constraints
  2 Decide on how to exploit the system's constraints
  3 Subordinate everything else to the above decision
  4 Elevate the system's constraints (change the design)
  5 If in the previous steps a constraint has been broken, go back to 1

2.5.2.3 Theory of Constraints

3.2.3.3 Theory of Constraints

- The process of Change:
  - What to Change?
    - Pinpoint the core problems!
  - To what to change to?
    - Construct simple, practical solutions!
  - How to cause the change?
    - Induce the appropriate people to invent such solutions!
    - This is called the Socrates method and uses questions to define the problem in such a way that the desired solution will form in the decision maker's own mind. It ensures ownership of the solution.
    - It is only required if the inventor does not have the authority to implement design decisions, which is typically the case for functional area managers reporting to the system engineer.

2.5.2.3 Theory of Constraints

3.2.3.4 Theory of Constraints

- To prove effect-cause-effect logic:
  - This is a scientific method to find the root cause of the problem

    Common sense is the highest praise for a logical derivation, for a very clear explanation
  - Speculate a cause for a given effect and then predict another effect stemming from the same cause, and testing the hypothesis, is called effect-cause-effect
  - In systems thinking we have to think cause-effect and Mill's methods give us a powerful, razor sharp inference rules by which scientifically valid inductions can be made.
  - Once a root cause of a problem has been identified, a solution process can be initiated. One such a process is the "Evaporating Clouds"

2.5.2.3 Theory of Constraints

3.2.3.5 Evaporating Clouds

- A method to invent simple solutions
  - Whenever we face a situation which requires a compromise, there is always a simple solution that does not involve compromise.
  - The tyranny of "or", and the magic of "and"
    - God does not limit us, we are limiting ourselves (no compromise)
    - You can't have your cake and eat it (need to compromise)
    - Define a problem precisely and you are halfway to the solution
  - Example:

Objective	Requirement	Prerequisite
Reduce cost per unit ←	Reduce setup cost per unit ←	Large Batch
		↕ Conflict!
Reduce cost per unit ←	Reduce carrying cost per unit ←	Small Batch

2.5.2 System Design Methods

2.5.2.4 Genetic Algorithms

- A method to evolve system design solutions
  - Based on evolutionary theory using models of DNA, breeding, mutation and population with a natural selection process.
  - The design parameters that have to be evolved are coded as binary numbers that are strung together, each number becoming a gene in the “DNA”
  - These numbers are started randomly, for an arbitrary population size (say 50-100). The population then “breeds” randomly and the DNA is randomly crossed over for each pair. Some mutation may also occur by flipping bits randomly. The next generation is thus borne. They are tested against a fitness measure and ranked. Only the top, say 50% survive. The cycle is then repeated for many generations until an acceptable design is generated.

INCOSE International Symposium

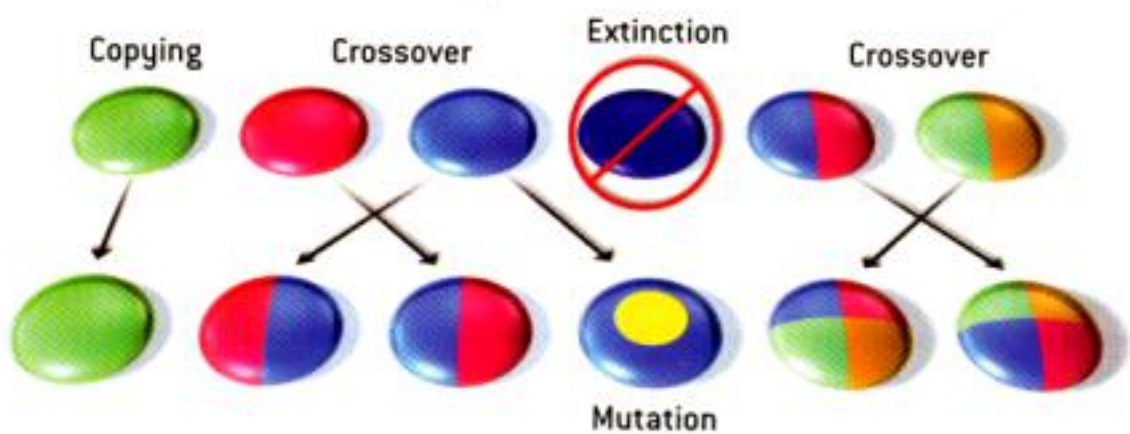
Genetic Algorithms Example

- Evolving inventions – Scientific American Feb 2003.

UNNATURAL SELECTION

Evolutionary Processes

THREE PROCESSES propagate "organisms" (*represented here by colored disks*) from one generation to the next in a genetic programming run. Some of the better organisms are copied unaltered. Others are paired up for sexual reproduction, or crossover, in which parts are swapped between the organisms to produce offspring. A small percentage are changed randomly by mutation. Organisms not chosen for propagation become extinct. The crossover operation is applied more frequently than copying and mutation because of its ability to bring together new combinations of favorable properties in individual organisms.

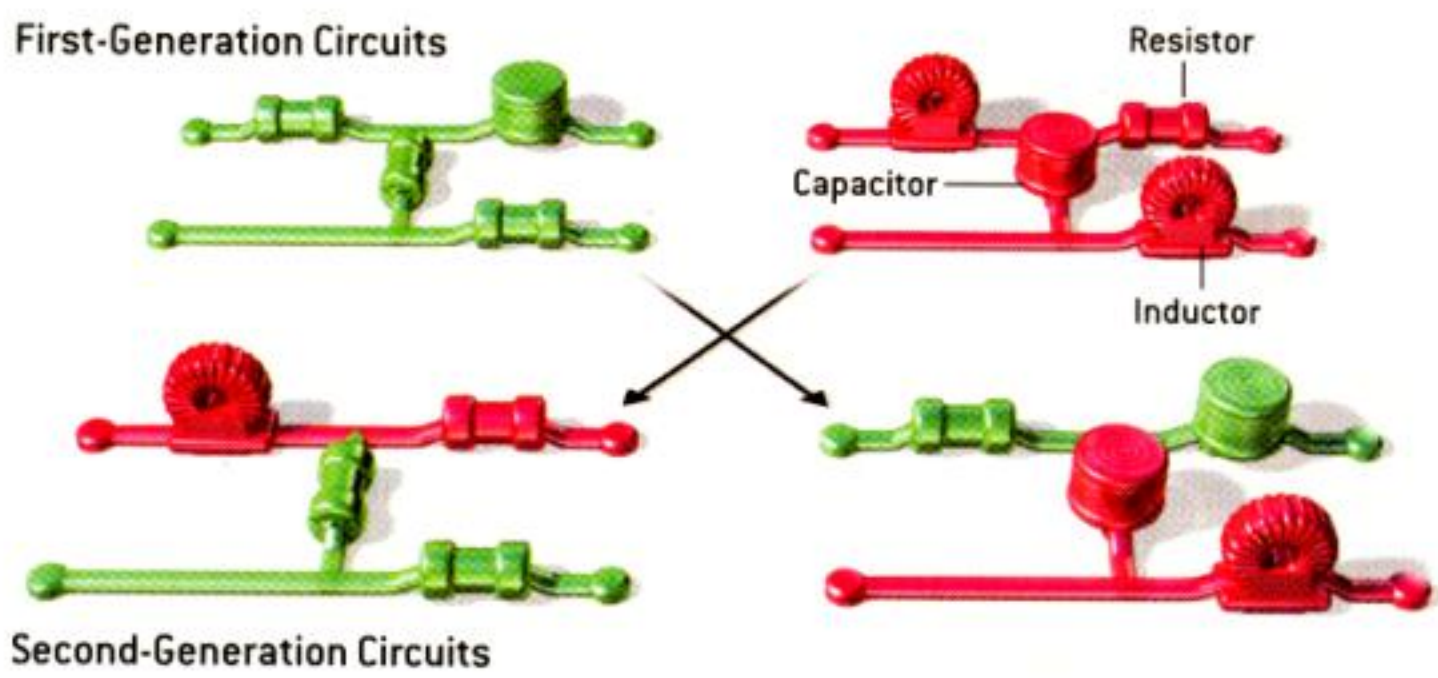


2.5.2.4 Genetic Algorithms

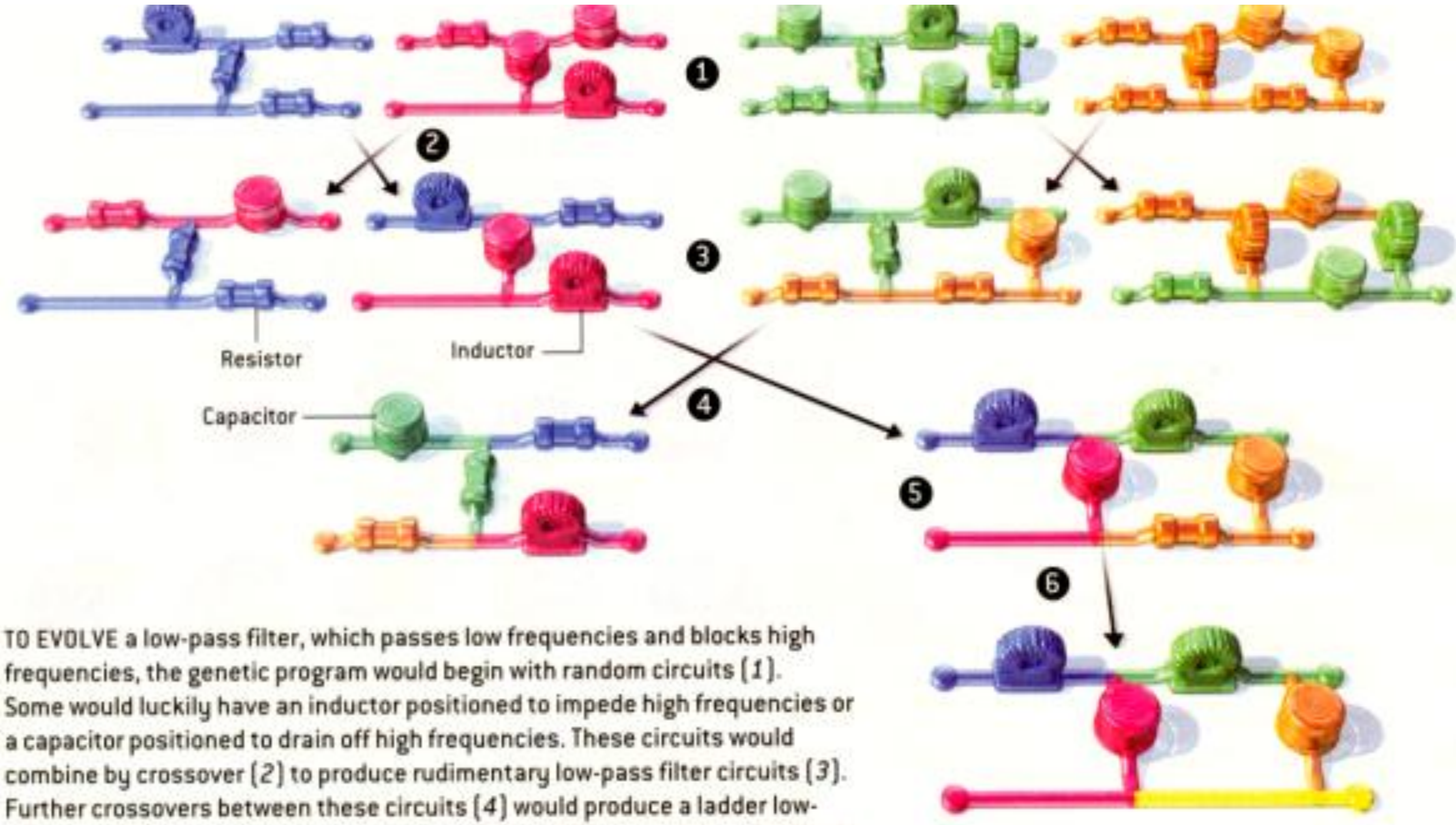
Evolving inventions (Cont.)

Crossover of Electronics

ACTING ON electronic circuits, the crossover operation takes two circuits and swaps some of their components, producing two new circuits.



2.5.2.4 Genetic Algorithms

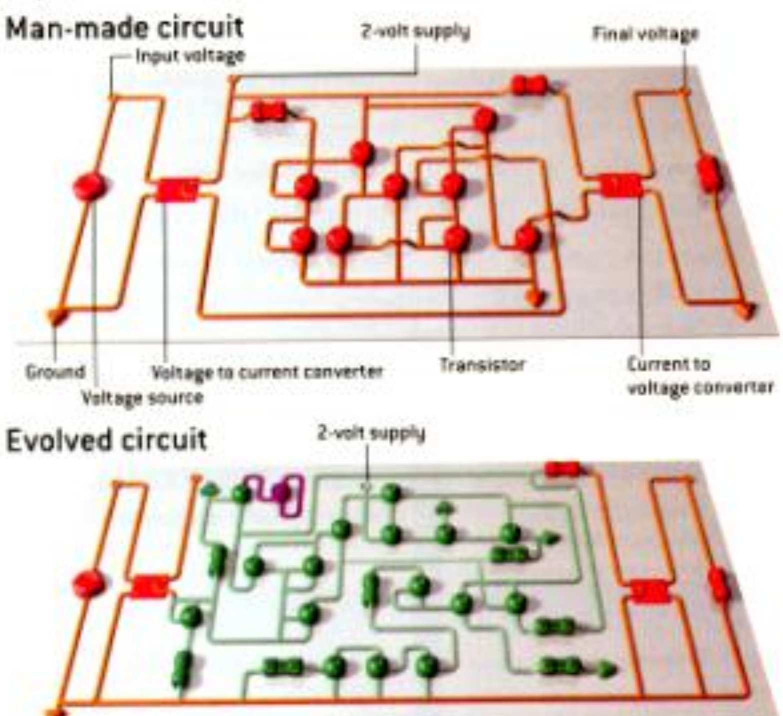


TO EVOLVE a low-pass filter, which passes low frequencies and blocks high frequencies, the genetic program would begin with random circuits (1). Some would luckily have an inductor positioned to impede high frequencies or a capacitor positioned to drain off high frequencies. These circuits would combine by crossover (2) to produce rudimentary low-pass filter circuits (3). Further crossovers between these circuits (4) would produce a ladder low-pass filter (5). Mutations (6) would eliminate superfluous resistors and would fine-tune the values of the components.

2.5.2.4 Genetic Algorithms

HUMAN VERSUS COMPUTER

THE TWO CIRCUITS shown below are both cubic signal generators. The upper circuit is a patented circuit designed by a human; the green and purple parts of the lower circuit were evolved by genetic programming (the other parts are standard input and output stages). The evolved circuit performs with better accuracy than the human-designed one, but *how* it functions is not understood. The evolved circuit is clearly more complicated but also contains redundant parts, such as the purple transistor, that contribute nothing to its functioning.



2.5 System Design Methods

2.5.3 Evaluation methods

- User feedback
- Heuristics
- Modeling and Simulation
- Decision making models

2.5.3 Evaluation Methods

2.5.3.1 User feedback

- If one has to decide on options that have a relative small effect on system performance or cost, it is easiest to ask the user directly what he prefers.
- If there is more than one user, one may get more than one answer! What then? It is better not to ask the user if there is multiple clients.
- Remember, asking advice from the user creates expectations!

2.5.3 Evaluation Methods

2.5.3.2 Heuristics

- Heuristics can also be used to trade-off options.
  - For instance, the heuristic that elements should be chosen to be as independent as possible can be used to decide which option is better.
- It is especially the heuristics that have been developed inside the organisation, through many years of experience, that will have the best application to trade-off analysis:
  - "We have tried option A before and it gave problems"
  - These heuristics can also be untrue, the reason for the problem or failure must also be given.
  - "We have tried option A before and it gave problems because we could not control process B under conditions C"
  - If process B and conditions C differ in this application, or there are technological fixes for process B, then the system design option may be valid, and should not be discarded on heuristic grounds.

2.5.3 Evaluation Methods

2.5.3.3 Modeling and simulation

– Hierarchy of models
  - Simple models (to be used exceedingly carefully)
    – *Far each complex problem there is an intuitive, simple solution that is completely wrong!*
  - Intermediate complexity models
  - Full, high fidelity models (rarely used for system design)

– Simulation
  - Used to get relative results to enable the designer to choose the better option.
  - The model has not been validated to predict absolute results.

2.5.3 Evaluation Methods

2.5.3.4 Decision models

– The decision evaluation matrix

Pj / Fj / Ai	P1	P2	…	Pn
	F1	F2	…	Fn
A1	E11	E12	…	E1n
A2	E21	E22	…	E2n
…				
Am	Em1	Em2	…	Emn

Where Ai = an alternative available for selection by the decision maker
Fj = a future not under the control of the decision maker
Pj=the probability that the jth future will occur
Eij = evaluation measure associated with ith alternative and jth future

2.5.3.4 Decision Models

Decision models – Evaluation matrix example

– Decisions under risks. The evaluation measure is Profit in k$

Probability	(0.3)	(0.2)	(0.5)
Future	C1	C2	C1+C2
A1	100	100	400
A2	-200	150	600
A3	0	200	500
A4	100	300	200
A5	-400	100	200

Where Ai = an alternative available for selection by the decision maker

- Problem: which alternative to choose?

2.5.3.4 Decision Models

Evaluation matrix example

- A5 does not provide any advantage for any of the futures and can be dropped from consideration
- The decisions under risk now depends on the following criteria of the decision maker:
  - Aspiration level (Say loss of not more than 100k and a profit of more than 400k) criterion implies A1 and A4 can both be selected
  - Most probable future criterion implies A2
  - Expected value criterion, weigh the profits and losses by the probability of the possible future:
    - » A1: 100(0.3) + 100(0.2) + 400(0.5)=250
    - » A2: -200(0.3) + 150(0.2) + 600(0.5)=270
    - » A3: 0(0.3) + 100(0.2) + 500(0.5)=290
    - » A4: 100(0.3) + 300(0.2) + 200(0.5)=190

  Implies A3 is the best solution

2.5.3.4 Decision Models

Decision models – evaluation matrix example

- If all criteria are taken together, option A3 occur more often than any other
- Decisions taken under uncertainty (that is when it is not possible to assign a probability to the future outcomes).
  - Laplace criterion:
    - Take an average of the possible futures, that is assume equiprobability
      - » A1: 100(0.33) + 100(0.33) + 400(0.33) = 200
      - » A2: -200(0.33) + 150(0.33) + 600(0.33) = 183
      - » A3: 0(0.33) + 100(0.33) + 500(0.33) = 233
      - » A4: 100(0.33) + 300(0.33) + 200(0.33) = 200

    Implies A3 is the best solution

2.5.3.4 Decision Models

Decision models – evaluation matrix example

- Maximin and Maximax criterion:
  - Maximin rule

    $$Payoff = \max_i \left\{ \min_j E_{ij} \right\}$$

    - » A1: 100
    - » A2: -200
    - » A3: 0
    - » A4: 100
  - Maximax rule

    $$Payoff = \max_i \left\{ \max_j E_{ij} \right\}$$

    - » A1: 400
    - » A2: 600
    - » A3: 500
    - » A4: 300
  - The conservative decision maker selects the best of the worst possible outcomes of each of these rules. That is A4.

2.5.3.4 Decision Models

Decision models – evaluation matrix example

- Hurwicz criterion:
  - Select a level of optimism $0 \leq \alpha \leq 1$
  - Compute

$$Payoff = \max_i \left\{ \alpha \left[\max_i E_{ij} \right] + (1-\alpha) \left[\min_j E_{ij} \right] \right\}$$

  - E.g. for α=0.2
    - » A1: 160
    - » A2: -40
    - » A3: 100
    - » A4: 140
  - Plotting vs. α gives insight. For α around 0.5 to 0.7 A1, A2 and A3 gives similar results. A4 looks bad for all α. At α =0 and α =1 the rule reverts to the Maximin and Maximax rule respectively.
- Comparing decision rules show that not option gets favoured, the decision still relies on the value system of the decision maker, there is no getting around this.

Tutorial Outline

3. SYSTEM DESIGN – Example (Session 3, 15h30-17h00)
   - Example Exercise background
   - Breakout session into groups (15h30-16h30)
   - Feedback from groups 16h30-16h50
   - Discussion and Summary 16h50-17h00

Example

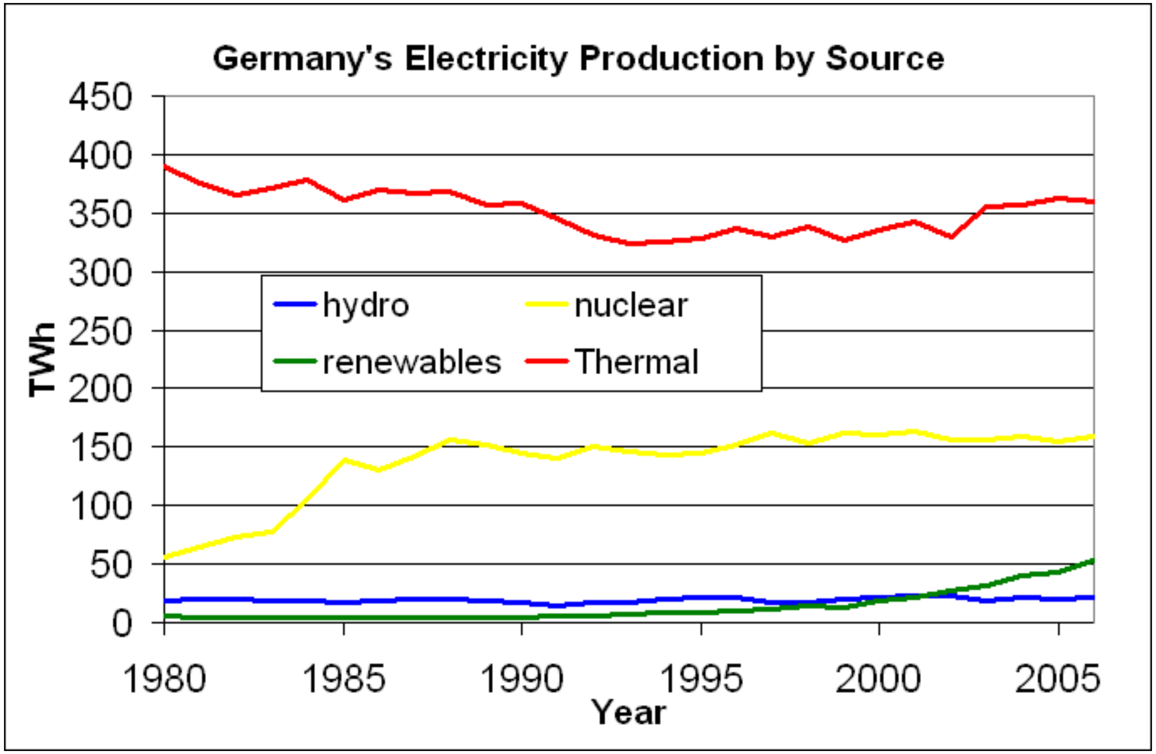
- The System Design Process given is not intended for trivial problems. The example we are going to work with is a real world, complex problem!
- The problem is the

'German electrical power supply gap problem'

- On 30 May 2011, the German government announced a plan to shut all nuclear reactors by 2022
- The reasons are complex, mainly political, and in response to the Fukushima disaster.
- That is 17 Nuclear power stations of which 8 have already been shutdown at the time of the announcement

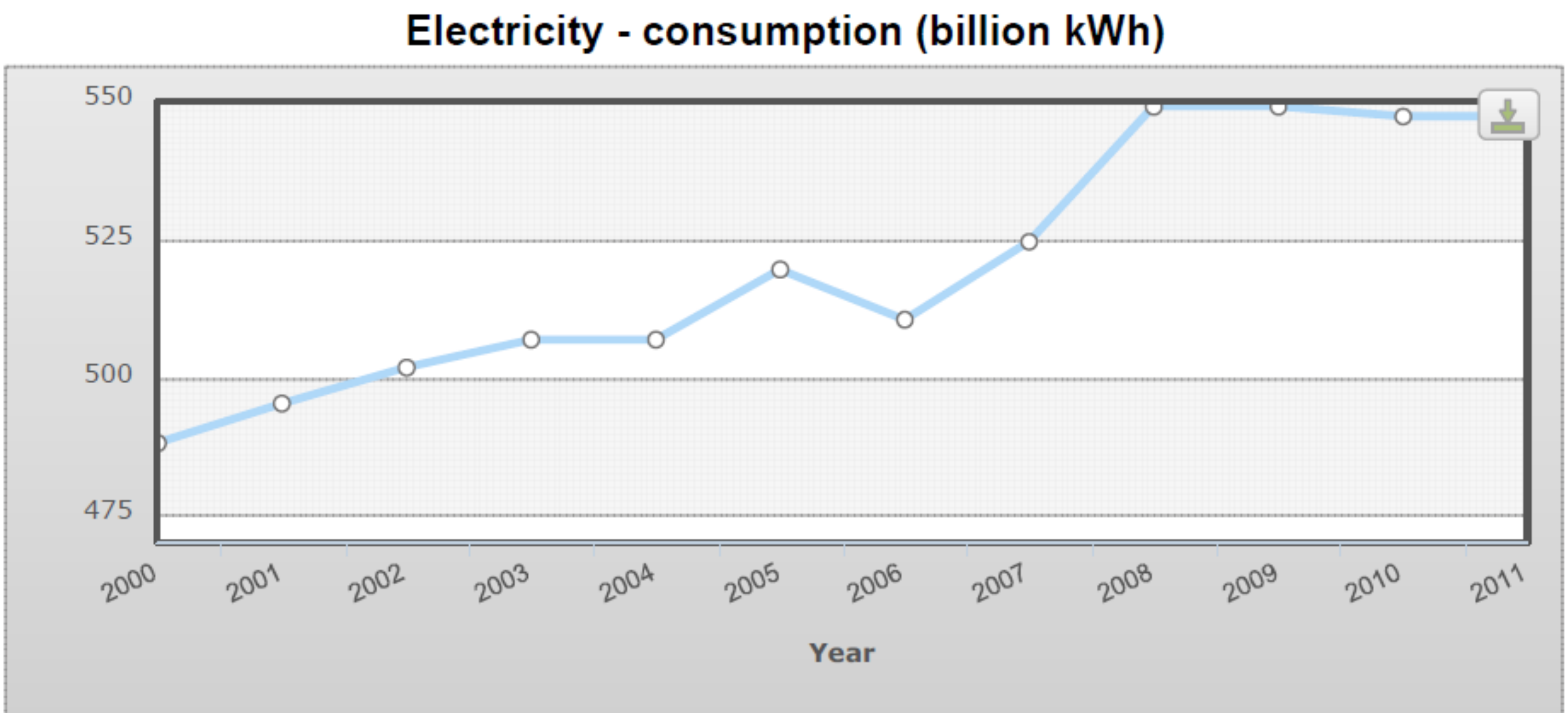
Example Background

- Production of energy is as follows for Germany



Example Background

- Energy demand is not expected to increase much due to the gradual shift from energy intensive manufacturing to a knowledge economy



- The question is now: "how to make up for the 25% loss in energy production expected by 2022 as a result of the nuclear plants being shut down?"

Example Background

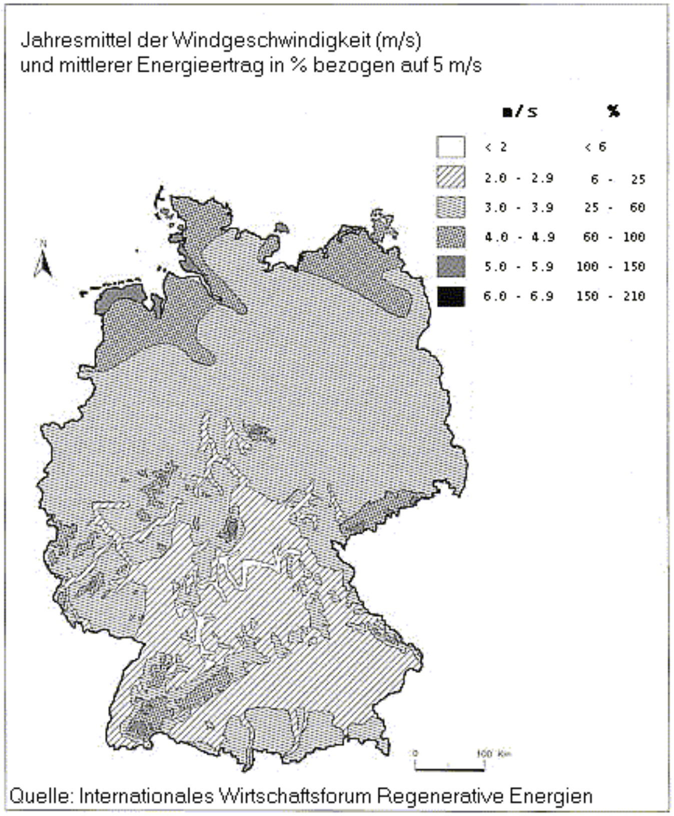
- The official reason given for the decision is concerns about nuclear safety.
- The German government wants power generation that is:
  - As independent from imports as possible,
  - Sustainable
  - Non-polluting
  - with 40% less carbon emission by 2022
- The current sources of thermal power are mainly coal, and that is brown coal with very bad emissions. Coal is also imported. A smaller percentage of power comes from imported oil and gas.

Example Background

- Germany is a world-leader in solar and wind power technology.
  - The share of electricity produced from renewable energy in Germany has increased from 6.3 percent of the national total in 2000 to over 20 percent in the first half of 2011. Renewable energy share of gross electricity consumption rose from 10 % in 2005 to 20 % in 2011. Main renewable electricity sources were in 2011: Wind energy 38.1 %, biomass 26.2 %, hydropower 16.0 %, photovoltaics (solar) 15.6 % and biowaste 4.1 %.
  - In 2010, investments totaling 26 billion euros were made in Germany's renewable energies sector. According to official figures, some 370,000 people in Germany were employed in the renewable energy sector in 2010, especially in small and medium sized companies. This is an increase of around 8 percent compared to 2009 (around 339,500 jobs), and well over twice the number of jobs in 2004 (160,500). About two-thirds of these jobs are attributed to the Renewable Energy Sources Act
  - Germany has been called "the world's first major renewable energy economy". In 2010 nearly 17% (more than 100 TWH) of Germany's electricity supply (603 TWH) was produced from renewable energy sources, more than the 2010 contribution of gas-fired power plants.

Example Background

- Here are some info about alternative energy sources:
  - It takes about 8 years to build a modern coal power station
  - Wind energy potential in Germany is as follows:

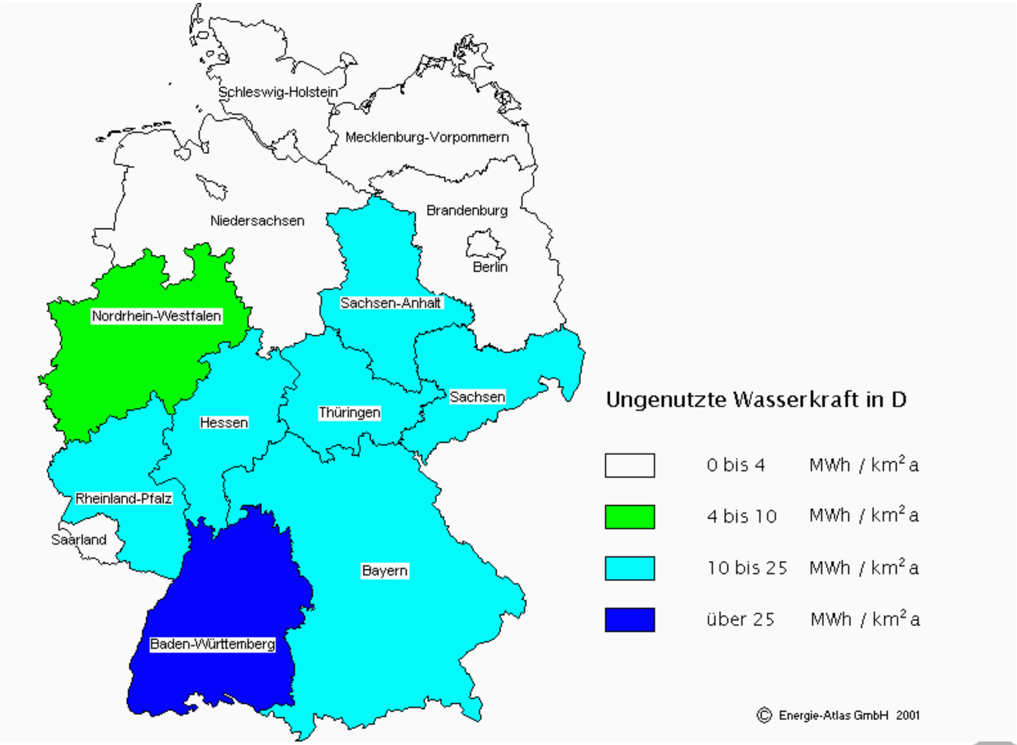


The first values represent the annual mean wind velocity (in m/s) and the second values the energy output that could be obtained by a conventional turbine (in % of the output at a wind velocity of 5 m/s). For an installation of 600 kW and a rotor diameter of 44 m, an output of 100 % corresponds to 730'000 kWh.

On some islands and some coastal regions of the North Sea and the Baltic Sea the wind velocity is adequate for the installation of conventional wind power plants. In the Center and South of the country some isolated regions can probably also be used for such installations.

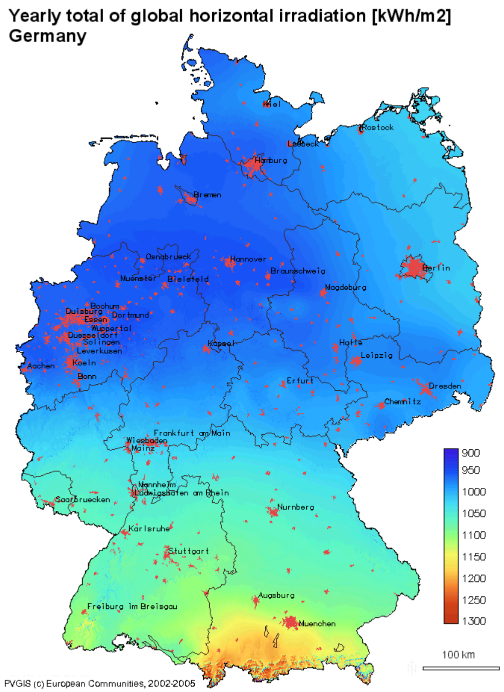
Example Background

- Hydropower potential:
  - The total installed capacity in Germany at the end of 2006 was 4.7 GW. Hydropower meets 3.5% of the electricity demand.

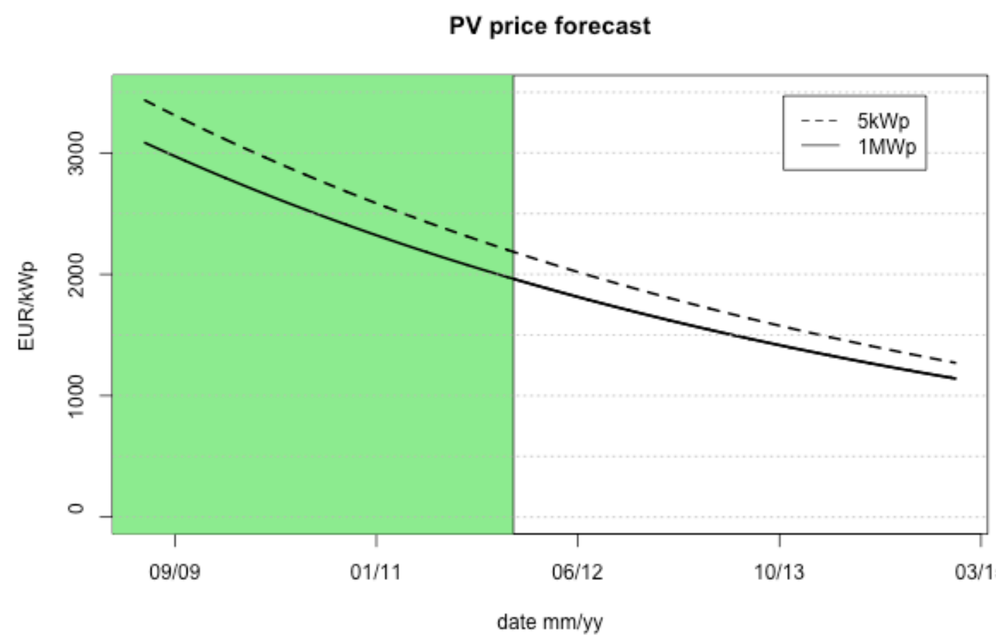


Example Background

- The German solar PV industry installed 5.9 GW in 2011, and solar PV provided 18 TW·h (billion kilowatt-hours) of electricity in 2011, about 3% of total electricity. Some market analysts expect this could reach 25 percent by 2050
- Solar power potential

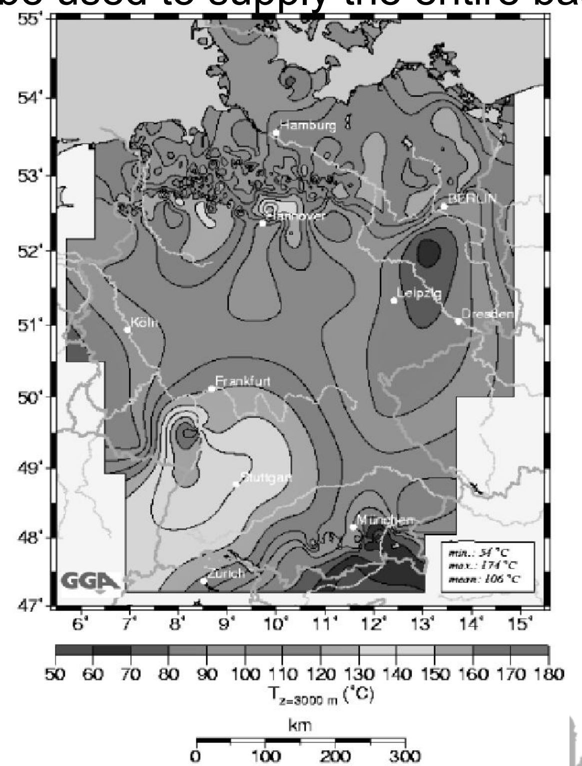


Solar Power Cost



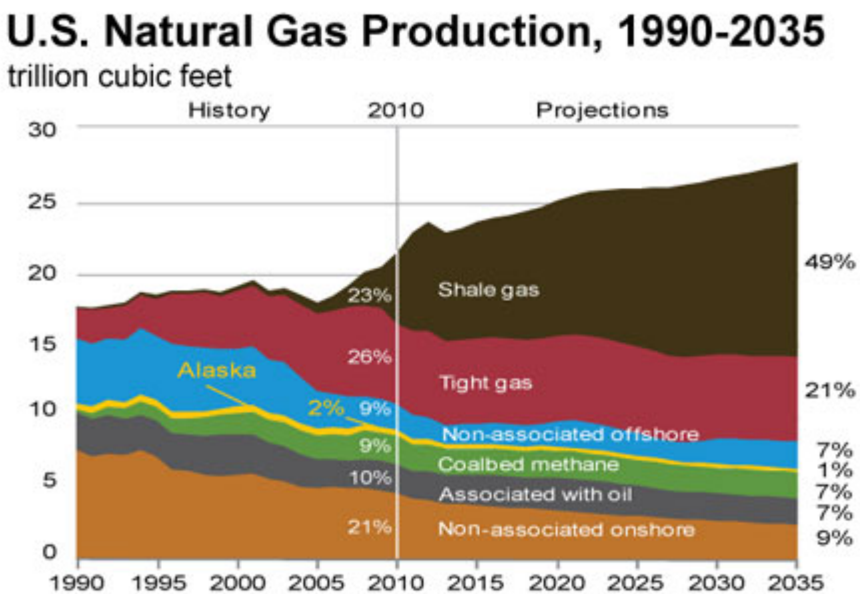
Example Background

- **Geothermal power in Germany** is expected to grow, mainly because of a law that benefits the production of geothermal electricity and guarantees a feed-in tariff. Less than 0.4 percent of Germany's total primary energy supply came from geothermal sources in 2004. But after a renewable energy law that introduced a tariff scheme of EU €0.15 [US $0.23] per kilowatt-hour (kWh) for electricity produced from geothermal sources came into effect that year, a construction boom was sparked and the new power plants are now starting to come online. In 2003 the bureau for technological impact assessment concluded that Germany's geothermal resources could be used to supply the entire base load of the country.

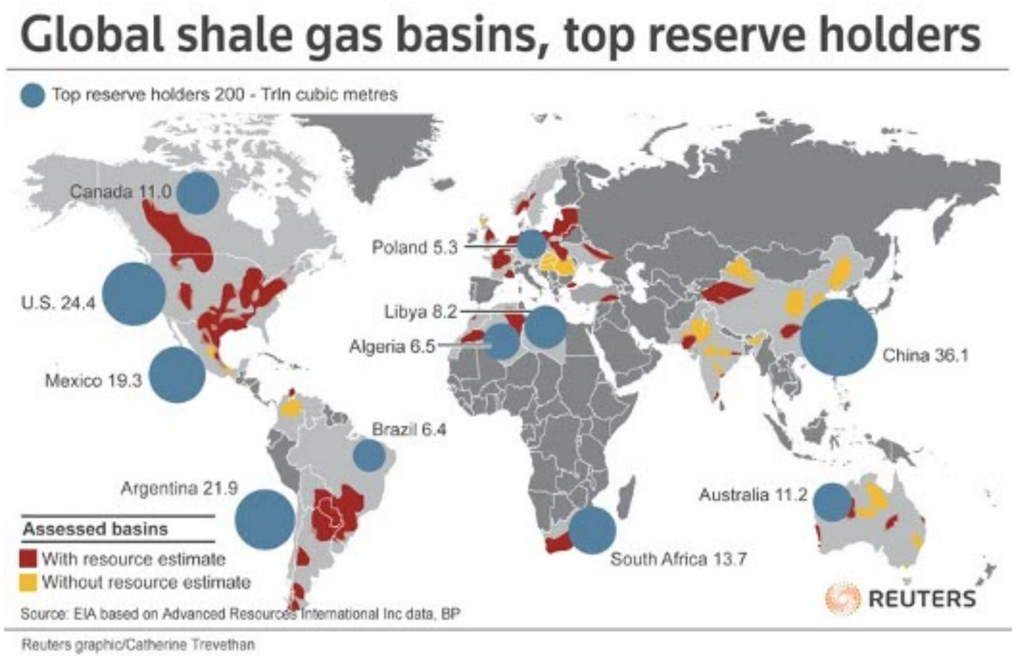


Energy Background

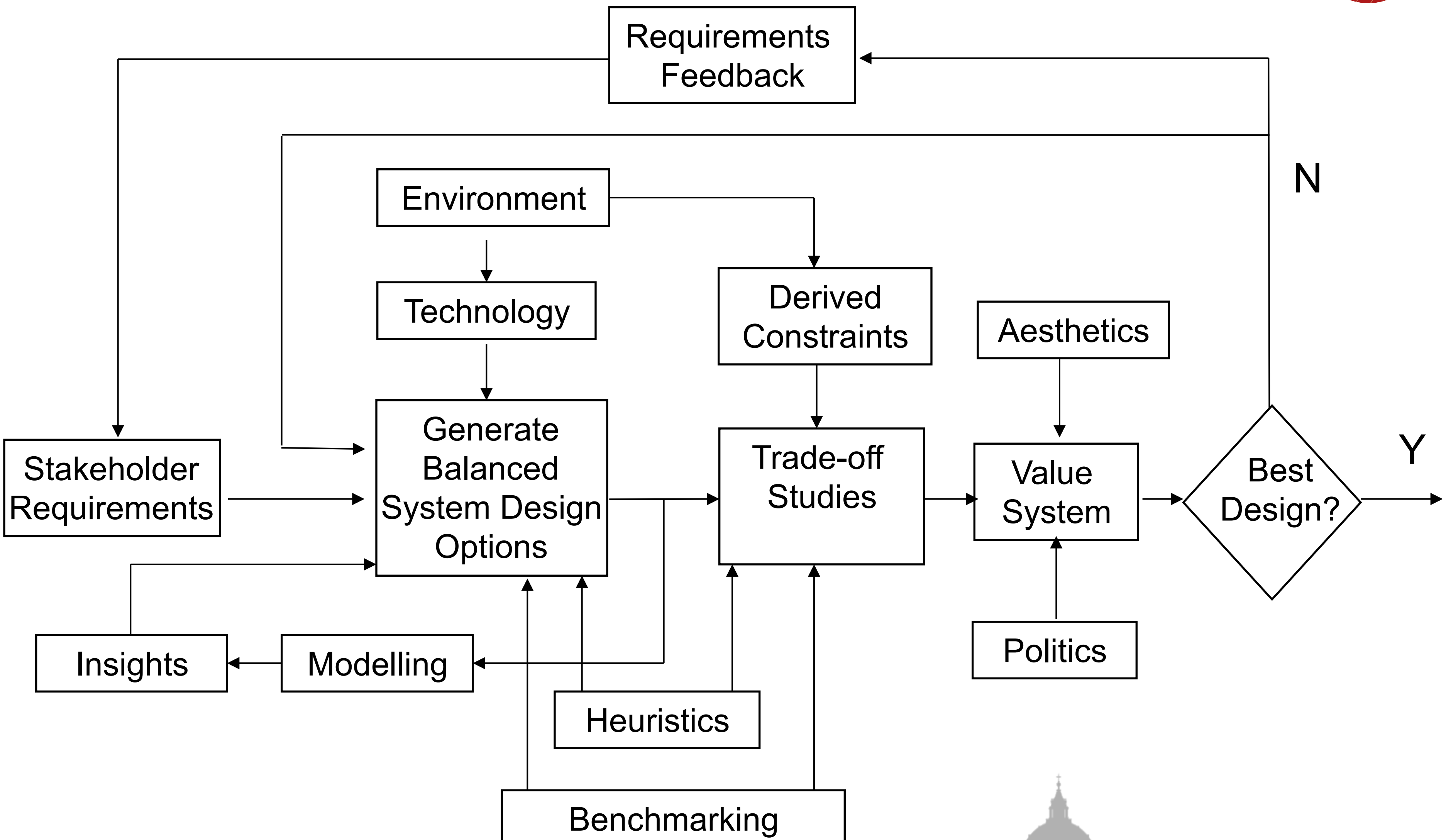
- Natural Gas Availability
  - Fracking technology is opening up vast resources of shale gas:



Source: U.S. Energy Information Administration, AEO2012 Early Release Overview, January 23, 2012.



2.4 System Design Process



Example - Breakout

- The problem will now be tackled in two teams by the students, red and blue.
- Each team has one hour after tea to analyze the problem and to suggest a solution
- Remember to use the methods taught earlier in the day (process diagram attached)!

Summary

- Today you have been introduced to “system design” a specialist field under system engineering
- You have learnt about system theory and systems thinking, and how to apply it to the problem at hand
- You were introduced to a system design process that guides the synthesis of designs
- You were given an overview of tools that can be used for system design
- You participated in solving a real world system problem, using these techniques

THANK YOU!

Dr Gerrit Viljoen

Gerrit.viljoen@deneldynamics.co.za