# INTELLIGENT DESIGN & INDUSTRIAL LOGIC

Attitudes in Engineering, Architecture, Design & Strategy [Objectivity + Subjectivity]

notionpress.com

# INTELLIGENT DESIGN & INDUSTRIAL LOGIC

Attitudes in Engineering, Architecture, Design & Strategy [Objectivity + Subjectivity]

STEX Advanced Design Bureau

notionpress.com

Notion Press

Old No. 38, New No. 6,

McNichols Road, Chetpet,

Chennai - 600 031

First Published by Notion Press 2015



ISBN 978-93-8404-948-5



THE Architect at STEX can be reached at Sourav.stex@yahoo.com/Commercial@stexinternational.com for any Freelance Consulting Needs, Intellectual Property Licensing or Project Collaboration Efforts.

# Contents


1. The world is a complex space 1
2. Factorizing complexity by learning to observe 5
3. Thinking objects & systems 11
4. The Eight Steps to Brilliance 16
5. Breaking the IQ barriers by cognitive fluidity 31
6. Engineering and Design of Products 41
7. A Brief History of Time – A Historical Analysis of Innovation 52
8. A Systematic World Space: Patterns of systematic Innovation 59
9. Rudimentary Economics and Financials 65
10. Outlining the future – Array of Innovative Possibilities 69
11. Industrial Logic & Industrial Design 81

# Intelligent Design & Industrial Design - General Introduction

This technology book is targeted at Technology Architects and Engineers who are interested in

- Intelligent design Object Paradigm of systems [not OOPS]
- Brand New Paradigm - Advanced Systems Engineering & Design
- Industrial Fractal Engineering
- Industrial Patterns & Practices
- Matrix Methods and Reduced Instruction Set Thinking
- Industrial Enterprise Architecture
- Automated Systematic Innovation
- Concept of Spatial Projections through technology
- Information Grid project & Geo Spatial Search
- Industrial Logic & Industrial Design - Machine Usecase Factorization/Learning Training/PassportSolutions/(Alternative) Turbine Design
- Software automation & Transactional Systems Design
- Artificial Intelligence/Machine Intelligence
- Industrial Domain Segregation/Listings for planet Earth
- Economics|Money Market|Stock Market - Principles/Complexity Analysis/Lateral Angle View Points
- Advanced Enterprise Finance and Cash Flows Systems
- Selected advanced Technology projects
- Decoding the matrix that we live in (The Film Matrix - Have you stood and stared at it, marvelled at it's beauty, it's genius?" )

---

Title of New Revised Version (Version 2) : INTELLIGENT DESIGN & INDUSTRIAL LOGIC (New Book Version 2) Date of Revision & Print: Estimated 30 September 2015
Total Pages in the NEW Book:155

Old Title of Book (Version 1): Intelligent Design Techniques To Build a smarter Planet.
Total Pages: Approx 95,
Date of Print: 24 June 2014

---

Though the book is intended for a technical audience, a general audience may be in a position to appreciate STEX Advanced Design Bureau's point of views and directions of guidance. The book being an Original Author book, is brief, to the point and has Hand drawn Diagrams as Illustrations. If you are Looking for a Industry/Industry Logic Workout Book with Brief Explanations & Original Logic Work (a Life Book) - This book is for you. If you are looking for impeccable linguistics & Beautiful Diagrams & Elaborate Explanations, this book may not fit into, that bracket.

Note: This Book/Manual Involves a Deep Focused Workout for the mind. (You may call it an Industrial Excercise)

The Book has served a second Purpose to STEX. Some of the technology Sets and Paradigms Invented by STEX ADVANCED DESIGN BUREAU are completely brand new in the FIELD of Software Technology & Industrial Engineering/Industrial Logic & Technology Design. Patent Filing and grants usually takes a long period to be validated therefore STEX has enumerated and Illustrated the New Technologies in the Book so as to Protect, It's Intellectual Property Rights.

More over New Technology Paradigms are too wide in spectrum to protect through patents, therefore The book serves the purpose of IP protection, IP Evangelism and IP Distribution. Most of the Technology IP assets are available for Industry Licensing.

## Chapter 1

# The world is a complex space

An analysis of this world quickly reveals the inherent complexities that constitute this real world. The real world is a world created by the accomplishments of humans through the knowledge and experiences gained through scientific thinking and socio-cultural behavior. Our observations however are strictly from an industrial & enterprise perspective and it involves a system's thinking methodology in analysis of this world. The reason why a system's thinking approach is the best qualified approach is because it uni-dimensionally connects and converges disparate or heterogeneous styles of scientific thinking.

By heterogeneous styles of thinking, we mean to suggest the division of information and knowledge into subdivisions of science and engineering for the convenience of comprehension and specializations.

A system's analysis of the world reveals complexities arising from three critical constituents of the enterprise ecosystem:

1. People
2. Work
3. Knowledge and information.

Let's first discuss the **people**'s end of the spectrum. People natively belong to different nations and cultures. They speak a wide variety of languages to communicate.

While we affirm this statement as a fact, it is important to understand how humans developed cognitive faculties over an evolutionary period of time. The human mind uses external symbolic storage and representation as a foundation to language and communication. Once humans had a modern language faculty, they had the potential to apply themselves in building a material culture.

You may understand that symbolic representation's interpretation in the form of a language meant drawing on a lexicon of sounds that we call words and each word standing for a significant idea. The critically important thing about language is that the sentence has more meaning than the sum of the meanings of the words of which it is composed. Words operate more significantly in relation to their individual signified meanings, and there are different ways, grammar and syntax among them which forms the complex web of symbolic relations and eventually, interpretations.

This makes languages a very complex medium of communication. In today's world, the language faculties have been extended to new functional skill sets and specializations, from the industry and enterprise point of view.

Now let's discuss the work end of the spectrum where work tasks range from low skilled mundane tasks to extremely specialized industrial tasks.

Work from an industries perspective is classified into a multitude of domains viz., software, aviation, petroleum, engineering, chemicals etc. Each of these domains requires its own level of specializations and skill sets. This makes it virtually impossible for the human mind to think holistically ignoring the specializations.

Therefore, the word specialist reflects a very narrow context of domain. Naturally, this gives an impression to the human mind that the whole world around, consisting of disparate domains is actually a very complex space.

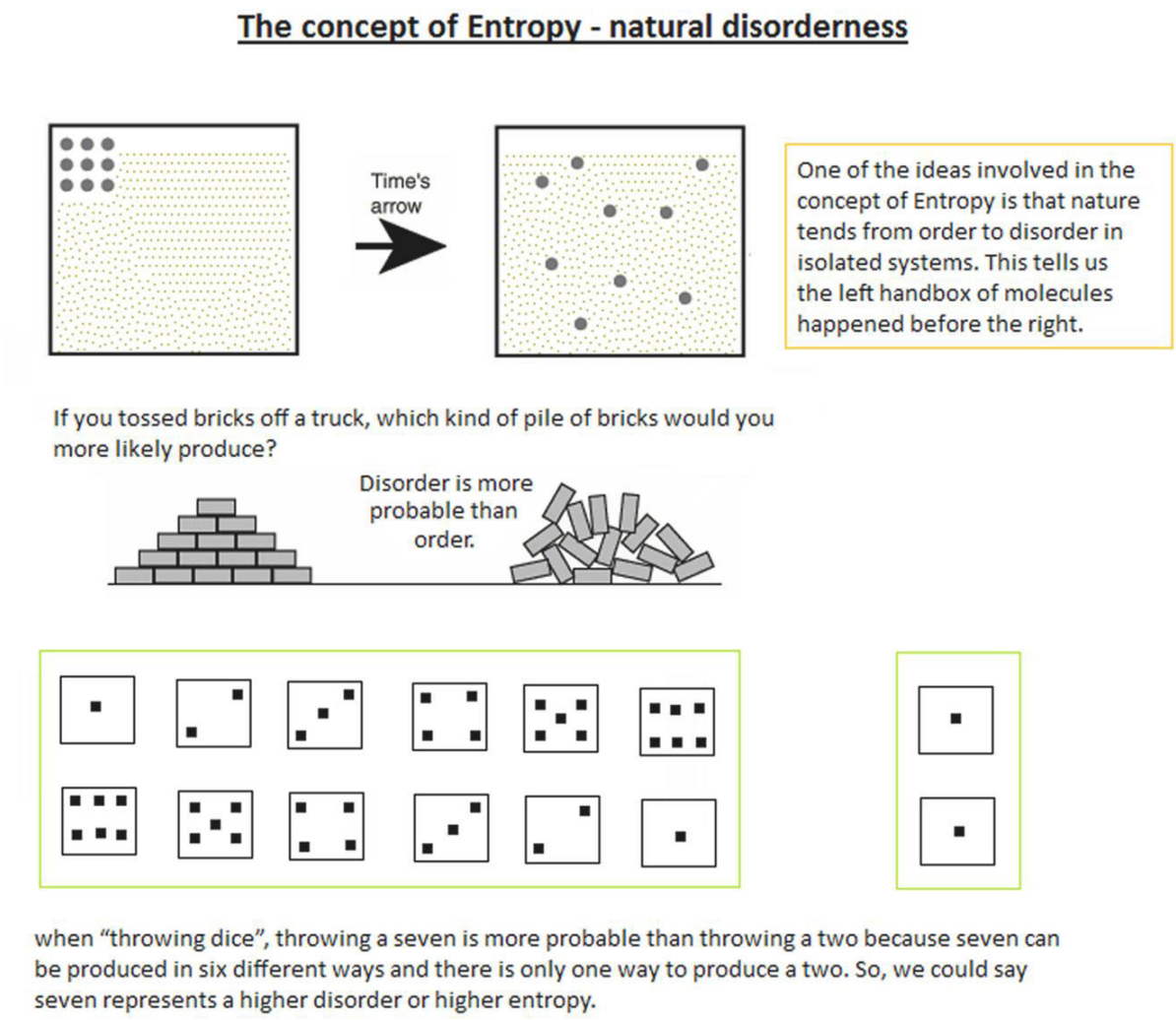
Now let's discuss the knowledge and information end of the spectrum. Knowledge as we understand is structured information about learning and inferences gathered from scientific experiments. As in the work spectrum, knowledge exists in a multitude of subjects and specializations.

It is because of these inherent complexities of the world space that we use systems engineering technique to systematically simplify perspectives and impressions of the world.

Before we move further, we would like to introduce a scientific term called entropy which theoretically defines a degree of randomness or the disorderliness of a system. This sets us into thinking in terms of real world forces that either add to the disorderliness or tackle disorderliness.

The forces that add to disorderliness can be called as decay forces. Similarly, the forces that fight disorder are called evolutionary forces. E.g. Fragmentation of nation states into smaller territories or emergence of a wide variety of language dialects can be considered as decay forces. Similarly, new inventions, discoveries, information systems and formation of larger nation states like the Eurozone can be considered as evolutionary forces.

A more precise way to characterize entropy is to say that it is a measure of multiplicity associated with the state of objects and also a test of complexity.




If a given state can be accomplished in many ways, it is more probable than one which can be accomplished in only a few ways. E.g. when "throwing dice", throwing a seven is more probable than throwing a two because seven can be produced in six different ways and there is only one way to produce a two. So, we could say seven represents a higher disorder or higher entropy.

Comparing this to the tasks we do at work, some tasks are more disordered because they can be accomplished in many alternative ways.

Why are we discussing this complex world space?

STEX Advanced Design Bureau has been set up to necessarily lay the foundation for a new beginning, wherein the complex world space as discussed can be systematically simplified to allow a simpler and clearer comprehension of the world, in turn instigating forces that are evolutionary in nature, leading to the creation of a smarter world.

The evolutionary forces that we've been discussing can be summed up by a simple word "change" – change in thought, action and existence. Physics students may interpret this word in the form of differentiation in world space with respect to change in time.

The objective statement of purpose in the STEX Advanced Design Bureau vision document states, “it intends to solve equations of changing times.”

The word change is scientifically called a fluctuation function that occurs before a sea change is observed in the entire world space. In retrospect, consider the time before computers were invented and consider the time today. Consider a fluctuation function like the advent of internet or the world after the invention of search engines.

The reason why we reiterate these change functions is to inculcate patterns of historical consciousness in human mind. Collective memory only simplifies, sees events from a committed perspective and is impatient with ambiguities of any kind. It reduces events to mythic archetypes.

In contrast, historical consciousness by nature focuses on the historicity of events that took place then and not now, that they grew out of circumstances different from those that are now. Memory by contrast has no sense of passage of time. It denies the pastness of its objects and insists on their continuing presence. It is this behavior of the mind that disregards the significance of these change functions that occur over long periods of time.

Historical consciousness helps keep in perspective change functions over a discourse of time and aids in a prepared adaptation to change and gives every significant change a momentous recognition.

## Chapter 2

# Factorizing complexity by learning to observe

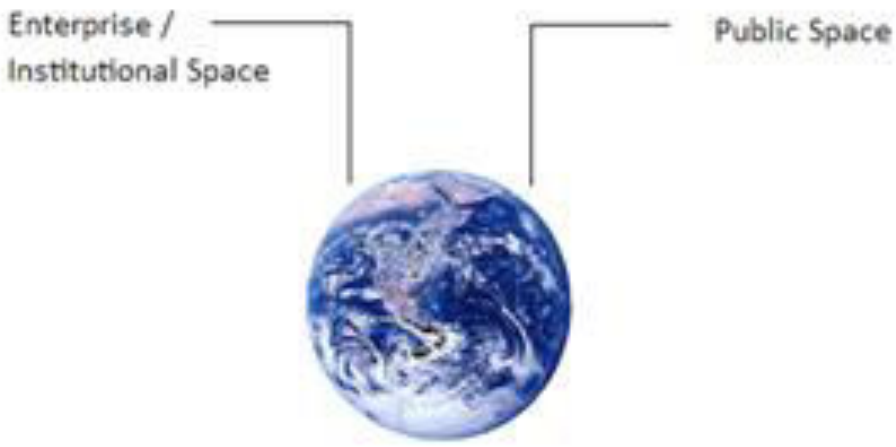
As the chapter name suggests, the goal of this chapter is to factorize the complexity that was discussed in the previous chapter. We used observation as recourse to the factorizing process.

Observation is an activity of assimilation/recording of data through human senses or scientific instruments. Scientific method requires observations of nature to formulate and test a hypothesis.

We also use the process of visualization or a visual paradigm to convey an inference. In this context, our observations would be visually represented using varieties of matrices.

By first observations the world can be divided into two spaces:

1. The enterprise/institutional space
2. The public space.




Enterprise space is the space occupied by individuals carrying out industrial or work activity e.g. factory.

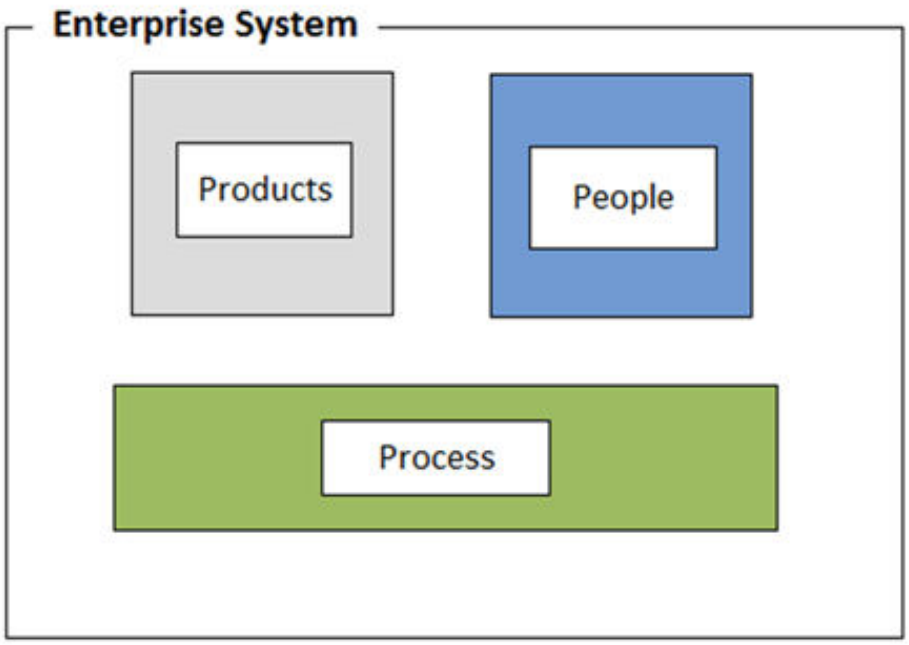
Public space is the space occupied by an individual with respect to his private or personal living space e.g. home.

Our second step to factorizing complexity involves the classification of the enterprise space and the public space into minor entities which we define as factors.

The enterprise space is factorized into the following factors:

1. People - people who work in enterprise/ institution
2. Product/service – products built by enterprises/ services offered by enterprises
3. Process – the methodology of carrying out any particular activity.

**Enterprise System Constituent Entities**




Public space is again factorized into:

1. People – people in the private/personal living space
2. Process – the process by which people receive information about products/services
3. Products – products that people purchase.

Further, enterprises are of three types:

1. Product/ OEM enterprises
2. Service enterprises
3. Distributor/Trade enterprises.

After our first level of factorization, we're now in a position to draw out the first matrix called the world matrix. The world matrix shows the possible interactions or transactions between the two spaces, the enterprise space and the public space.

**World Matrix:**

| | |
|---|---|
| Enterprise space | Enterprise space |
| Public space | Public space |

The inference from the world matrix is that the transactions that occur in the world space can be classified into:

1. Enterprise – Enterprise transactions
2. Enterprise – Public transactions
3. Public – Enterprise transactions
4. Public – Public transactions.

In our next step to factorizing complexity, we draw out another matrix called the transaction matrix. Transactions can be of many types as described below:

**Transaction Matrix (Involving People, Processes and Products)**

| | Products/Service | People | Process |
|---|---|---|---|
| Enterprise to Enterprise | Buy and sell, Distribute. | Support, Knowledge transfer. | General communication channels. |
| Enterprise to Public | Buy and sell. | Hire and fire. | Advertising/Media/ other channels. |
| Public to Enterprise | Buy and sell. | X | General communication channels. |
| Public to Public | Buy and sell, exchange | Communicate, socialize, inform. | In person, socializing forums. |

The Transaction matrix describes the possible types of transactions that may occur between the entities of Enterprise and Public spaces.

1. **Enterprise to Enterprise [PRODUCTS/SERVICES]:** This describes the Products/Services that may be bought and sold between enterprises.
2. **Enterprise to Public [PRODUCTS/SERVICES]:** This describes product buy and sell transactions between the enterprise space and the public space.
3. **Similarly, Enterprise to Public [PEOPLE]:** This describes the transactions between the enterprise space and public space wherein people are hired for jobs or fired from jobs
4. **Public to Public space [PROCESS]:** This describes transactions between people through socializing platforms and channels.
5. **Public to Public [PRODUCTS]:** This describes transactions where there is a purchase, sale or exchange of products or services between people.

The other transaction types are self explanatory.

Analysis of the transaction matrix reveals the following examples of companies, individuals and their activities in the world space:

1. Enterprise to Enterprise [PRODUCTS/SERVICES] - product for use by other enterprises
   a) E.g. IBM – servers for datacenters and enterprises.
   b) GE – Aircraft engines for OEMs like Boeing/Airbus.
   c) CISCO – Networking equipment for enterprises.
2. Enterprise to Public [PRODUCTS/SERVICES] –products for use by people
   a) Microsoft – Windows operating systems.
   b) Apple – Ipods, Iphones, Ipad.
   c) Sony – Laptops, Video cameras.
3. Public to Enterprise [PRODUCTS/SERVICES]
   a) Specialized consultancy services.
   b) Art offerings.
4. Public to Public[PRODUCTS/SERVICES]
   a) Houses, cars - sale and exchange.
5. Enterprise to Enterprise [PEOPLE]
   a) Technical Support.
   b) Knowledge transfer.

6. Enterprise to Public [PEOPLE]

a) Hire People.

b) Fire People.

7. Public to Enterprise [PEOPLE]
   a) Not applicable.
8. Public to Public [PEOPLE]
   a) Communicate
   b) Socialize
   c) Inform
9. Enterprise to Enterprise [PROCESS]
   a) Direct communication channels.
   b) Advertising.
10. Enterprise to Public [PROCESS]
    a) Advertising
    b) Media, other channels.
11. Public to Enterprise [PROCESS]
    a) Direct communication channels.
12. Public to Public [PROCESS]
    a) In person – word of mouth.
    b) Socializing forums.

We now introduce you to a term called “Circle of influence”. It is the area or region in which each of the transactions described above have a valid scope.

For example, Sony sells laptops to the entire world. Therefore, its circle of influence is total world. Similarly, companies may be region specific or nation specific. In that case, their circle of influence would be regional or national.

We’ve now learned how a simplified world looks through the world matrix and the transaction matrix. However, let’s not forget this simplification is only a first step. We’re yet to deal with the inherent complexities associated with each of the entities of the world matrix or the transaction matrix viz., [ Enterprise systems, Public systems, People, Products, Processes ].

Let’s quickly run an exercise to observe each of the entities described above, from a complexity perspective.

[People] – Belong to an array of nationalities, are native speakers of different languages, follow different cultural habits and must still co-exist in the globalised world.

[Enterprise Systems] – Exist as multi sized enterprises in a variety of domains. Enterprises exist as protected, public or private enterprises and each having its own level of innovations and specializations.

[Products] – Product constructs range from mildly complex to extremely complex systems. They range from a pen or a chair to highly complex systems like gas turbine engines, operating systems, nuclear power plants, aircrafts and naval ships. It may be wise to remember that each of these complex products or systems go through decades of iterative cycles of development in the evolutionary cycle.

[Processes] – Processes exist in the enterprise domain as domain specific processes or general enterprise processes. In the people's space, these processes exist as a learning process or a process by which an individual gains awareness about a particular product or service through a media method.

[Public Systems] – Consist of heterogeneous spread of public spaces into urban and semi-urban regions. It also consists of different regions having various degrees of public infrastructure and information system infrastructures. It also consists of differential educational systems imparting knowledge and skill for a significant contribution in the enterprise space. It also consists of government systems structured in a variety of ways. E.g. Center controlled or state controlled system. Public systems also consist of a variety of heterogeneous markets.

## Chapter 3

# Thinking objects & systems

As with our previous chapters, we begin our understanding with an exercise in observation. An observation of the world space reveals that the world around us can be classified into two:

1. Objects
2. Systems.

Observed from a distance, a pen is an object, a calculator is an object and a car is an object. Usually objects can be classified as simple or complex. Simple objects are self explanatory. Complex objects however involve interdependence of other simpler objects forming an integrated whole. Such complex objects are also called systems.

For example, a car is a complex object, but is also referred to as a system. The car's engine is itself a complex object and therefore is called an engine's system. Although computer science teaches us car is an object of the car class, from a system's engineering perspective, the car can behave as an object as well as a system. This is called polymorphicity of objects and systems. Thus the world has many such complex systems viz., an Airplane, a Ship or an industrial enterprise etc.

All complex systems have interactions of object entities as mentioned earlier. Therefore a system can be defined as a function as follows:

System = function of (internal objects, processes)

If we define an engine system, it has many internal objects like cylinders, sparkplugs, pistons etc., all working in a sequence of steps that is called a process.

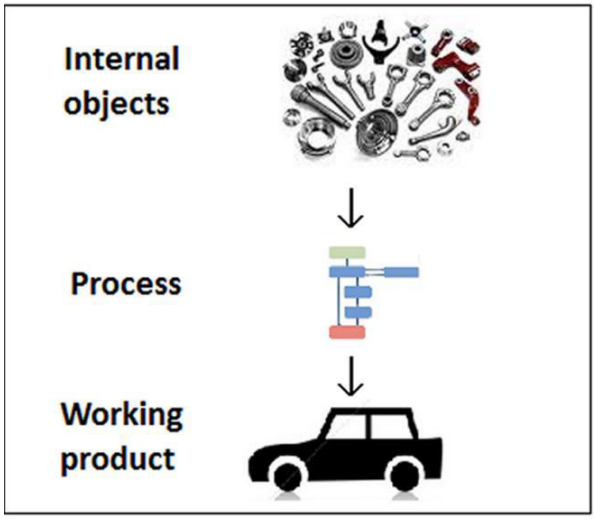
Similarly, in the world of computer science, internal entities work in a sequence of steps to complete a specified process or task.

This principle of systems can be extended to a variety of complex systems.

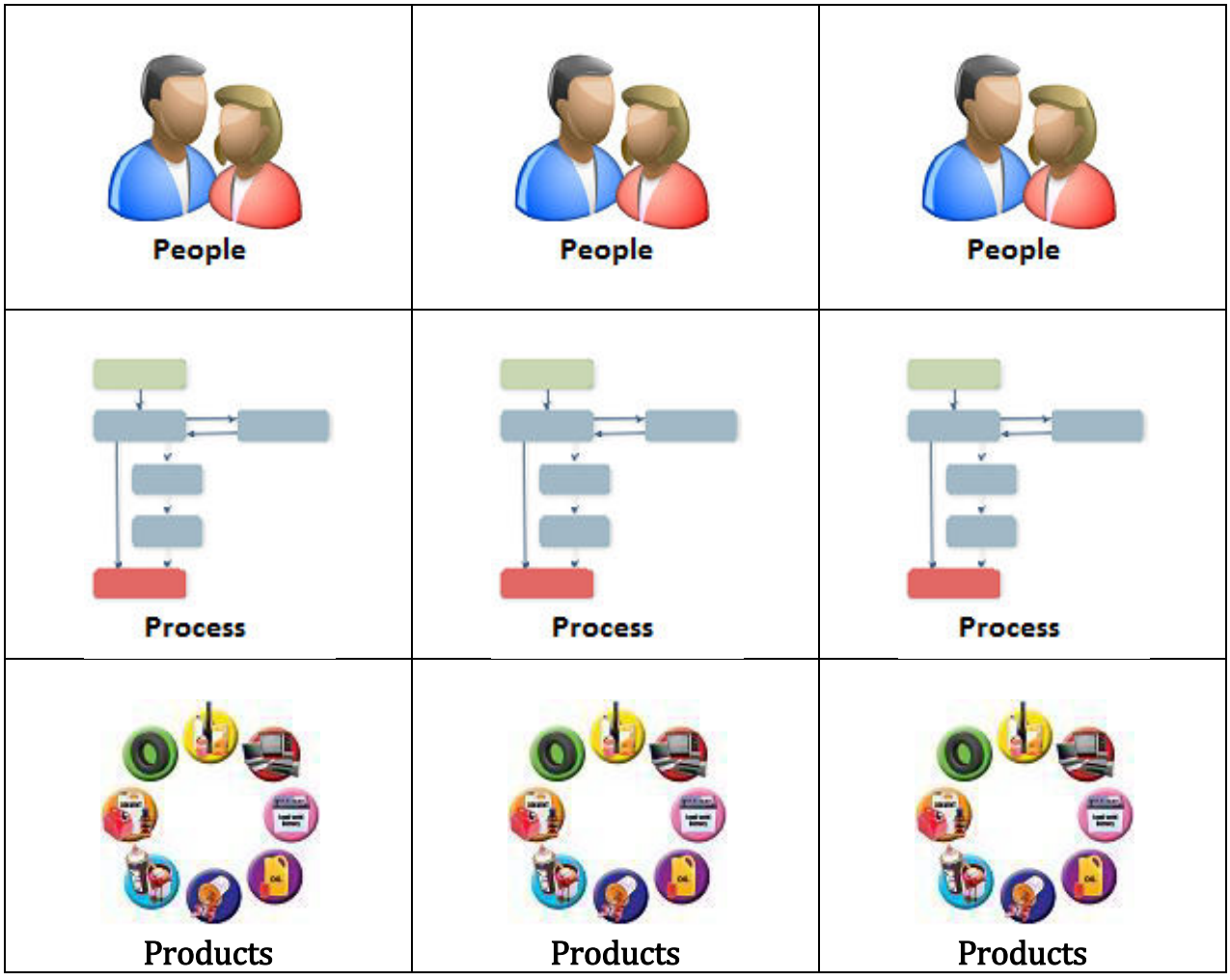
When we extend the system from a machine system to an enterprise system, another additional parameter is added to the function i.e. people. Therefore,

Enterprise System = function of (people, objects[products], processes)

In an enterprise system, people use processes to use products or to develop products.



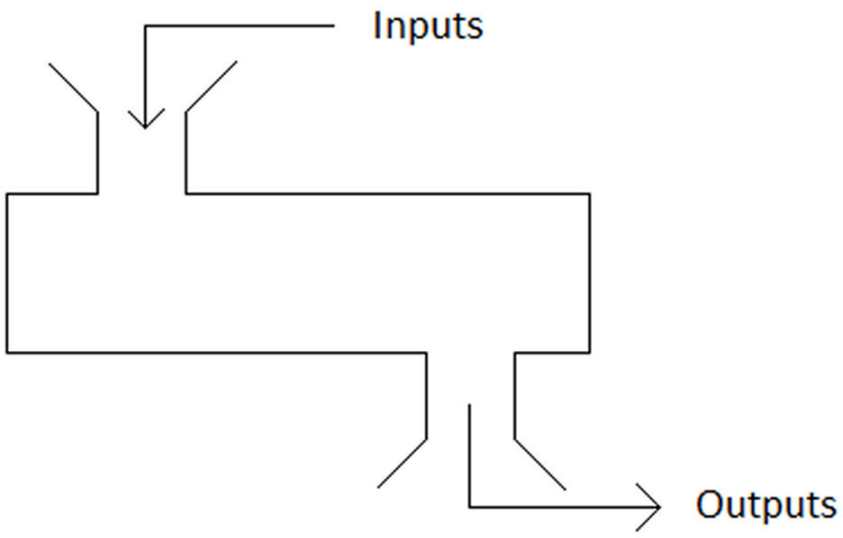

Machine System < Internal Objects, Process >




Enterprise System <People, Process, Products>

For example, in an enterprise system, people use the software development process to develop a software product or people use auto components to produce an automotive engine in a factory.

We're nearly halfway down being declared as systems engineer but we're not there yet. Lets study a few working systems to decide a scientific purpose or a design goal of a system.




Every system normally has an input and an output function. The input and output functions depend on the design goal of the system. This implies every functional system belongs to a specific domain and the input and Output depend on the domain.

| | Input | Output |
|---|---|---|
| Aviation enterprise system | Assembly of Aviation components | Aircraft |
| Automotive enterprise system | Assembly of Auto components | Car |
| Petroleum refinery system | Crude oil, refining process | Fuels (petrol, diesel etc.) |
| Computer mfg system | Electronic components assembly | Computer |

Every system has a process and these processes can be classified as:

1. Specialized Processes
2. Generic Processes.

Specialized processes are those processes that are specific to a particular domain, for example, automotive firms have a specialized process for the manufacture of cars. Petroleum refinery has a specialized process for refining crude oil into fuel.

Generic processes are those processes which are common to every kind of industry for example, marketing, sales, human resources (HR) etc.

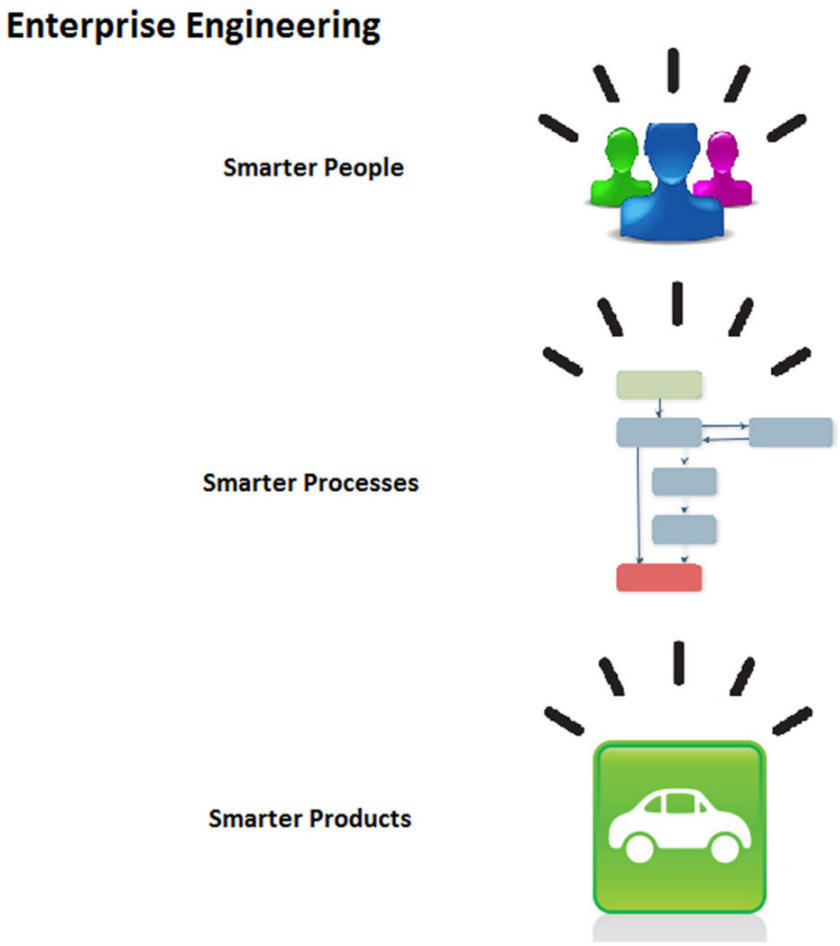
Science and engineering foundations teach us to build the most efficient systems. In other words, every system has a primary design goal to be able to extract maximum useful work from the system and therefore, act as an efficient system.

We're now in a position to freely use the term enterprise engineering. Enterprise engineering is a discipline of systems engineering in that it applies the knowledge and methods of systems engineering to the system and architecture of enterprises. Its core purpose is to increase the value of:

1. People
2. Processes and
3. Products.

Our goals henceforth would be to:

1. Build a smarter world by making or building smarter processes.
2. Making people smarter
    a) The way they learn, think, work.
    b) The way they communicate.
1. Making smarter products or services.




We've already learned how complex the people, process and the product ends of the spectrum are. So, let's work our way through.

## Chapter 4

# The Eight Steps to Brilliance

Before we move ahead, we need a ground up course in a function oriented thought. We presume you understand the definition of the term 'function'.

For example,

Function Draw Circle (Radius r) – is a function that draws a circle with input parameter 'r'.

Similarly, function Eat (Item I and Weight Xgms) – is a function that represents the process of consuming a particular food 'I' in quantity 'X'gms.

Similarly, a function Work (Task T, Hours H) – is a function that represents a process of doing H hours of Task T.

From these functions, we infer that any activity or task performed by a machine or a human can be represented in a function oriented terminology. You may also infer that this terminology is a departure from the usual language constructs used by humans. It is this deviation in the thought process that differentiates a normal world from a programmed world.

Software professionals and scientific personnel may be aware or acquainted with this style of thought process. Those professionals who are in the software field may further recognize a variety of styles of programmed thinking or functional thinking in the form of a variety of programming languages that are available as an option today.

The programming languages differ from each other on a scale whose one end represents a verbose style and the other end represents a purely functional style.

Example - Visual Basic is a language that's closer to the verbose end of the scale while the programming language C# is closer to the functional style of the scale.

This re-orientation in cognitive faculty is the first step into building smarter people. We can infer that people who have a background in functional thinking are better suited to be professional programmers.

The functional style of thinking actually disregards fluency in natural languages or traditional symbolic representation of language. It is with this assumption that the measurement of intellectual capacity differs from traditional methods of intellectual capacity measurement whose basis has a very strong traditional language orientation.

Having learnt basic theory of function oriented thinking, we move ahead to learn about a proprietary set of functions introduced by STEX Advanced Design Bureau. This set of functions is called the TEFS [T-Enterprise Free-Scaling Smart] Functions or the INTELLIGENT DESIGN function set. This function set has two representations:

1. Simple
2. Complex (Recursive)

Let us first define the simple function set and understand the meaning of each individual function.

1. **Execution Function** - The process of doing any task or activity. In the context of software development process, execution would mean programming. In the context of automotive industry execution would imply the process of manufacturing or assembling auto components to build a car or automobile.
2. **Design/Structural function** - The process of creating a blue print or a template using creativity, innovation, engineering and science.
   In the context of automobile industry, it would mean creating a template or design blue print of a car or an automobile. In a software context, it would mean architecting an application based on design theory or structural engineering theory. The same principle may be extended to other domains like architecture of a building.
3. **Data/Storage functions** – This function represents the process of storage in a physical world or storage of data in the digital world. In the context of an automotive industry, storage would imply storage of inventory, materials etc. In a digital context like software, information would be represented as data.
4. **Risk function** - The process of assessing risk or mitigating risk in any domain. In the software context, this would mean security functions in a software application or in the context of automotive industry, it would mean safety features and characteristics incorporated into manufacturing of a car or automobile.
5. **Optimization function** – This is the process of completing a task in the most efficient way. In the software context, it would imply optimization algorithms. In automobiles context, it could mean fuel efficient way of running an automobile engine. From a logistics & transportation point of view, it could mean shortest route delivery channels.
6. **Enabler function** – This is a pre-requisite process or task required to complete the succeeding task. In the software context, it could mean a pre-requisite framework like the .NET framework for the running of an application. In a computer hardware context, an electrical power supply is an enabler for the hardware machine. In the automobile context, the robotics, welding machines and hoists are classified as enablers to the Manufacturing process.
7. **Rules and Boundary Condition Functions** – As the name implies, it is the set of rules or principles which govern the working of any process or system. In the software context, it would imply the business logic and rules of a software application. In the automotive context, the rules would imply the principles of Otto cycle in the functioning of an engine or the physics rules of vehicle stability.
   The boundary conditions imply the maximum permissible values of variables of a system or process. E.g. In the automobile context, it could mean the maximum weight of the vehicle or the maximum dimensions of the vehicle or its maximum speed. In a software application context, it

could mean maximum number of users who are permitted to use the application or maximum speed of data flow.

8. **Exception Functions** – These functions represent an unusual occurrence or exceptional event occurring in a process or a system.

We have now completed the description of the simple form TEFS functions or the INTELLIGENT DESIGN functions. Our focus now shifts to what these function sets can be used for in different domains and systems.

We understand we are in a pursuit to make the people, processes, products and the system on the whole a smarter space. This function set has the power to intelligently define products, people, processes and the system as a whole.

Before we move ahead, describing the complex form of the intelligent design function set, we would study the applications of the simple function set in a variety of domains.

**Automotive Engine System:**

**Execution** – The continuous running of the engine.

**Storage** – The storage of fuel during fuel injection.

**Structure** – The forged framework of the engine where the pistons move up and down in the cylinder generating power.

**Risk** – Gaskets, bushings, washers to disallow the spillage of oils, hydraulics, fuels etc., or engine cooling system.

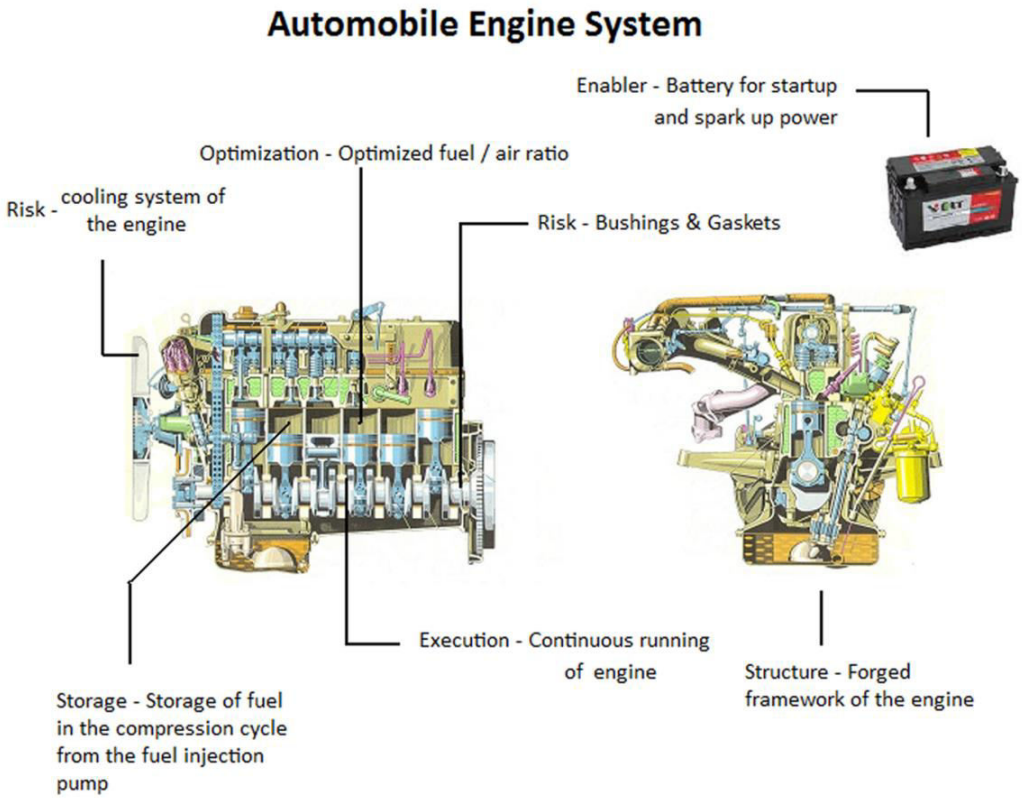
**Optimization** – Optimized fuel/air ratio, optimized sequential process of sparkler timing.

**Enabler** – Battery power to allow spark up or start up power.

**Rules/Boundary conditions** – Otto cycle principle/process of combustion using programmable processors. Boundary condition implies engine capacity of 1.2Ltr or 1.4Ltr.

**Exceptions** – Exceptional heat or temperature rise in the engine or exceptional fuel consumption.

The functions and the examples that we have described above are for a sample purpose only. A real engine function set would contain far more functions.




**Computer Hardware Systems:**

**Execution** - Operation of the computer processor for computation and calculation.

**Design /Structure** – The structure is represented by the cabinet or the outer casing.

**Data/Storage** – Storage of data is in the hard disk or in the memory, in the physical computer system

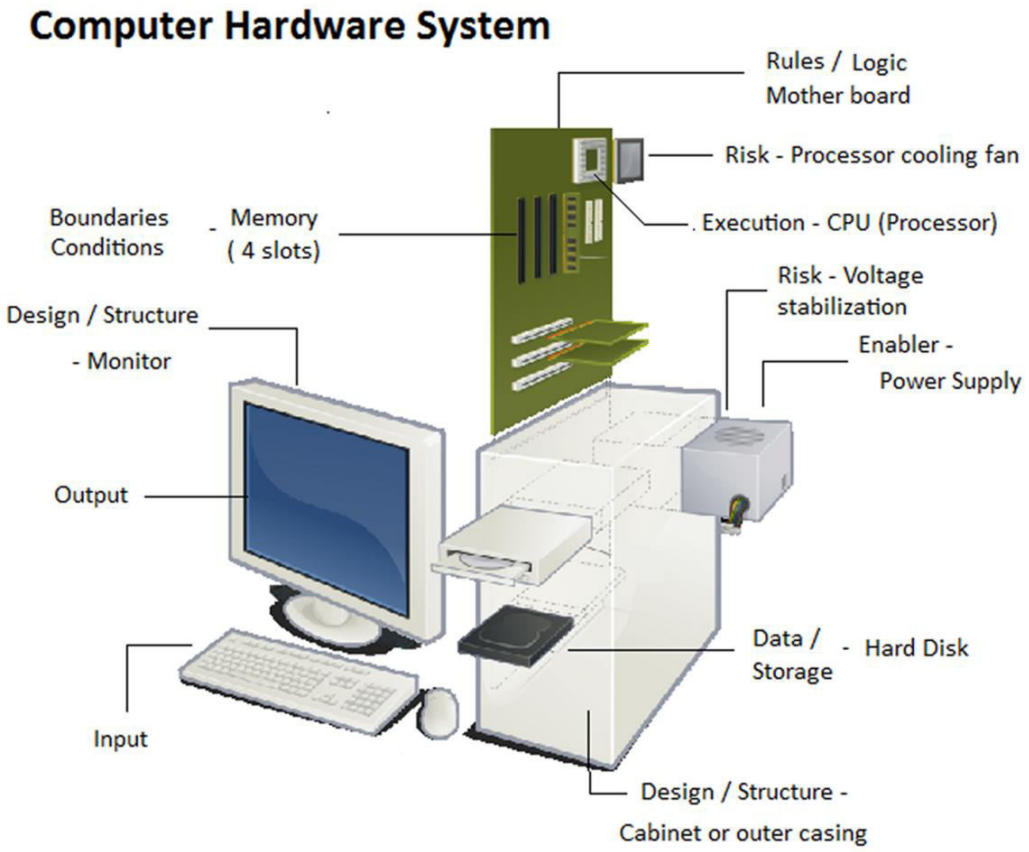
**Risk** - Cooling systems for the processor and the electronics. Or regulated voltage stabilization or RAID - replicated data in duplicate disks.

**Optimization** – Optimization of electrical power consumed by the processor, optimized storage and indexing of data in hard disks.

**Enabler** – Power source (AC source)

**Rules/Boundary Conditions** – Rules and logic are governed by the mother board or an Operating System to run the computer by management of memory/process scheduling.

**Exception** – Memory faults, hard disk failure, processor overheat.




We have now seen two examples of using the TEFS Functions or the INTELLIGENT DESIGN Functions to define the working of two machine systems. A similar principle can be used to describe software systems and a variety of other systems.

We would now move ahead to learn the definition of an enterprise system.

**Enterprise Systems:**

**Execution** – Manufacturing/Product development by the people working in an enterprise.

**Design/Structure** – design of the enterprise building or organizational structure or the hierarchy of people.

**Data/Storage** – Storage involves storage of inventory, materials and finished products. Data involves information related to people, tasks, products etc.

**Risk** – Information security or production quality and physical security of the enterprise.

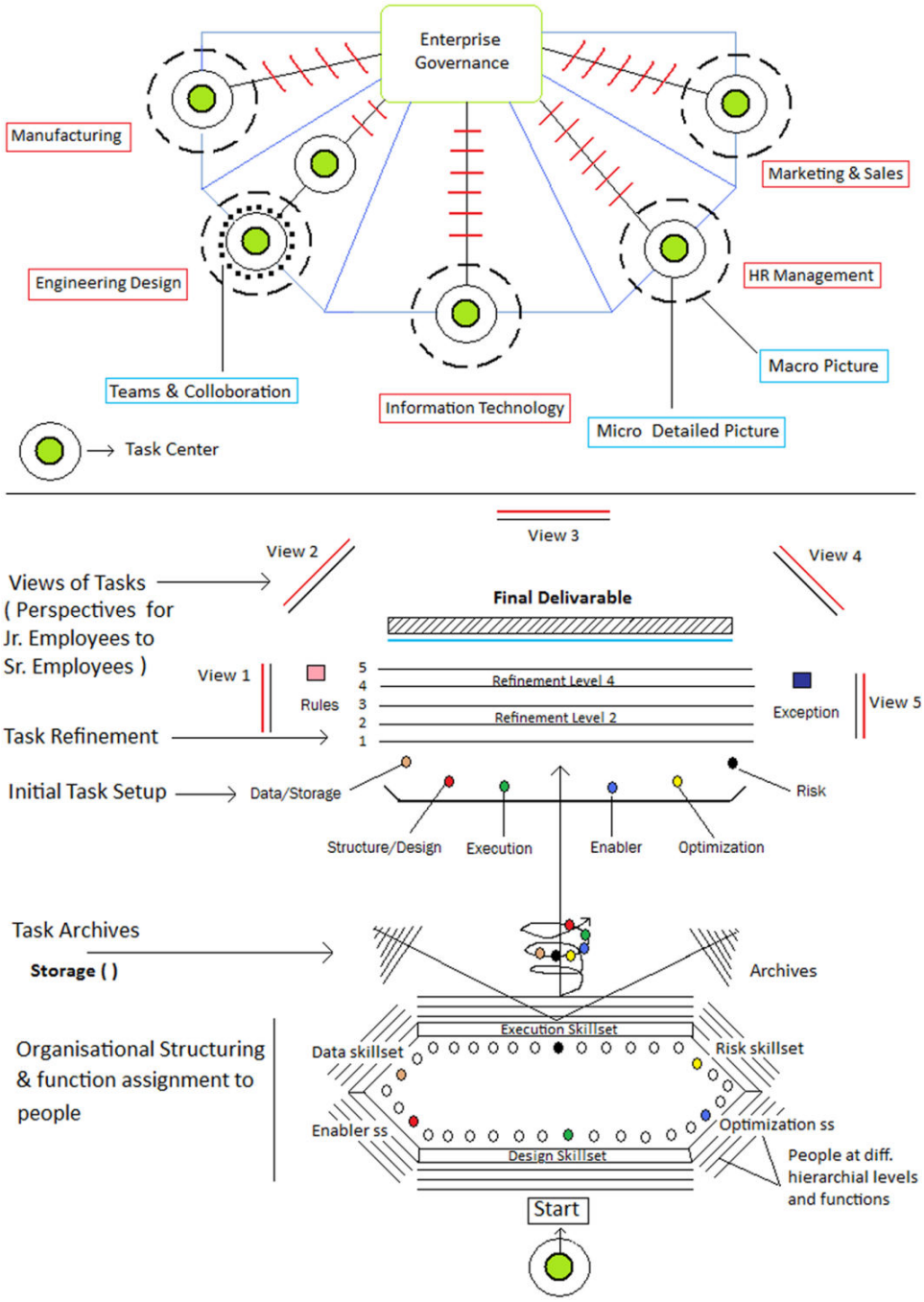
**Optimization** – Efficient process of manufacture or development of products e.g. Lean manufacturing.

**Enabler** – Office working space or working capital.

**Rules** – Enterprise, governance, rules and policies.

**Exceptions** – Unusual incidents in the enterprise space.

We can now represent this function set in a pictorial diagram representing the enterprise system:




**Enterprise workflow system definition**

We now extend this intelligent design function set to define a nation system.

**Nations Systems or Nations expressed in systems parlance:**

Let us take a country or a nation on earth. E.g. USA.

**Execution** – is represented by Industries, enterprises and national economy.

**Design/Structure** – Division of nations into nation states with governance at national level as well as local level.

**Data/Storage** – All the information related to citizens, establishments etc.

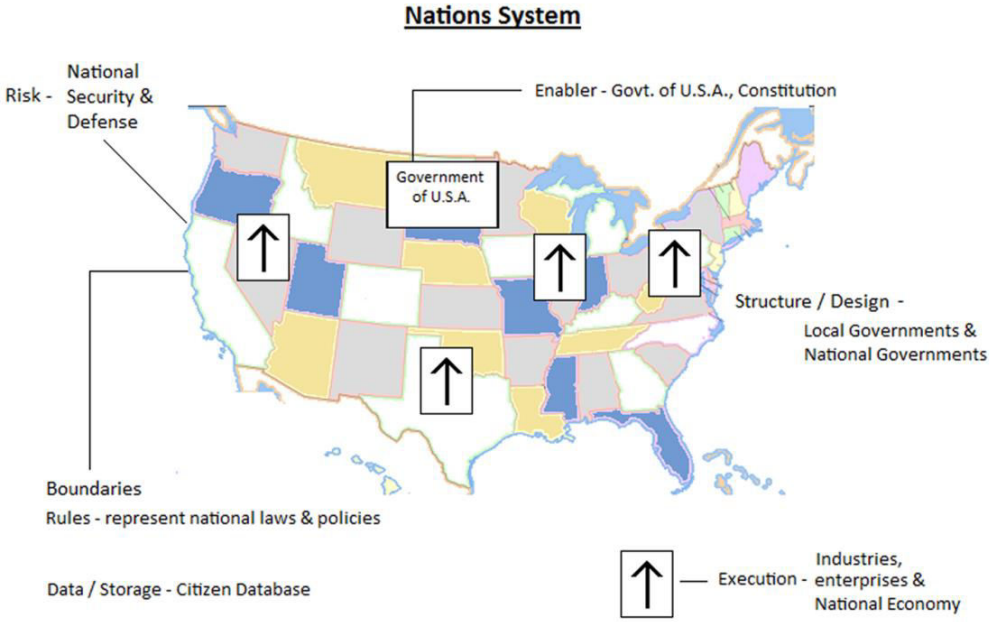
**Risk** – National security and defense services.

**Enabler** – Government frameworks and constitutions.

**Rules/Boundary conditions** – Geographical boundaries of the nation and various national laws for the public and private space.

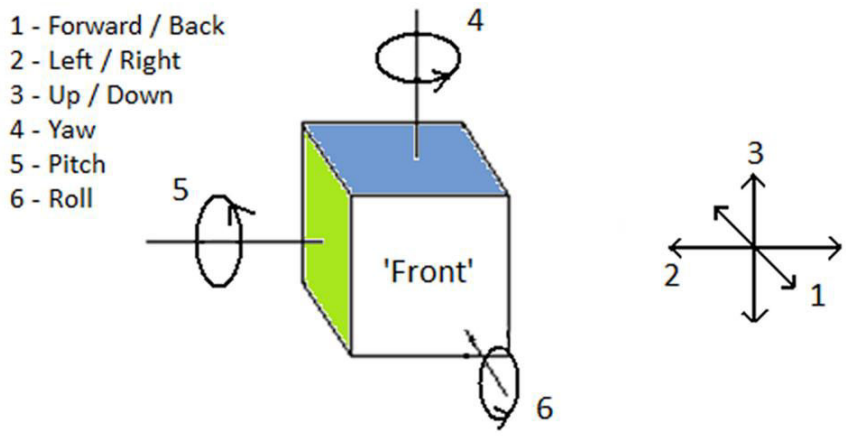
**Optimization** – Optimization of industrial space, housing space and forest space.

**Exceptions** – Untoward incidents, Recessions or wars (External or internal).




We have worked out system definitions at machine level, enterprise level and Nation systems level. The same principles may be used to define systems of any other kind. Mathematics students may understand this approach through the principles of mathematical induction.

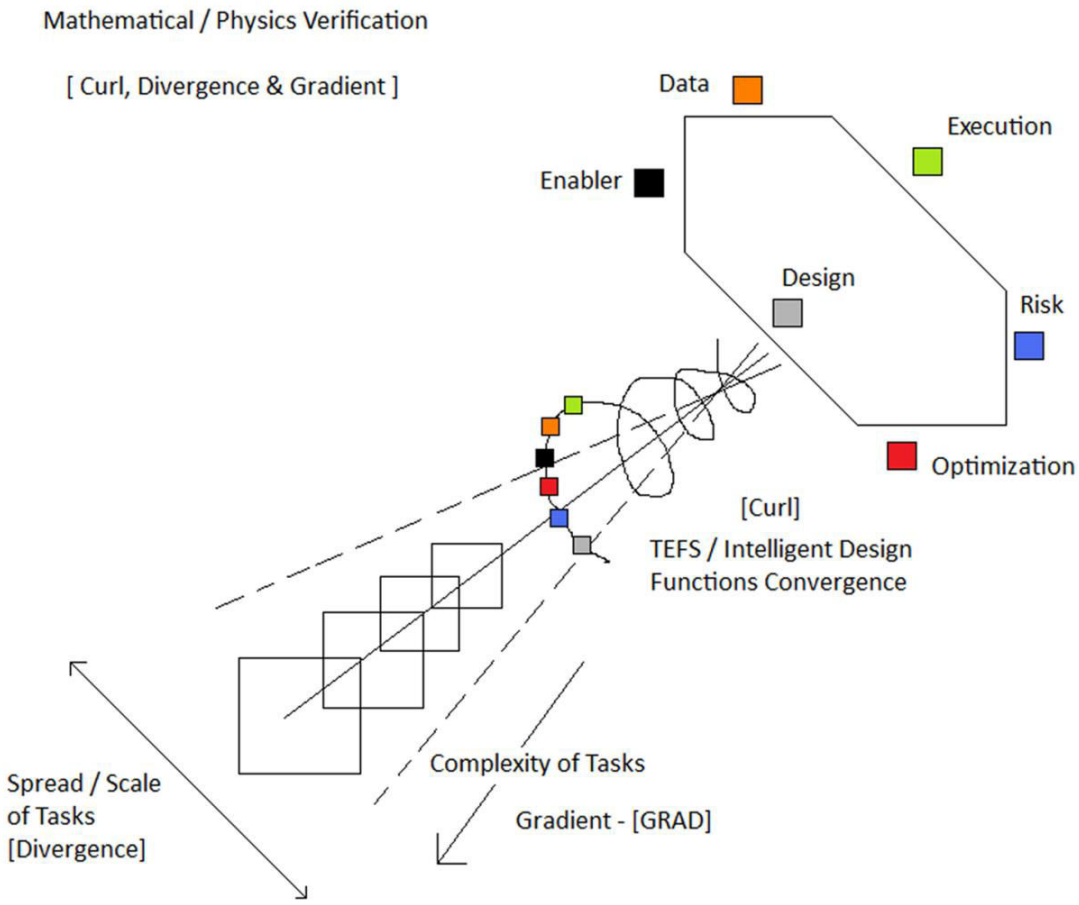
Physics students may be aware of the concept of six degrees of freedom which refers to the motion of a rigid body in three dimensional space viz., the ability to move forward/backward, up/down and left/right combined with rotation about three perpendicular axes (pitch, yaw, roll).




**The Six Degrees of Freedom**

Similarly any system whether a machine system or an enterprise system are governed by six degrees of freedom viz., < Execution, Risk, Optimization, Data/Storage, Design/Structure, Enabler >

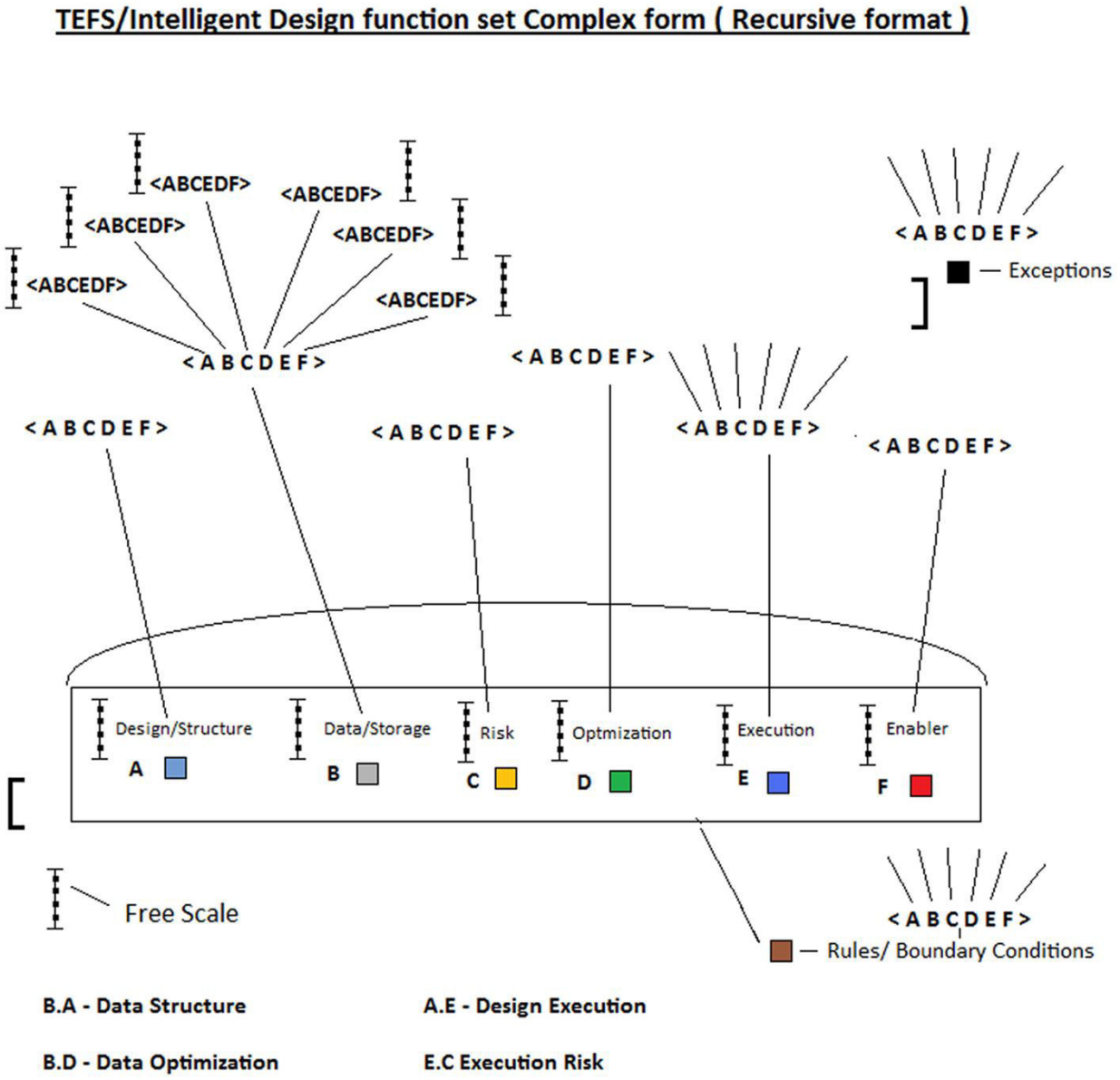
Mathematics and Physics students may also understand the depiction of the Intelligent Design functions in the form of Vector Calculus functions – Gradient, Divergence and Curl.




**Mathematical/Physics Concepts Demonstrating Intelligent Design or TEFS functions**

Please note that for all the examples that we have described, we have defined each of the systems, specifying only a few points for each intelligent design function e.g. Execution or risk etc. A detailed description may involve definition of multiple such execution or risk functions like e.g. execution1, execution2 and execution3 or risk1, risk2, risk3.

Till now we've been discussing the simple form of TEFS or INTELLIGENT DESIGN Function set to define systems. Let us now analyze the complex form (Recursive form) of the TEFS functions.




**Intelligent Design Recursive Function Set**

You may have noticed this recursive or complex function can be expressed as a second order, third order or nth order function set. Using the third order functions and higher order functions is not a very simple task unless used in the field of software programming. Therefore, to explain the concept, we go ahead using the second order TEFS functions/INTELLIGENT DESIGN Functions.

Since the recursive functions appear complex, our approach would be to transform the functions into a 8x8 matrix which we call as the INTELLIGENT DESIGN Matrix or the TEFS Matrix.

## Intelligent Design Matrix or TEFS Matrix:

| | Design/ Structure | Execution | Data/ Storage | Risk | Optimization | Enabler | Rules/ Boundary | Exceptions |
|---|---|---|---|---|---|---|---|---|
| Design/ Structure | Design-Design | Design-Execution | Design-Data | Design-Risk | Design-Optimization | Design-Enabler | Design-Rules | Design-Exceptions |
| Execution | Execution-Designed/Structured | Executable | Execution Data | Execution Risk | Execution Optimization | Execution Enabler | Execution Rules | Execution Exceptions |
| Data/ Storage | Data Structure | Data Execution | Meta Data | Data/ Storage Risk | Data Optimization | Data/Storage Enabler | Data rules | Data Exceptions |
| Risk | Risk Design/Structure | Risk Execution | Risk Data | Risk of risk | Risk Optimization | Risk Enablers | Risk rules | Risk Exceptions |
| Optimization | Optimization structure/ design | Optimization execution | Optimization Data | Optimization Risk | Optimization of Optimization | Optimization enablers | Optimization rules | Optimization Exception |
| Enabler | Enabler Design | Enabler Execution | Enabler Data | Enabler Risk | Enabler optimization | Enabler of Enabler | Enabler Rules | Enabler Exceptions |
| Rules/ Boundary | Rules Design | Rules Execution | Rules Data | Rules Risk | Rules optimization | Rules Enabler | Rules about Rules | Rules Exception |
| Exceptions | Exceptions Design | Exceptions Execution | Exceptions Data | Exceptions Risk | Exceptions Optimization | Exceptions Enabler | Exceptions Boundaries | Exceptions among Exceptions |

Once we execute the matrix, it may become very clear that the elemental functions of the matrix can be easily used to describe or define any concept in the enterprise space (By application of the functional oriented thought process).

Let us take a few of these elemental joins and see what they mean or represent in the real world space.

**Data Structures** – Subject in Computer Science, dealing with data or Structure of a product/Object or Database design in Software development or Algorithms representing a variety of data structures.

**Data of Data** – That is data about data, also called metadata. In computer science, it could mean header information about a data packet. In general context, it could mean the synopsis or a précis about some detailed information.

**Execution Risk** – Execution risk implies quality of execution or an executable. If execution risk is 0 (Zero), then it can be inferred that the deliverable of the product is of 100% quality.

**Design Execution** - drawing of graphics on a computer screen or any other media. Ina manufacturing context, it could mean development of a prototype.

**Design Data** – design data is the blue print or design templates which can be replicated during object creation or object manufacturing process.

**Execution Optimization** – represents the most efficient process or the sequence of executing a task. E.g. in a software development context, it could mean tuning of an application for performance or in an automobile context, tuning the engine for maximum efficiency.

**Execution - Execution** – It represents an executable or a deliverable. In computer science context, it represents an executable binary. In automotive space, it represents the assembly of a car or the manufacture of an auto component/machine part.

**Structured Execution** –It represents a systematic execution process. In computer science context, it could mean structured programming languages like PASCAL, C++ or Ada. In manufacturing process, it could represent Cellular Manufacturing Techniques.

**Rule Design** – It represents the design of systems rules for effective functioning of the system. In the context of a game, it could mean the design of rules of the game. In the context of an enterprise, it could mean the rules and policies of the enterprise.

**Rule Execution** – It is the process of compliance of rules in any system, machine or enterprise. In the software terminology, it represents functioning of a rule engine.

**Execution exceptions** – It represents an exceptional condition during the execution process. E.g. Machine failure. In software context, it could mean occurrence of an illogical statement like 'Divide by zero'.

**Exception - exception** - It represents an exception among exceptions. Out of hundred executions instances, if 5 values of exceptions are in the range 100-200 and one value of exception is in the range 1000-2000, such an exception would be termed as an exception among exceptions.

We have taken up a few cases of these matrix elemental joins and given a few examples. The rest of the elements could be worked out by you as an exercise.

You may observe that in some systems certain of these elemental functions appear as strong functions and some appear as weak functions.

**Strong functions** are those which have a strong significance in the particular system or process.

**Weak functions** are functions with less significance. For example, if we are talking in the context of building a high performance machine, then the execution optimization function may become less significant or may be classified as a weak function because we may focus on a higher performance of the machine rather than a more efficient performance.

As stated above, the third order and higher order INTELLIGENT DESIGN functions are not easily applicable. However, it appears feasible in software development process. An example is as follows:

```
System.Design.Draw Circle( )
  {
      System.Execute.Fetch_Radius ( );
      {
          System. Data.Database_Call ( )
          {
              System. Data.Use_Connnection_Pooling ( ) { Conn.Open … }
          }
      }
  }
```

We have now covered all the magical eight steps to system definition. You may now understand why the chapter name has been called so.

We now discuss a few 'Not so Brilliant' functions which support or act in conjunction with the primary TEFS or INTELLIGENT DESIGN Functions.

1. **State** – State functions help define that exact state of a system at a given time.

2. **Time** – Time is a sequence of timestamps at which any function may execute. E.g. time T1 - execution, Time T2 - optimization, time T3 – execution, time T4 – Data etc.

3. **Iterations** – Iterations represent number of cycles of design or execution functions.

4. **Threads/Parallels** - These represent simultaneous executions of any task or activity.

5. **Measurement** – It is a function to measure or quantify an execution or any other function. E.g. measurement of execution percentage or measurement of execution risks showing quality of deliverables.
6. **Medium** – Medium functions are used to convey the medium of execution or material medium. E.g. Medium of storage could be Hard Disks or tapes, medium of data could be print medium or digital medium on the web.

7. **Views** – Views represent a perspective of a process in execution from a variety of angles.eg Perspective of a junior engineer, Perspective of CEO etc.

8. **EDIC (Explicitly Distinct Implicitly Coherent)** – This function is used as a technique in intelligent design keeping the type of functions explicitly distinct like e.g. explicitly distinct execution functions ,optimization functions etc., yet keeping all the functions cohesively coherent.

We have now completed our discourse involving the TEFS functions or INTELLIGENT DESIGN functions.

Let's analyze a few Information Technology companies and the space they represent in the TEFS matrix or the INTELLIGENT DESIGN matrix:

Google is a formidable search engine company operating in the data space. It may be worthwhile to note that while there are many search engine companies around, Google has a lead in the search space because of its strength in the DATA-DATA (METADATA) / Indexes space. Over a period of time, Google has managed to translate large sets of search queries into huge indexes. While the search process remains nearly the same for all search engines, it is the metadata functions that give Google its strength.

Another company like Microsoft occupies the software enabler space. This space is represented by its operating system software which allows or is a pre-requisite for a wide variety of applications to run on it. E.g. Industry software, web applications, games etc.

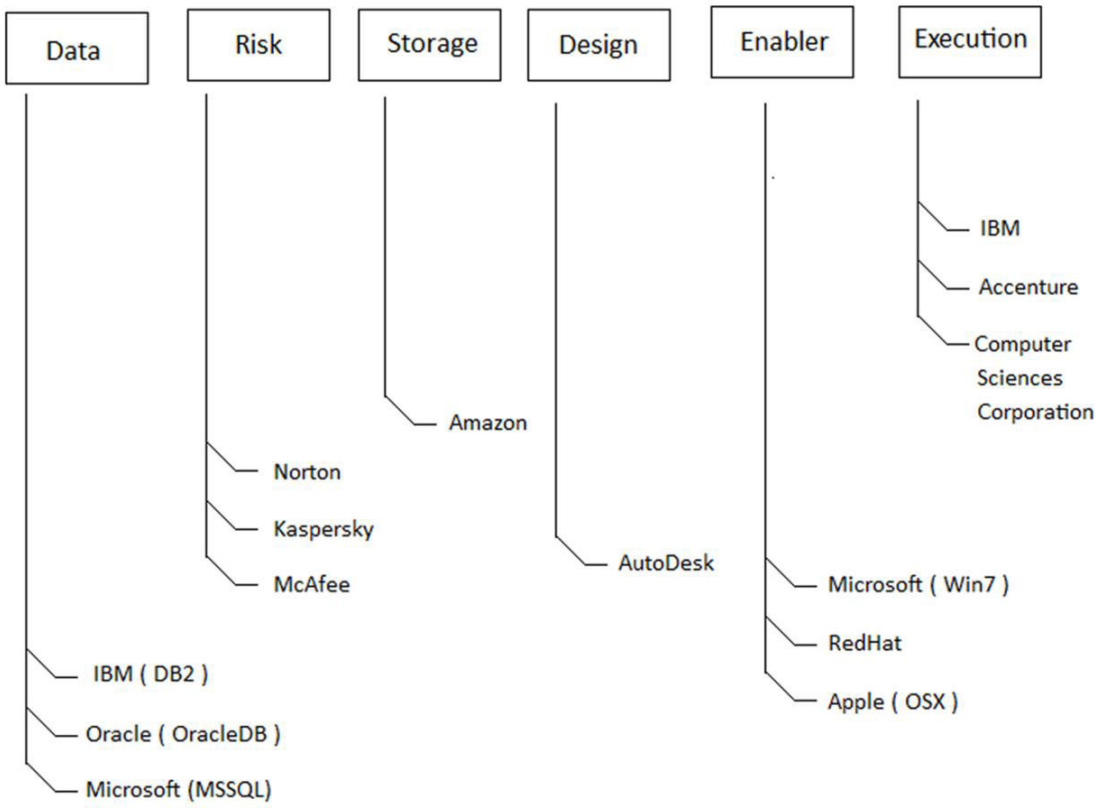
Companies like Autodesk etc., occupy the structure design space.

Companies like IBM, Oracle and Microsoft occupy the Data Execution space represented by their products DB2, OracleDB and MSSQL.

Companies like Norton, Kaspersky, and McAfee occupy the Risk Execution space represented by their products - Virus scanners.

Amazon is an e-commerce company occupying the Storage Execution space represented by its fulfillment services, although it is classified as a distribution or trade enterprise.

## Information Tecchnology Companies Worldmap



## Chapter 5

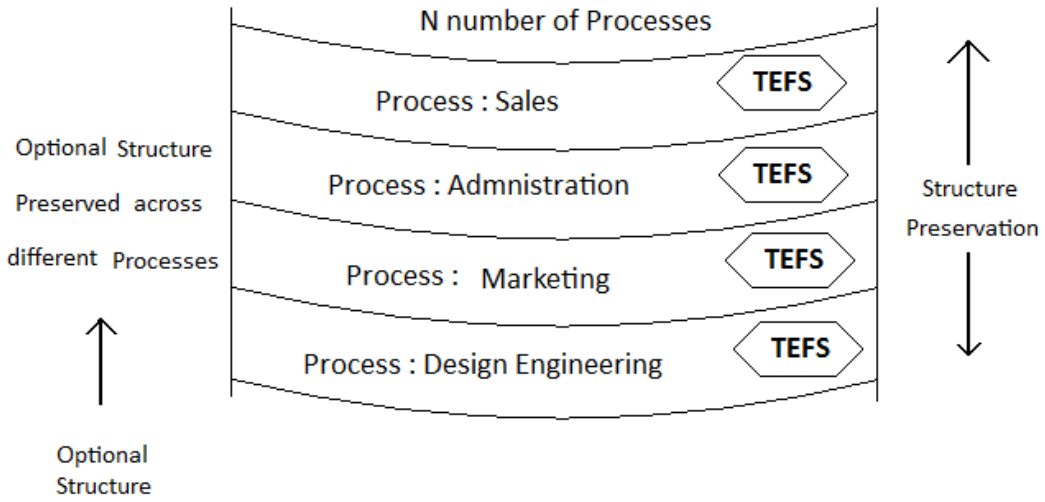
# Breaking the IQ barriers by cognitive fluidity

This chapter deals with two constituent entities of the enterprise system:

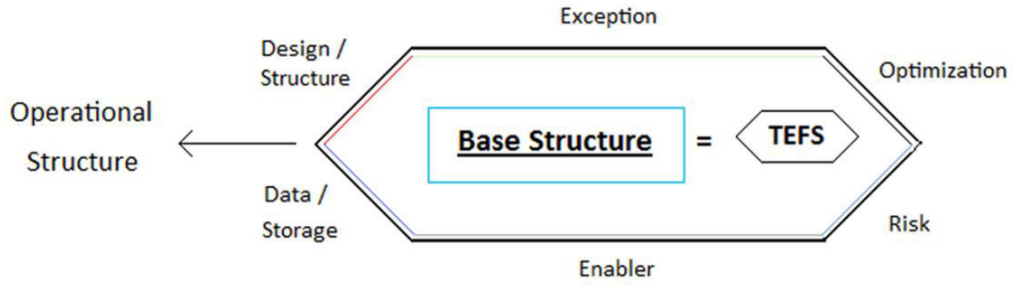
1. Processes
2. People.

Let's first deal with the process aspect of the enterprise space. You may remember processes are classified in the enterprise space as specialized and generic processes. Specialized processes are domain specific processes like Software development or automotive manufacturing process. Generic processes are processes which are typically common to all types of industry e.g. HR process, Marketing and sales.

Mathematics teaches us few concepts in abstract algebra. Abstract algebra studies sets with operations that generate interesting structure or properties on the sets. The most interesting functions are those that preserve the operations. These functions are called Homomorphism's. Homomorphism is a structure preserving Map between two algebraic structures. Applying these mathematical principles in the enterprise space leads us to the following enterprise process structures:




**Preservation of Intelligent Design functions across Processes**




**The TEFS functions or The Intelligent Design Function set**

We now are aware that the TEFS or INTELLIGENT DESIGN functions can be used to describe the specialized process as well as the generic process. To understand this better, we take up a process in marketing and see how the TEFS functions defines the process.

**Marketing Processes:**

**Design/Structure** – In the marketing process, design functions involve design of advertising campaigns.

**Data/Storage** – Data functions represent the process of collecting data from market research as well as product data.

**Execution** – It is the execution of the marketing campaigning.

**Optimization** – Allocation of capital to different mediums of advertising.

**Risk** – Risk is assessment of all the negative reactions that might originate as a part of the marketing campaigning.

**Enabler** – It is all the information about the products and the intended target audience.

**Rules/Boundary Conditions** – Compliance with all advertising rules and laws, regulations existing in the enterprise space.

**Exceptions** – An unusual condition requiring out of the ordinary actions.

Now let us discuss the discipline of developing software.

**Software Development Process – Development of a web/windows application:**

**Data/Storage** – It involves database design, SQL queries and store procedures.

**Design/Structure** – Web form/ Windows form Design.

**Execution** – It is the project compilation and build.

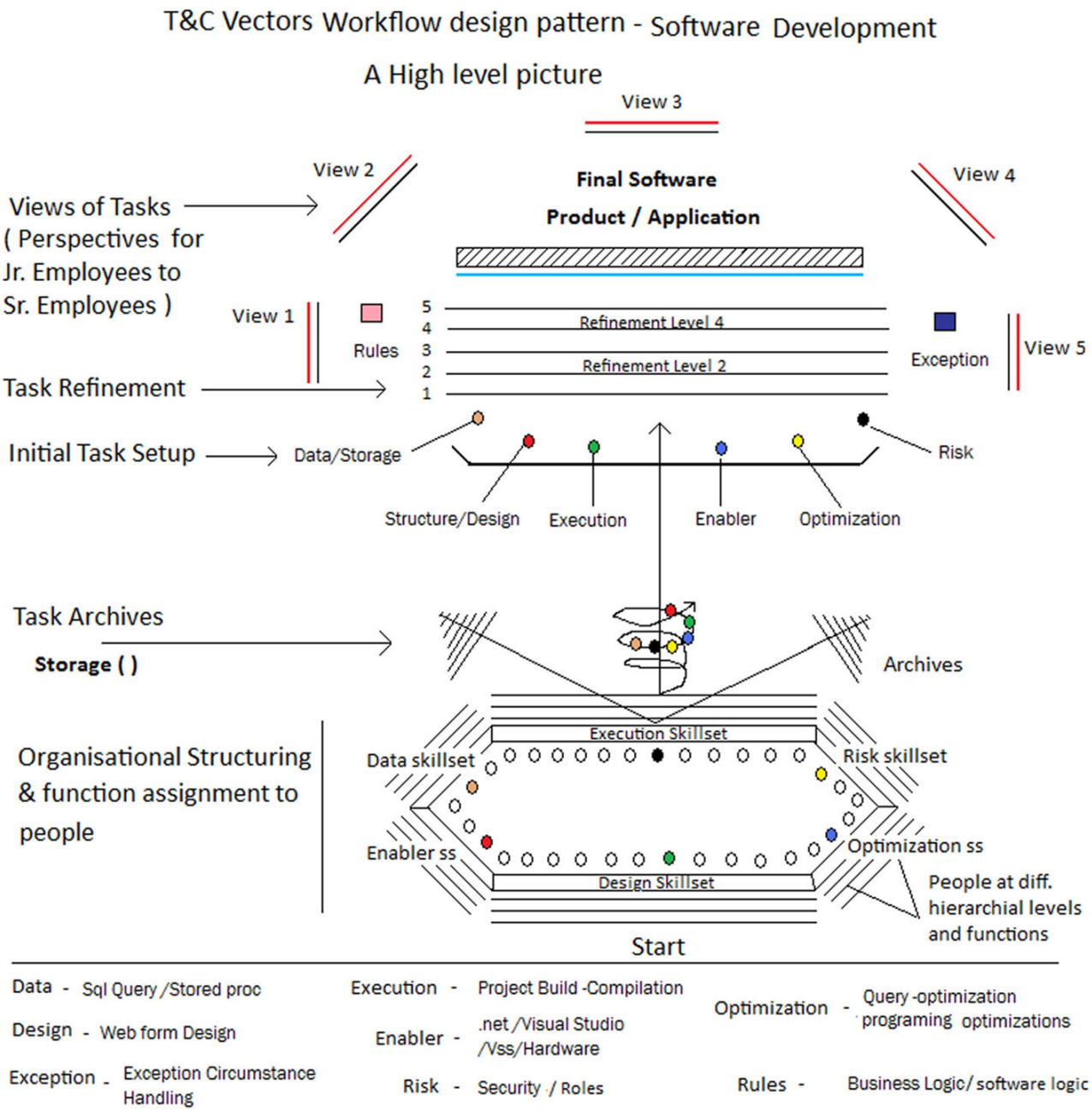
**Enabler** – It includes hardware, Net framework, integrated Development Environment (IDE) like Visual Studio.

**Optimization** – Programming optimizations involves .Net programming optimizations or Database query optimizations.

**Risk** – Involves all the security essentials related to the software.

**Rules** – Involves the codification of all business logic and process logic.

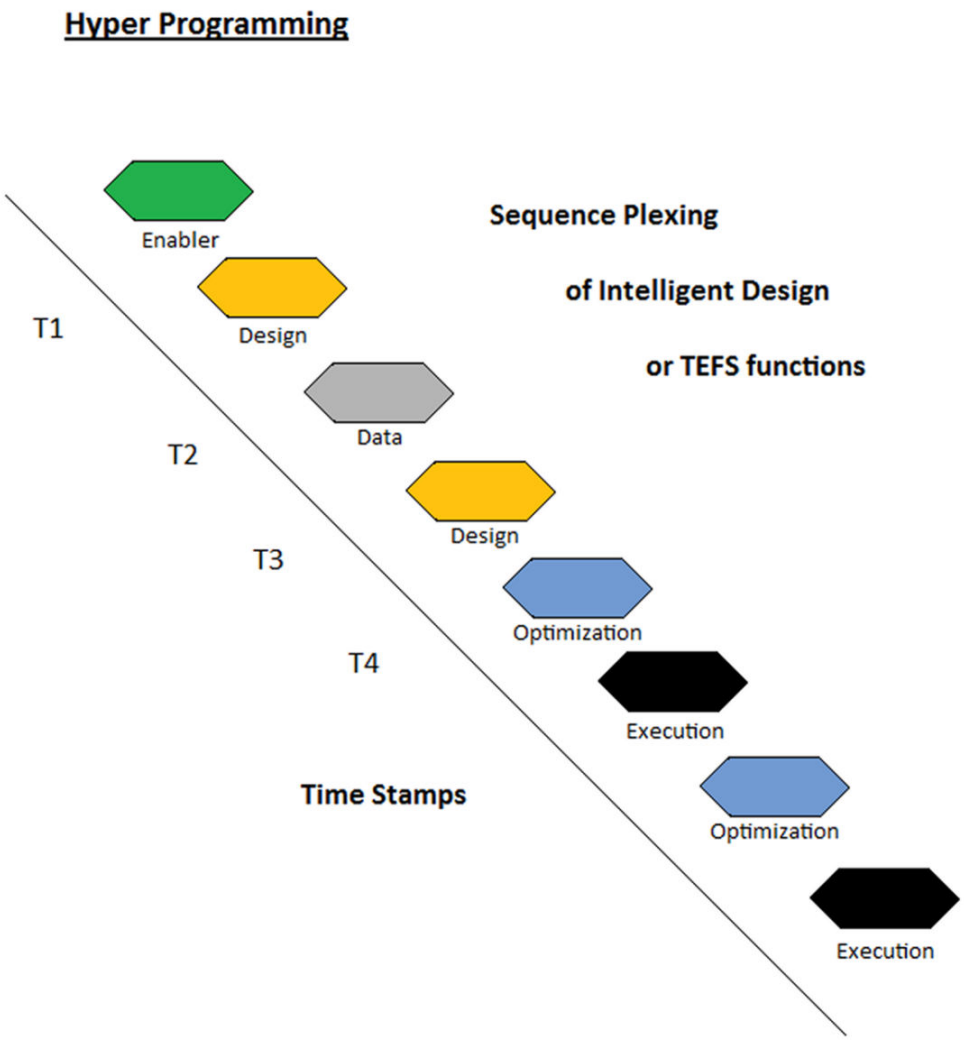
**Exceptions** – Exception handling within the application.



Until now, the standard software development methodologies included SDLC (Waterfall model, Spiral Model etc.) and techniques of Agile development. TEFS foundation suggests a next step in the software development evolutionary life cycle which would be known as Hyper Programming or Programming by Intelligent Design.

Initially, in SDLC, the focus lay on the process. In agile techniques, it shifted to people and teams. Hyper Programming or INTELLIGENT DESIGN methodology suggests a remarkable shift in both the process space and the people's space. It involves sequence plexing of TEFS functions or INTELLIGENT DESIGN functions.

It also by principle necessitates entire team cohesion towards a task of common concern. The teams are structured so that the team members are split up into factored teams focusing on one particular aspect of the software development. For example, a team may be solely responsible for all tasks related to optimization of the software or risks aspects of software. The software development technique is represented by the sequence-plexing diagram as shown below:



**Auto manufacturing Process:**

**Design/Structure** – Involves Product design blue prints interior plus exterior.

**Data/Storage** – Storage involves material storage and machine component storage.

**Execution** – Involves assembly of all components to produce the final product.

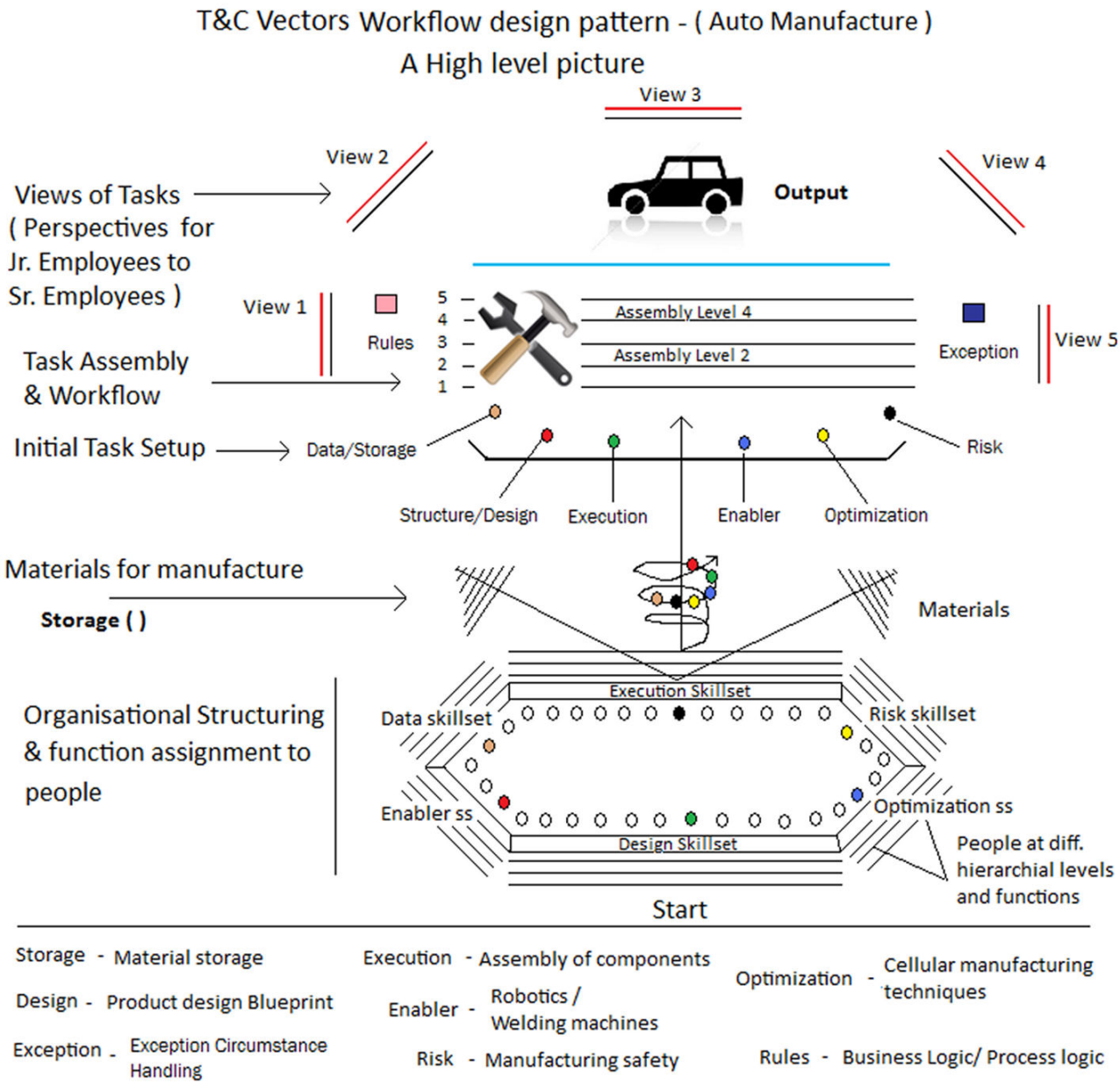
**Enabler** – Includes Robotics and welding machines for the assembly process.

**Optimization** – Includes cellular manufacturing techniques where similar types of products are produced together to reduce material flow through the enterprise space leading to an optimized production.

**Risk** – Involves all activities related to manufacturing safety.

**Rules** – Represents Logical rules and sequence to building a product.

**Exceptions** – Exceptions are incidents or events that take place over the entire manufacturing cycle.


T&C Vectors Workflow design pattern - ( Auto Manufacture )
A High level picture
View 3
View 2
View 4
Views of Tasks
( Perspectives for
Jr. Employees to
Sr. Employees )
Output
View 1
Rules
5
4
3
2
1
Assembly Level 4
Assembly Level 2
Exception
View 5
Task Assembly
& Workflow
Initial Task Setup
Data/Storage
Risk
Structure/Design
Execution
Enabler
Optimization
Materials for manufacture
Storage ( )
Materials
Organisational Structuring
& function assignment to
people
Execution Skillset
Data skillset
Risk skillset
Enabler ss
Optimization ss
Design Skillset
People at diff.
hierarchial levels
and functions
Start
Storage - Material storage
Design - Product design Blueprint
Exception - Exception Circumstance Handling
Execution - Assembly of components
Enabler - Robotics / Welding machines
Risk - Manufacturing safety
Optimization - Cellular manufacturing techniques
Rules - Business Logic/ Process logic

**Enterprise architecture concepts:**

Enterprise architecture is a rigorous description of the structure of an enterprise, its decomposition into subsystems and the relationships between the subsystems. The description includes business processes, roles and organizational structures.

By the introduction of TEFS functions or the INTELLIGENT DESIGN functions we have simplified the complex terminology of enterprise architecture wherein every system consists of processes and every process, be its specialized or non-specialized, follows a design pattern in line with the TEFS functions. After observing processes, the other points necessary to complete an entire enterprise architecture process are organizational structures and roles.

Excerpts from the work "The management of Innovation" by Burns and Stalker have provided the following characteristics of static and dynamic systems:

| | Static Systems | Dynamic Systems |
|---|---|---|
| The Environment | Static | Dynamic / Ever changing. |
| Distribution of tasks | Specialized differentiation of functional tasks. | Contributive in nature to common task of concern. |
| Task Scope | Precise definition of rights and obligations and technical methods attached to a role. | Shedding of responsibility as a limited field of Rights. Problems may be posted Upwards, sideways, Downwards. |
| How is Task Conformance Ensured | Translation of rights and obligations and methods into responsibilities of a function. | Spread of commitment concerned beyond any technical definition. |
| Location of Knowledge | Reinforcement of the Hierarchic structure by the location of knowledge of actualities at top of Hierarchy. | Omniscience is no longer imputed to the head of concern. |
| Communication between Members | Vertical communication | Lateral communication |
| Values | Loyalty & Obedience | Commitment to the concern's tasks and methods of material progress and expansion values more. |

You may observe from the diagram shown below that the Intelligent Design architecture inherently handles all the characteristic methodologies of dynamic systems. The most prominent of them being, the enablement of cohesive teams working towards a common goal or task of concern and an Enterprise nervous system connecting the people, processes and the products.

Let's move ahead from the process end of the spectrum to the people end of the spectrum. It is now imperative that we learn about two terms which may render our discussions much simpler:

1. **Intelligence Quotient** – is a metric to measure a person's intellectual capabilities in comprehending knowledge or structured information to be able to effectively utilize it.
2. **Cognitive Fluidity** – is a term to describe the ability of a person to learn and comprehend knowledge or structured information belonging to diverse domains and processes.

The human cognitive process occurs at two levels:

1. Through the educational systems at school, college and university.
2. In the enterprise space through research and work experience.

It has been observed that the gaps that exist between the cognitive process at school, college and between those in the industry are fairly large. It is near impossible for enterprise knowledge and information to flow down to educational systems in a systematic way primarily because of lack of suitable channels and secondly because much of the intellectual learning that happened at the industries research and development environment are classified as proprietary intellectual property.

We at STEX Advanced Design Bureau are attempting to bridge this gap in a small way by imparting a functional thought process that may find relevance at the industry level. This portion of our exercise involves making people smarter. We have already introduced the concept of functional thinking paradigm in the previous chapter. Our attempt to make people smarter involves two steps:

1. Better learning and cognitive process
2. Better communication ability.

**Better learning and cognitive process:**

Better cognition or learning involves learning to specialize in a particular domain and additionally building enough cognitive fluidity to master associated domains. For Example, a software engineer may learn to specialize in software programming/ development, as well as learn other associated disciplines like enterprise architecture and enterprise engineering.

STEX Advanced Design Bureau proposes the TEFS based functional learning techniques in the cognition process while starting from ground up at school. The reason being this functional and systems oriented approach facilitates an absolutely clear learning technique in a variety of subjects ranging from mathematics, science to engineering.

**Better communication ability:**

We are very aware that people are native speakers of a multitude of languages prevalent in diverse geographical areas. We might agree that English is a globally accepted language that's used in most forms of technical communication today. But could there be a more technical oriented language that connects people from a variety of fields and specializations?

Our approach to introducing a new form of communication takes root from the fundamental behavior of how diverse and disparate machine systems communicate through language neutral formats of data e.g. XML.

People in the computer science industry may well be aware that today's programming language compilers translate a variety of programming languages like C#, Visual Basic, C++, Delphi to intermediate forms of language called an intermediate language.

The TEFS or the INTELLIGENT DESIGN functions again prove worthwhile in the construction of an intermediate technical language that people may utilize to communicate and convey a very sharp context.

**The Intermediate Language Matrix:**

| | Design/ Structure | Execution | Data/ Storage | Risk | Optimization | Enabler | Rules/ Boundary | Exceptions |
|---|---|---|---|---|---|---|---|---|
| Design/ Structure | Design-Design | Design-Execution | Design-Data | Design-Risk | Design-Optimization | Design-Enabler | Design-Rules | Design-Exceptions |
| Execution | Execution - Designed /Structured | Executable | Execution Data | Execution Risk | Execution Optimization | Execution Enabler | Execution Rules | Execution Exceptions |
| Data/ Storage | Data Structure | Data Execution | Meta Data | Data/ Storage Risk | Data Optimization | Data/Storage Enabler | Data rules | Data Exceptions |
| Risk | Risk Design/Structure | Risk Execution | Risk Data | Risk of risk | Risk Optimization | Risk Enablers | Risk rules | Risk Exceptions |
| Optimization | Optimization structure / design | Optimization execution | Optimization Data | Optimization Risk | Optimization of Optimization | Optimization enablers | Optimization rules | Optimization Exception |
| Enabler | Enabler Design | Enabler Execution | Enabler Data | Enabler Risk | Enabler optimization | Enabler of Enabler | Enabler Rules | Enabler Exceptions |
| Rules/ Boundary | Rules Design | Rules Execution | Rules Data | Rules Risk | Rules optimization | Rules Enabler | Rules about Rules | Rules Exception |
| Exceptions | Exceptions Design | Exceptions Execution | Exceptions Data | Exceptions Risk | Exceptions Optimization | Exceptions Enabler | Exceptions Boundaries | Exceptions among Exceptions |

Let us, for example, set up a few dialogues that showcase an intermediate technical language:

**XYZ –** ABC! What is the **EXECUTION** status of project X?

**ABC** – XYZ, project X **EXECUTION** is 80% complete. However, the cost percent is 95% due to advanced payments.

**XYZ** – I heard there were two critical **EXCEPTION**s last month. Had they been handled?

**ABC** – Yes! The first critical **EXCEPTION** was a server fault due to a faulty hard disk. The second critical **EXCEPTION** was due to failure of the internet gateway. Both the **EXCEPTIONS** were handled in urgency adhering to all **RULES** of operations and also in adherence with all the **RISK** procedures.

**XYZ** - Other than the **EXCEPTIONS**, have you observed any **EXECUTION RISKS** in the application deployment and **EXECUTION** process?

**ABC** – We've collected all **EXECUTION DATA** and have not encountered even a single instance of **EXECUTION RISK**.

**XYZ** – Did you know, we've called in some IT Engineers to **DESIGN** the next module of the manufacturing process?

**ABC** – I heard that. We have the **DATA** ready. I guess they would be insisting on starting with the **DATA DESIGN.**

**XYZ -** Did you hear the latest story about the car crashes on Manhattan road due to faulty chassis?

**ABC** – Yes, I did hear. They must be taking in the cars for a **STRUCTURAL RISK** check. First report suggests a **DESIGN RISK** in the front suspension.

This ends our discussion of an intermediate technical language.

We now have insights into domain neutral and process neutral techniques theoretically explained with the help of TEFS functions or INTELLIGENT DESIGN functions. Additionally, we've also developed an intermediate technical language with the help of the TEFS function sets.

Our journey in learning has gone through the following discourse. In the chapter "Thinking Objects and Systems", we learnt about how to define a machine system and an enterprise system. At the end of the chapter we've set a goal to build a smarter world by making people, process and products ends of the spectrum smarter.

In the chapter "Eight Steps to brilliance", we learnt about the TEFS functions or the INTELLIGENT DESIGN function set and learnt its applicability in the systems space.

In this chapter we've fulfilled a part of our goal to build a smarter world by learning to apply the TEFS functional methodology to the process and people ends of the spectrum.

The next succeeding chapter would deal with making the product end of the spectrum smarter.

## Chapter 6

# Engineering and Design of Products

Having covered the people and the process end of the spectrum, we move to the product end of the spectrum where we use advance techniques to construct or develop smarter products.

Products are basically tangible or intangible systems. We have already learnt about the concept of object to systems polymorphicity. Products that are not overly complex can be simply called as objects. Every product that has been invented or built has a singular design goal. A complex product may have multiple functionalities to present a slightly complex design goal.

A car is designed to transport people on the ground.

A commercial Aircraft is designed to transport people by air.

While both cars and aircrafts transport people, their medium of execution differ i.e. one uses road as a medium while the other uses air.

Let's take another example.

A phone is designed to help people communicate with voice.

A smartphone is designed to help people communicate with voice as well as data.

We again notice that while both the products are used for the purpose communication, they differ in their medium of execution.

Therefore it becomes very necessary that we clearly define the design goal of the product that we plan to develop or create.

For ease of understanding, products can be classified into three types:

1. Physical/Material products
2. Software products and
3. Hybrid products.

Further, each of these products can be classified again as

1. Static products (objects or systems)
2. Dynamic products (objects or systems)

These two classifications result in the formation of a product matrix:

| | Static Products | Dynamic Products |
|---|---|---|
| Physical / Material Products | Physical Static products | Physical Dynamic products |
| Software Products | Software Static products | Software Dynamic products |
| Hybrid Products | Hybrid Static products | Hybrid Dynamic products |

Physical/Material products are products that are tangible, have a physical presence and their workings or creations are governed by scientific laws of physics, chemistry and engineering.

Software products are intangible products which run in a computer or server system and provide information/systems solutions in a specific domain or in a specialized process.

Hybrid products are products which have hardware as well as a software element to them. Examples of hybrid systems are smartphones and computers.

All products naturally go through the basic evolutionary cycles of design and execution. When constructing products in the physical world, the rules of science and engineering are so rigid that it inherently never tolerates any fundamental design flaws which go against the scientific laws. If flaws exist, they very "observably" show up as exceptions and failures.

Our approach to building physical world objects uses the TEFS functions or the INTELLIGENT DESIGN functions to cover all functional aspects of development or construction.

Let's for example, take the construction of a physical world building. Let us now use the TEFS functions or the INTELLIGENT DESIGN function set in the creation of the building.

**Structure/design** - Structural functions take care of all the materials and procedures required to create a stable structural foundation. Design functions cover all the user friendly aspects as well as the aesthetic and sanitation aspects.

**Risk functions** – Risk functions cover all the risk factors and security functions related to the building.

**Data/Storage** – The data functions cover all the information and data channels to the building. The storage functions cover all aspects of external storage spaces like warehouses for material storage.

**Enabler functions** - The enabler functions cover the aspects of suitable soil tested land availability worthy of construction activity as well as Power/Energy supply.

**Optimization functions** – Optimization functions cover all aspects of creating as much open space or work space within the limited boundaries of the building.

**Execution** – Execution functions cover tested material and construction procedures to erect a building.

**Exceptions** – Exceptions functions cover all the untoward incidents or events that may occur during building construction or execution.

**Rules/ Boundary conditions** - Rules/ Boundary conditions cover all the aspects of compliance to building construction norms.

You may notice that many formal design and construction techniques inherently adhere to some, if not all functions of Intelligent Design. While this is so noticeable in physical world products, the software world products do not necessarily follow such a pattern.

If we approach the building of software products through an Intelligent Design approach, then this software would naturally adhere to all principles of systems theory and such software may be called a software system. Software systems which do not follow or cover all the aspects of intelligent design may be called merely software.

Let us now move our discussion to hybrid systems.

Hybrid systems as we mention have software aspects as well as hardware aspects. But for a stable system to function, the software and the hardware must co-exist and co execute compatibly.

Ill designed software can easily ruin the best hardware systems. We may agree that while designing hardware systems or hardware aspects, some, if not all, rules of Intelligent Design are naturally adhered to while, this is not so in case of software. The reason being the design and structural flaws are more observable in physical or hardware systems than in software. It is for this reason that software bug is a very acceptable term while discussing software.

Another point to be noted is that hardware product design is a more cohesive activity than a software development activity because the surface area of flaws that may occur in hardware systems is comparatively less than while building software systems.

While building hybrid systems our approach should follow the process of intelligent design in both, the aspects of hardware and software.

**Design Sense:**

Design is a very frequently used term which very naturally conveys the meaning of creating a template or blueprint for a product, based on aesthetic scales. However, Design in the form of Intelligent Design is actually a rarity in this world because the intelligent design approach dwells into establishing the science of art or aesthetics.

Talking about the software world, we may agree that the software world has a very strong foundation with the usage of elements of abstraction in the software development cycles.

Abstractions very frequently signify an artistic aspect. Therefore, many a time, the activity of software development has been described as – “The Art of Programming”.

The STEX Advanced Design Bureau approach to building software suggests a far more scientific design behavior of software, a deviation from the artistic or abstract behavior. The reason being, art or abstraction is “not always meant to be understood”.

Many a time, we might have heard debates “comparing Windows operating systems to Apple Macs” or “Software as a science or an art”. Such debates necessarily use the art scale of measurement to differentiate.

If, however, software were developed as software systems, like hardware systems, the room for debates would narrow down to scientific scales of measurements like Execution Performance, Execution Risk or Risk mitigation ability etc.

**Interface based Technique and Visual Paradigm:**

We will now move to apply the interface based technique to product development.

Every object or system is necessarily perceived through its interface. Therefore, it may not be hard to convince that an interface is really the object. It is with this foundation statement that the software world is acquainted with the term or concept of “Interfaces”. For clarity in discussion, we address interfaces as object interfaces. Though the use of the word interface conveys user interface, we must be careful with the terminology. The user interfaces may be treated as explicit object interfaces for the purpose of showing graphics or design on a visual screen.

For those of you who are acquainted with the software terminologies may know the definition of an interface object which is nothing more than a collection of abstract functions, which gain definition and logic during class definition.

For those of you who are not acquainted with the software world, you may understand that an interface object in the software terminology is actually a file which contains a collection of functions that the object or the system is capable of doing. It is merely the depiction of an object or a complex system in the form of a set of functions.

Interfaces can be classified into two types:

1. Macro interfaces
2. Micro interfaces.

Macro interfaces are interfaces which contain macro functions or functions which are abstract in nature.

Micro interfaces are interfaces which contain functions at a micro level or a detailed level.

You may observe that in the present stage of technical progress, design systems like Computer Aided Design (CAD) or Computer Aided Machining (CAM) use the technique of translating software blueprints into physical world products/objects. It is because of this principle, that we are essentially discussing this interface based approach to designing far more precise, sophisticated and yet simpler products.

Let us now move ahead in describing a few real world interfaces.

Example of a Macro interface: A car engine.

The macro functions that the car engine interface contains are as follows:

| |
|---|
| Start() |
| Stop() |
| Accelerate() |
| Decelerate() |

Now a micro interface of a car engine is as follows:

| |
|---|
| cranking() |
| Fuel_injection() |
| Air_intake() |
| Compression_cycle() |
| Sparkler_ignition() |
| Combustion() |
| Piston_cycle() |
| Exhaust() |
| Accelerated_engine_cycle() |
| Decelarated_engine_cycle() |

Now let's discuss an Aircraft interface.

| |
|---|
| Aircraft_taxing( ) |
| Aircraft_Takeoff( ) |
| Aircraft_fly_forward( ) |
| Aircraft_dive( ) |
| Aircraft_roll( ) |
| Aircraft_yaw( ) |
| Aircraft_carry_goods( ) |
| Aircraft_seat_passengers( ) |
| Aircraft_maintain_ambient_temperature( ) |

We have now learnt the concept of defining an interface for any product or system. Our goal now is to make these interfaces smarter using the Intelligent Design techniques.

The intelligent design approach uses the “what you see is what you get” principle to design fully factorized interfaces. Please note that we used the term fully factorized interface to convey every aspect of the systems functionality.

In the interfaces that we have defined above, the interface functions are ad hoc in nature and there is no way for us to decipher whether the function was an execution function, a data storage function or a risk function.

As described in the product matrix, the systems may be classified as static or dynamic. This suggests that the interfaces that build up to a system may also be static or dynamic in nature. Let us take an example of a dynamic system using the example of an aircraft.

The wings of an Aircraft in today’s technology can be classified as dynamic systems. This is so because the aircraft wings are designed to dynamically increase or decrease surface area of the wings during takeoff or landing.

Similarly, automobiles or cars may have the functionality of retracting and folding away the car roof as in a convertible.

This leads us to the concept of fully factorized and free scaling interfaces. In simple words, free scales are the dynamic functionality added to the functioning of any system or product. Free scales are dynamic scaling ability of the intelligent design functions defined in any interface.

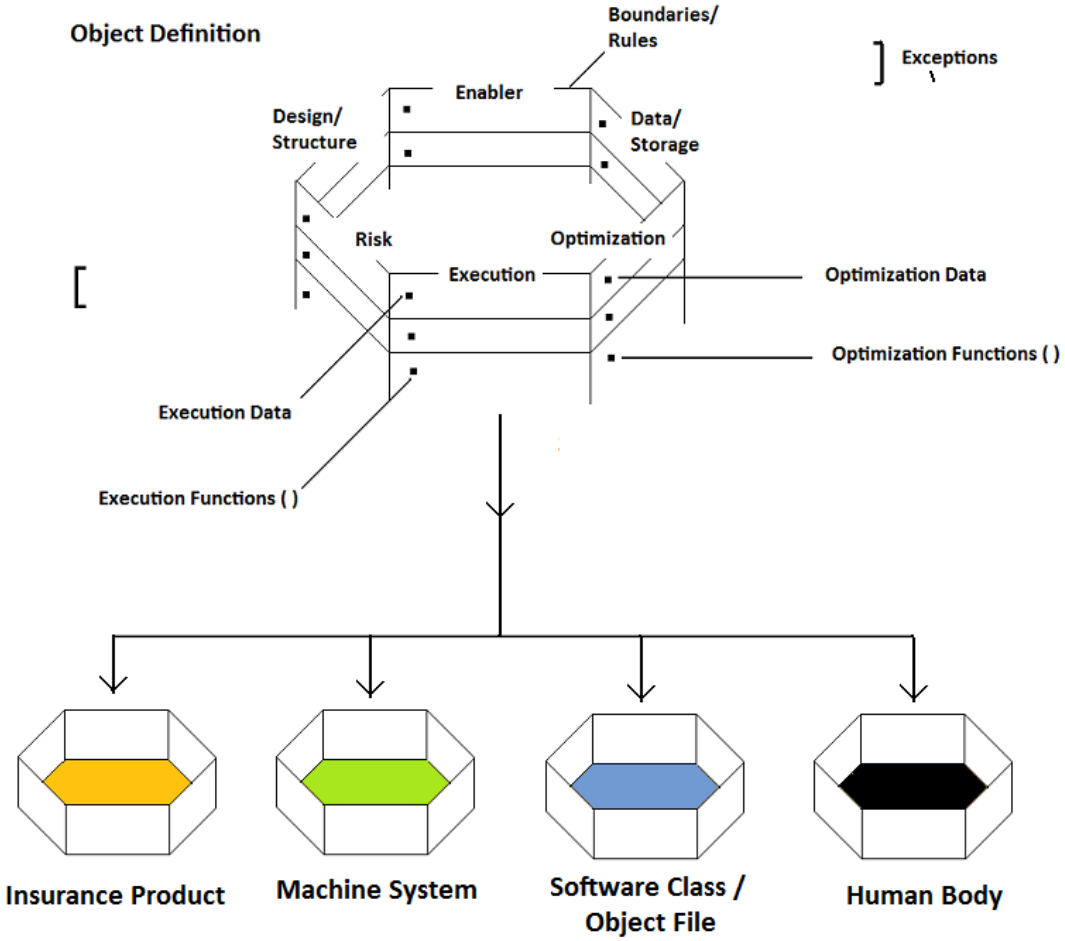
E.g. A server/Computer system may have free scaling / expandable memory slots to accommodate more memory for the hardware as and when the need for scalability arises

Similarly Boot space of a car may be increased dynamically by folding away seats.

**Polymorphicity of objects/interfaces:**

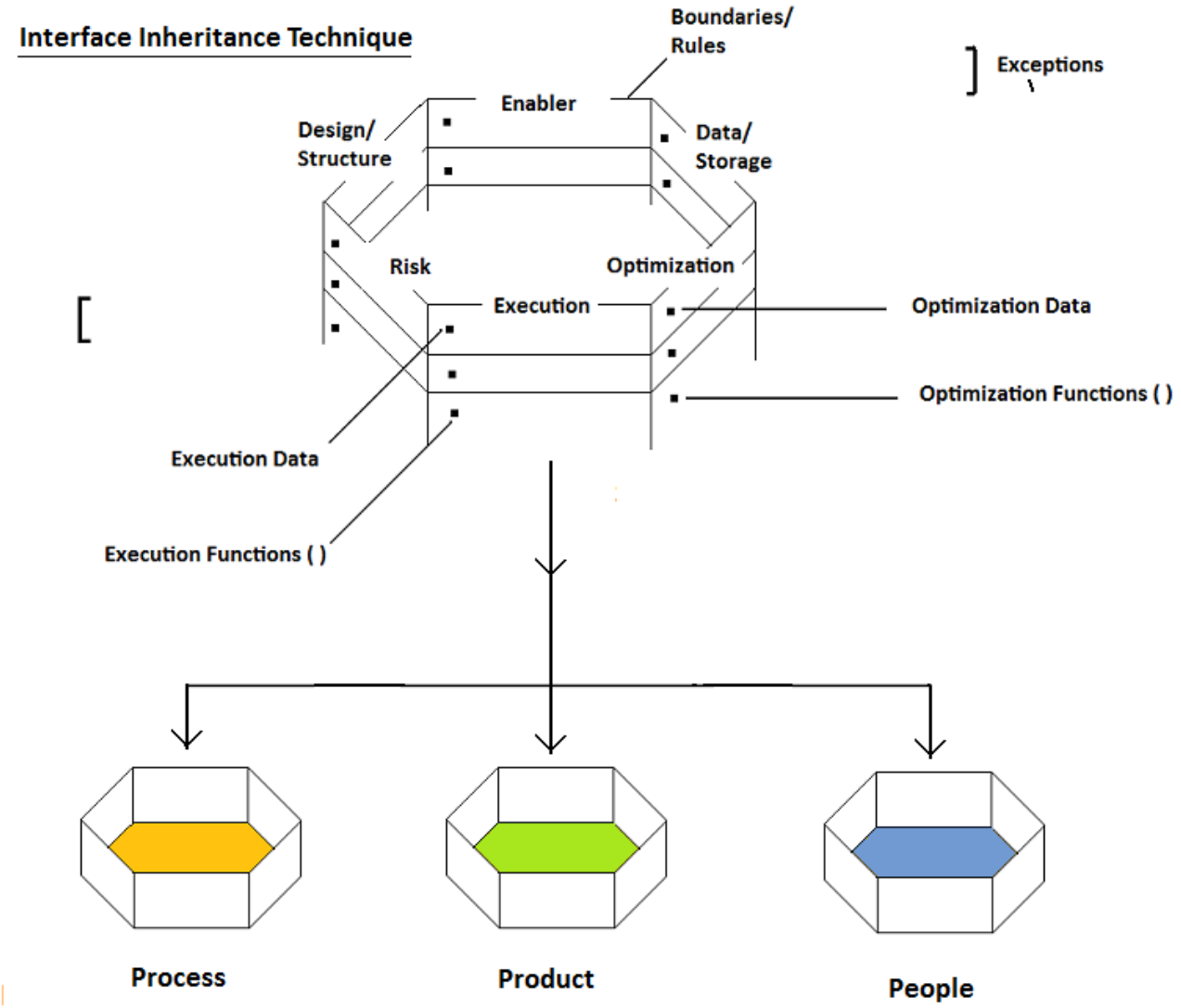
Those acquainted to the computer science world may be aware that in the object oriented world, all objects inherit from the parent or base object. This is true even in the case of the interface based technique that we are presently discussing.

The Intelligent Design fully factorized free scaling interfaces form a formal interface or the base interface for every or any kind of product or system.




Polymorphicity of objects/interfaces

We have already discussed how the TEFS function sets or the intelligent design function sets are used in the people and process ends of the spectrum. This interface technique may aptly represent how this Intelligent Design interface describes all the three ends of the spectrum. The pictorial representation is given below




Interface Inheritance Patterns

Let us now create a few interfaces based on the fully factorized functionality of the intelligent design interface. The simple example of the car engine interface that we discussed earlier translates to the following:

The fully factorized car engine interface:

```
System.Execute.Start()
System.Execute.Stop()
System.Execute.Accelerate()
System.Execute.Decelerate()
```

The full factorized interface of software that draws a circle on the computer screen is as below:

```
System.Design.Draw_Circle( )
System.Data.Fetch_Radius ( );
System.Execution.Database_Call ( )
System.Execution.Use_Connnection_Pooling ( )
```

Now let's use the fully factorized interface of a software web application (Macro Interface):

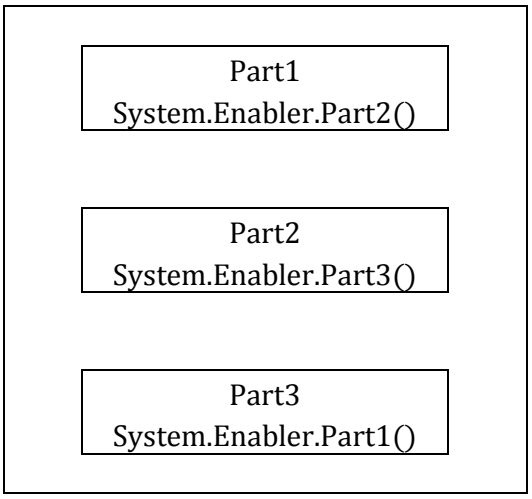
```
System.Design.Web_form_design ()

System.Data.SQL_Query()
System.Data.Stored_procedures()

System.Execution.Project_compilation ()
System.Execution.Project_build()

System.Enabler.Visual_studio_IDE()
System.Enabler.DotNet_software_framework()

System.Optimization.Program_code()
System.Optimization.Queries()

System.Risk.Security()
System.Risk.Roles()

System.Exception.Exception_circumstance_handling()

System.Rules.Business_logic()
```

We have now dealt with interfaces of a variety of types. Now we move ahead into discussing the build or construction of systems which consists of individual, interdependent parts.

The theory of reductionism is an approach that is normally used to understand the nature of complex systems by reducing them into interactions of the subparts. The Intelligent Design functions very easily handle reductionism because they have a built-in function type called the enabler function.

In a hypothetical machine with three constituent parts, if part1 is dependent on part2 and part2 is dependent on part3 and Part3 is dependent on Part1, the interface enabler functions would be as follows:

Complex Machine with three constituent interdependent parts:




**Verifications with Science:**

We have already learnt about the concept of entropy in one of our previous chapters. The phenomenon of entropy is the process by which disorder sets in naturally in any system. We also learnt about the multiplicity effect, where the probability of throwing a dice and achieving a seven is far more probable than achieving a two.

The Intelligent Design approach is a value added approach similar in analogy to achieving a two with a throw of dice (i.e. there is one way to the best solution).

Products normally follow evolutionary iterative cycles of development. And this leads to a process where there is a chance or probability of missing aspects or functionalities conveyed by the intelligent design approach.

This approach is normally associated in analogy with the throw of dice and achievement of a total of seven in a multiple ways. The end product could be characterized by elements of disorder or complexity.

Therefore we can conclude that the intelligent design approach helps in systematically tackling the disorder forces and allowing the building of sophisticated interfaces which are naturally very revealing about their functionalities.

These functions would thus lead to building smarter and far more systematic products.

**Conclusion:**

We have now come a long way in our learning exercise – having dealt with the three core ends of the enterprise spectrum viz., People, Processes and Products.

In our successive chapters, we would be outlining various examples of smarter systems that may be built in the distant future.

## Chapter 7

# A Brief History of Time – A Historical Analysis of Innovation

Until now, we have been discussing Intelligent Design techniques to build smarter people, processes, products and smarter enterprise ecosystems. This can be summarized by a single word – Innovation.

This chapter deals with analyzing a brief history of innovation or rather the evolution of modern life per se.

Innovation is a term we frequently used to represent or identify inventions, but it may not always be the case. The process of bringing about a change by introducing new processes or methods, products and higher forms of communication and thought in individuals is a more apt definition of Innovation. We may recollect that we used the term fluctuation functions that signify an advent of change.

But change itself is not a new phenomenon. Change can be more aptly put forth as constant phenomenon over the ages or cycles of time. We as stated, dwell into historical insights of great changes that took place in society over a period of time. The eventual purpose is to go back in time and celebrate the past, because that's the only way that can prepare us for an appreciation of the future.

STEX Advanced Design Bureau is only attempting to enlighten its readers to recognize these forces of change and align with them for the primary goal of evolution of the individual and the society on the whole.

We may make a note that change can easily be classified as evolutionary or degenerating in nature. Evolutionary change is usually accompanied by forces that make the world we live in a more systematic system. The degenerating forces are those forces that do not have substantial purpose or a higher goal of contributing to the system on the whole.

Although man is classified as a social animal and social percussions have a very strong role in human society, individual perseverance and pursuit have for long contributed to evolution of the masses in societal systems.

From a historical perspective, the above statement stands true very strongly because the innovations that took place then could be achieved by singular effort. However in the modern age the innovations and systems that are created today are so complex that it requires individual as well as a team effort in the process of creation.

While we could easily identify the inventor of a ball point pen, it is nearly impossible to identify the inventors of the computer. For complex systems like computers were invented over a period of time by contributions from many individuals.

Before we go into studying historical timelines of innovation, let's identify a few critical innovations that change the world and impacted society in a very strong way. Research by FPRI had concluded the most important innovations of all times.

1. True semantic, syntactic and phonetic languages that allowed humans to communicate and co-operate with each other.
2. The taming of fire allowed humans to inhabit cold places, work after dark, scare away wild animals and cook food.
3. Simple tools that allowed people to multiply their mechanical effort to getting the work done.
4. Farming was a long chain of activities that involved growing crops and domestication of animals.
5. Clothing permitted people to protect themselves from inclement weather.
6. Earthenware and pottery when ancients discovered hardened clay in their fire pits.
7. Metallurgy permitted people to make hardened tools and spears.
8. Food preservation meant humans to store the food for later consumption.
9. The system of writing that made possible written records and calculations.
10. Religion shaped up human culture and behavior.
11. Invention of wheel changed how man could move goods from one place to another.
12. The manufacture of paper led to explosion in publication and spread of knowledge.
13. Specialization of labor work force classified people into various skilled classes.
14. Discovery of fossil fuels permitted people to harness energy.
15. The concept of science came in only by the 19th century before which it was only considered as natural philosophy.
16. Harnessing of water power or water mills permitted one to carryout heavy work.
17. The invention of steam engine multiplied man's ability to do work and made practical the activity of coal mining which provided fuel for the steam engines.
18. Electromagnetism led to the harnessing of electricity and the running of electric motors.
19. Diesel and fuel based engines converted potential energy stored in fuels to work.
20. The gas turbine engines allowed propulsion of large transportation systems like ships and aircrafts.
21. Invention of computers created the foundation of a digital world which runs in parallel to the physical/material world.
22. The advent of internet that allowed information to flow freely across geographical distances.

This analysis gives us a fair glimpse of how the world changed over a period of time and gives us insights about how the future could play out in due course.

We could conclude that from a few of the points above, the most important directions of innovation that changed the world over a period of time could be classified as:

1. Information and communication.
2. Energy and utilization.
3. Invention of transportation.
4. Creation of the digital world.

We now study the timelines of each of these individual directions of innovation to draw conclusions and, if possible, extrapolate to the future.

**Historical timeline of Information and Communication:**

3500 B.C. - Phoenicians develop and alphabet. Sumerians develop cuneiform writing - pictographs written on clay tablets.

1775 B.C. – Greeks use a phonetic alphabet written from left to write.

1400 B.C. – Record of writing in China on bones.

1270 B.C. – The first encyclopedia is written in Syria.

900 B.C. – The first postal service for government use in China.

776 B.C. – First record of homing pigeons used to send message.

530 B.C. – Greeks started the very first library.

500 B.C. to 170 B.C. – papyrus rolls and early parchments made of dried reeds – first portable and writing surfaces.

200B.C. to 100B.C. – Human messengers on foot or horseback common in Egypt and china.

14 A.D. – Romans established postal services.

100 A.D. – First bound books.

105 A.D. – Tsai of China invents paper as we know it.

305 A.D. – First wooden printing press.

1450 A.D. – News papers appear in Europe.

1455 A.D. – Guttenberg invents the printing press.

1560 A.D. – Camera Obscura invented – primitive image making.

1650 A.D. – First daily newspaper.

1714 A.D. – The invention of the typewriter.

1793 A.D. – First long distance semaphore telegraph line.

1814 A.D. – First photographic image.

1821 A.D. – First microphone.

1835 A.D. – invention of Morse code.

1876A.D. – The invention of the telephone.

1888A.D. – First role film camera.

1899 A.D. – first recordings using magnetized steel tape.

1910A.D. – First talking motion picture.

1923A.D. – First television camera.

1925A.D. – First experimental television signals.

1944A.D. – Computer like *Harvard's mark I* – the age of information science begins.

1969A.D. – The first internet started – ARPANET.

1979A.D. – First mobile phone.

1994A.D. – World Wide Web (WWW) taking communication at light speed.

We may note that it was form of communication as well as the medium of communication that changed over a period of time. If we extrapolate the patterns of communication into the future, we are likely to find only two possible forms of change.

1. Likely change in Format of information / Form of communication.
2. Likely Change in the Medium of communication.

The format of information of communication that exists as of today exists in the form of language, written and spoken. The STEX Advanced Design Bureau Intelligent Design methodology is working on deriving a new form of intermediate technical language that we've already suggested. The format or structure of the information that we convey through education and learning, in a variety of subjects could also use the intelligent design methodology.

As for the medium of communication is concerned, the world is moving ahead into digital classrooms and Visual communication, further extrapolations may lead to more sophisticated digital mediums of communications.

STEX Advanced Design Bureau however suggests that great value added work lies in the conversion of unstructured forms of information and theory into structured, intelligently designed theory. It may also be noted that if the format of information or theory is closer to a functional based approach as suggested by the Intelligently Designed theory of intermediate language, the process of conveying or relaying this structured information through a digitized medium may be far more feasible.

**Historical timeline of Transportation:**

3500 B.C. – First wheeled vehicles in history.

3500B.C - River boats with oars.

2000B.C. – Domesticated horses used for transportation.

770 A.D. – iron horseshoes improved transportation by horse.

1492A.D. – First illustrated theories of flight by Leonardo Da Vinci.

1783A.D. – Invention of the first steam boat.

1783A.D. – Invention of Hot air balloons by Montgolfier brothers.

1790 A.D. – Modern bicycles.

1801A.D. – First steam powered locomotives.

1862A.D. – Gasoline engine automobile.

1867A.D. – The first motor cycle invented.

1885A.D. – The first practical automobile to be powered by an internal combustion engine.

1903A.D. – The first engine airplanes.

1926A.D. – Liquid propelled rockets.

1940A.D. – Modern helicopters invented.

1947A.D. – Supersonic Jet flight.

1970A.D. – First jumbo jet to carry passengers.

1990A.D. – Electric hybrid automobiles.

Having seen the timelines of innovation and transportation, it may be inferred that the transportation products domain is now fairly matured with availability of very efficient engines and transportation systems. With further research in the Intelligent Design approach, superior transportation systems could be developed. .Again, transportation can be divided into:

1. The core product / medium of transportation e.g. Cars, rail, flight.
2. The processes that govern the Transportation system.

The Intelligent Design approach could easily work out new structured processes that optimize the entire domain of transportation processes. We may note that algorithmic flight scheduling and rail system scheduling already exist in many developed nation systems, however there is still scope for an intelligent design approach in optimizing transportation and traffic systems.

**Historical timelines of harnessing of energy:**

1500 B.C. – Geothermal power used for cooking, bathing and heating.

200B.C. – Mining of coal.

644A.D. – Vertical axes windmill recorded in Iran.

1765A.D. – Invention of steam engine.

1780A.D. – Coal is used to generate electricity.

Till 1800 A.D., mankind's use of energy relied of muscular and biomass sources. Energy was provided by manual labor, animals and biomass (wood).

1800 A.D – Invention of the Battery

1821A.D. – First natural gas drilling in New York.

1885A.D.- Invention of Gasoline powered automobile engine.

1890A.D. – Solar energy used to run a printing press.

1890A.D. – First operating wind turbine.

1891A.D. – Invention of Tesla Electric coil by Nicholas Tesla.

1892A.D. – Invention of Diesel engine.

1938A.D. – invention of Nuclear fission (Otto Hann and Fritz Strassman).

1944A.D. – First nuclear reactor.

1950A.D. – First alternating current wind mills.

An analysis of timeline of energy reveals maturing of energy producing techniques capable of generating enough power to sustain working of large enterprises, societies and nations. Scope for innovation in the field of Energy could be classified into:

1. Production / Generating techniques.

2. Energy consumption patterns and processes.

The Intelligent Design techniques as suggested can help in finer research in the energy generation space, but could be far more effective in improving energy consumption patterns or processes.

We have discussed three critical timelines of historical innovation that covered the different ages of history viz., The Early age, The Medieval age and The Modern age. The present age that we live in today is often addressed to as the information age.

If patterns of history could convey what the future beholds, we could well be moving into what could be called as The Intelligent Age or The Systems Age.

STEX Advanced Design Bureau is envisioning the advent of a systems age, or more commonly a smarter world.

The systems approach concentrates on structured and intelligent governance of systems and its constituent entities in a manner, so as to allow a methodical, structured and sustainable future, a future that encourages multiple enterprise systems co-existing in equilibrium, through standard methodologies and protocols of techniques and behavior.

Having discussed the nuances of intelligent design and historical consciousness, we are in a position to successfully prepare for the times to come.

## Chapter 8

# A Systematic World Space: Patterns of systematic Innovation

For the purpose of a systematic comprehension of the world, we have already discussed the people, the process and the product ends of the spectrum. However, the comprehension is incomplete without a chapter on systematic innovation that deals with innovation at the people level, the product level and the process level and is measurable through a generic framework of innovation.

Industry always faced a challenge of each industry measuring innovation in its own way. It is because of this reason that there are no standard metrics that define innovation or measure innovation. STEX Advanced Design Bureau has developed the innovation framework based on the intelligent design theory or the TEFS function set that gives a standard framework for the enterprise in any domain to measure its level or accomplishment in innovation.

Locating a particular place in a geographical map requires two co-ordinates, the latitude and the longitude. Similarly, the procedure for measuring innovation involves two core steps:

1. Identifying the innovation space that the enterprise occupies.
2. The level of or the degree of innovation in < products, people and processes space >.

### Identifying the innovation space:

To evaluate the first point of identifying the innovation space, we outline two simple matrices. They are as follows:

| |
|---|
| Products, OEM (Original Equipment Manufacturers) |
| Services |
| Trade or distribution |

The Enterprise Matrix

The enterprise matrix divides an enterprise into three distinct types:

a) Product/OEM – Companies which build products or equipment for use by public or other enterprises.

b) Services – Companies that offer some form of service e.g. accounting services, financial services, software services or hotel accommodation services.

c) Trade or Distribution – Companies that distribute products from manufacturers to the public or other enterprises e.g. E-Commerce stores, retailers, distributors, shops.

| Enterprise | Software/Digital world | Physical world |
|---|---|---|
| People | a | b |
| Process (internal or external) | c | d |
| Products | e | f |

The Enterprise Medium – Software World or Physical World

The interpretation of the above Enterprise Medium matrix is as follows:

a) Enterprises that have "people" as its core entities e.g. Social networking and social media sites like Facebook, Twitter, Orkut etc.

b) Enterprises that have "people" as its core entities, but conduct their exercise in the physical world by group gathering and one to one, in person meeting.

c) These are enterprises which have "process" as their core entity of offering, in the software world.

d) These are enterprises which have "process" as their core entity of offering, in the physical world.

e) These enterprises are "product" companies which build software products e.g. Microsoft, Apple etc.

f) These enterprises are "product/OEM" companies which make products which are tangible, in the physical world.

We use the first matrix (the Enterprise matrix) and the second matrix (The Enterprise Medium matrix) to pinpoint the space that the enterprise occupies in the systematic world map.

**The level or the degree of innovation:**

Having identified the innovation space that the enterprise occupies, we move into the process of identifying the degree of maturity of innovation of the individual entities - people, processes and products.

To solve this, we outline the second degree intelligent design or the TEFS functions and call the same as Innovation matrix.

This innovation matrix forms the framework that we mentioned earlier, that can be used to measure innovation in different domains and enterprise spaces.

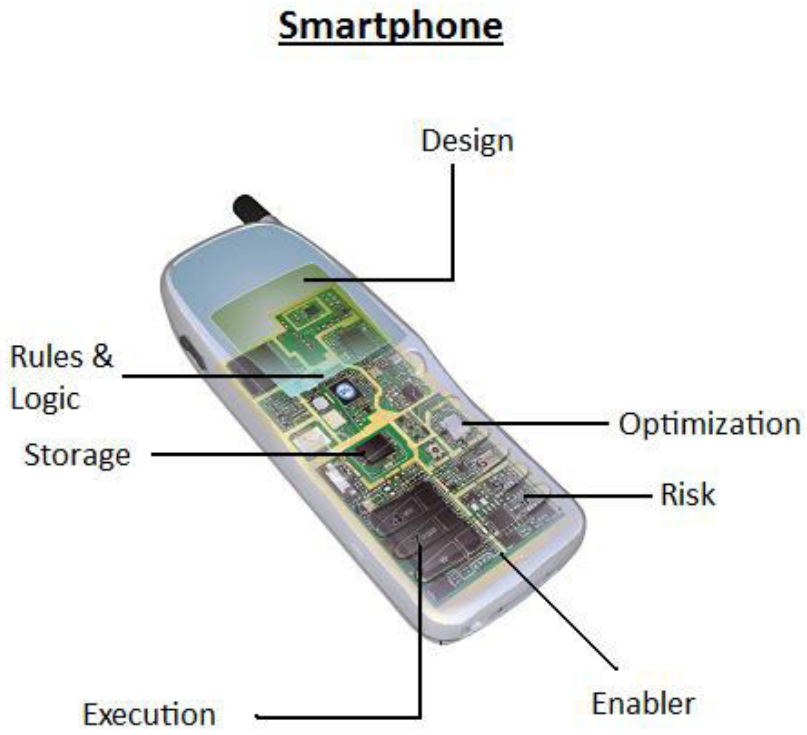
Systematic Innovation Framework

| | Design/ Structure | Execution | Data/ Storage | Risk | Optimization | Enabler | Rules/ Boundary | Exceptions |
|---|---|---|---|---|---|---|---|---|
| Design/ Structure | Design-Design | Design-Execution | Design-Data | Design-Risk | Design-Optimization | Design-Enabler | Design-Rules | Design-Exceptions |
| Execution | Execution-Designed/Structured | Executable | Execution Data | Execution Risk | Execution Optimization | Execution Enabler | Execution Rules | Execution Exceptions |
| Data/ Storage | Data Structure | Data Execution | Meta Data | Data/ Storage Risk | Data Optimization | Data/Storage Enabler | Data rules | Data Exceptions |
| Risk | Risk Design/Structure | Risk Execution | Risk Data | Risk of risk | Risk Optimization | Risk Enablers | Risk rules | Risk Exceptions |
| Optimization | Optimization structure/ design | Optimization execution | Optimization Data | Optimization Risk | Optimization of Optimization | Optimization enablers | Optimization rules | Optimization Exception |
| Enabler | Enabler Design | Enabler Execution | Enabler Data | Enabler Risk | Enabler optimization | Enabler of Enabler | Enabler Rules | Enabler Exceptions |
| Rules/ Boundary | Rules Design | Rules Execution | Rules Data | Rules Risk | Rules optimization | Rules Enabler | Rules about Rules | Rules Exception |
| Exceptions | Exceptions Design | Exceptions Execution | Exceptions Data | Exceptions Risk | Exceptions Optimization | Exceptions Enabler | Exceptions Boundaries | Exceptions among Exceptions |

In conjunction with this matrix, we also use a measurement scale to measure the degree of each function of innovation.eg 70% Execution optimization or 10% Execution Risk.

Let us work out a few examples to see how the systematic innovation framework helps in the measurement of innovation and outlining new directions of innovation. We discuss systematic innovation in the product space and the Enterprise space.

**The systematic innovation approach to building a smartphone:**




**Execution** – Building a smartphone with a faster processor.

**Execution Optimization** – Building a smartphone using an optimized power saving processor.

**Storage** – Building a smartphone with higher capacity of data storage e.g. 1GB, 2GB, 4GB etc.

**Design** – Building a smart interface for the phone like a tiled interface or a touch interface etc.

**Structure** – Building a smartphone form factor with a variety of blends of plastic, carbon fiber or metal.

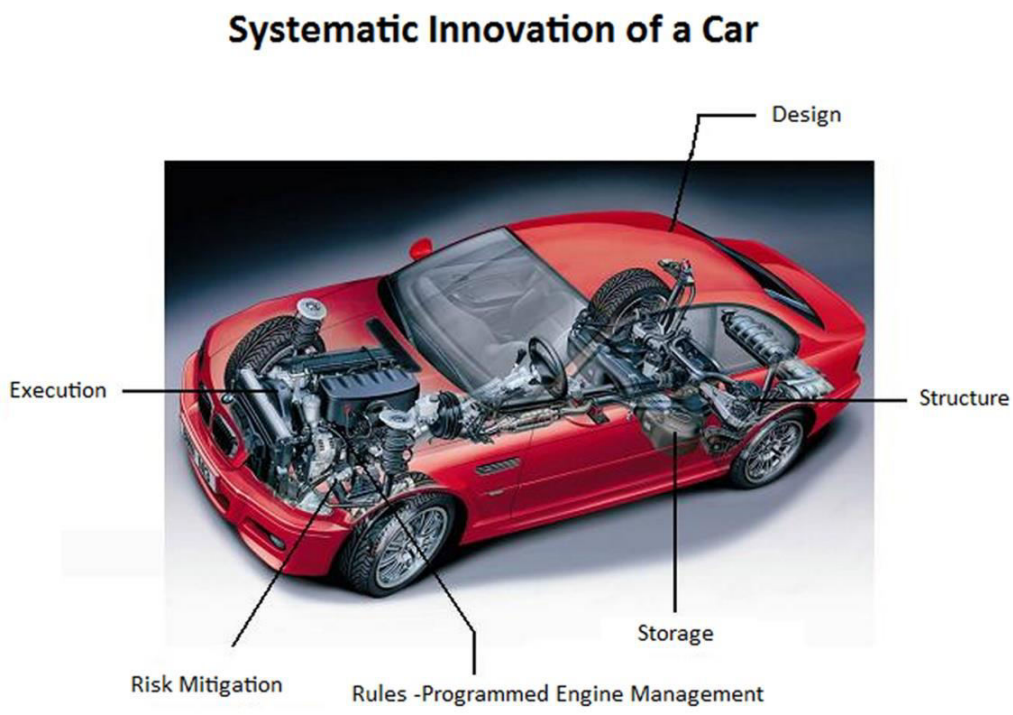
**Risk** – Risk mitigation in a smartphone by incorporating bio-metric security for authentication.

**Enablers** – Building a smartphone with long lasting, high capacity batteries.

**Rules** – Building a smartphone with proprietary or open operating systems that run all the system logic and functionality.

**Risk Execution** – In case of smartphone theft, the system would lock up and send alerts.

**Example 2 - Systematic Innovation of a car:**




**Execution** – Building a car with more sophisticated, powerful engine.

**Execution Optimization** – Building a car with a fuel efficient engine.

**Storage** – Fuel storage and boot space or luggage storage space.

**Design** – The exterior design of the car.

**Structure** – Building the car chassis out of sophisticated metal alloys or carbon fiber reinforcements.

**Risk** – Design crumple zones in the front and exteriors.

**Rules** – Efficient engine programming and management processors.

The innovation process can be classified into:

1. Fundamental out of the box innovation.
2. Systematic Iterative innovation.

Fundamental out-of-the-box innovation involves building a breakthrough product from scratch while, systematic innovation involves innovating in cycles or iterations to bring out the next version of the product.

Let us analyze the software world space to get a perspective on how things have evolved systematically over a period of time.

Talking about computer programming languages – languages were initially unstructured (unstructured execution) succeeded by procedural and structured programming languages (Structured

execution). It was further followed by the object orientation methodology. If patterns could forecast a future - it could well be an “object” programming paradigm in the near future.

Talking about evolutions of nations – Nations were formed after long histories of war. People learnt to form Governments and created law. Nations evolved to build national security. Nations built education systems to spread knowledge. Industries and Enterprises were built over time. Nations engaged in global Commerce. If patterns could again guide a future – it could well be the formation and governance of “nation systems”, through sophisticated frameworks of intelligent design.

Today, data analytics and data search is a common place term. However, the data or information that is dealt with is largely in the form of unstructured data. A transition or evolution in data spectrum would extrapolate to analytics and search of “structured” data. An example of a structured data paradigm is represented by the Wikipedia.

Search engines like Google, Bing operate on unstructured data. STEX Advanced Design Bureau is currently working on a world space intelligent search engine that operates on structured data.

According to the systematic Innovation framework, progress in innovation could be possible in any of the directions guided by the 64 Elemental Functions.

In Enterprise systems – Innovation could take place in the redesigning of the rules of the enterprise.

Avionics systems could be programmed to allow structured exception handling sequences in cases of emergency.

Search engines could operate on meta-data (Data about Data) instead of operating purely on data.

Project Management systems could encompass the entire spectrum of intelligent design activities defined in the Innovation function set.

Thus we can conclude this chapter by inferring that innovation is the process of inventing the next new by breaking the boundaries set by the previous iteration.

Innovation actually follows patterns and trends represented by the innovation matrix and applications of the innovation matrix in specialized domains may yield systematic results.

## Chapter 9

# Rudimentary Economics and Financials

Our discussions till now have been following a systems perspective in an analysis of the world. It goes unsaid that commerce and finance are equally important functions related to the enterprise space that usually gather attention as subjects of study in courses like business studies, economics etc.

From the systems engineering angle, these courses and fields of study are classified as abstract fields of study. Although subjects like economics used in conjunction with subjects like statistics have a slightly higher basis of scientific foundation, they are still considered imprecise by many factors. The reason being, these subjects deal with heterogeneous data OR Imprecise data.

The intelligent design approach or the systems engineering approach to studying economics and commerce follows one single tenet that, it classifies and structures data, to a synthesized smarter data, that eventually leads to precise measurements of the state of a system.

The world that we live in is actually classified into a variety of ecosystems or systems like the enterprise system or nation system. Each of these systems actually follows very scientific working principles of nature for their execution and governance. In other words, a system runs well when people, processes and products follow efficient trends of evolution in execution and are governed by systematic rules of governance.

Any deviation or inefficiency in processes, people or products leads to a deterioration in the working of the system. It is because of this reason that we observe breakdowns in multitude ecosystems resembling a phenomenon called a recession. We may highlight that enterprise systems and public systems are interdependent for their mutual growth (intellectual growth as well as commercial growth). Whenever, either of these spaces, the enterprise system or the public system undergoes significant deterioration, they lead to a cascading breakdown in the world ecosystem. This is a systems interpretation of the economic term recession.

This ultimate goal of any system therefore, translates to operating in equilibrium with other interdependent systems. It is with this reason that a systems engineered enterprise differs very largely from a business enterprise. Business enterprises allow room for human emotions like greed and capitalism to prevail over the larger principles of eco-systemic equilibrium, while systems engineered enterprises allow no room for forces that instigate deterioration in equilibrium.

**The world of Finance:**

Since finance is the foundation of commerce, it is an imperative subject for one and all occupying the enterprise space or the public space. Though finance is extremely complex subject of study today, a systems engineering approach simplifies finance into three simple categories:

1. Product Finance.
2. People Finance.
3. Process Finance.

Product finance deals with all the cost price in building a product and selling price of the product.

People finance deals with cost of people required to building products.

And process finance deals with financial aspects associated with the processes that lead to the manufacture and supply of products.

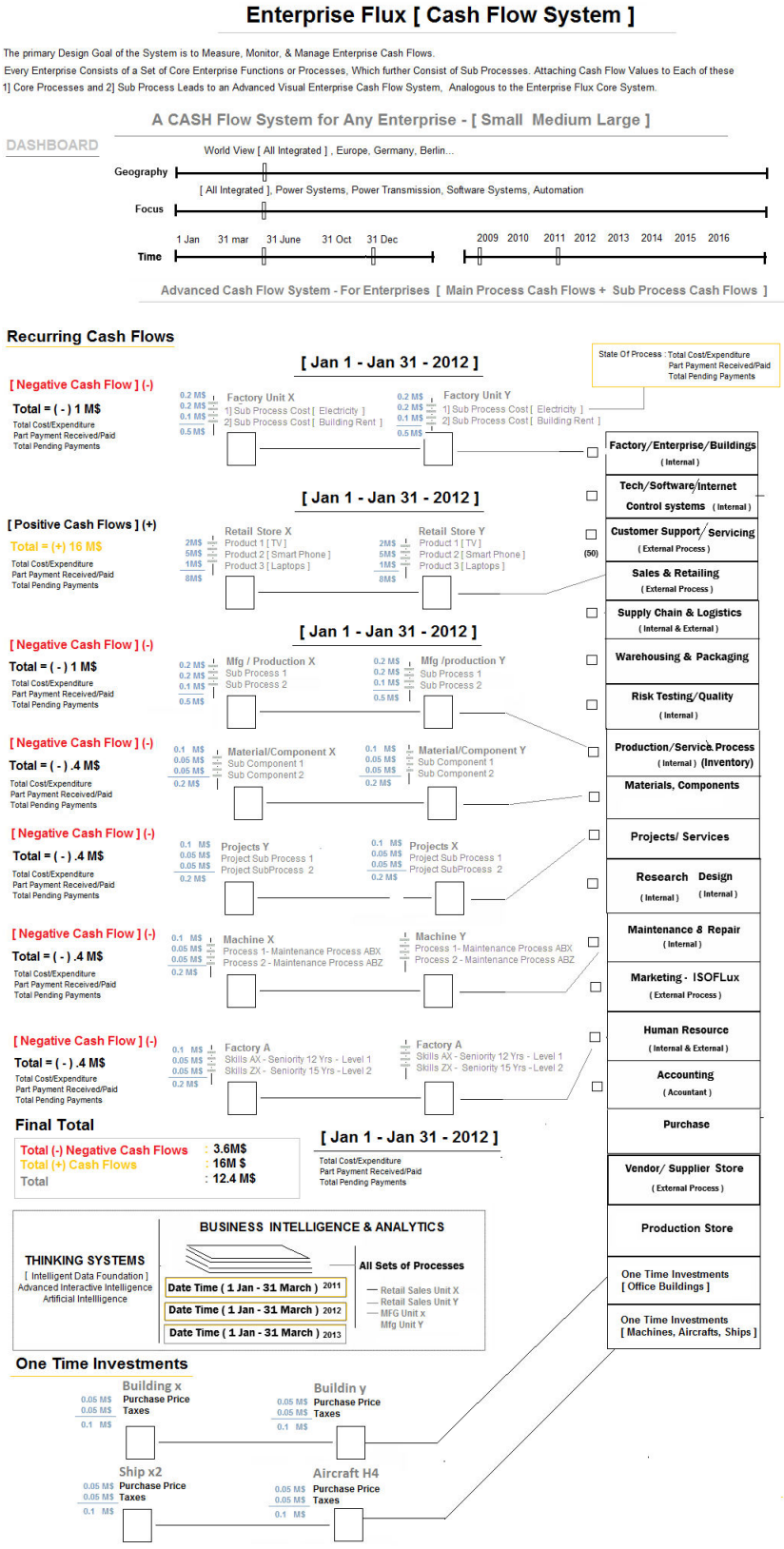
**Let us take the building of a computer as an example.**

Product cost – cost of individual components.

Process cost – Cost of the assembly process.

People cost – Cost of the employing people for various tasks.

The summation of these three costs gives us the total input cost. The price that the product sells at multiplied by its volume gives us the output cost. The difference between the input cost and the output cost is the amount of wealth created by the enterprise.




A simple analysis of the stock market principle leads us to infer that the price of a stock should have relevant dependence on the payout of the enterprise rather than on the notional values or price to equity multiples that the stock sells at or is valued at.

From a systems engineering perspective, terms like market capitalization and wealth created are false financial jargons. The only true measurement that ultimately qualifies is the amount of cash acquired and amount of cash paid back by an enterprise. In this context, if an enterprise promises not to give away any payback, the value of the stock in the stock market can only be considered notional. There is an alarming trend of firms in the world today who do not contribute to paying back to the public systems.

Sometimes, it is these sorts of imbalanced working or governance principles that lead to deterioration of commercial wealth in the world ecosystem. E.g. Thus if a company borrows debt and never pays back, the public system that loans the debt may find itself in a financially deteriorated position, unable to further support newer capital to upcoming enterprises.

It may also be inferred that notional wealth or market value of stock prices is only created by successive generations of investors forcing up or escalating stock prices rather than the payout from an enterprise system doing so.

Keeping this in mind we wind up our rudimentary yet complete analysis of economics and finance and understand all the important governing parameters that lead to ecosystem stability or instability.

## Chapter 10

# Outlining the future – Array of Innovative Possibilities

This chapter is a collection of possible areas of innovation that could possibly be demonstrated in the future. Since STEX Advanced Design Bureau does not have specialization in each of the possible areas of innovation, it is only outlining a template or guiding theme. The areas that are outlined are as follows:

- ERP Systems – Multi Domain / Multi Discipline.
- System Software – Development e.g. compilers, operating systems.
- Object Programming Paradigm – Extension to OOP Methodology.
- Object Based Database Systems.
- Game Design.
- Artificial Intelligence.
- Computer Hardware Systems.
- Applications in Government/Nations Systems.
- Standards framework.
- Applications in Machine Systems.
- Enterprise to enterprise, Enterprise to public Digital engagement.
- Digital Economic Lifeline Systems

**ERP Systems – Multi Domain / Multi Discipline:**

According to our systems engineering theory, we have divided the enterprise space into people, processes and products. ERP systems are enterprise resource planning systems which help in managing the processes of an enterprise. ERP systems as of today manage specialized processes like human resources, payrolls, manufacturing etc.

Future ERP systems are envisioned to be not only domain neutral but also process neutral i.e. the ERP systems could fit into any kind of industry and any kind of process. We have already demonstrated how domain neutrality and process neutrality are feasible through the introduction of the intelligent generic process design. The following picture would convey the generic methodology.

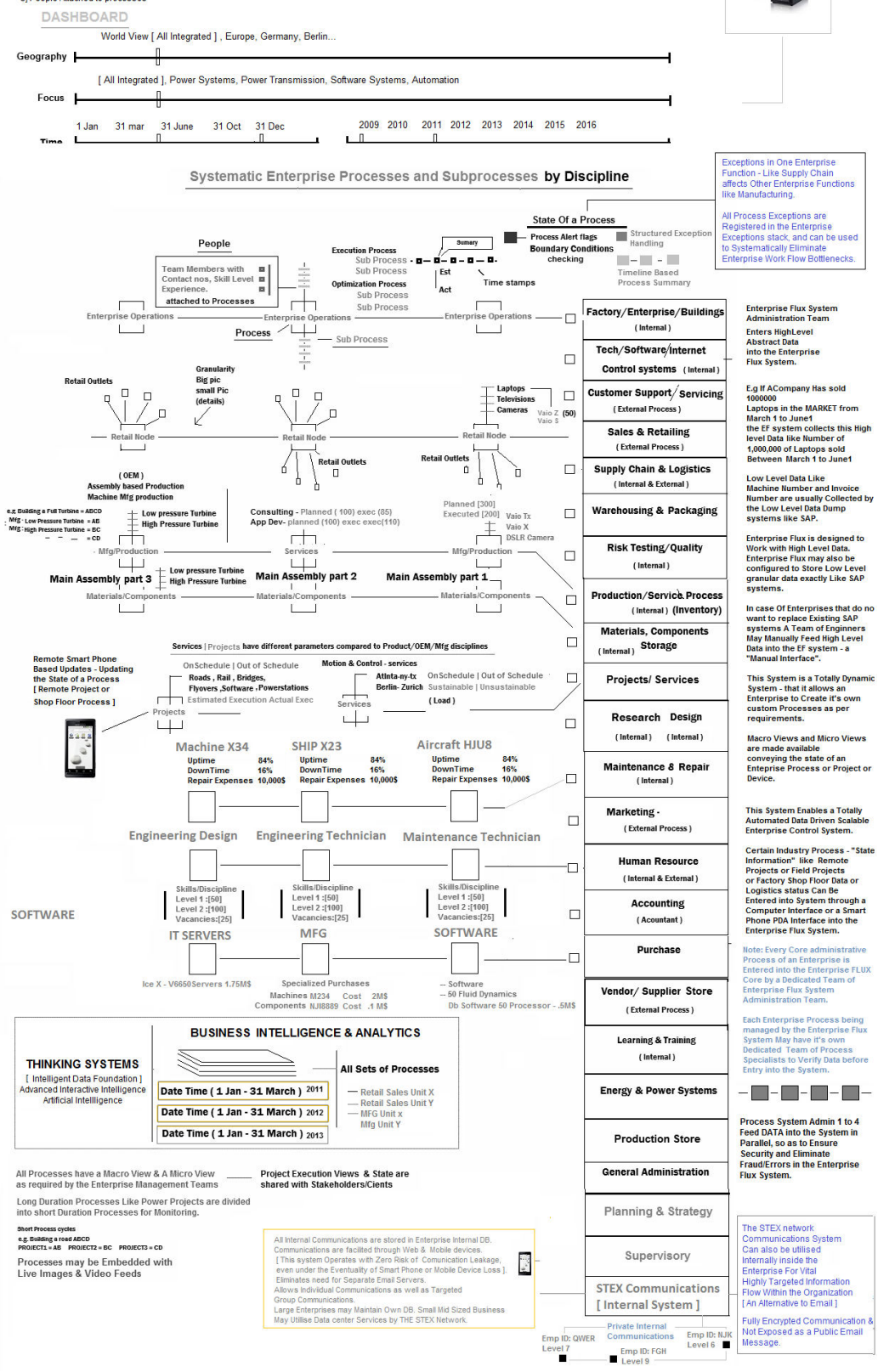
# Enterprise Flux [ Core System ]

The Primary Design Goal of Enterprise Flux is to Measure, Monitor & Manage Enterprise Processes.
This System Leads to Enterprise System Automation, Enterprise System Intelligence & Enterprise System Efficiency.

The Data associated with the Enterprise Flux Systems is High Precision Intelligent Data - That Forms the Foundationof Enterprise Intelligence and Artificial Intelligence.

Every Enterprise Consists of Core Functions or Disciplines which are Further Divided into Enterprise Sub Processes. Further Each Process has People Attached to Processes. In Summary the Enterprise Consists of

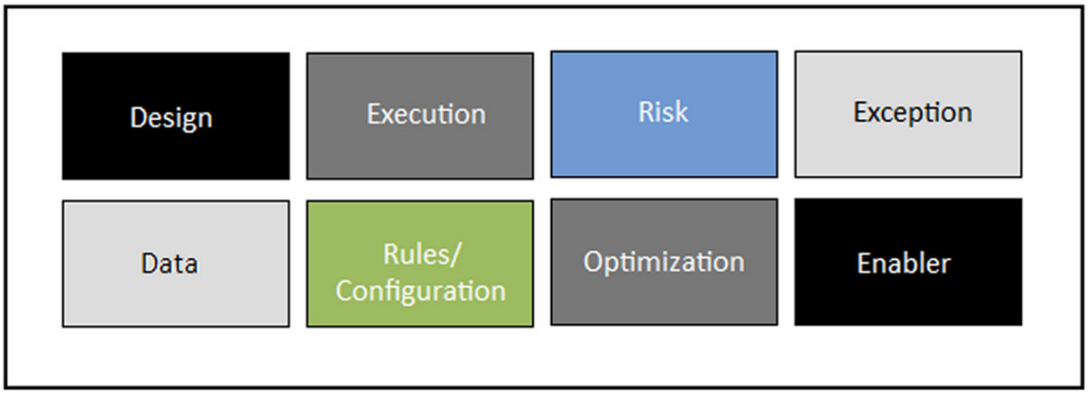
1] Enterprise Core Processes
2] Enterprise Sub Process
3] People Attached to processes



**System Software – Development:**

We have a wide variety of system software available today, but most of the system software is proprietary in nature and in features. The Intelligent Design approach helps in eliminating proprietary standards and leads to building standards based software, each varying from the other in scientific functional features. We have compiled a sample operating system interface for an appreciation.

**Operating System Interface Sample**




- Design Software
  - D.E - CAD/CAM
  - D.E - Photoshop
  - E.D - Visual Studio
- Data Software
  - B.E - DB2
  - B.A - Winword / Adobe PDF
  - B.B - File Storage
- Risk Software
  - C.E - Bitdefender Antivirus / Norton
- Enabler Software
  - F.A - DirectX / OpenGL / PhysX Libraries
  - F.E - .NET / JAVA Framework
  - F.C - BioMetric Authentication
- Optimization Software
  - Harddisk Defragmentation s/w
  - Clean up software

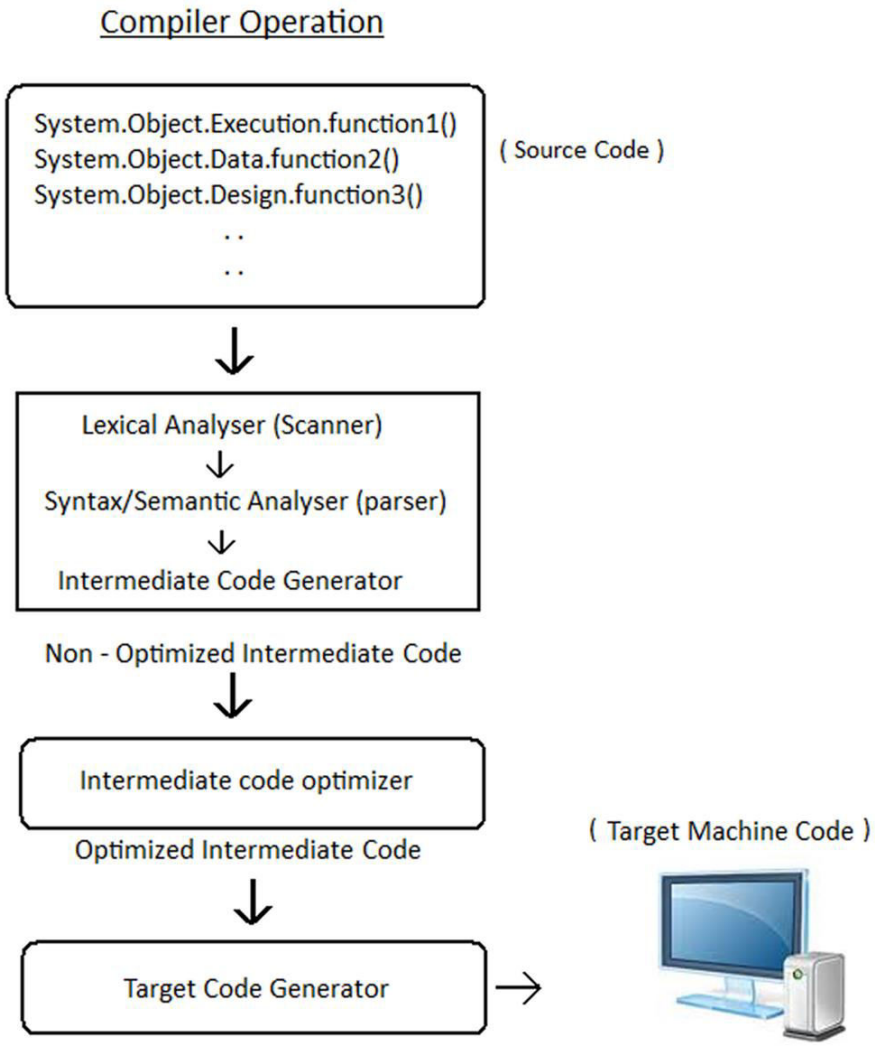
- Execution (General Purpose)
  - Communication - Mail / Messenger
  - Modem Connect
  - Exception Software (Not default to an Operating System)
    - Games
- Rules / Boundaries
  - OS configuration
- Exception — Events / Errors Viewers

| | |
|---|---|
| Design / Structure **A** | Data / Storage **B** |
| Risk **C** | Optmization **D** |
| Execution **E** | Enabler **F** |

Compilers can be designed adhering to the principles of intelligent design such that all functions describing a class can be represented in a factorized manner as suggested by the interface based approach. In a truly object oriented software, all objects inherit from the base class object called “System.Object”. In case of Intelligent Design Software, the class files would be further classified as “System.Object.Execution.xyz_function” depending on the classification of the function.

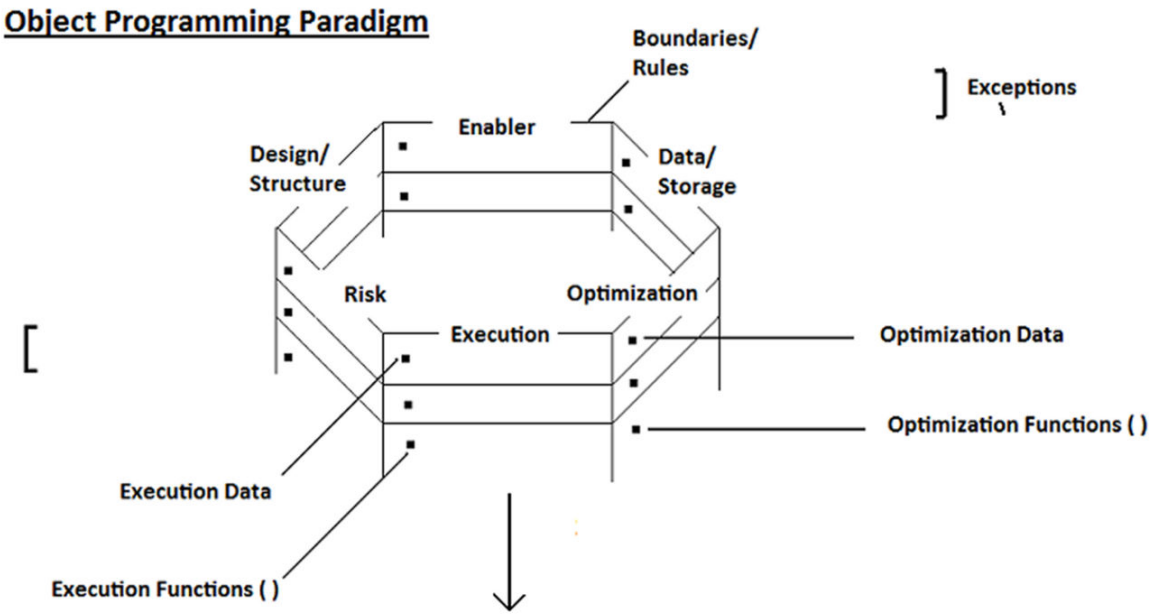
This would improve readability of the code as well as allow certain types of functions to be dedicated to specialized hardware suc1

has the GPU (Graphics Processing Unit). The "**System.Object.Design**" functions can be directly allocated to the GPU. Mathematical functions of execution could be allocated to the CPU (Central Processing Unit).




**Object Programming Paradigm:**

As of today, the highest programming paradigm that exists is the Object Oriented Methodology or Object Oriented Paradigm (OOP). The programming technique has been described as "object oriented" rather than "object" programming because, structured programming evolved into the methodology of describing objects or systems through classes. With the breakthrough in intelligent design approach, the object oriented paradigm may be evolved further into object programming paradigm where all the functions and data of object oriented design are fully factorized and represent pure objects.




Software class file

```
Namespace
{
        Class_name class
        {
                System.Object.Execution.function_name();
                System.Object.Design.function_name();

        }
}
```

**Object Based Database Systems;**

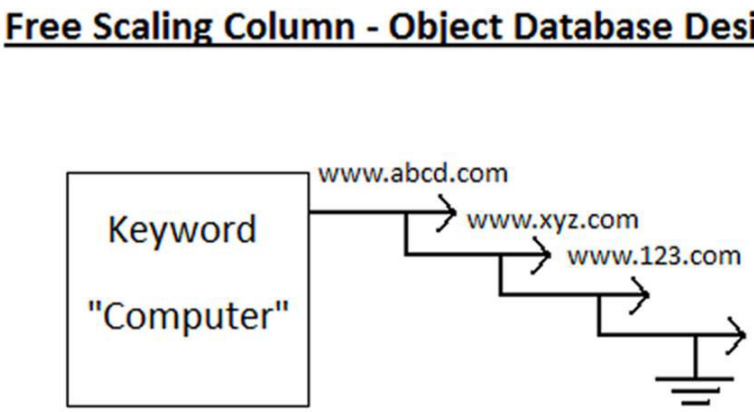
A large variety of Database systems are available today namely relational databases, object oriented databases etc. The Intelligent Design approach can lead to the development of object databases where the primary entity of storage is a free scaling object. Let us take an example of relational database where we need to map a list of URL links to a specified key word, for example, the keyword computer should map to links www.abcd.com, www.xyz.com, www.123.com etc. The way the database would be structured is as follows (ignoring XML key field methodology):

| Primary key (Keyword) |
|---|
| Computer |
| Server |
| Interface |

| Keyword | URL Link |
|---|---|
| Computer | www.abcd.com |
| Computer | www.xyz.com |
| Computer | www.123.com |
| Server | www.abcd.com |
| Server | www.a1b2.com |

**RDBMS table structure representing keywords to their URL Links**

If we approach the intelligent free scaling approach to design an object database, the database structure would function as follows:




The database field consists of scalable linked lists that store data in an optimized way.

**Game Design:**

Today's game design software's are graphically astounding. And there are a wide range of titles available with various types of game play. The intelligent design approach could improve upon the game play of a variety of games by involving all the TEFS functions or the Intelligent Design functions in a game sequence.

The game play sequence could be evaluated such that structured execution maneuvers yield better results than an unstructured maneuver.

Games could be designed for simulating enterprise systems and world systems .Younger

Players could be introduced to advanced intelligent approach to learning rather than games acting as only a source of entertainment.

**Artificial Intelligence:**

Artificial Intelligence is the field of Engineering involving, building of intelligent systems that resemble Human Intelligence or human intellectual capabilities.

The Intelligent design Functions are very useful in the design of intelligent systems, because the functions help in building a very sharp context.

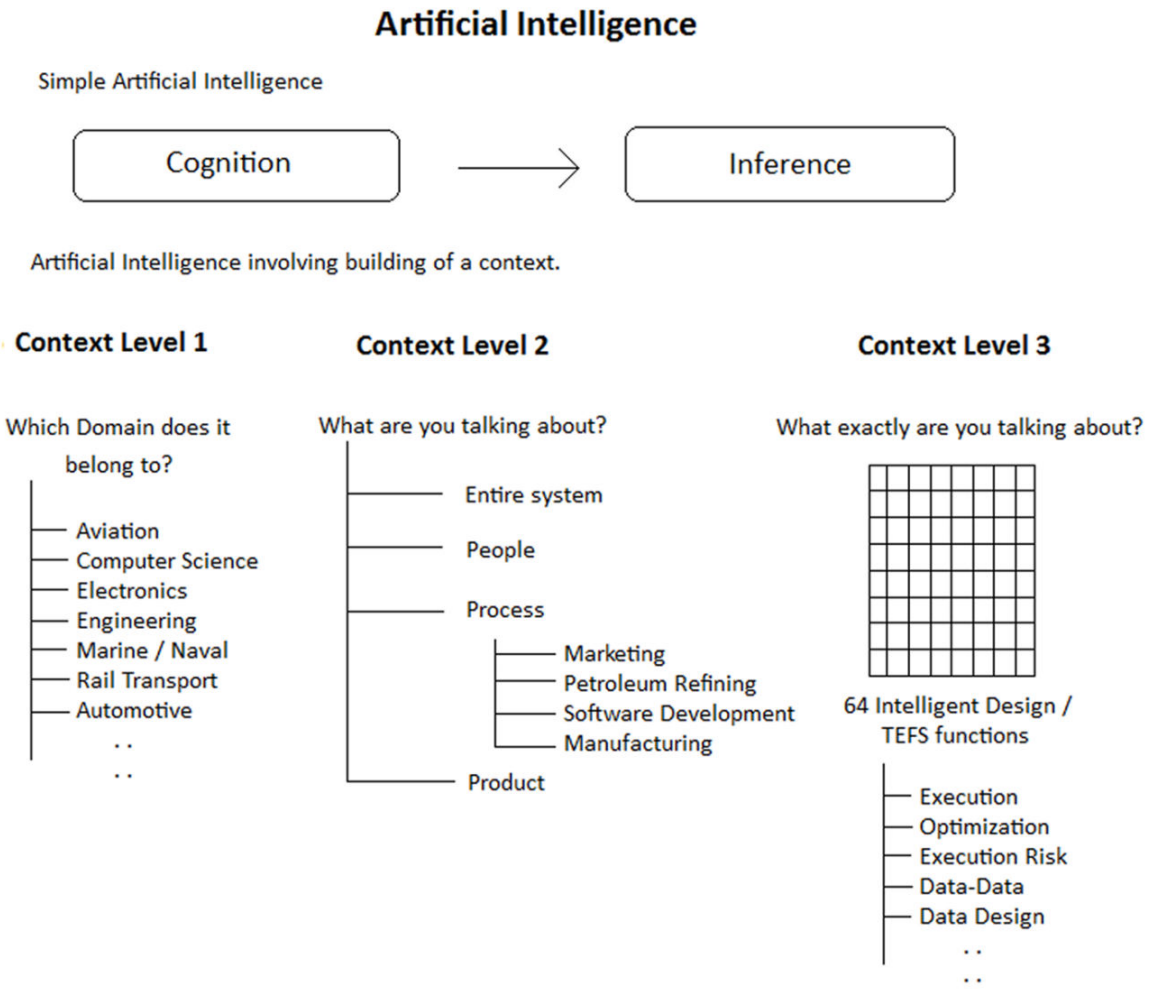
Much has been criticized about machine systems not being able to think like humans do. We beg to differ. Human thinking is not structured in many ways. Humans make assumptions that their thought process is superior to the ways of the machine because they use intuition for judgment. It may be noted that intuition is nothing but an emotional process. E.g. an assessment of Risk in a scenario can lead to a

variety of judgments by different individuals, depending on their emotional coefficients. A conservative nature or emotion would lead to a different assessment from that of a risk taker.

In contrast Machine thinking or logical thinking infers judgment through an assessment of all the variables of data it has in context.

It may therefore be improper to draw inferences of superiority of human intelligence over machine thinking.

Given a proper context machine logic is far superior. STEX Advanced Design Bureau is currently working on a Search Engine that tends to use smarter Data to build contexts and offer relevant results, a process similar to artificial Intelligence.




Examples of Artificially Intelligent Queries :

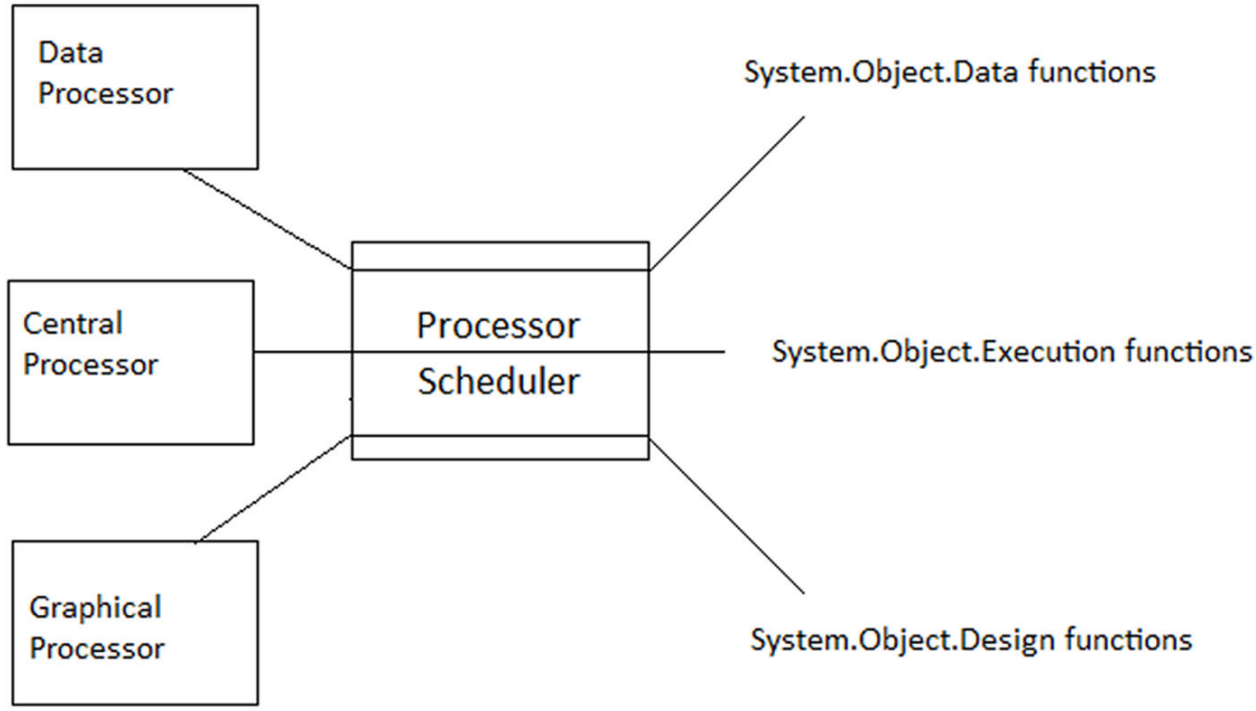
1) Software - People - Execution => We are talking about people's programming skills in software like C#, C++.

2) Aviation - Product - Execution => The process how the product works. If the product is an aircraft, the inference is "how an aircraft flies?".

3) If you are talking about Marine / Naval - Process - Design => Architecture and design of Ships / Marine products.

**Computer Hardware Systems:**

Computer architecture could follow the EDIC (Explicitly Distinct Implicitly Coherent) principles of design such that we might have specialized processors for graphical processing, for mathematical execution and for data fetch and store functionality. The processors could be coordinated by the help of schedulers allocating data tasks to the data processors, execution tasks to execution CPUs and Design tasks to Graphical Processors. The reason for such a split or such an explicitly distinct design could be because different kinds of tasks require different frequencies of operation or execution, for example, data fetching tasks are much slower compared to GPU tasks. Such architecture could be useful in Servers.




**Applications in Government/Nations Systems:**

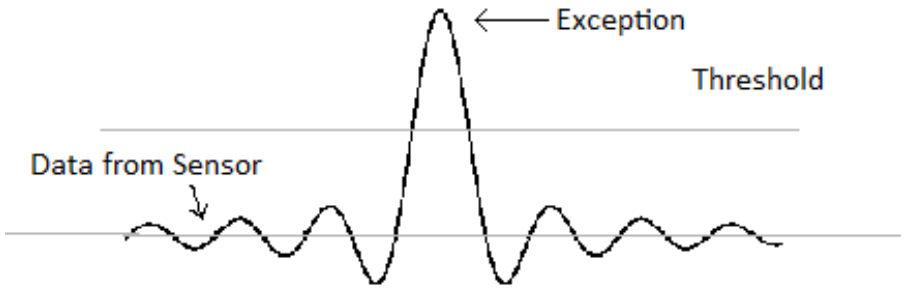
The TEFS functions have wide range of applications in the Government and Nations systems space. The applications could involve codification of national laws searchable by search engines or structuring of patent data in a format searchable by search engines. Other applications could be right to information systems that broadcast important information to the public.

**Standards Framework:**

The TEFS functions or the Intelligent Design functions can be used in an open standard framework that can be utilized by industries belonging to varied domains. For example, it could be used by manufacturing industry or software industry as a standard.

**Applications in Machine Systems:**

Applications of Sensors in Machine Systems to measure exceptions occurring during machine runs. Similar principles could be applied in a variety of systems where measurement of data is feasible.

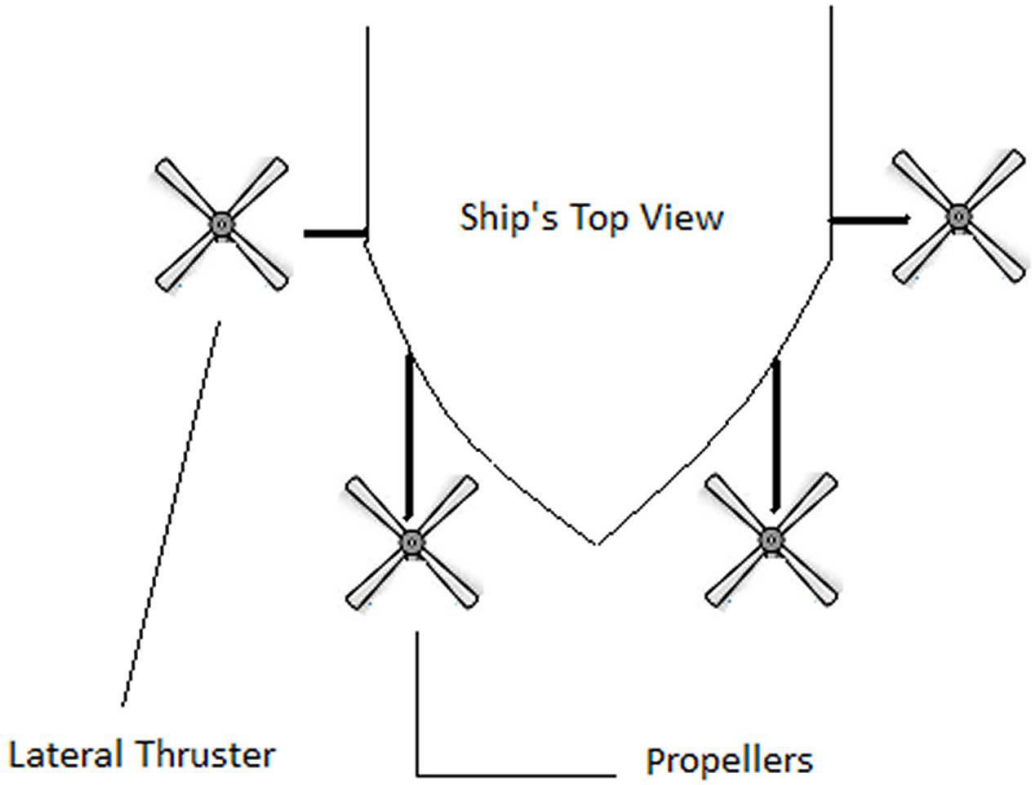



a) Possible futuristic fuel injection systems in automobile engines where the engines could be programmed to fire on individual cylinders or multiple cylinders depending on the need of the situation. For example:

| engine RPM | Executing cylinders |
|---|---|
| Idle engine RPM e.g. 900RPM or below | Execution of only cylinder 1 sustaining minimum idling of engines. |
| >900 RPM up to 2000RPM | Higher fuel air mixture in cylinders 1,2 &3 |
| >3000 RPM | Highest fuel air mixture in cylinders 1,2,3 & 4 |

b) Tilting seats in automobiles – The seats of an automobile or a car could be designed to accommodate higher degrees of freedom so as to allow the rotation of seats during centrifugal push while taking steeper turns or maneuvering higher uphill and downhill climbs.

c) Horizontal lateral thrusters in ships would allow easier maneuverability of ships during yaw or turn motions.




a) Applications in braking of trains – Harmonic exponential timed brakes can be used in train wheel brakes while reducing wear and tear of brakes. Either the time parameter could be used or the surface area parameters can be used while exposing the brake pads to the wheel. Timing or surface area Patterns could be like $(x + \frac{x^2}{2!} + \frac{x^3}{3!})$.

b) In computer hardware machinery if a drop in threads or tasks is observed multiple cores of processors could be shutdown so as to reduce power consumption .This is different from the "Always on" cores of processors.

Section - Part 2

# INDUSTRIAL LOGIC / INDUSTRIAL DESIGN

**Date of Revision / Addendum Date : 15 Sep 2015**

# ADDENDUM : Enumeration & Description of Design Bureau Projects

Chapter 1.

**The Information Grid Project ( The Stex Grid Project )**

Chapter 2.

**GEO Spatial - 3D Map Search**

Chapter 3.

**Selected Advanced Technology & Design Bureau Projects**

Chapter 4

**Software Automation - An Approach**

Chapter 5

**Finance by Structure & Design**

Chapter 6

**Industrial Logic & Industrial Design**

Chapter 7

**Physics & Patterns, of Colours and Colour Combinations**

Chapter 8

**Transportation & Logistics - Real time Telemetry Information to Public Users**

Chapter 9

**STEX Projects & Concepts - The Complete Big Picture**

Chapter 1

# The Information Grid Project
# ( The Stex Grid Project )

## THE STEX GRID Projects

## What is the STEX Industrial Network/StexGrid ? What are it's commercial Use Cases?

You are an Enterprise, an Individual or an Agency, TRYING HARD to reach an Audience for any one, few or all of the following needs.

You MAY have a need

1) To take a product/service to market and reach an audience,

2) To promote your Article/Technology Papers/Blogs,

3) To Promote your Event/Seminar/Exhibitions to bring in people as audience,

4) To find a Tech cofounder or a Partner online,

5) To hire a Good Team for your firm or project,

6) To fill a job vacancy,

7) To showcase your resume/profile,

8) To showcase Your talent through a youtube video,

9) To Offer a Discount on your products/services,

10) To Publish/Promote/Market your book,

11) To make a press release to reach an audience or newspaper,

12) To Launch a NEW product/service or a NEW company that you have started.

13) To find a Sponsorer to fund your projects/Events,

14) To find or promote work contracts or Sales lead,

15) To Find an Investor to Fund your/ projects

16) To find an Exporter/Importer for your goods,

17) To dispose your old used goods ( cars, machines,house hold items ),

18) To dispose your property ( buildings/offices/houses/land ) without any hassels.

and you have tried out a number of ways to reach people - Through Facebook marketing, Google search, Paid Press releases, Paid search, Pay per click Marketing, Twitter marketing, LinkedIn Networking, Flyer Distribution, Forum listings, Craigslist, coupon marketing, Exhibitions/events ( product Demonstration) etc....

## The Problems we Face using the tools mentioned

**Facebook & Twitter** Network systems were essentially designed to help people socialize, but are being improvised for product marketing through supplementary advertisement technologies. These network systems allow you to reach an audience consisting primarily of friend circles or followers. E.g. Twitter/Facebook marketing works well in celebrity circles, and in case of users with Large fan bases, but lacks audience reach for general users, without a large fan/friend following. Honestly - Most information or content on the networks is fairly general gossip with an occasional news trend.

**Google Search** requires you to have a Good Rank, based on website back links, to rank high on search results. The first ten Google search results are reasonably great, but the rest are not so fine. Google was never actually designed to help people find jobs or showcase people's profiles or promote events. Google search for industry category information actually leads you to 3rd party jobsites or social networks fulfilling people's needs of promoting their identity. Advertising technologies are supplementary technologies that have been improvised to sell space on the search pages and these Pay per click advertising campaigns are quiet popular, however a little expensive. The Search Audience is mostly untargeted & conversion of clicks to sales is not a guarantee.

**Press releases** can be an expensive affair costing up to 500$, without a real guarantee of a

newspaper article Print. Can you imagine millions of News releases vying for a space of 10-15 News spots on Google News or Yahoo news or other News Blogs? It's a fact that News journalism today has manifested itself as a medium for paid publicity.

**Posting job** ads on jobsites was the traditional old way of finding people for job vacancies - but jobsites are a dime and a dozen in number. Many of the sites that people visit, are City centric or Country centric. People today have moved into showcasing their Talent/skill/profiles online by hosting their online resumes/profiles with dynamic content demonstrating or showcasing their talent & accomplishments.

**LinkedIn** allows you to get in touch with people in your business/Industry circles but is still a tool for business networking only. Finding a Technology architect or Technology Specialist on LinkedIn involves searching through their paid closed databases. The free Google listings of LinkedIn profiles are only searchable on Google - when people search using an individual's name (proper nouns), not exactly by their skills or experience.

**Coupon Marketing sites** are a nice way to offer discounts and promotions however the internet is filled with a large listing of expired and unexpired Coupons and probability of finding a valid coupon is not so easy.

**Flyer marketing** is an inexpensive way to reach an audience, but works well for SMALL Enterprise products, services and solutions. It is difficult to market larger scale Industry products or services through this medium.

**Events, Exhibitions & Product demonstration events** is a nice way To reach an audience, however most exhibition marketing events involve - International Logistics and stall hosting fees and have a limited geographic audience ( foot falls ). This way of marketing can be expensive as the process of reaching an audience is a fairly elaborate & is expensive format. (As it involves event marketing & advertising expenses). Universities and educational institutions regularly conduct scholastic seminars and involve the same old advertising or Agency marketing techniques to reach an audience. (THE STEX WAY - removes all hassles of promoting your event, through a single stex industrial network information post, cutting down on all advertising costs).

**The DOT COM way of solving problems** - (Sarcastic sense of Humor - going by the Dotcom Trends as of today) when looking for a house you might need to check a listing on housing.com. When looking for a job you might need to check jobs.com. When looking for work contracts you might check workcontracts.com. When checking events you might check events.com and finally when disposing of your old car you might be checking out usedcars.com.

## The Proposed solution to the above mentioned problems - A Planet scale Intelligent Information Grid

Just the way we have built Electricity grids - supplying "electricity" to large number of End Users ( Electricity being used for a Variety of Functions- Air Conditioning, Television, Computing, Machining, Water purifying.....), we have designed an Information grid ( A Single Large Intelligent Information Platform ) to allow exchange and discovery of Vital real-time Information ( useful for a variety of Functions ) - for Industrial /Commercial or Personal use.

This implies that ELECTRICITY is ANALOGICALLY EQUAL TO = INFORMATION, therefore you may PLUG in to the STEX GRID ( www.stexgrid.com) at any Time, to pull or push ( produce or consume ) the vital information that you need, and use it just like electricity. This Information Grid is geography agnostic & consists of three main Parts.

1) The STEX Industrial Information Network ( An Information exchange system for collecting and Intelligently Tagging Essential Information, leading to a planet scale intelligent data repository. )

2) A Built in or Attached - Nework Search Engine called STEX Search ( For Information discovery in a variety of Categories ).

3) A Built in 360 Degree Communication system for all Network members to communicate with each other ( Breaks communication barriers, facilitating solicited and non solicited Industry related communication between any network member to any other network member ).

Note: This Information grid system was not designed for socializing, or creating business circles, or Information Listing, but rather for the purpose of Multi-Dimensional/Multi purpose - Information

exchange, creating Vital Information opportunity for it's users. Further, this system is designed to integrate well & play well with well known search technologies like Google, Bing, Yahoo. The Information grid system is completely free of Cost - on the working model of WIKIPEDIA. OH ! Yes Information search results are [ open ] for Industry sponsorships. E.g THE STEX SEARCH RESULTS ARE SPONSORED BY ABB - www.ABB.COM

## STEX Industrial Network & STEX Search - How it Works ?

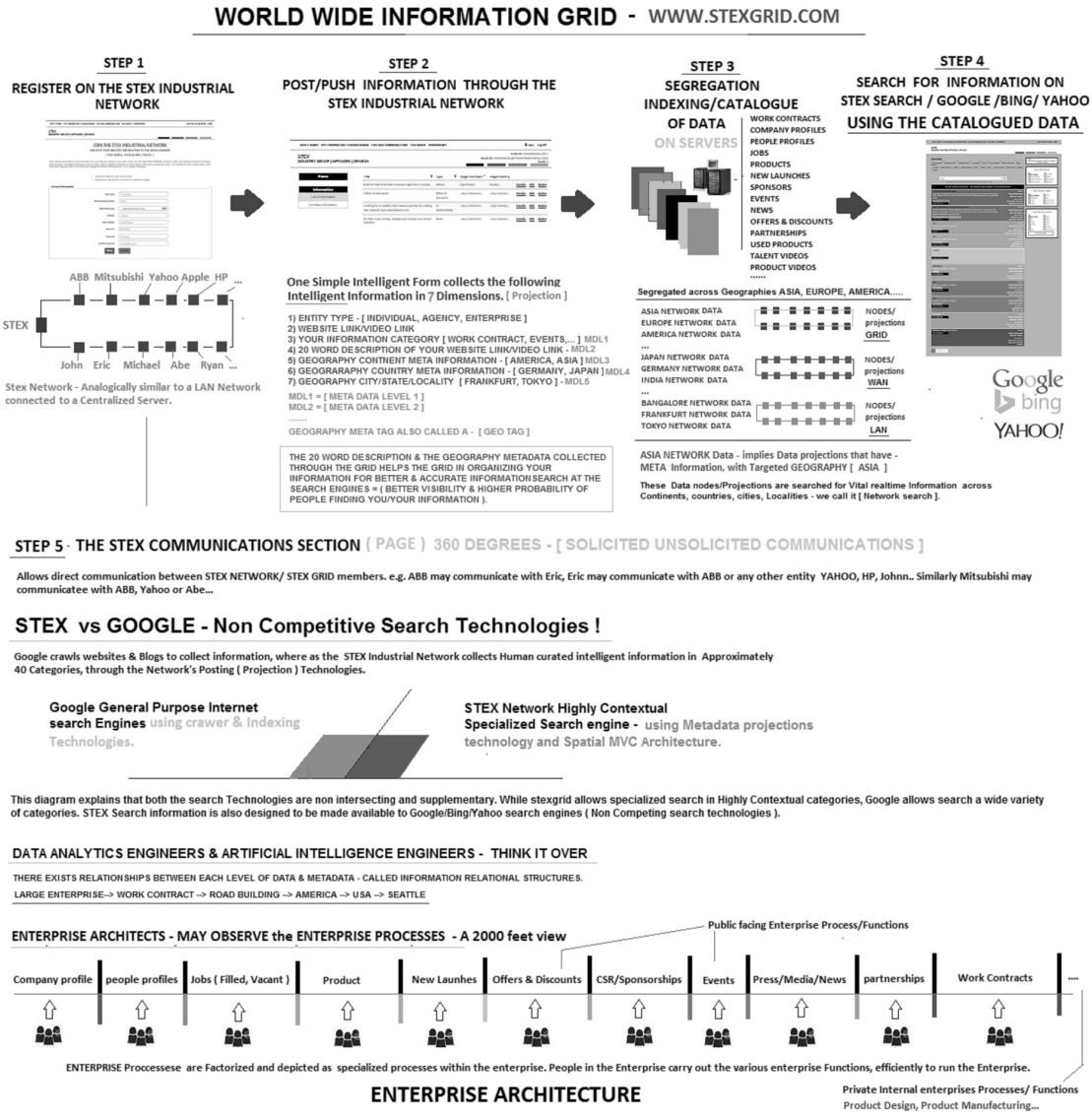
## There are 4 simple Steps.

1) Individuals ( People) or Enterprises ( small, medium, Large, Governments,Universities, Agencies ) join or register on the STEX Industrial Network, and become STEX Network members.

2) Network Members Post/Push "Industry grade Contextual information" to the STEX Industrial Network Datastores/DataRepository in realtime. ( Using a new technology called metadata projections ).

3) The STEX Network Search Engine & Search Engines like Google/Bing/Yahoo help in "Information Discovery" or "Information search" from the STEX Datastores/ Data Repository.

4) The STEX Industrial Network Members have the facility to Communicate with Other STEX Network Members, through a built in STEX Communication system ( An Email Alternative ).



## Internet search Engines Vs Network Search Engines - [ A Defence ]

Most people we meet tell us that the google search works well and collects information automatically and has petabytes of Data, COMPARED to your technology that collects only META information through your Network and has little data.

So what we may do straightaway is - dissect the google/STEX technology, from a Technology perspective and let you think, and draw your conclusions.

Did you Know that all Browsers like Internet Explorer, Google Chrome and Firefox use a Web METHOD OR web function ( called HTTP REQUEST/RESPONSE ) to fetch data from a website server to show/display the websites on your web browser ?

GOOGLE IS SMART. It uses, the same ( called HTTP REQUEST/RESPONSE ) funtions, provided by the operating system web framework to download and collect all Data/information from the websites, on the world wide Web. Instead of displaying the information on a web browser, it collects & piles up the data (after stripping of all the HTML Tags ) in a data repository or data base, called the Google Database.

As a working example Google servers - go to websites like stexinternational.com, mitsubishi.com, yahoo.com, stanford university.edu, microsoft.com, stackoverflow.com, codeproject.com, Tech Blogs... use the HTTP Request Response functions, and collect all the website data, and pile it up on Massive hard disk stores in computer data centers.

This pile of Data in the Google Data base is indexed or catalogued and is now eligible for search. The Google Search Engine page or the search interface is now attached to this data repository, and allows search, for end users like you and me.

Technology enthusiasts may understand that while GOOGLE crawls websites to collect website data, STEX grid Uses a Network Interface to collect Important Meta data. Both the Technologies Categorise, index and catalogue the data, and finally both the Systems have the built in or attached search engines interfaces that allow search of the Catalogued data.

Now you may have to think ? What Type or kind of INFORMATION is available on websites and Blogs the world over ? Product descriptions, articles, News etc etc. This limited website/blog information is collected in google data repositories, and, is available for search at the search engines.

# THIS IS WHERE STEX COMES IN.

There are about 30 to 40 Categories of Industry information that is never put on your company websites - For example events, Realtime offers & Discounts, Used products for disposal, Resumes etc etc'. Since, this kind of information would never exist in the google Datastore or repository, it is not possible, for anyone to find this kind of information, on google search.

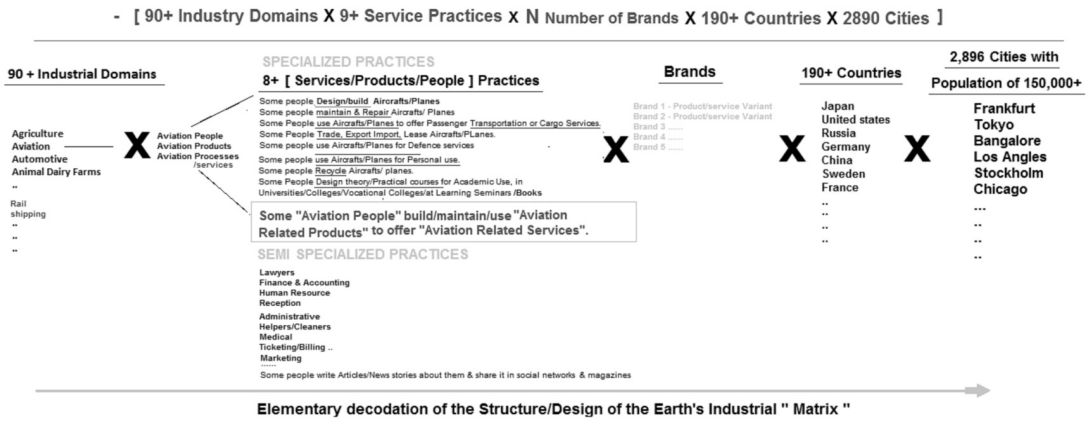
That is the exact problem that we are solving. We help people search or find the information that GOOGLE can never help you find.

With this in mind, STEX Has designed the stex NETWORK interface to collect vital Intelligent Industry Information Data from it's members, and store it in a STEX Data Repository. The STEX NETWORK SEARCH INTERFACE is now attached to the STEX Repository, to allow search of important Information/data. You might Now understand, that both the Technologies are not exactly competitive and are supplementary technologies.

Do remember one thing that Google is your friend and So is STEX. We are technologists too, and we use google day in day out for our programing needs. We admire Google as much as we admire STEX ( OUR own technologies ).

## Earth is a More Complex Story than, it meets the eye.

DID YOU KNOW that there are 90+ industry/industrial categories on this planet system ?




Lets take Aircraft/planes For an Example

Some people build Aircrafts/Planes.
Some people maintain & Repair Aircrafts/ Planes
Some People use Aircrafts/Planes to offer Passenger Transportation or Cargo Services.
Some People Trade, Export Import, Lease Aircrafts/PLanes.
Some people use Aircrafts/Planes for Defence services
Some people write Articles/News stories about them & share it in social networks & magazines
Some people use Aircrafts/Planes for Personal use.
Some people Recycle Aircrafts/ planes.
Some People Design theory/Practical courses for Academic Use, in Universities/Colleges/Vocational Colleges/at Learning Seminars.

Now Think of what you may do with Trucks/Buses/Cars ... or the 90+ Industrial types.
The STEX GRID empowers you to WORK and promote you/ your services without spending a Dime on Marketing.

For Example -

When there is a dearth of jobs - you might just procure a few Used Washing machines and set up a Laundry service center for your city/Locality - without having to think - how will people find us or our services ? THE STEX GRID - should help you out with service discovery, without you spending a penny on marketing.

You could do the same with grocery delivery services at Frankfurt, Tokyo, Los Angeles, Bangalore or any of the cities world wide.

A HR Services firm or an HR Consultant could offer to build your customized resumes and host it online for you (costing 20$-50$) - so that you could market or showcase your profile better, to find a better job....

How would it feel to - find 3-4 Job offers ( Solicited/unsolicited) in your STEX Communications Mail box, when people find your profiles/ or read your articles on the grid, and contact you directly through the STEX Communication system.

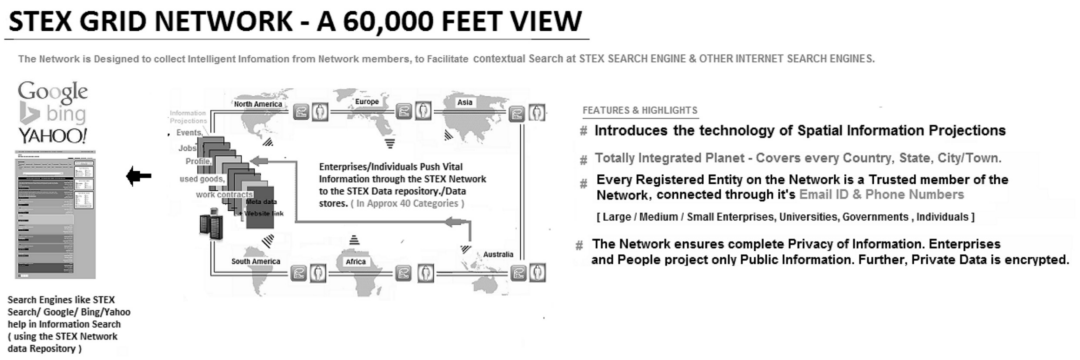
## What is a Metadata Projection ?

**Note:** The STEX Industrial Network Members Push ( 300 Characters ) of Brief Information about your Enterprise profile, Individual profile, jobs, work contracts, etc... in 40 categories, along with a link to your web page containing vital information or information descriptions.

The 300 Characters of Brief Information is called META Deta & is stored in the STEX Datastores/ Data

Repository in realtime and Indexed. ( It is as simple as Tweeting ).

This meta data information stored in STEX Datastores/ Data Repository, is made available to the STEX Network search Engine and search engines like Google/Bing/Yahoo for Accurate contextual search.




## Why Post Information through the STEX Industrial Network ?

1) Posting Information through the STEX Industrial Network leads to - Accurate Information Search on Search engines like ( STEX Search/Google/Bing/Yahoo) ! .

2) Posting Information through the STEX Industrial Network leads to reduced Marketing & Advertising Expenses on the Internet. ( It is an Alternative to Internet Advertising ).

3) Internet search engines like Google, Bing, Yahoo are absolutely fantastic Information discovery technology sets, however there are about 40 Specialized Industrial Categories of Information that are not easily Searchable on Internet Search Engines.

## Internet search Technology vs Networked Metadata search Technology - A Technical Perspective.

1) Internet search Engines have limited strengths in Abstract search queries, Queries without proper nouns/Names. E.g.- A Search for Enteprise Architects in Boston or a search for Latest Innovations and breakthroughs in the aviation industry or search for companies/Enterprise profiles offering Java development services in Switzerland / A technology event happening in your City.

2) The Webpages that Search Engines collect or crawl do not have many vital Industrial category information. If this Information could be posted to the search engines through the STEX Industrial Network, search engines could help in more accurate information search/discovery. e.g. Events/Seminars

3) Internet search engines have little Space (Geography) sense or Time Sense. That means that Geography based search is not accurate on internet search Engines.

4) Internet Search Engines do not actually collect/show Real Time Information, because they crawl periodically. In contrast the stex network collects realtime ( Live ) Information from it's Members.

5) Specialized "Enterprise related" Search Capabilities are not built into Internet search engines, as they operate on the world wide web, which is more of a general purpose information repository. Moreover the search engines also collect vast amounts of Internet Junk. ( While the stex network collects clean contextual information from it's members, in a way, as simple as tweeting ).

6) Internet search Engines lack features to control privacy.E.g. Casual Forum comments are displayed in search results.

7) Additionally, Internet search engines collect massive amounts of Data, however, only 10 -15 search results have user relevance, and the rest of the 99% of the results turn Irrelevant. This Implies that the Information/data system operates a very low levels of system efficiency and requires extremely large

datacenter infrastructure. In Contrast all stex search results stand to be relevant ( whether the first result or the Last) .

8) Finally Internet search engines are supplemented through advertiwsements. STEX Search allows you to control Advertising costs, if not eliminate them.

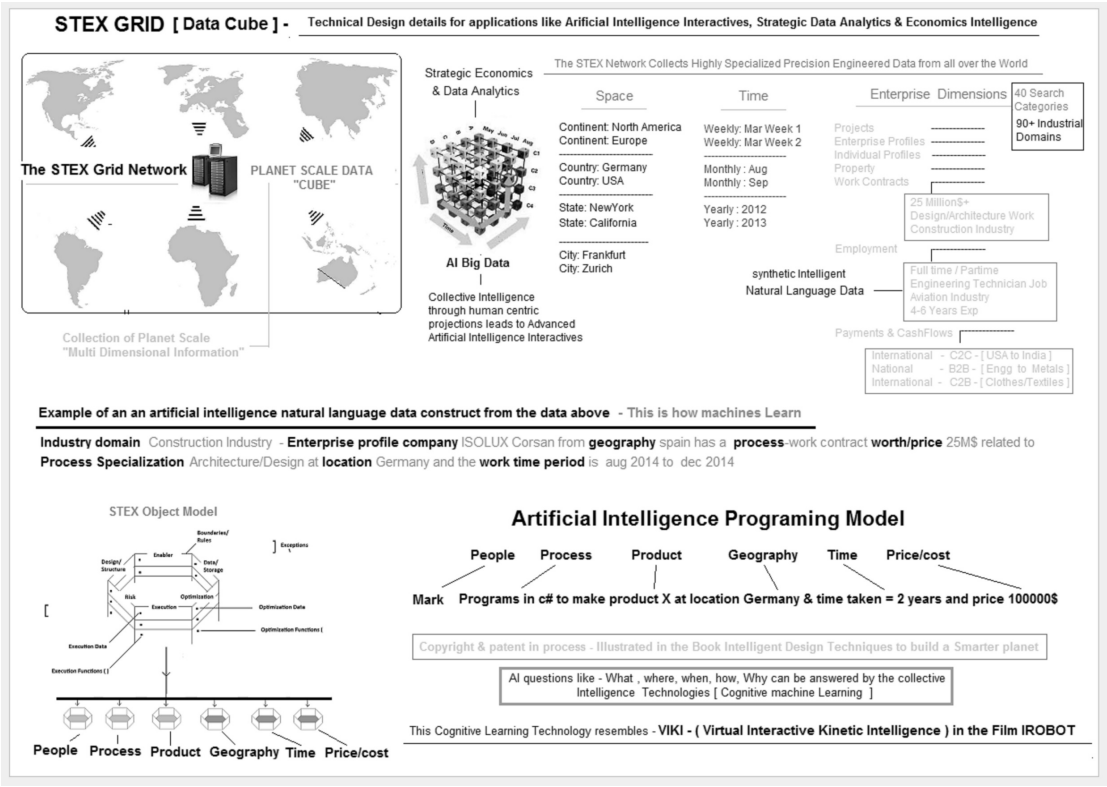
## 30+ Categories of Industrial Information - not easily searchable on Internet search Engines ( Google/Bing/Yahoo).

The Categories/Types of Information & services that cannot be easily searched on Internet search Engines are listed below. STEX Advanced search has been specifically designed to facilitate Accurate search in the following Commercial Information Categories.

- Enterprise Profiles
- Individual profiles
- Work/Project Contracts
- Finance Offerings
- Industry Products
- Software Products
- Gaming Software
- Services
- Internet Services
- Industry Offers & Discounts
- Consumer Offers & Discounts
- Property
- News
- Events/Seminars
- Sponsorship/Scholarships
- Contests/Competitions
- Acquisitions
- Talent Search Videos
- Used Products Personal
- Used Products Industrial
- Specialized Information
- JV Partnerships
- Innovations/Breakthroughs

- Jobs
- Software jobs
- Learning Training
- Reviews
- Video's
- Books
- Mobile Apps
- Internet Services/Tools
- Exports & Imports
- Travel & Tourism

# STEX GRID THE BIG PICTURE



Chapter 2

# GEO Spatial - 3D Map Search

# The STEX 3D SCAN - 3D Map Search Engine

**A Brand New - Digital Experience using Technology**

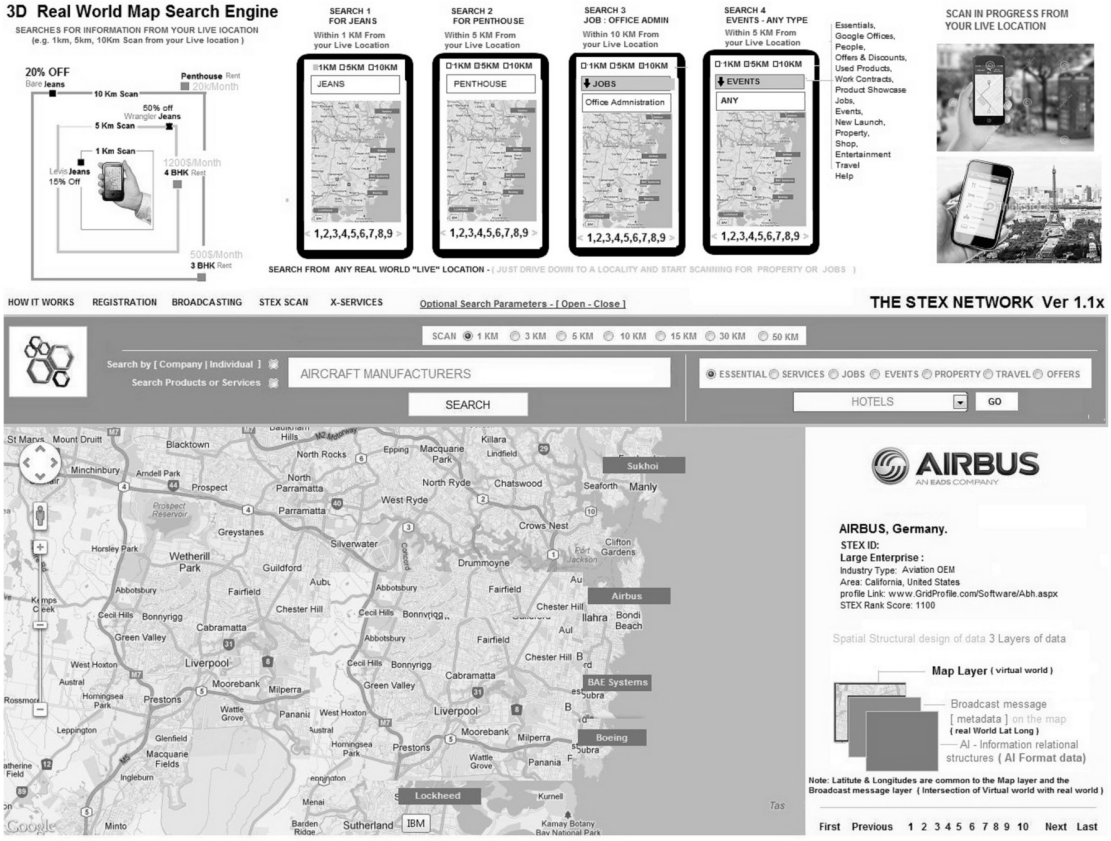
The STEX GRID Facilitates - Fluid, Real Time, Spatial 3D Information Access through a Map based Search Engine on the Web or a Smart Phone. This Real world geographical search engine that can be used to search for [ All kinds of realtime Information that cannot be found on digital Maps Bing/Google Maps ]. E.g. The Search Engine allows Local Search within a scan a radius of 3km/10 km/20km/50km from an individuals live location to find vital information like

- Discount and Sales of Jeans in a Radius Scan of 5 Km from your present Location
- A House For Rent in any Particular Locality - in a 3 Km Radius.
- Nearest Hotel with Hotel Room Fee - in radius of 1 Km from your Present Location.
- Events or Exhibitions in your Locality
- A restaurant with a Deal within 3 Km
- Vacant Property for Sale - Within 10 Km
- Used Furniture for Sale in a Locality of 5 Km Radius.

Another unique technology in this map scan is that markers are "Image Markers" with Text on them. Users may choose the Image Text marker they prefer, and read the Information Content meta data associated with the marker. This process is a single step process, different from 2 Step process where the markers on the maps have no frontline text associated with the marker Image.

Finally The Last Piece of unique Technology in this search engine is that only 5-10 map search Results are shown on the map for each search page.( This solves the problem of having 50+ or a large number of Search Results crowded on one Map Page ).

**Some of Search Use Cases are defined at our website http://www.stexinternational.com/MapProject.aspx, for you to understand the power of 3D MAP SEARCH & Locational Search**


3D Real World Map Search Engine
SEARCHES FOR INFORMATION FROM YOUR LIVE LOCATION
(e.g. 1km, 5km, 10Km Scan from your Live location )
20% OFF
Bare Jeans
10 Km Scan
Penthouse Rent
200$/Month
50% off
Wrangler Jeans
5 Km Scan
1 Km Scan
Levis Jeans
15% Off
1200$/Month
4 BHK Rent
500$/Month
3 BHK Rent
SEARCH 1
FOR JEANS
Within 1 KM From
your Live Location
SEARCH 2
FOR PENTHOUSE
Within 5 KM From
your Live Location
SEARCH 3
JOB : OFFICE ADMIN
Within 10 KM From
your Live Location
SEARCH 4
EVENTS - ANY TYPE
Within 5 KM From
your Live Location
1KM 5KM 10KM
JEANS
PENTHOUSE
JOBS
Office Admnistration
EVENTS
ANY
1,2,3,4,5,6,7,8,9
Essentials,
Google Offices,
People,
Offers & Discounts,
Used Products,
Work Contracts,
Product Showcase
Jobs,
Events,
New Launch,
Property,
Shop,
Entertainment
Travel
Help
SCAN IN PROGRESS FROM
YOUR LIVE LOCATION
SEARCH FROM ANY REAL WORLD "LIVE" LOCATION - ( JUST DRIVE DOWN TO A LOCALITY AND START SCANNING FOR PROPERTY OR JOBS )
HOW IT WORKS
REGISTRATION
BROADCASTING
STEX SCAN
X-SERVICES
Optional Search Parameters - [ Open - Close ]
THE STEX NETWORK Ver 1.1x
SCAN 1 KM 3 KM 5 KM 10 KM 15 KM 30 KM 50 KM
Search by [ Company | Individual ]
Search Products or Services
AIRCRAFT MANUFACTURERS
SEARCH
ESSENTIAL SERVICES JOBS EVENTS PROPERTY TRAVEL OFFERS
HOTELS
GO
Sukhoi
Airbus
BAE Systems
Boeing
Lockheed
IBM
AIRBUS
AN EADS COMPANY
AIRBUS, Germany.
STEX ID:
Large Enterprise :
Industry Type: Aviation OEM
Area: California, United States
profile Link: www.GridProfile.com/Software/Abh.aspx
STEX Rank Score: 1100
Spatial Structural design of data 3 Layers of data
Map Layer ( virtual world )
Broadcast message
[ metadata ] on the map
( real World Lat Long )
AI - Information relational
structures ( AI Format data)
Note: Latitute & Longitudes are common to the Map layer and the
Broadcast message layer ( Intersection of Virtual world with real world )
First Previous 1 2 3 4 5 6 7 8 9 10 Next Last

Chapter 3

# Selected Advanced Technology & Design Bureau Projects

# 1) Spatial Information Projections/Information Broadcast - Technology for the STEX Grid project

Stex Advanced Design bureau has Invented this special new Information/Data Architecture or Paradigm for specialized network applications, search Applications as well as Artificial applications. Information collected on the Internet, Web Portals, search engines or social networks is called Traditional uni dimensional Data and Lacks many Technological Features such as

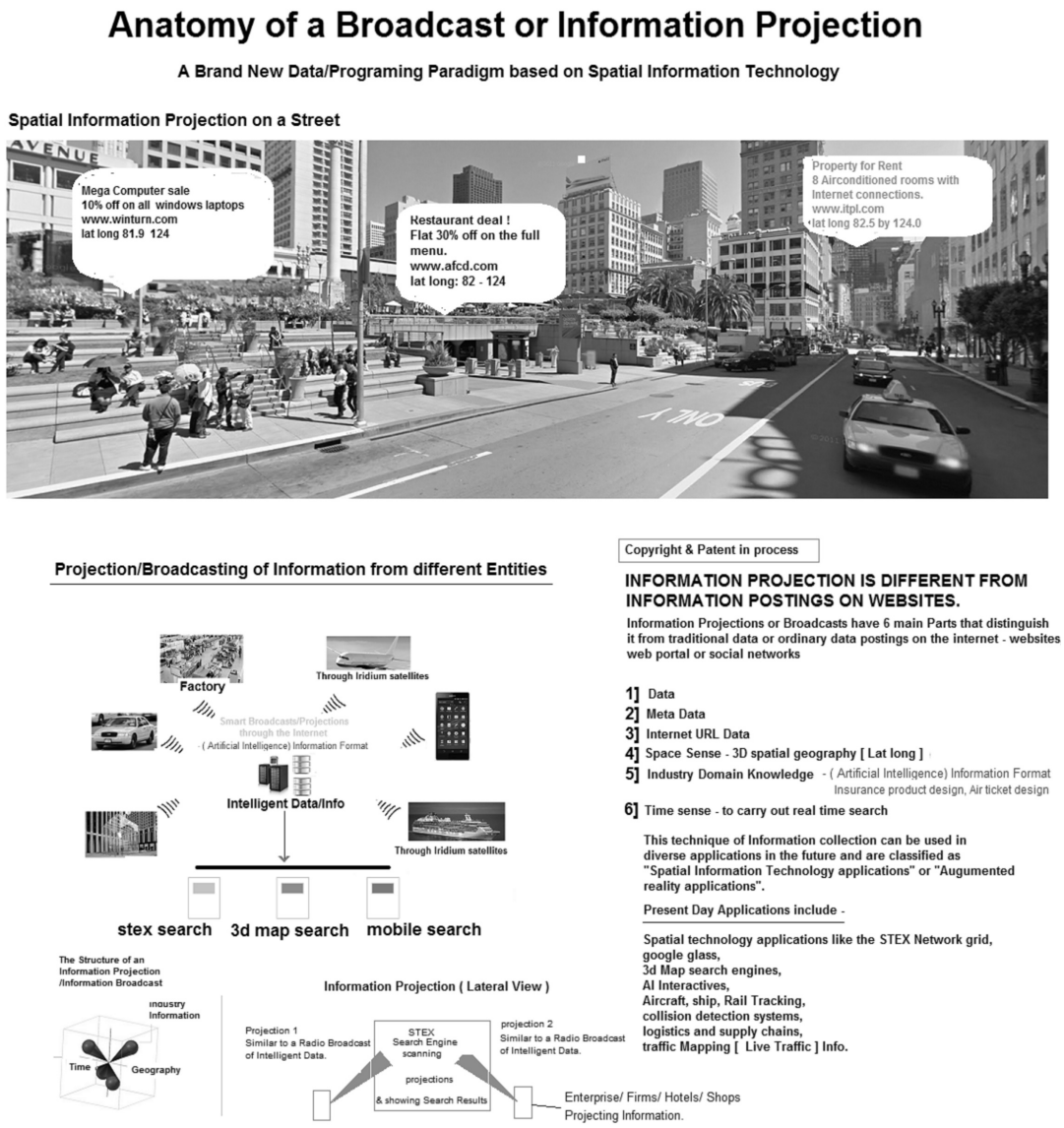
Multi Dimensional Meta Data associations

Sense of Space/Geography

Sense of Time

3Dimensional spatial structure

Domain Knowledge

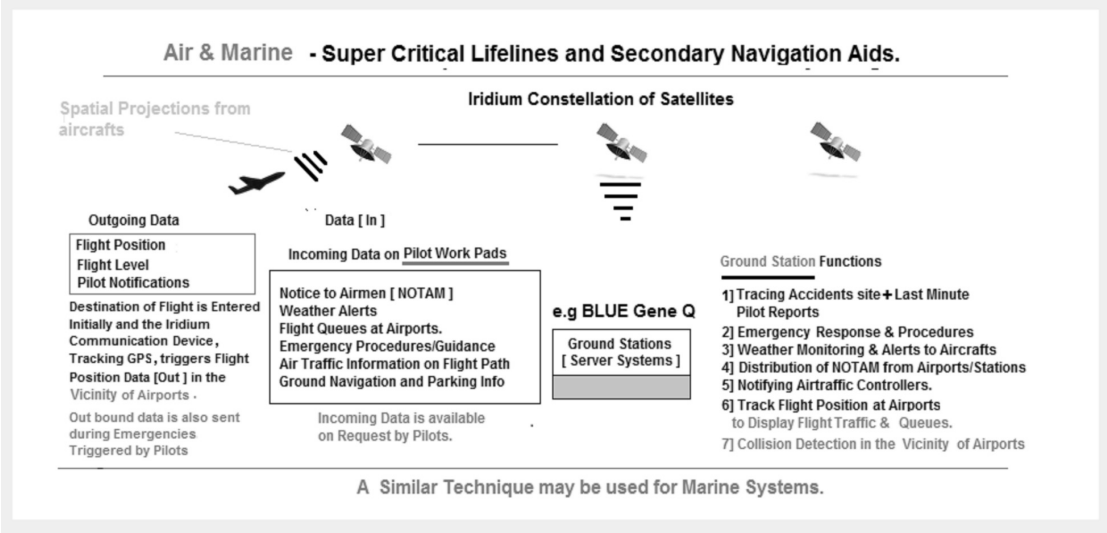



This Technology has Many Futuristic applications as shown in the Illustration above, For E.g. STEX GRID project, STEX Advanced Technology Services Project, STEX Map Search project.

## 2) Aircraft/Marine Rail Navigation& Tracking systems - using spatial Information Projection Techniques.

Flight Telemetry Data is collected through Internet enabled Hand held Devices - like Smart Phones or IPAD'S or Location Sensing Devices.( From Aircrafts to Ground Stations ).

Internet Technologies are provided by LEO Satellite systems like IRIDIUM.




How is this Flight Telemetry Information Useful?

- Lets Pilots know of Exact flight queue's so as to reduce fuel costs during Flight start ups.
- Allows Aircraft tracking beyond radar controls or ATC's upto the last mile parking bay.
- Displays Flight Queues at Viscinity of airports so as to calculate expected flight delays and Collision Risks, even during low visibility conditions
- Emergency notifications can be sent to ground stations with Engine and Flight parameters information.
- Allows ground stations to monitor Aircraft movement - as a Route Optimization function for fuel savings and Flight Delays.
- Note: For efficiency on the Internet satellites, each broadcast packet is compressed and sent to ground stations in burts every 1 minute, 2 minutes or 5 minutes. Ground stations operate in realtime mode on Inmemory Aircraft objects on Background maps.

## 3) STEX Advanced Technology Grid Services :

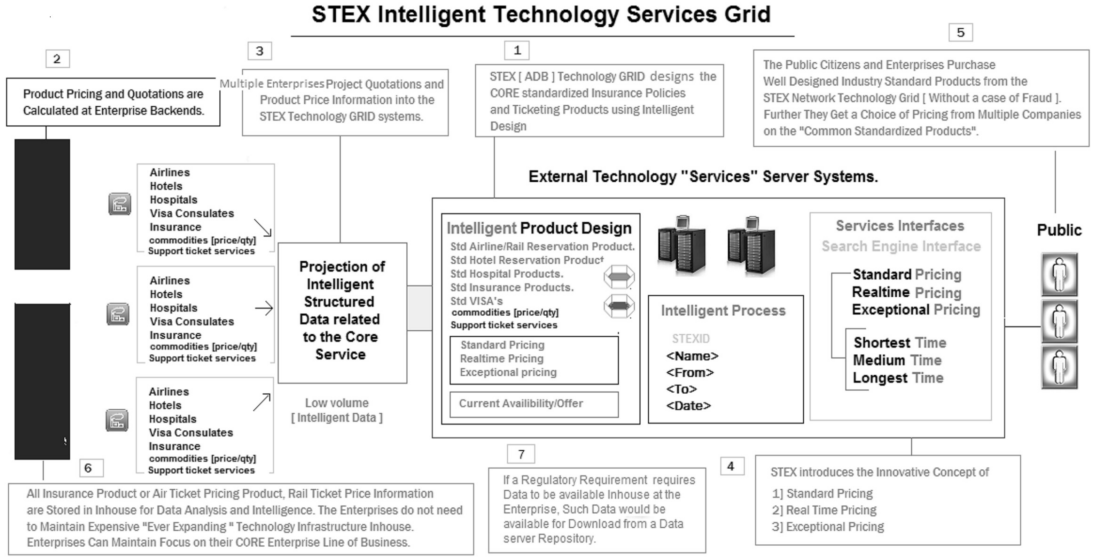
The Primary Design Purpose of Building a Technology Grid Offering Insurance Products, Financial

Products, Air Ticketing Facility, Rail Ticketing Facility, Hotel Booking Facility is to Standardize Idustry products.

Industry Products are Very Widely Heterogeneous in Design, so Every Insurance Policy or Financial Product in the Market has Varying Parameters, Clauses and Rules. The Products are So Many in Number that Consumers often Face the Challenge of Comparing Different Products with Many Variations in Terms, Conditions and value.

STEX Design Bureau aims at STANDARDIZING Insurance and Financial Products/ Air Ticket /Rail ticket/Hotel Reservation products by Designing The Core Product, The Policy and Rules and placing it on a Technology Grid server. Any Insurance Firm may Calculate The Product Quotations related to the Product and project the Information into the STEX Technology Grid.

This Saves Insurance Companies with a Huge amount of Ever Expanding - Inhouse Technology Spending, thus allowing them to Focus on their Core Line of Business. The STEX Grid System Integrated with the Technology Grid allows Members to Purchase a "Industry Standardized Product/Policy" with ease. The Same Applies to the Airline/Rail Industry, Logistics Industry and Health Industry.




## #4 Enterprise Control systems : Enterprise Flux

Enterprise Flux is an Advanced Enterprise Control System, also called an Enterprise Operating System. It is a Next Generation System that Introduces the Concept of a Totally Managed, Automated Enterprise – based on the principles of Advanced Enterprise Architecture, Spatial Systems Engineering, Enterprise Collaboration, Pervasive Information.

It is Domain neutral, therefore it fits into any enterprise in any domain and allows effective & Intelligent Enterprise Governance.

It is designed to work with Enterprises having remote divisions Remote site projects & Transnational Operations. It facilitates Measuring, Monitoring and Managing an Entire Enterprise from One Single DashBoard. It connects People Process and Product across the Enterprise, makes Enterprise Information Pervasive and creates a [ Fluid & Focussed ] Engineered Enterprise System.

It is also designed to offer Artifical Intelligence Assistance [ Advanced Interactive Intelligence - A program designed to Collect Enterprise Data, Think and Provide Strategic/Intelligent Supervisory Insights and Guidance ].

**What is the Working Principle and Foundation for Building an Advanced Enterprise Control System ?**

Enterprise Flux is similar in working principle to SCADA [ Supervisory Control and Data Acquisition systems ] and works at a Abstraction layer above traditional Enterprise Process systems like ERP software and IT systems.

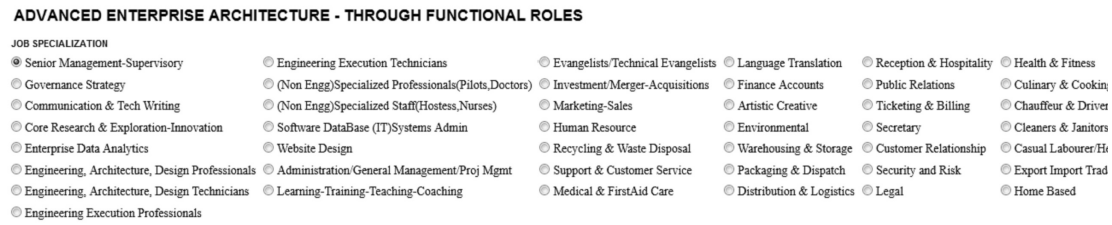
In anology Windows and Unix systems run the Computer Hardware system, while Enterprise Flux runs Enterprise Systems. Note: ERP SYSTEMS like SAP collect highly detailed info like Invoice number, product numbers etc which makes ERP systems a Low level granular Data Dump. Enterprise control systems inturn work with High LEVEL Data that provides effective Enterprise Governance mechanisms/tools.

Enterprise Flux Systems may Collect Enterprise Process and Supervisory Control Data from Users through traditional computing Terminals or even Handheld Microcomputers like the STEX pad. [ STEX Pad are Connected devices and can be usd with ease at Remote Locations ].

**ADVANCED ENTERPRISE ARCHITECTURE**

The Enterprise Flux system acts as a Functional Control system, as well as a Reporting system to the Enterprise Work Force. ( This method implies that Individuals in the workforce don;t report to a Hierachially superior Boss, but report to the Enterprise Flux system Instead.

Note: People and Heads of Organizations leave and move out, but the Master system control Enterprise Flux remains as an Irreplaceable part of the Organization allowing constant control of the Enterprise. All Enterprise Commands, control structures & actions are made through the Enterprise Flux System through Qualilfied Members and Administrators.




**ADVANCED ENTERPRISE ARCHITECTURE - PART 2**

Static, Aged, Large organizations may operate in a new "structurally designed Enterprise Structure/model" called the "Derived Enterprise model". A new Enterprise Division is "derived" ( from the Parent Organizations and Accomodates the Highest Performing workforce in the division" ). Senior Supervisory professionals act as Supervisors and facilitators at the Parent Division/Enterprise, controlling the Sub divisions. Note there could be More than 1 Sub Division to accomodate a large numbers product segments or Functional disciplines. E.g Sales division work for a Single division and sell for multiple Product Segments or services cohesively and act as an Intermediate cost center.

# Enterprise Flux [ Core System ]

The Primary Design Goal of Enterprise Flux is to Measure, Monitor & Manage Enterprise Processes.
This System Leads to Enterprise System Automation, Enterprise System Intelligence & Enterprise System Efficiency.

The Data ssociated with the Enterprise Flux Systems is High Precision Intelligent Data - That Forms the Foundationof Enterprise Intelligence and Artificial Intelligence.

Every Enterprise Consists of Core Functions or Disciplines which are Further Divided into Enterprise Sub Processes. Further Each Process has People Attached to Processes. In Summary the Enterprise Consists of

1] Enterprise Core Processes
2] Enterprise Sub Process
3] People Attached to processes

## DASHBOARD

World View [ All Integrated ] , Europe, Germany, Berlin...

Geography

[ All Integrated ], Power Systems, Power Transmission, Software Systems, Automation

Focus

1 Jan 31 mar 31 June 31 Oct 31 Dec 2009 2010 2011 2012 2013 2014 2015 2016

Time

## Systematic Enterprise Processes and Subprocesses by Discipline

Filed for Patent & Discussed in the Book - Intelligent design techniques to build a smarter planet

Intelligent Design functions - Design/Structure, optimization, Risk, Data/storage, Execution , Rules, Boundary conditions, Enablers.

State Of a Process
Process Alert flags
Boundary Conditions checking
Structured Exception Handling
Timeline Based Process Summary

Exceptions in One Enterprise Function - Like Supply Chain affects Other Enterprise Functions like Manufacturing.

All Process Exceptions are Registered in the Enterprise Exceptions stack, and can be used to Systematically Eliminate Enterprise Work Flow Bottlenecks.

People
Team Members with Contact nos, Skill Level Experience.
attached to Processes

Execution Process
Sub Process
Sub Process
Optimization Process
Sub Process
Sub Process
Sumary
Est
Act
Time stamps

Enterprise Operations — Enterprise Operations — Enterprise Operations
Process — Sub Process

Granularity
Big pic
small Pic
(details)

Retail Outlets — Retail Node — Retail Node — Retail Node — Retail Outlets — Retail Outlets

Laptops
Televisions
Cameras
Vaio Z (50)
Vaio S

( OEM )
Assembly based Production
Machine Mfg production

e.g. Building a Full Turbine = ABCD
Mfg - Low Pressure Turbine = AB
Mfg - High Pressure Turbine = BC
= CD

Low pressure Turbine
High Pressure Turbine

Consulting - Planned ( 100) exec (85)
App Dev- planned (100) exec exec(110)

Planned [300]
Executed [200]
Vaio Tx
Vaio X
DSLR Camera

Mfg/Production — Services — Mfg/Production

Main Assembly part 3 — Low pressure Turbine, High Pressure Turbine
Main Assembly part 2
Main Assembly part 1

Materials/Components — Materials/Components — Materials/Components

Services | Projects have different parameters compared to Product/OEM/Mfg disciplines

Remote Smart Phone Based Updates - Updating the State of a Process [ Remote Project or Shop Floor Process ]

OnSchedule | Out of Schedule
Roads , Rail , Bridges, Flyovers ,Software , Powerstations
Estimated Execution Actual Exec
Projects

Motion & Control - services
Atlnta-ny-tx
Berlin- Zurich
OnSchedule | Out of Schedule
Sustainable | Unsustainable
( Load )
Services

| | Machine X34 | Ship X12 | Aircraft Y123 |
|---|---|---|---|
| Uptime | 84% | 84% | 84% |
| DownTime | 16% | 16% | 16% |
| Repair Expenses | 10,000$ | 10,000$ | 10,000$ |

Engineering Design — Engineering Technician — Maintenance Technician

Skills/Discipline
Level 1 :[50]
Level 2 :[100]
Vacancies:[25]

IT Servers — Mfg — Software

Ice X - V6650Servers 1.75M$

Specialized Purchases
Machines M234 Cost 2M$
Components NJI8889 Cost .1 M$

-- Software
-- 50 Fluid Dynamics
Db Software 50 Processor - .5M$

Factory/Enterprise/Buildings ( Internal )
Tech/Software/Internet Control systems ( Internal )
Customer Support/ Servicing ( External Process )
Sales & Retailing ( External Process )
Supply Chain & Logistics ( Internal & External )
Warehousing & Packaging
Risk Testing/Quality ( Internal )
Production/Service Process ( Internal ) (Inventory)
Materials, Components Storage ( Internal )
Projects/ Services
Research Design ( Internal ) ( Internal )
Maintenance & Repair ( Internal )
Marketing - ( External Process )
Human Resource ( Internal & External )
Accounting ( Acountant )
Purchase
Vendor/ Supplier Store ( External Process )
Learning & Training ( Internal )
Energy & Power Systems
Production Store
General Administration
Planning & Strategy
Supervisory
STEX Communications [ Internal Systems ]

Enterprise Flux System Administration Team
Enters HighLevel Abstract Data into the Enterprise Flux System.

E.g If ACompany Has sold 1000000 Laptops in the MARKET from March 1 to June1 the EF system collects this Hi level Data like Number of 1,000,000 of Laptops sold Between March 1 to June1

Low Level Data Like Machine Number and Invoice Number are usually Collected the Low Level Data Dump systems like SAP.

Enterprise Flux is designed t Work with High Level Data. Enterprise Flux may also be configured to Store Low Leve granular data exactly Like SAI systems.

In case Of Enterprises that do want to replace Existing SAP systems A Team of Enginner May Manually Feed High Leve Data into the EF system - a "Manual Interface".

This System is a Totally Dyna System - that it allows an Enterprise to Create it's own custom Processes as per requirements.

Macro Views and Micro View are made available conveying the state of an Enteprise Process or Project Device.

This System Enables a Totally Automated Data Driven Scalab Enterprise Control System.

Certain Industry Process - "S Information" like Remote Projects or Field Projects or Factory Shop Floor Data or Logistics status Can Be Entered into System through Computer Interface or a Sma Phone PDA Interface into the Enterprise Flux System.

Note: Every Core administrativ Process of an Enterprise is Entered into the Enterprise FL Core by a Dedicated Team of Enterprise Flux System Administration Team.

Each Enterprise Process being managed by the Enterprise Fl System May have it's own Dedicated Team of Process Specialists to Verify Data befo Entry into the System.

Process System Admin 1 to 4 Feed DATA into the System in Parallel, so as to Ensure Security and Eliminate Fraud/Errors in the Enterprise Flux System.

THINKING SYSTEMS
[ Intelligent Data Foundation ]
Advanced Interactive Intelligence
Artificial Intelligence

BUSINESS INTELLIGENCE & ANALYTICS

All Sets of Processes
— Retail Sales Unit X
— Retail Sales Unit Y
— MFG Unit x
Mfg Unit Y

Date Time ( 1 Jan - 31 March ) 2011
Date Time ( 1 Jan - 31 March ) 2012
Date Time ( 1 Jan - 31 March ) 2013

All Processes have a Macro View & A Micro View as required by the Enterprise Management Teams

Project Execution Views & State are shared with Stakeholders/Cients

Long Duration Processes Like Power Projects are divided into short Duration Processes for Monitoring.

Short Process cycles
e.g. Building a road ABCD
PROJECT1 = AB PROJECT2 = BC PROJECT3 = CD

Processes may be Embedded with Live Images & Video Feeds

All Internal Communications are stored in Enterprise Internal DB.
Communications are facilited through Web & Mobile devices.
[ This system Operates with Zero Risk of Comunication Leakage, even under the Eventuality of Smart Phone or Mobile Device Loss ].
Eliminates need for Separate Email Servers.
Allows Individual Communications as well as Targeted Group Communications.
Large Enterprises may Maintain Own DB. Small Mid Sized Business May Utilise Data center Services by THE STEX Network.

Private Internal Communications
Emp ID: QWER Level 7
Emp ID: FGH Level 9
Emp ID: NJK Level 6

The STEX network Communications System Can also be utilised Internally inside the Enterprise For Vital Highly Targeted Information Flow Within the Organization [ An Alternative to Email ]

Fully Encrypted Communicatio Not Exposed as a Public Email Message.

# 5) STEX FCX Reality Compilers [ FCX Programing Language ] - For Dynamic System Programing - Theoretical design

---

Reality Compilers are a Special Type of Compiler that can can be used to Facilitate Building of Robust, Complex and General Purpose Software systems like - Next Generation Operating Systems, Network Protocols, Games Development, Enterprise Control Systems & Realistic Artificial Intelligence Systems. The FCX compiler uses a High Level programing Language called FCX. It Includes an Imperative programing model along with a meta data model and a functional programing model.

1.FCX is a Visual Intutive Programing system with a Number of built in Programing Models. ◦ Object Lifecycle Model

◦ Object DATA Model

◦ Object Functional Model

◦ Object Relational Model

◦ Imperative Programing model

◦ Object Interaction Model

◦ Spatial View modelling

◦ Virtual Fabric Memory store -non file system memory model

◦ Fluid datastore Model

**2.Object Life Cycle Model** - Used to create,destroy and Clone Objects.

**3.Object Data Model Programing** - Consists of Object Data Points and [ 1] Primitive Datatypes like Int, Float, String, Boolean, Date time etc. 2] Abstract Datatypes like: Array, Linked List, Lists, Stacks, Queues, Trees, Indexes. It may Include Other special Data Types that allow easy System Software Development. ◦ Freescaling Structured Object List[] - Objects with Multiple columns and Dynamic Column datastore. [ Freescaling Data without creation of an additional object record ]

◦ 3D Array Object Data type - To Store Custom Datapoints like X,Y,Z Pixel coordinates, Lighting, Colour etc.

◦ Object Pattern Data Type[], String Pattern Data Type [] - To Store Patterns of Data.

◦ Time Stamp Datatype[] - a Dynamic Array of Time Stamps that are auto generated and Cannot be changed.

◦ Object State Stamp[] - Can be used to save the state of the current Object in a Dynamic List or File. A list of Object States may be used for Analysis. It May also be Incorporated into the Object Patterns Data Type for a variety of applications.

◦ Virtual Fabric Memory Address [ Main Memory | Flash Memory | Hard Disk | External Card ] - Used to Set Memory address of Objects in a Particular Disk Medium

◦ Geographical Location[], Space Coordinates[],

◦ CPU Core [] - For Parallelizing Operations &, Energy [] - To detect Battery energy Levels for Process optimizations

◦ Object Reference DataType[] - Is a File Pointer that can be used to Reference Objects and store them in Special Graph Data Structures or other Data Structures like Linked Lists.

◦ File Data Type[] - Used to Read or Create a Physical File on the Virtual Fabric Memory. These Files may Store Objects of a Custom Data Structure type and read Objects of a Custom Data structure - E.g. Like the Freescaling Object. This also allows creation, Modification and Storage of Open Standard File Structures.

◦ File Reference Data Type[] are Used as Physical File Pointers that may be stored as Linked Lists, Maintained in Transaction Logs and Indexed.

3D Array Data Type is used to set Object Structural/Design Boundary Points using the X,Y,Z Coordinates. The Polygons, Colours, Textures, Shading & Lighting Functions may be applied on the Data Points. Application areas for this this Data type is

1.Game Object Programing - Which May be Dynamically Transformed through Structural Transformation Functions.

2.Operating System Windowing Manager.

3.Interactive Movie Programing.

4.CAD - CAM - 3D Design Objects for Electronic for 3D Prototype Printing or Machining.

Patterns Data Types are [ 5,6,7,8,9,X Variable ] Data Point Patterns of [ Dimensions like - String of Words, Object Columns ] and are called Pattern Strings. A Strings of Words [Text] may be stored as Pattern string inside a Pattern List, with an Unique Pattern Identifier and this UniQue Identifier may be attached to a Complex Object Record as a Pattern Dimension for Facilitating a Pattern search. Indexes of Pattern Data Types may be created for these Pattern Dimensions or Normal Object Dimensions for High Performance Search Queries of Complex Contexts & Synthetic Data. Pattern Indexes/Lists may be build at Two Levels or Multiple Levels, with each Having a Different Number of constituent Data Points. Pattern Matches could be of the Type:

1.Exact Pattern Match

2.Synonym Pattern Match

3.In Synonym Pattern Matches the Entire Input string is converted to a String of Representative Synonyms before creation of a Synonym based Pattern. When Searches are conducted for a Pattern we May use Both the types of Patterns for Finding a Record Match.

Note: These Pattern Matching Technique can be used to Map A Natural Language Pattern to the FCX Object Relational Patterns for Fluid natural Language Processing in Realistic Artificial Intelligence Interactives. Internal Mechanisms may include Converting a Natural Language String to an Object Pattern of Synonym ID's where a Synonym ID is an Unique Synonym ID representing a of Pattern of Synonyms of Words. The object Pattern of Synonym ID's is mapped to Pattern of Synthetic Contexts in the FCX object Relational World. Note: String of Words may be stored as a Single Pattern or an ObjectSet of Multiple Patterns.

Pattern Based Data Points can be used to Store a set Latitude Longitude Datapoints that Consitutes a Road. This Can Provide Accurate Point to Point Map Navigation and Alternate Traffic Routes when, Integrated to Artificial Intelligence Interactives. [ Note: Data Points can be collected by Driving a Car

through a Particular Road ]

Data Types like Time Stamp[] May be Used to Store The Multipoint Process Time stamps, e.g. Aicraft Landing Times at a Specific Airport or delivery Time of a Parcel. Delays may also be stored as Time Stamps and Ultimately as pattern Datatypes and These Patterns could be used to spot Delays in a Process or Low Load Factors of in an Aircraft route.

Virtual Memory Address Data Types can be used to Create Tiers of Memory Within a Program. It Can be used to create and Manage Lists of Free Memory Blocks, that may be used for Purposes like Automatic Memory Management Processes and additionally for Physical File Storage & IO on Disks.

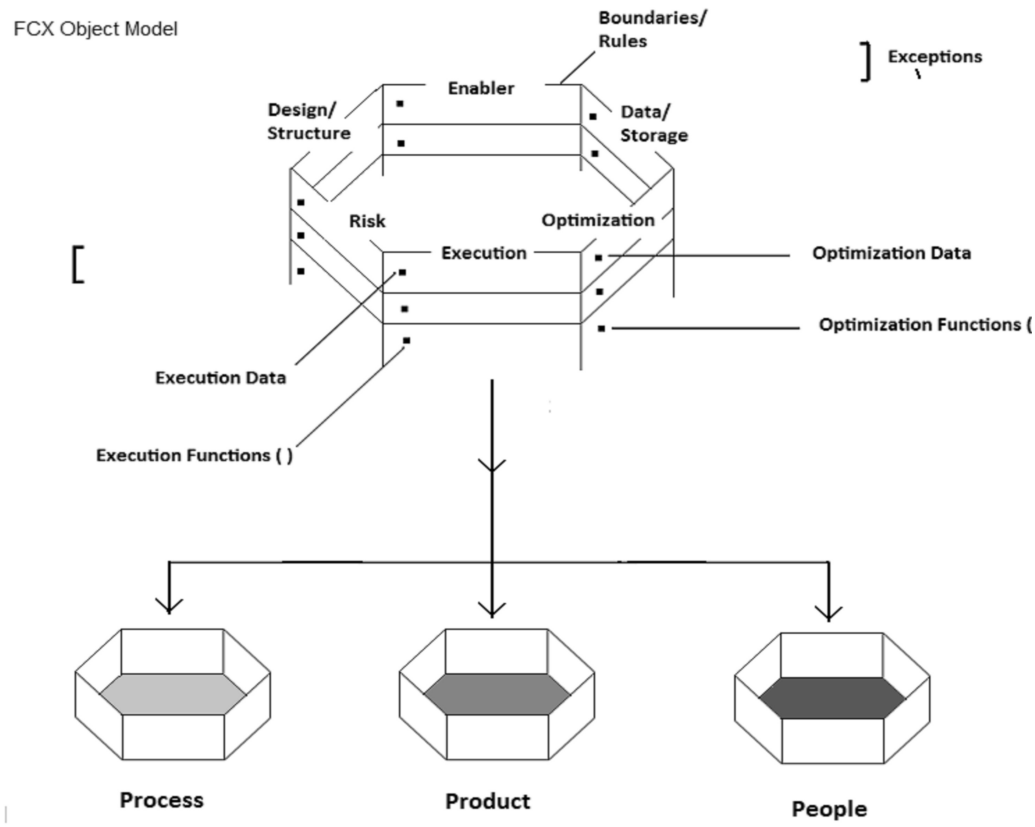
Geographical Memory Addresses may be used to Store Latitude Longitude and Height of an Object. Space Data Type may be used to store XYZ Central Reference Space Point of an Object - used for applications like Graphic Object Positioning on Screens depending on a Form Factor. An Object may use a 3D Array of XYZ space Points to create a 3D Object Data Structure. Other Application specific Dimensions like Pixel Colour, Lighting may be used in conjuction with the Space Points[] in a 3D array. e.g write an algorithm to capture all the distinct pixel points in a [ realistic picture ], and place each distinct color's pixel coordinates in a single linked list array of 3d objects. it may look like : Blue list - [1,2,3],[ 1,2,4],[ 6,7,5]..... green list - [5,2,8],[ 7,9,10],[ 46,77,58].....

CPU cores and Battery Energy are used for Parallel Processes allocations and VIKM type intelligent Process Scheduling and Energy Conservation.

**4.Object Function Model** [ Explicitly Distinct Implicitly Coherent Factorized Intelligent Design Functions ]. E.g. Design[ ], Execution [ ] etc, External Physics Engines/Libraries [ ]. Note: EDIC Functions may be [ Public ] or [ Private ]. E.g. A player Has the Ability to Initiate Action Functions like "Shoot [ ]" at enemy which is classified as an Execution Function - "Execution.Shoot[ ]". Another Function could be "Design.DrawTextures[]"

Note: An object may Constitute of Multiple Execution Functions, Graphic draw functions( state transform functions ), data points , optimzation functions etc. Storing Object functions repeatedly within each object may waste inmemory space therefore objects may store "pointers to the Function" in the Inmemory objects. These Object pointers point to the real functions, and in principle, is similar to code reuse in C++ (oops).e.g. Multiple game objects point to a single exec or draw functions. This could be achieved by adding a metadata attribute [Pointer] to any function in the Game object, in order to recognize that the function is a Pointer to the "Real Function".

Note: Data Points of the objects always remain within the object , because the datapoints represent the state of the object. These datapoints may undergo state transformations, allowing state changes of objects.




**5.Object Relationship Model** - This design principle allows creation of Object - Object & Object-System Hierarchial relationships, and is represented in the system by an Object relational Graph Data structure. The Graphs are created during Program Compilation but are dynamically processed and kept "upto date" to allow "High Context" Object State Updates taking place in the system. Note: State Updates are Handled by Object State Management Functions. E.g. In the context of a game - weapons are child Objects to a Parent Player. When a Player drops a weapon, the player Object's Weapons State is updated by accesing the object through the object graph.

Objects with No Relationships [ Neutral Relationships ] are also maintained in the Object Graph. E.g. Collision Physics between two Objects with neutral relations. Priority attributes are Priorities attached to the Object Variables [ Dimensions ], to allow sequential ordering of Object Graphs, which would facilitate instant State Updates of objects, depending on the Priority Dimensions. E.g. In a Game Scenario, if Priority is Set on the "Space" dimension, and a Blast/Collision Occurs, the Nearest Objects to the blast/collision would need a State Update [ Collision physics ]. This would be possible through traversal of Object Graphs which are Sorted by Space or Distance and represent the actual Objects in memory. 1.Relations between Objects are Created by attaching Intelligent Metadata to them. This Process is termed Intelligent Object Life Cycle.

2.All Objects and MetaData Attributes Have Names and an Identificaltion Code.

3.The Objects are of the Type -- [ Simple Object | Complex Object | Simple Process | Complex Process | Intermediate System | Complete System | Organism ]

4.Objects have Relationship Metadata associated between them -- [ Parent-Main | Child-Sub | Partner | Opponent | Friend | Neutral ]. Default Relation is "Null".

5.The Objects have additional Metadata - [ Private | Public ].

6.The Objects have "Multiple" Type Metadata Points associated with the Objects - Simple 8 Point Matrix or 8 by 8 Intelligent Design matrix consisting of Types like [ Design ] [ Structure ] [ Data ] [ Storage ] [ Risk ] [ Enablers/Resource ] [ Optimization ] [ Exceptions ] [ Rules ] [ Boundaries ] [ None/Null ].

7.They may have Inbuilt Object Lifecycle/State Level 2 Metadata types [ Medium, Measurement, Lifespan, Current Iteration, Current or Previous State, Priority, Parallel Tasks/Threads, Views, Current Location [From], Current Location [To], Created Time, End Time, Complexity Mode of Object/Process, Size or Scale of Object ]

8.The Objects have additional Level 2.A Metadata like -- [ 1] Primitive Datatypes like Int, Float, String etc. 2] Abstract Datatypes like: Array, Linked List, Stacks, Queues, Lists, Freescaling XML Data List 3] Undetermined 4] Null 5] Custom Enumerations like: 1.Size: Small | Medium | Large.

2.Duration Task: [ Short | Medium | Long | Unknown | Continuous ]

3.FileTypes:Program Registry, Directory, File

4.Current Location: From [ X,Y,Z ] to [ A,B,C ] Coordinate or File Location [ X to Y ] Memory address or Parking Bay 1 to 2. From and Two Locations may also be the same.

5.Transaction: Voice, Realworld Handshake, Online

6.Internal Storage: BIOS Memory, HardDisk, Main Memory, HighSpeed Cache

7.External Storage - USB|CD|DVD|Omega Drive or Physical World Storage: Storage Shop1, Shop2, Godown etc

8.Mode of Object/Process: Simple Functions | Standard Functions | Complex Functions | Trail & Error Functions etc ]. Note: an Object be a Complex Process consisting of Multiple Transaction Functions or Multiple Complex Functions.

9.Size or Scale: [ Very large | Large | Medium | Small ] -

E.g. Large Gas Turbines, Mid sized Gas Turbines | Small gas Turbines

E.g. Large Video File | Mid Size Video file | Small Video File

10.Priority: System | Very High Priority | High Priority | Medium Priority | Low priority

11.State: OnSchedule | Delayed | Exception | Active[ Unknown]. Note: Detailed Custom State of a process may be obtained by writing Execution.State[] functions that collect datapoints to show detailed state.

9.Objects may have Optional Customizable Level 3 Metadata called Domains -- [ Homosapiens | Animal Species | Aviation | Computing Hardware | Software | Operating System Memory Management etc. ]

10.The Metadata attached to objects are Synthetic Intelligent Data Collection/Patterns to convey a Sharp Context about the Object. The Context Synthesis happens by Combining Attributes aross levels of MetaData like [ Organism + Opponent + Private + Lifespan + Virtual Computer Game Player ]. Additional MetaData May be Added as Datapoints to the Object as per Requirement of the system E.g. Body Damage or Energy Levels etc.

11.All MetaData Values may be Stored as Enumeration OPcodes, therefore they would, Occupy Very Little Memory and Database Space. It may be Noted that the Objects are in memory Objects, however access to the Objects and State Changes are Facilitated through the Object Graphs [ Pointers to the Objects ]

**6.Imperative programing Model** - They are essentially Object State Management Techniques to allow Cascaded Triggered State change of Related Objects in a Systems using "Object Relational Graphs". The object graphs are continuous functions that reorder themselves to maintain an ordered list of High Context Object Pointers. This allows Predictable "Paths" for State change of nearby objects

by another Related Object. E.g When a System Event occurs, High Context Objects are affected but other Objects remain unaffected. In the Example of a Game Play -When One Player explodes a Bomb, the Bomb Explosion Event may affect the State of Nearby Objects, while distant Objects may Remain unaffected.

The Object Relational graphs may be generated from the relationship Metadata data values & and are Continuously sorted according to Dimension Priorities, so as to Facilitate an action/state updates of High Context related Object Variables. This Programing Paradigm is called Programing with Side Effects, where "ONLY" a small set of "Related cascading Objects" undergo state Change, when triggered by an object/system event, thus forcing nearby related objects to undergo state change. The state change of "Objects" or "Processes" or the "Entire system" may finally translate to a change in system behaviour on the graphics screen.

Note: Object Lifecycle Functions are used for creation, destruction and duplication of Multiple Objects, while State Functions are used to control realtime "States" of objects. Every Object May Have a Function marked with a Attribute [ Event ] and may have Conditional rules of State Data, which helps in Triggering Events. When The State of an Object is Changed in a Manner, that it matches the Conditional rules of State change described in the [ Event ] attributed function - a sequence of cascading state change functions may be triggered into nearby related objects. This Happens as a Sequence Chain in a Sequential manner, so that no deadlock conditions arise. This implies that State updates of objects through object Relational graphs are sequential processes instead of parallel processes, however Functions within the object, that trigger a state change, may be executed in parallel. In principle, this resembles a Sequence of parallel tasks. [ It is similar to the Physics principle of Nuclear Fission ].

Note: Whenever the Object State Data Changes, the Functions attributed as [ Event ] are checked for an Event Triggering Boundary Condition. If Conditions are Valid an Event is Triggered.

In principle this Compilation technique leads us to create Dynamic software Systems where, the Quantum state of a System evolves gradually or varies, based on the actions of Individual Constituent Objects or Players of the system. This dynamic system has no Fixed Final outcome or state. The Immediate Next "States" of the System are however predictable based on the Current Action of the objects or Actors of the System.

Example:Processes within an Enterprise are "interrelated" to other processes. A Component Storage Unit may Have the Capacity of 100 Components. When a Component is removed from the Storage Unit the Number of Total Components is checked and if low in number, an event may be triggered to to another Related Parent Storage to supply it with Further Components. On Unavailibility of Components, a Process Exception May be raised and Recorded into Dynamic Exception Handling Stack Data Structure. The Flow of Exceptions across processes would enable Systems to Point at the Error Source, so as to enable corrective Action. Exception Flows may be [ Intra Process or Inter Process ]. Example : Manufacturing Exception May lead to a Component Storage Exception and this may further lead to a Supply Chain Exception. These Patterns of Exceptions can be analysed for Smoothening a System Workflow.

E.g 2: EARTH is a Dynamic System. It's Constituents are people, enterprises, processes .State changes in People and enterprises or processes leads to cascading state changes in other constituent elementsor nearby objects ( other people, other enterprises etc ) and this leads to different "Quantum State" outcomes in the Earth system. ( Think !- about economic conditions Like Recessions )

Advanced Artificial Intelligence Interactives may also Quickly Trace Exceptions and help in Trouble Shooting of Problem Areas, leading to Fully Automated Enterprise Systems.

**7.Object Interaction Model** - COLD Machine Technique - Interactions are "Process Interactions" between two Objects [ or Two Processes ]. This Interaction is Invoked by One Object Changing the

State of the Next Object by calling the Second Objects Internal Method. The Internal Method may choose to change the State change of Invoked Object by Rules and Logic and this state Change may cause a Cascading set of Event Triggers that Change the State of Other Related Objects [ e.g A Domino ].

Cold Machine Technique - This Interaction Model Saves Game Objects from Going through External Game Loop Cycles and Internal Object Game Loop Cycles, listening for Events. Out of Context Game Objects may need to be ready for State updates, but may save CPU cycles by allowing only Chained Triggered Events. Note: Saving CPU Cycles in Mobile Devices Saves Battery Life. Some Processes like Graphics Rendering may need to be run Rendered Continuously, Though Background Processes may Run "Cold". On Larger Server Systems on Public Networks A Huge Amount of server CPU Cycles may be saved, adjusting to User Demand - Inturn Saving Energy Consumption. CPU Strategies may involve shutting down maximum CPU Cores while keeping a spare core active - for a Responsive system Design.

Intelligent Operating Systems & FCX Programs The Virtual Fabric Memory Filesystem technique offered by FCX has been Described in sections Below. , offers a Facility of Accessing a Mixture of Main Memory, Flash Cards [ like XQD, Compact Flash Cards ] and Harddisks in Programs. Next Generation Operating Systems and other Programs written by using FCX Compilers can use Intelligent kernel manager Techniques to, use cheap non electricity consuming - Flash Card based Virtual Fabric Memory [ As a Buffer memory for Idle system condition/state ], while Putting Hard Disks to Idle. With Drop down in Process Loads - using the cold Machine Technique, General Purpose Programs may also run and reside in Main Memory while Hard Disks are set to Idle. On Increase in Process demands Hard Disks could run at Max Speeds. This Technique would Drop Down General purpose machine & Datacenter Energy Consumption needs.

**8.Parallel & Sequential Model** - This system may be designed for Automated Parallel or Sequential Execution of Functions of an Object, with the Default Functionallity being Parallel Execution. Every Sub Function of a Complete Function within an Object may be Described by an Attribute called [ Sequential | Parallel] to facilitate Parallel or Sequential Exection of the Functions. The Compiler has the Internal Inbuilt Mechanism to intelligently handle Parallelism, by tracking callbacks. The Compiler would allow Programs to enable Parallelism by passsing on Information about total number of Processors and Processor Cores. This Implies that Aprogram Created with FCX would Autodetect CPU Cores on a Computer before executing a Parallel Execution Strategy. No there may be No Explicit Threading Libraries Needed to Execute Parallelism. 1.Example 1: Five distinct DB Fetch queries may be run in Parallel and subsequently Merged in a Sequential operation

2.Example 2: A Game Object May Involve Automatic Parallel Execution of Execution Functions and State Changes. Graphics Rendering Functions may Run at System Level.

**9.Spatial View Modelling** - Views are Functions that may generate Views/Perspectives of an Object or a System e.g. [ Objects, Process, Total System ]. Total System Views may Consist of: [ Priority 1 Processes, Priority 2 Process ] based on The Process MetaData. Other Views May be Of Custom Types like 360 Degree Stitched Panaromic Views. Another Example is the "STEX GRID search" displays complex views of data projected as Spatial Projections. Finally, Some Views may be Complex Rendering Views generated during Game Play. Games and Windowing Mangers may Use the Technique Pixel Lighting of Objects and Scalable Vector Graphics to Render Images on the Screen [ For Any Form Factor - Based on the Reference X,Y,Z Axis Values of Objects. ].

## 10.Virtual Fabric Memory

The FCX compiler uses a [ Virtual Fabric Memory ] System to allow Objects & Processses access to Memory. The system May demarcate Memory Zones in the Virtual Fabric Memory - High Speed Memory, Flash Cache memory and Hard Disk Memory so as to allow High Level Objects/Processes to use specific Processes to run in Specific Memory Zones.

Operating Systems may run Dedicated Processes that Move Objects from One Memory Zone to another as a Part of the Memory Management Process. System Processes may have a Fixed Memory Zone and Other Objects may be allocated to High Speed Memory Zones and Deallocated to LowSpeed memory Zones as a Part of the Memory Management Process of an OS or a Program.

Note: Internal Working Mechanism of the Compiler is that, when Working on Existing Operating Systems like Windows or Unix, It Detects Memory Addresses for the Main Memory and Memory

addresses of Other Devices though their Drivers to Create a Total Virtual Fabric Memory, with Demarcated Memory Zones.

The FCX Compiler may also Detect Total Number of Processors and Processor cores and Additional Hardware Device Info. The Advantage of the FCX Virtual Fabric Memory Technique is that it allows a Programing Model to use Demarcated memory Zones, which is not offered by present day General Purpose Operating Systems or Programing models in Compilers.

In the Program Model an Object may use the Virtual Fabric Memory using Enumerations like: Fastest Tier 1 Memory, Tier 2 Memory, Tier 3 memory etc. The Developed Programs autodetect Tiers of Memory Available in any System and make available the best Mmemory Options for the Program to Utilize. If the Computing System has additional Memory Cards like XQD cards, the Program would autodetect the High Speed Memory and utitize it to speed up Programs.

The Advantage of Such a Model is that Hardware Technology Innovations [ Faster Devices newer Devices ] may be Accomodated into Existing Hardware Systems incrementally as per requirement without the need for a Full system Overhaul. Newer Systems may have a Wide Variety of Configuration Options available [ Low End to High End Machines ]. The Hardware appears Tiered and Abstracted to the Operating System or the Programs - for them to Make The Best utiliation of available Resources. E.g A Program may Commit a Transaction on a Flash Memory if available and then transfer the data to a Hard Disk for a Final Commit.

## Virtual Memory Fabric Based Physical File Storage - No File System !

The FCX compiler Uses No File system to Store Physical Files or Objects. Every Object Whether a Video File or a DataBase Record is stored using the Virtual Memory Fabric [ HardDisk Demarcated Zone ]. The Object Created from the Compiler or the Program may be stored as a File using the Virtual Fabric Technique. The Pointer to the Main File [ File ID and Name ] are Stored in Tier 1 Zone of the Virtual Fabric as Ordered Lists of a Particular File Type.

This File Pointer Has the File Start Address and File End Address File, pointed to a Tier 2 File Storage Zone. The File start Address and File End addresses are Virtual Memory Fabric Addresses containing the Actual File. The Ordered Lists may be traversed by Search Algorithms to locate the Start & End address of the File, to fetch the Actual Physical File. This Virtual Fabric Address can be reached Directly on the Hard Disk, for a High Speed Buffered File read into a Special Tier Main Memory, from where the Data may be accessed by the CPU. All Data belonging to a Particular File are Stored in Contiguous Virtual Fabric Memory Locations.

A Virtual Fabric File "Dynamic Lists [ Where Pop Happens at One End and Push Happens at the Other ]" Stores the Current Contiguous Blocks of Free Virtual Memory Addresses along with their Size [ Contiguous Blocks May be greater than a Reasonable Size of 3 Mb to be Placed on this List ]. When a New File is to be written, an Algorithm Finds the Most Fitting Contiguous Location to write the File, and it's Start and End addresses are updated to the File Pointer.

When a File is deleted the [ File Pointers ] are Updated. The Immediately Freed "File Start" and "End" Address Locations are updated to the Virtual Fabric File "Dynamic Lists" as a Contiguous Free Memory Block [ Along with Size of Block & only if Block is greater than 3 Mb in Size ]. A Second Minor "Dynamic List" of 1000 [ Customizable ] Memory Addresses with Contiguous Blocks of Virtual Fabric Memory upto the size of 3 Mb are also Maintained.

This Minor List is used to Write Files of Small Sizes. A Dedicated Process May run to Rearrange Small Blocks of Defragmented Files to make Them Contiguous. This Virtual Fabric Based File Pointer Based System, is a Fast Technique to Read Files from the Disk.

## Fluid Data Store

The Internal Construct of the STEX Fluid DataStore is Described as Follows. Every record in the Fluid Datastore may be Represented by a Single File, and this File is represented by a File Pointer. The File Pointer is Stored as a Ordered List Object in a Tier 1 zone of the Virtual Fabric [ Hard Disk Zone]. Records may be Added, Deleted and Searched on the List by Maintaining an Ordered Ordered List.

A File Index may also be maintained for High Speed traversing of the Ordered List to Locate a File. This Helps in Extremely Fast Datastore Record/File Pointer Searches, Which lead to the Actual Data address Location. A separate Ordered List May be Maintained for Other Dimensions of the Object that are Frequently Used for Searching a Record. E.g The STEX Grid ID or an Email Address.

E.g If a PersonID has Multiple Bank Accounts and Each Account has Account Records, the Datastore Allows Direct Querying of Bank accounts Object or Accounts Records as The PersonID is stored as Associated MetaData with the Bank Account Object.[ Parent Object and Metadata Objects are Separate and Metadata Objects have Parent Objects id's as Metadata atatched to them - thus Querying does not Involve hierarchical Object Relations in Search Queries ]

Every Object in the STEX Fluid Datastore is Alloted a Dedicated Virtual Memory Fabric Space of Minimum 1 Mb. [ This Size is Configurable depending on the Object Size to be Stored ]. The Maximum Size May Have no Upper Limit. This Implies that When Small Objects or Files Are Written to the Disk They Occupy a Minimum of 1Mb, even though the File Size May be less than 1Mb. A Freescaling Object Record Belonging to a Person may occupy 0.5Mb of Data, while 0.5Mb of space is available for Future data addition.

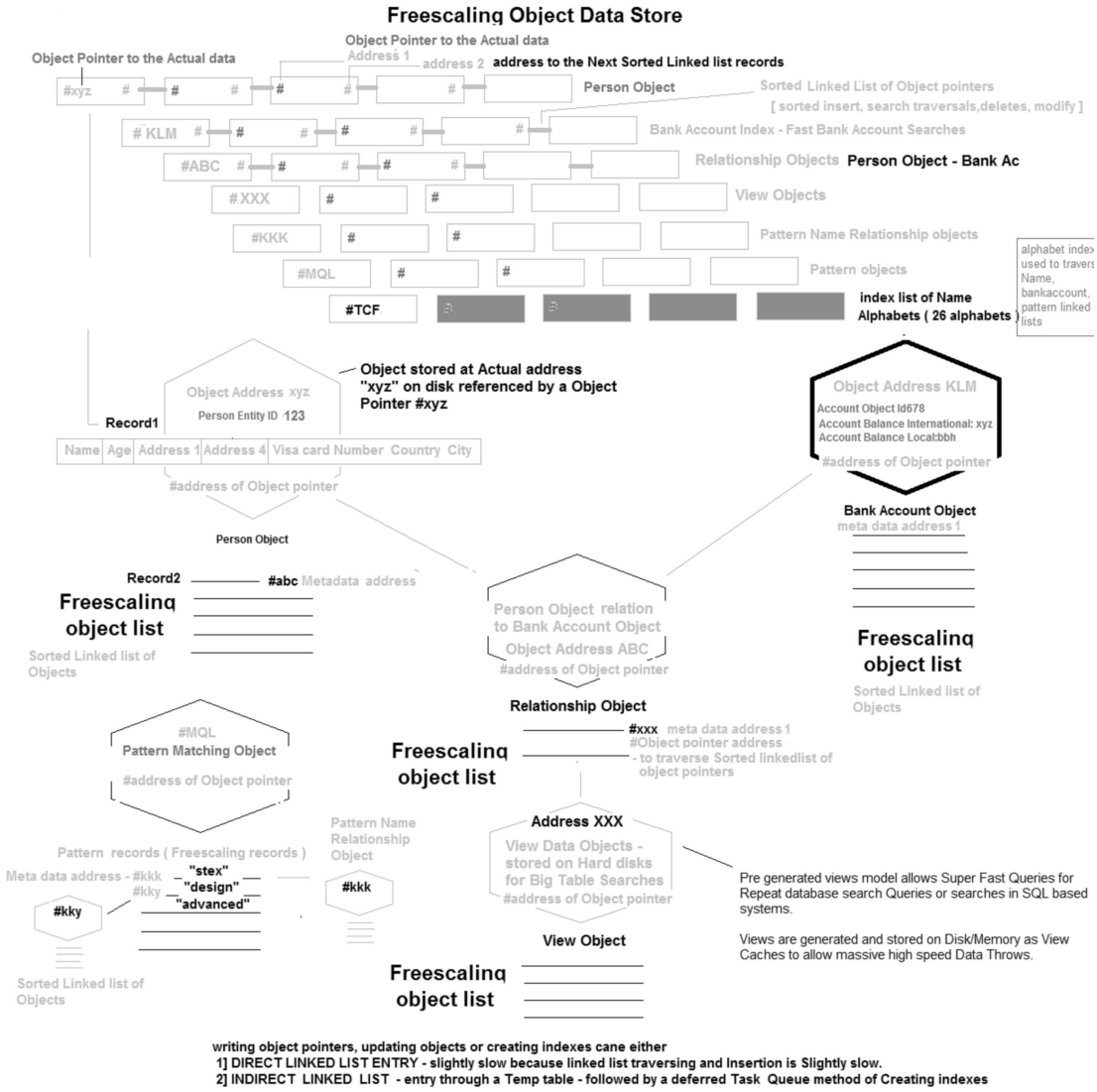
When File Size Updates are Larger than the Total Available Space for the Object, the Entire Object is written into a New Virtual Fabric Location with the Addition of 0.5MB Free space for the Future use. By this Technique Maintaining a Record of 6.5 Billion People on this Planet would require only 6.5 Petabytes of Average Total Datastore Memory space. This Technique Involves Very Little Fragmentation and may additionally be healthy for Flash Based Storage devices which suffer from Constant Writes.

Note: Normally Transation Based Objects may require only 10 to 50 Kb of Object Dataspace instead of the 1Mb mentioned. Large "Contiguous & Free" Virtual Fabric Space Addresses are mained in Lists for the Purpose of New File Additions or File Relocations.

A File Transaction Log is used to maintain Milestones of a File stream being written to a Disk or is being relocated from one disk zone to another. Transaction Logs are maintained to check File Integrity in case of a Power Failure, and are always checked during a System Startup to check an improper Disk shutdown during a filewrite process.

Note: A File stream being written onto a Disk Consists of a Set of Milestones which are maintained in a Transaction Log along with a Milestone Start and End address and an End of File Descriptor. A large file of 5 Mb May consist of 5 Milestones, each with a Start and End address. A small file of 2 bytes May have 1 Milestone with a start and end address.

If the Transaction Log is Intact the Ordered List File Pointers and Contiguous Free Space Lists may be Updated to the Disk. In Case of a Power Failure Scenario, Between a File Write Process, The System may check the Transaction Log for Data Integrity of File Milestones. If Milestones are in the right Sequence and the transaction log has a File end descriptor, the Linked Lists and free contiguous File lists are updated, else the stream of Bytes on the Disk are updated into the Ordered List as a Broken File, which may be deleted if Required. The Same Principle is used to store Files in Storage Area Networks instead of a Small Disk.




## Massively Parallel File or Database Record Reads & Failovers

Finally, Object Record Writes or File Writes may Happen Parallely only Limited by the Disk IO Capacity. This system Design principle Allows Fluid Datastore to Run Disk Replication Strategies that are extremely Fast - Where a File Write or Record Write Process involves Copying one Set of Bytes from One Disk Area to Another, without the use of a File system based Disk Replication. In such a Strategy The File Pointer may have Address locations of two or More Replicated File Data, Located in Different Disks Zones. This Technique may be used for failovers. During Parallel Disk Writes - The Transaction Log is Verified before selecting a Start Address for Contiguous File Writing because There may be a Possibility of Contention between two Disk Writes trying to Occupy the Disk Zone [ Therefore in Principle the Transaction Logs are only Disk based dynamic lists that hold the Start addresses for every Current Parallel Write Disk operation ]. Duplicate File Pointer Linked Lists may also be stored for Security and Robustness and Scalability [ by allowing Parallel Data access from Disk Zones or Disk Machines through Loadbalancer algorithm - Higher IO Capacity ]. This Technique allows Massively Parallel Record Reads and Writes on a Disk or a Storage Area Network, by using Infiniband connects

for high speed IO.

The Fluid Datastore Facilitates Pre-Generated [ Pre Sorted ] Data Views ( View cached on a Hard disk ) and Data Models that allow Allow Fast Queries for Pre Determined Repeat search Queries. Fluid Datastore also facilitates Data Caching or Data View Caching for Repeated Searches. This Cache may be on disk or in Memory and removes the need for a separate Inmemory Caching system or a Key Store.

When Running on Proprietary File Systems of Operating Systems like Windows - the FCX based Programs Mark Certain regions of Hard Disk Space for Exclusive Virtual Fabric Based Usage - Similar to a Disk Partition.

## 6) View Caching Technology [ Pre Sorted ] for SQL/NoSQL based Database Systems.

This model allows Super Fast Queries for Repeat database search Queries or searches in SQL/NOsql based systems, which do not have the Technology concept of a "pre-generated Views -( View cached on a Hard disk ).

The Existing Technologies create a real time view of data,in memory.
Presorted Views are generated and stored on Disk/Memory as large View Caches to allow massive high speed Data Throws.

If these features are built into SQL databases - they would provide extremely high speed data fetches for Big Data Queries ( Queries with Large Data Results ). This technology feature makes sql/No sql server's fit for "large throw" data search applications.

Chapter 4

# Software Automation - An Approach

# Software Automation

**Software Automation for - Resource Planning, Enterprise Control & Processes Control systems**

An Entire Enterprise can be controlled or Governed by ( Monitoring & Management ) of People, Enterprise Machines, Enterprise Processes and Enterprise Products. This Technique is called the STEX - PPP Intelligent Design, and is, a part of it's "COMPLETE SYSTEM AUTOMATION" or "COMPLETE SOFTWARE AUTOMATION" solution.

## Software Automation: System Elements or System Components for, Automated Software Systems.

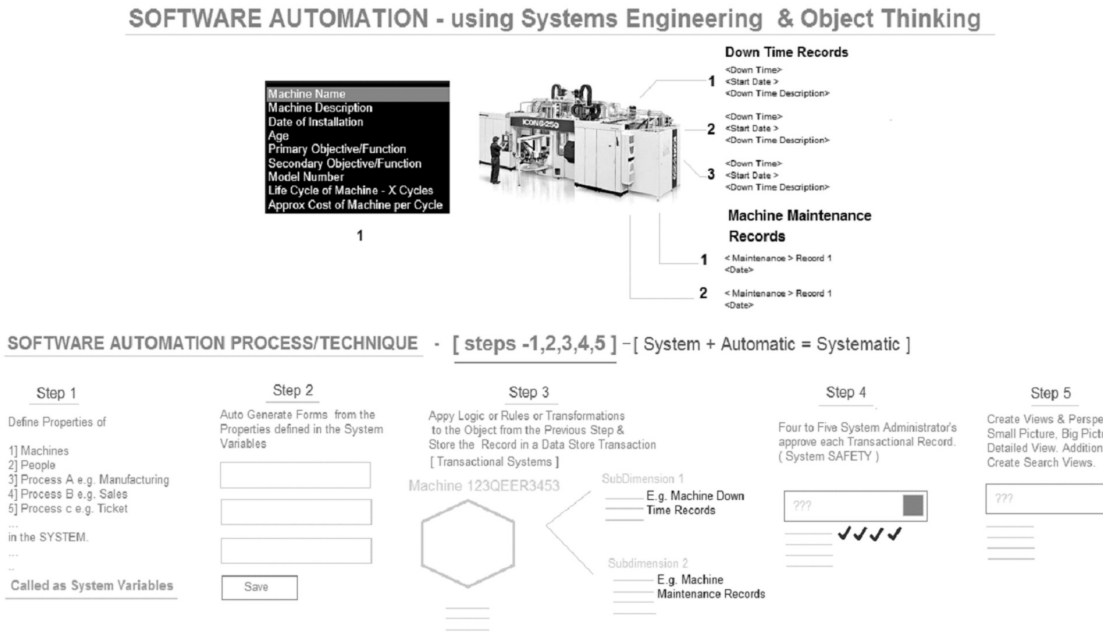
Complete System Automation for - Enterprise Resource Planning / Enterprise Control systems/ Enterprise Process Control Systems, Consists of 4 Main Parts

1. Enterprise Super Administrator screen, where The Entire System's - Properties/Variables are defined. ( E.g.A Class File )

2. Autogenerated Data Screens - are used to Feed Transactional data into the system through Process System Admins. The Screens are auto generated from the Properties and Definitions defined by the Super Administrator in step 1.

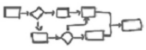
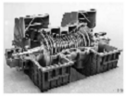
3. Approval Screens for System Safety - Allows approval Process for all Data transactions ( Fed into the System ).

4. View Screens - for Governance & Supervisory control.



# 1) Enterprise System Variable Definition : Super System Administrator Section

**Define Properties/Variables of < people, Enterprise Machines, Enterprise Processes, Created Products, Sold Products >**

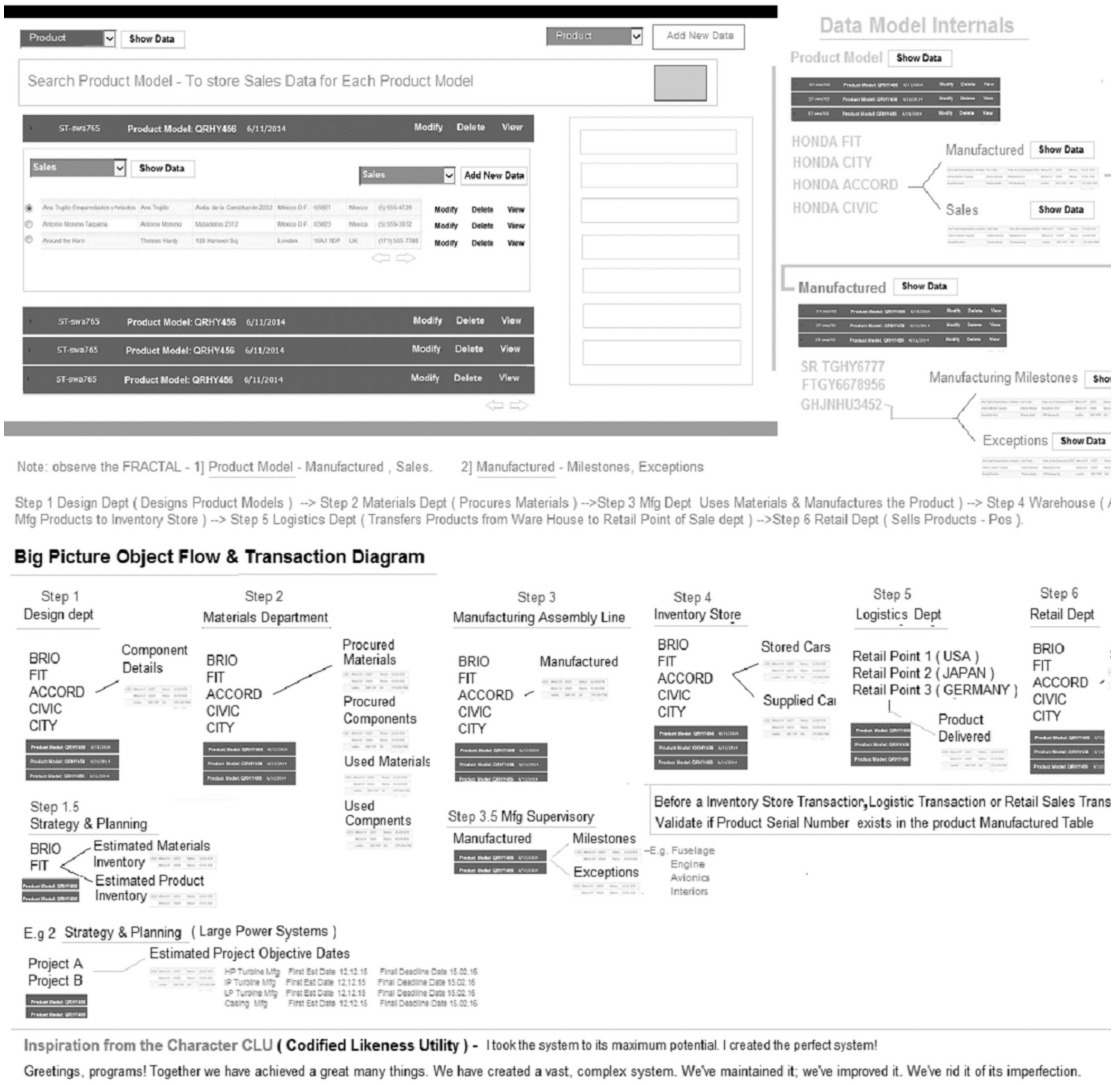
| People/Human Resource | Enterprise Machines | Enterprise Core Workflow Processes | Enterprise Non Core Processes( Transactions ) | End Product |
|---|---|---|---|---|
|  |  |  |  |  |
| **Properties Definition**<br>People Definition | **Properties Definition**<br>Enterprise Machines | **Properties Definition**<br>Enterprise Core Processes | **Properties Definition**<br>Non Core Processes | **Properties Definition**<br>End Product |

**E.g Process Sales**

Sales Process Properties:
1) Invoice Number
2) Invoice Date
3) Sales Item Name
4) Sales Item Quantity
5) Sales Item Price
6) Retail Outlet Name
7) Taxation Number
.....

## 2) Auto Generated Data Input Screens, for Feeding Data

**Feed in data through Forms/Screens defined by the Super System Administrator.**
**These Screens are System Auto Generated Screens generated from System Variables and properties defined by the Enterprise Super Administrators.**




## A c# .net Example to demonstrate - Autogeneration of Screens

Code Demonstrating - Complete Automated Form/Screen Generation & Data Storage - from a Data Definition Class

Do You Ever Imagine - Building a Complete ERP SYSTEM or a Control System or a Piece of Enterprise software in 20-30 Days ?

**3 simple Steps**

---

**a) Data Definition in the Data Definition Class / Definition of the Enterprise PROCESS Properties E.g Sales Invoicing**
**b) Auto Generate the Forms/Screens**
**c) store the Screen Data in a datastore**

**Code Part 1**

**App_Code/DataDefinition.cs**

```
public class ProcessProperty
{
    public string String1;
    public List<string> SubList;
    public ProcessProperty(string String1, List<string> Li)
    {
        this.String1 = String1;

        this.SubList = Li;

    }
}

namespace DataDefinitionnamespace
{
public class DataDefProcessA
{
    public List<ProcessProperty> DataDef( string id )
    {
        //Data Definition
        List<string> CashflowDirection = new List<string> { "Expenditure", "Income" };
        List<string> CostsRelatedTo = new List<string> { "People", "Process Enablers", "Process Material
        List<string> Domain = new List<string> { "Engg", "Aviation", "Automotive", "Energy", "Oil & Gas"
        List<string> Department = new List<string> { "Execution/Mfg", "Design/Prototype", "Maintenance &
        List<string> Currency = new List<string> { "Euro", "Dollar", "Yen", "Yuan", "Sterling" };
        List<string> Taxation = new List<string> { "5%", "10%", "15%", "20%", "25%" };


        List<ProcessProperty> ProcessX = new List<ProcessProperty>();
        ProcessX.Add(new ProcessProperty("Cashflow Direction", CashflowDirection));
        ProcessX.Add(new ProcessProperty("Costs Related to", Co(local variable) List<string> CashflowDirection
        ProcessX.Add(new ProcessProperty("Domain", Domain));
        ProcessX.Add(new ProcessProperty("Department", Department));
        ProcessX.Add(new ProcessProperty("SubProcess", null));
        ProcessX.Add(new ProcessProperty("Amount", null));
        ProcessX.Add(new ProcessProperty("Currency", Currency));
        ProcessX.Add(new ProcessProperty("Taxation", Taxation));
        ProcessX.Add(new ProcessProperty("Transaction Date", null));


        return ProcessX;
    }


   public List<ProcessProperty> DataDefDimensionSales(string Id)
   {
       //Data Definition
       List<string> ProductModel = new List<string> { Id.ToString() };
       List<string> Sales = new List<string> { "Ecommerce", "Retail" };
       List<string> Currency = new List<string> { "Euro", "Dollar", "Yen", "Yuan", "Sterling" };
       List<string> Taxation = new List<string> { "5%", "10%", "15%", "20%", "25%" };

       List<ProcessProperty> ProcessX = new List<ProcessProperty>();
       ProcessX.Add(new ProcessProperty("Product Model", ProductModel));
       ProcessX.Add(new ProcessProperty("Sales format", Sales));
       ProcessX.Add(new ProcessProperty("Serial Number", null));
       ProcessX.Add(new ProcessProperty("Sale price", null));
       ProcessX.Add(new ProcessProperty("Sale Currency", Currency));
       ProcessX.Add(new ProcessProperty("Sales Tax", Taxation));


       return ProcessX;

   }
       public List<ProcessProperty> DataDefDimensionExceptions(string Id)
       {
           //Data Definition
           List<string> ExceptionID = new List<string> { Id.ToString() };


           List<ProcessProperty> ProcessX = new List<ProcessProperty>();
           ProcessX.Add(new ProcessProperty("Exception ID", ExceptionID));
           ProcessX.Add(new ProcessProperty("Exception Name", null));
           ProcessX.Add(new ProcessProperty("Exception Date", null));
           ProcessX.Add(new ProcessProperty("Exception Description", null));

           return ProcessX;

       }
```

```
public class DataDefProcessB
{
    public List<ProcessProperty> DataDef( string z)
    {
        //Data Definition
        List<string> CashflowDirection = new List<string> { "Expenditure", "Income" };
        List<string> CostsRelatedTo = new List<string> { "People", "Process Enablers", "Process Material
        List<string> Domain = new List<string> { "Engg", "Aviation", "Automotive", "Energy", "Oil & Gas"
        List<string> Department = new List<string> { "Execution/Mfg", "Design/Prototype", "Maintenance &
        List<string> Currency = new List<string> { "Euro", "Dollar", "Yen", "Yuan", "Sterling" };
        List<string> Taxation = new List<string> { "50%", "60%", "70%", "80%", "90%" };

        List<ProcessProperty> ProcessX = new List<ProcessProperty>();
        ProcessX.Add(new ProcessProperty("Cashflow Direction", CashflowDirection));
        ProcessX.Add(new ProcessProperty("Costs Related to", CostsRelatedTo));
        ProcessX.Add(new ProcessProperty("Domain", Domain));
        ProcessX.Add(new ProcessProperty("Department", Department));
        ProcessX.Add(new ProcessProperty("SubProcess", null));
        ProcessX.Add(new ProcessProperty("Amount", null));
        ProcessX.Add(new ProcessProperty("Currency", Currency));
        ProcessX.Add(new ProcessProperty("Taxation", Taxation));

        return ProcessX;

    }
```

**Process C Code is repeat or Clone of  Process B Code**

## Code Part 2

---

Process.aspx.cs

```
public partial class Process : System.Web.UI.Page
{

    protected override void OnPreInit(EventArgs e)
    {
        base.OnPreInit(e);

        List<ProcessProperty> processYY =null;
        placeholder.Controls.Clear();
        string v = Request.QueryString["Process"];
        string x = Request.QueryString["SubDimension"];
        string z = Request.QueryString["ID"].ToString(); ;


        if (v == "processA" && x == "main")
        {
            DataDefProcessA DX = new DataDefProcessA();
            processYY = DX.DataDef(z);
        }
        else if (v == "processA" && x =="sales")
        {
            DataDefProcessA DX = new DataDefProcessA();
            processYY = DX.DataDefDimensionSales(z);
        }
        else if (v == "processA" && x == "Exception")
        {
            DataDefProcessA DX = new DataDefProcessA();
            processYY = DX.DataDefDimensionExceptions(z);
        }
        ////////////////////////////////////////////////////////////
```

```
//////////////////////////////////////////////////////////

if (v == "processB")
{
    DataDefProcessB DX = new DataDefProcessB();
    processYY = DX.DataDef(z);
}
if (v == "processC")
{
    DataDefProcessC DX = new DataDefProcessC();
    processYY = DX.DataDef(z);
}

Int32 j = 0;

foreach (var ProcessProperty in processYY)
{
    Label Labelx = new Label();
    Labelx.ID = "Label" + j.ToString();
    Labelx.Text = ProcessProperty.String1;
    placeholder.Controls.Add(Labelx);


    if (ProcessProperty.SubList == null)
    {
        TextBox tbox = new TextBox();
        tbox.ID = "TextBox" + j.ToString();
        placeholder.Controls.Add(tbox);
        placeholder.Visible = false;

    }
    else if
   (ProcessProperty.SubList != null)
   {
    if (ProcessProperty.SubList.Count >6)
    {
        DropDownList Drop = new DropDownList();
        Drop.ID = "Drop" + j.ToString();
        for (Int32 i = 0; i < ProcessProperty.SubList.Count; i++)
        {
            Drop.Items.Add(ProcessProperty.SubList[i]);
        }
        placeholder.Controls.Add(Drop);
    }

    else
    {
        RadioButtonList Radio = new RadioButtonList();
        Radio.ID = "Radio" + j.ToString();
        Radio.RepeatDirection = RepeatDirection.Horizontal;
        for (Int32 i = 0; i < ProcessProperty.SubList.Count; i++)
        {

            Radio.Items.Add(ProcessProperty.SubList[i]);
        }
        placeholder.Controls.Add(Radio);
```

## Code Part 3

Process.aspx.cs

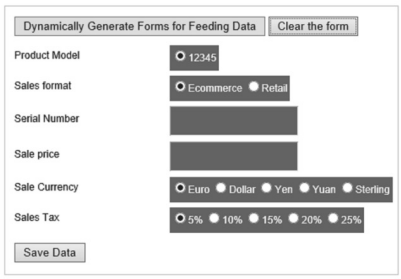
```
protected void Button1_Click(object sender, EventArgs e)
{
    Int32 i = 0;
    Int32 k = placeholder.Controls.Count;

    for ( i = 0; i < k; i++)
    {
        Label l = ((Label)placeholder.FindControl("Label" + i.ToString()));
        TextBox Y = ((TextBox)placeholder.FindControl("TextBox" + i.ToString()));
        DropDownList Q = ((DropDownList)placeholder.FindControl("Drop" + i.ToString()));
        RadioButtonList Z = ((RadioButtonList)placeholder.FindControl("Radio" + i.ToString()));

        placeholder.Controls.Add(new LiteralControl("<tr>"));
        placeholder.Controls.Add(new LiteralControl("<td>"));
        if (l != null)
        {
            l.Width = 180; l.Height = 30;
            placeholder.Controls.Add(l);
        }

        placeholder.Controls.Add(new LiteralControl("</td>"));
        placeholder.Controls.Add(new LiteralControl("<td align =left;>"));

        if (Y != null)
        {
            Y.Height = 30; Y.BackColor = System.Drawing.Color.FromName("#576484");
            Y.ForeColor = System.Drawing.Color.White;

            placeholder.Controls.Add(Y);
        }


        if (Q != null)
        {
            Q.Height = 30; Q.BackColor = System.Drawing.Color.FromName("#576484");
            Q.ForeColor = System.Drawing.Color.White;
            Q.SelectedIndex = 0;
            placeholder.Controls.Add(Q);
        }

        if (Z != null)
        {
            Z.Height = 30; Z.BackColor = System.Drawing.Color.FromName("#576484");
            Z.ForeColor = System.Drawing.Color.White;
            Z.SelectedIndex = 0;
            Z.RepeatDirection = RepeatDirection.Horizontal;
            placeholder.Controls.Add(Z);

        }

        placeholder.Controls.Add(new LiteralControl("</td>"));
        placeholder.Controls.Add(new LiteralControl("<tr>"));

    }
    placeholder.Visible = true;
    Labelxx.Text = "";
    Page.SetFocus(Button2);
    Button3.Enabled = true;
```

## Auto Generated Screens

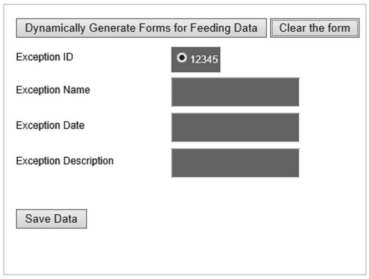
### Process A - CashFlow System


### Process B &C - CashFlow System


### Process A - Sales




### Process A - Exceptions

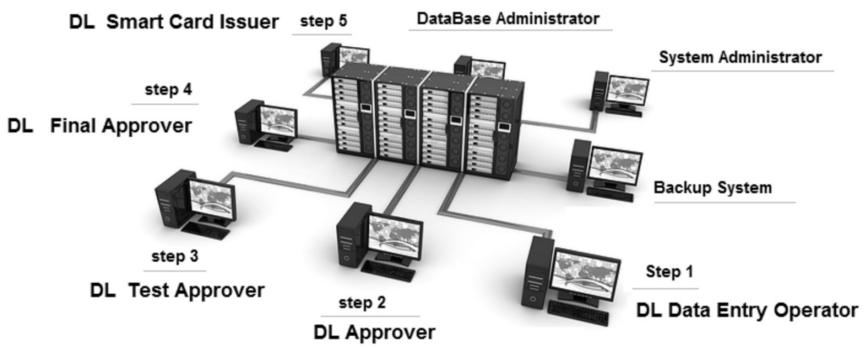


## 3) Enterprise Views for Monitoring & Control of People, Machines ,Processes, Products.

**View & Monitor < people, Enterprise Machines, Enterprise Processes, Product details >**

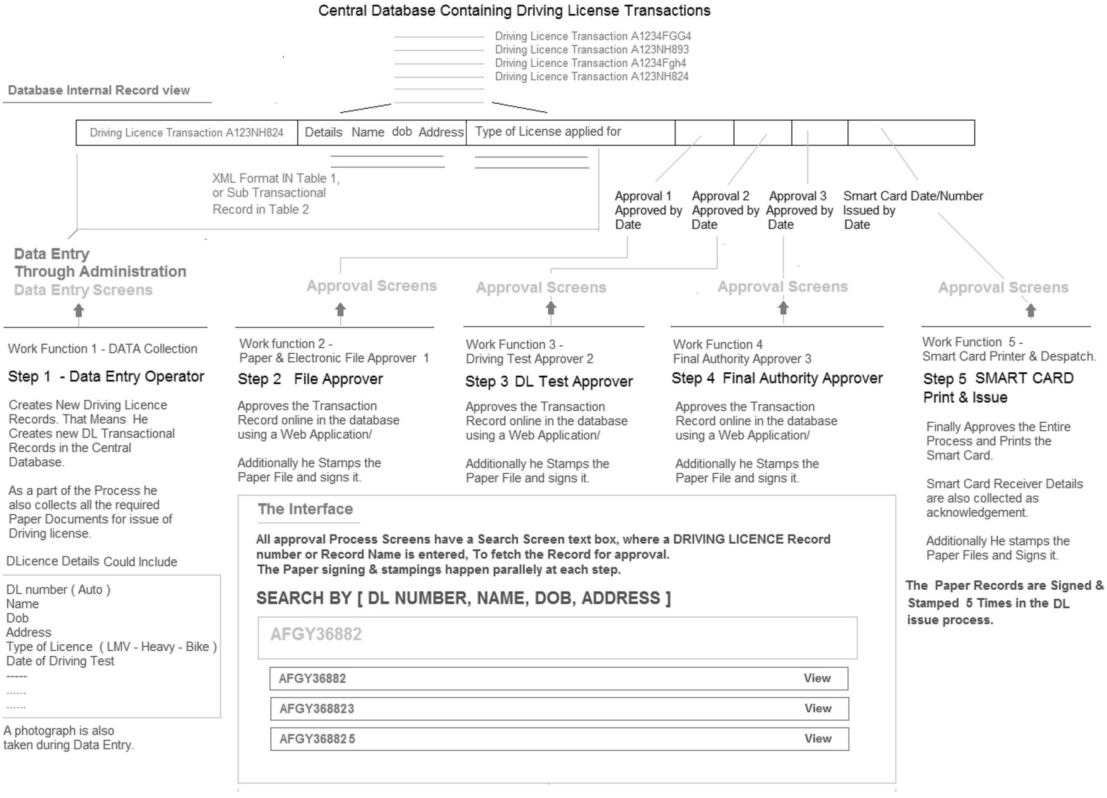
# Software Automation - Driving Licence Issue Process
## Using Transactional Systems Architecture.

### Hardware Architecture Picture




### The Software Architecture using - TRANSACTIONAL SYSTEMS


**Each Record in the database is called a Transcational Record. The Transactional record contains every detail, Variables of any Enterprise Process. Some times the Transactional records may be split into 2 Transactional Records [ Part 1 Containing Basic details and Variables & Part 2 Containing all the detailed variables ]**



## Chapter 5

# Finance by Structure & Design

# 1) Community Mutual Funds

Stock are instruments to hold partial ownership of stocks , transacted through Stock Markets. Mathematics proves that stock market trading through Stock market trading is a Zero sum problem ( GAME theory evaluation & Mathematical Induction techniques from a single player game to an n Player game. ). This implies that when one player makes money through stock gains, another loses money. Successive generation of stock owners increase liquidity in Mutual funds and stock exchanges to raise value of the stock. Stock which do not have a dividend component are not liable to pay out any cash to stock owners.

An Ideal condition requires cash flows from stock owners to Enterprises and vice versa. If enterprises do not pay back to the stock owners the cash flows from the enterprise to the public stock owners is nil and the stock ownership is notional. Additionally Money in stock markets is Floating Money ( Money Floats between the stock owners holding the public float ) and never reaches the Enterprise that provides the stock ( Stock Money reaches Enterprises only during an IPO, Bond issue or Loan ).

Stocks that are traded on stock exchanges exhibit a stock value that is created by a minor amount of liquidity ( Volume ) , which is a small subset of the total number of shares held by stakeholders. This stock Value representation ( created by a small section of traders ) may be grossly inaccurate in representation of the actual value of the stock. , which actually consists of a large number of subset of owners involving ( Traders, FII's, Mutual Funds, Institutional Investors, Hni's and small stock owners ).

The stock market problem can be solved by fixing the range of Future stock dividends during an IPO process or through an AGM ( meeting ) post IPO. E.g. Declaring 15% to 40% dividend equivalent of "the Net Profit earned yearly". The Stock market problem can additionaly be solved by a Buy Back note issued by enterprise quoting the value of the stock and the Quantity of stock that can be bought back by the enterprise. These two solutions answer the simple fundamental questions - what is the pay back amount or cost of capital raised from the stock market IPO ?

## Calculating the Cost of Capital

Why do enterprises go to the stock market instead of taking a bank loan?( It's an instrument designed to offer risk free capital ( money ) without any payback( dividend ) commitments. Stex Suggests that a payback ( dividend ) value in the range of 15% to 40% of the "Net Profit earned yearly", could balance the Risk free nature of the instrument, and The rest of the 60% "net profits earned yearly", could be used for growth of the Enterprise.)

Calculating the Value of the Stock Price

An A+ Technique can be used to arrive at value's of the stock price by including Stock Value calculation parameters like

- Premium value created by the services/Product ( Good to Have services ).
- "Essentiallity" of the services offered.( A service/Industry Solution that cannot be done without )
- Attractive or Innovative Revenue Models
- Other "Present Continuous, Future Continuous & Past participle" - Financial Parameters .
- Value of Industrial Machine Assets, Factories, Buildings etc.
- Potential Scale
- Cumulative Net Earnings/Net losses ( Net volume of money storage ) over a period of time.
- Net Money Borrowed during an IPO.
- Net Debt
- Order Books

This facilitates balanced sharing of business risks and business rewards between all stakeholders of the stock. e.g [ Bank, Insurance, machinery tools companies , Individuals ] and each of them have their mutual interests protected . The stock market Financial problem can be solved by a more systematic and Risk free Instrument called the Insured loan product( described in section #15 ) & community mutual Fund ( described in section #8 below )

Stablilizing the Stock Market by mitigating market risks of stocks.

Stocks are highly Liquid Assets, which implies stock = Cash.

if STOCK Equal to Cash, Stocks could earn Fixed deposit styled interest and Saving account styled interest from Banks, Industrial Enterprise and Financial services Industry. This stabilizes the stock price and opens three modes for earning income. This process design has advantages over the floating money principle of the stock market, where stock money, never reaches The Industrial Enterprise's account books.

- Dividends from stocks
- Interest yield on stocks by Banks or Industrial Enterprises. ( e.g a Stock price at 1500$ could earn "Interest yield" on a stock, at a value of 1200$ ).
- Price Escalation of stocks by Market Forces.
- A Stable Financial Instrument Implies Higher Participation in the Stock Market and Inturn Higher Liquidity and stability of Stock Prices.

## How does the Community Mutual Fund - "Patented" by STEX Work?

### There are Primarily 4 Participating entities

**Corporations like**
IBM
SIEMENS
Mitsubishi

**Community Mutual Fund Manager**
E.g. Deutsche Bank who may have exclusive license from STEX .

**[ Public at Large ] Community of Individuals**
People who might want to Invest in Projects.

**Governments**
State Level Governments and National/ Federal Governments

**Corporation and Fund Managers go to the Public proposing an Utility type of Project.**

Propose a Community Mutual Fund Solution for an Utility Project , to a National Level Community , State Level Community , City Level Community.

Declare Proposal Imperatives like :
Total Investment Size e.g. 1 billion Dollar
Ownership Ratio : e.g. 50% : 50%
Profit Sharing Ratio: Community: 20% : Industry 60% Government: 20%

Measure the Response from the Communities and Government and study Feasibility.

**A clear Response from the Public will mandate the Governments to support the projects that the public demands.**

**The Govt would also clear regulatory hurdles at State Level and National Level.**

Is Response Positive or Negative

**Negative** → End. Terminate Proposal

**Positive and Feasilble**

Issue Community Mutual Funds and setup the a Private Corporation

**Community Mutual Fund - Contract Format**

Ownership Ratio: Corporation 50% Public Public 35% Government:15%
Profit sharing Commitment : 15% of yearly Profits distributed among Public and Govt.
**Investment Risk and Investment Rewards are shared by all entities ( Corporation, Public and Govt ). Investments are also Protected - such that in case of Project failure All Investments will be returned in as is where is Condition.**

Corporations Conduct Operations Industrial Operations Directly or Indirectly. (by Subcontracting)

Evaluate yearly profits and Revenues. Share with Community

**Yes**

Are Operations profitable

**No** → Are Operations sustainable in the Future ?

**Yes** → Distribute Wealth in Pre Agreed Ratio as per CMF Agreements.

**No** → Dissolve Company. Return Money as per Ownership.

---

**e.g. Airport Infrastructure**

**If this Airport Infrastructure was built in Berlin and was a profitable operation due to large tourist activity - The Tech Corporation/Govt and People could gain together**

| Ownership % | | Yearly Profits sharing % | |
|---|---|---|---|
| Technology Corporation | 30% | Technology Corporation | 90% |
| Public | 70% | Public | 10% |
| National Govt | | National Govt | |
| State Govt | | State Govt | |

- **Corporations get Work Contracts + Take total responsibility + a Large share in profits**
- **Public gets - Modern Infrastructure + a commited share in profits + Employment Benefits in the Projects**
- **Governments - Clear Regulatory Hurdles + gain a committed share in profits.**
- **Projects are created at International Level, National Level, State level, City Level.**

**e.g. of an**

**International Project : STEX Network.**
**National Project : National Highways , Rail Networks ,School Education ,Tourism Development**
**State Project: State Highways, Drinking water projects ,College, Polytechnic Vocational Education**
**City level projects: Digital Infrastructure , Road Transport**

**Typical Types of Projects -** Airports, Highways, Tourist Destinations, Rail, Hydrogen Infrastructure, Digital Infrastructure, Agriculture, Healthcare

**Highlight**

- Corporations, People ( Investing public at Large ) and Governments have a Balanced and Mutual Ownership in the Projects and each one of them have their interests protected ( i.e each one of the entities stand to gain the rewards and share a fraction of the Risk. )
- Each Entity has it's Investments Protected.

**When a corporation signs a community Mutual Fund Contract - the Corporation is legally a Private Stock Corporation, which can be dissolved when the Corporation and it's Projects fail or are unsustainable.**

**The community Mutual Fund - creates a system where the Corporations, the Government and the Community of Individuals Share the Risk, the Reward and still have their Investments Protected.**

**The Community Mutual Fund Mode of Operation is unlike the IPO Process where the Corporation/Enterprise raising Money is not obligated to Payback the acquired funds, incase of an Enterprise or Project Failure.**

## 2) Insured Loans Instruments - Finance Loans with Integrated Insurance :

Stex has invented a new Financial Technique/Tool/Instrument called Insured Loans, to facilitate Colateral free Technology Projects/Machinery tools projects and other projects . ( using tensor maths and freescales ) . It is an Integrated Financial Loan endorsed with an Integrated Risk Policy ( Insurance endorsement to cover loan repayment failure risks ).
**E.g. A loan of 25000$ Towards a Machinery project/ Tool project/ Technology project may consitute of**

1. 22500$ as a Principal loan amount to be disbursed to the loanee.
2. 2500$ as Insurance Risk to cover the Loan Risks, payable by the Insurer.
3. The Loan can be disbursed for Technology projects, machine works projects, machine tooling projects and many other projects.
4. The Loanee is obligated to return the following
    5. Project failure scenario/loan default scenario: Tools/Machinery ( Depreciated Assets - which can be refurbished and Resold by the machine manufacturer ) & a principal amount of 10000$ ( 10000$ or 9000$ or 8000$ or 7000$ Freescale maths ) out of the 22500$ ( Loan amount) , on Project/Busines failure
    6. Project success scenario/ Non loan default scenario - ( 25000$ Principal + Interest )
7. Used Machines/Tools can be refurbished and marketed to People for projects.
8. The Loan amount can be as high as 500k$ to 1 Million $.
9. These loans are disbursed by Financial institutions in assosciation with machine tool manufacturers or by machine tool manufactureres as the sole authority. The Loans amounts depend on Type of Projects and Project machinery required. E.g. The Loan amount may be fixed for a Specific Market segment of Engineering tools - 25k$ for a Wood cutting machine type B.
10. All stake holder mutual interests( Tensors - Vectors forces & Relationships ) are balanced between the loanee, Insurance company, Technology machine maker, Individuals.

## 3): Single world Single International Currency ( IC14 ), A Futuristic New Paradigm.

This paradigm provides a INNOVATIVE new way, to use a Single International Currency IC14 for the Entire globe ( consisting of 100+ countries ), specifically for Trade & commerce.

IC14 is a digital Currency which implies that, The Local currency like Dollars, Euro, Yen, Yuan or Rupee can be exchanged or Converted/Excahnged to an International IC14 Currency through a Bank.

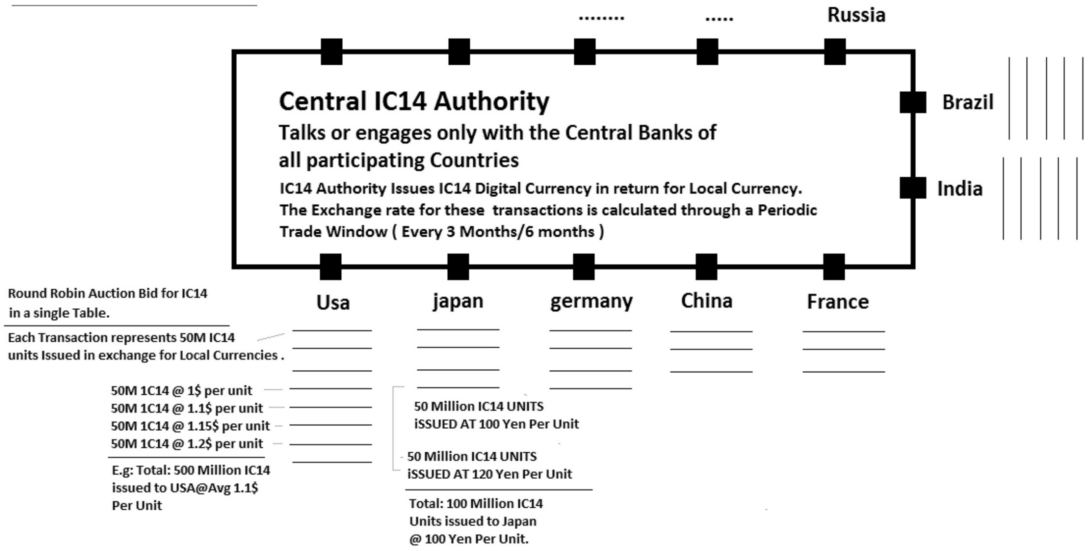
The IC14 international Currency can also be reconverted back to your Local Currency when there is a Need. The Actual Exchange rate for the Conversion of Local Currency to IC14 or IC14 to Local currency is fixed every three Months, by Market trade methodology, where Dollar, euros, rupees could be traded/exchanged, through demand supply equations, through a Single Window.

For E.g 1 Dollar = 1 IC14 Currency. 1 Euro = 1.30 IC14 Currency. 1 IC14 = 65 Indian Rupees. Practically/Theoretically - IC14 value is equal to a [ Average of 100 + Local currencies ].

At Present,( upto the year 2015 ), Different nations maintain Foreign Exchange in Either Dollars, Euros, Yen or Sterling as foreign exchange reserves. If IC14 were to be implemented Countries could maintain their Foreign Exchange Earnings in a non Devaluable Single Digital Currency - IC14.
So could All Individuals, choosing to convert their Local savings to IC14.

The Value of Money used to be measured through the Gold Standard. The IC14 System or Paradigm changes the Standard to what is called the IC14 Standard. The IC14 standard is basically an EXCHANGE Rate of International Currencies against the IC14 Currency. For Example 1 IC14 currency UNIT = 65 Indian Rupees, 1 IC14 currency UNIT = 1 Dollar, 1 IC 14 currency UNIT = 0.8 Euros.
The IC14 Central International Authority maintains this standard reference for all IC14 Financial Exchanges.

## HOW IC14 Works




**Periodic Trade Window - Round Robin Auctioning of IC14 at a Single Table [ Repeated every 3 months or 6 months ]**

**THE issue of IC14 IS conducted through a Periodic Trade window of Round Robin Auctions where Each participating Countries Bid For Issue of IC14 at a Price. For E.g USA Out Bids other Countries to Get 10 Transactions of 50 Million IC14 Units at an Average price of 1$ Per Unit.**
**This is how the Exahnge Rates are fixed. Different Countries Bid different Local Denominations currencies against the IC14 TO SET THEIR EXCHANGE RATES.**
**The Total Threshold for an Exchange Rate Standard to be Fixed is decided and made clear Before the Auction. E.g: 4 Billlion IC14 Units.**
**This ensures that the Exchange rate is Fixed through a Thick Volume Transaction.**

**While Bidding for the IC14 - bidders keep in mind the folowing Factors**

**1) Buying power of their Local Currencies ( $, Yen, Euro, Yuan, Rupee )**
**2) Power of IC14 as an Local & International Trade Currency.**
**3) Non Devaluable nature of thte currency**
**4) A system for Accurate Exchange Rates.**

**IC14 Control & Thresholds**

**TheVolume of IC14 Units issued in total is restricted and controled. The Volume is calculated using a factors like Population * Per capita savingsof Individuals * 70% PER INDIVIDUAL. OR Number of Business Firms * Net Value Additions/ * 70% Per Firm.**

**After the Excahange Rates are Fixed for a 3 or 6 Month Period - The Central Banks oF DIfferent Countries may Go to the IC14 Authority to Get more of their Local Currencies Converted to IC14. The Flow is as Follows . People Purchase IC14 from Banks. Banks Purchase IC14 from Central Banks. Central Banks Purchase IC14 from IC14 Central Authority.**

**RISK OF FRAUD.**

**Three Copies of IC14 Conversions are issued by Banks. One Copy is retained by the Customer. The Second Copy is sent to the IC14 authority. The 3rd Copy is sent to the Central Bank. This helps the IC14 authority to prevent Fraud in IC14 Currency.**
**The Issued IC14 at Bank level, Central Bank Level and at the IC14 Authority level are matched for Account Discrepancies or control of fraud.**
**Every transaction of IC14 Conversion by Individual or Firms is reported by a Triple Database Transaction [ Reporting to IC14 Authority/ Bank/ Central Bank ]**

a) An Individual or a firm choose to convert their savings of from $/Yen/Euro/Rupee Accounts to IC14. This is faciltated through their Bank - E.g Deutsche Bank America.

b) Deutsche Bank America issues IC14 in lieu of Dollars.[ E.g 1 Billion IC14 = 1 Billion $ ].

c) Fed/National Central USA Bank Collects the Dollars from Deutsche bank and Issues Deutsche Bank America 1 Billion IC14 currency.

d) Fed/Central Bank USA Passes on the 1 Billion Dollars to the IC14 Central International Authority accounts and Collects a Stamped Certificate in return - stating that IC14 Central International Authority Has Collected 1 Billion$ from the Fed/Central Bank at the Exchange Rate - IC14 Standard and has Issued 1 Billion IC14 through the FED/Central Bank.
( Dual Stamp or Twin Stamping - By Fed/Central Bank USA & 1C14 Authority )

e) IC14 Central International Authority is designed to collect Money or Currency in 100+ National Currencies.

f) When an Individual in the USA or A Firm in the USA, Chooses to Re Convert 1 Billion IC14 currency a Blossoming Currency like YEN, they Request their Private Bank for yen savings.

g) The private Bank Goes to the FED/CENTRAL bank USA for Yens. The Fed/Central Bank USA goes to the IC14 Authority and asks for 1 Billion IC14 worth Yens.

h) IC14 Authority may Issue Yens from it's Account reserve to the Fed/Central Bank. If it does not have the reserve, it asks the Japan Central bank for Yens and Issues 1C14 Currency to Japan central bank in return.

i) If the Demand for Dollar were to Dip and the Demand for Yen were to Go High - The Exchange Rates for the IC14 Standard are modified by the IC14 Authority to Give More Power to the Yen. E.G. 1IC 14 = 120 Yen may be revised to 1 IC14 = 90 Yen in the IC14 standard charts.

j) If there were to be a significant Downturn in the Dollar or a significant lower Demand for the dollar, The IC14 Authority may revise the dollar Exchange rate to the IC14.
E.g 1 IC14 2US $. However Before revision of the Dollar, the IC14 Authority may approach the fed to check for a Asking rate of the IC14 against the Dollar. If the Fed/CentralBank US aggrees to take back the dollar at IC14= 5$, then then IC14 exchange rate, also follows to revise the exchange rate in IC14 Standard Reference. This activity of Dollar Buy Back is done through a Periodic (3 Monthly ) Trade Window to fix the IC14 Standard reference.

k) In a Hypothetical Situation of Dollar going down and Yen also going down ( Devaluation of Buying Power ) - The IC14 Authority Reevaluates the IC14 vs US$ and the IC14 against the Yen. The IC14 maintains it's strength by allowing convertibility to other basket Currencies,until dollar or Yen Demand Picks up ( This ia s Gloom scenario ). If every Currency in the basket has a low demand or Buying Power, then in that case IC14 has a relatively lower Strength or Value. ( This is a Doom Scenario ).

1. The IC14 Systems has two approaches to CALCULATION of Exchange Rates.

a) The Participating Countries are Gathered together to a Single Window for exchanging /Trading Currency, every 3 Months or 6 months. ( Thick Volume Transaction Through a single Periodic Window ) and the central IC14 Authority fixes the Resultant Exchange rates as the IC14 Standard.

b) The Central International IC14 Authority Fixes the Exchange Rate ( using Last 3/6 Month Daily Moving Average ) and this is called the IC14 Standard. E.g. 1 IC14 = 65 Indian Rupees, 1 IC14 = 1 Dollar, 1 IC 14 = 0.8 Euros.

2. An example of a commercial Transaction is as follows. USA builds F-35 Lightning Aircrafts in the USA, at a cost 100 Milion$ Per UNIT. If India were to purchase the unit a Quoted price of 120 Million$. India could need to Convert X amount of Indian Rupees to 120 Million IC14 Currency and send it to a USA IC14 account.

3. All Banks Acounts may be equipped to a add a single new currency column to their Databases called IC14 Currency, with a functionality allowing convertion of Local Currency to IC14 and vice

versa.

4. Since IC14 is a Trade currency, all International & Local Commercial activites may be conducted in IC14 =( X Local Currency ) format. International Contracts can be agreed upon through IC14, for example an Engg Work Contract in China, could be fulfilled by an Indian Company at a IC14 Value of 1 Million IC14 units. Bank Accounts could maintain Local / International transactions in their Account books as per the classification of Transaction.

5. IC14 is non devaluable and, people and enterprises, may store their assets in IC14 with no currency risks and further could use IC14 as a Trade currency.( Note Central banks buy back local currency to allow conversions of local currency like dollar to IC14 .)

E.g. if China were to maintain a Foreign Account : 1 Trillion $, and the Dollar is Devalued by America, due to Difficult Economic conditions, The Foreign Exchange held by China would also loses the Buying Power or strength. To Prevent such a scenario, CHINA could maintain Foreign Exchange in IC14 Currency accounts, which in reality can be exchanged to Yen, or the Sterling or Euro, if there were to be a need.

6. Excess money printing by any country does not devalue the IC14 currency and further does not affect Other countries economics. Most Currency Economic maneuvers made by countries are kept blind or are hidden from other countries.

7. Expensive Derivatives based Options hedging can be avoided to mitigate currency fluctuation risks.

Note: Currency fluctuation happen between each economic currency in permutations ie -
eg. dollar rupee, rupee dollar, dollar yen, yen dollar, dollar euro, euro dollar.

Therefore Currency Fluctuations between a large Number of Permutations ( 100+ countries * 100+ countries/2 ) can be avoided.e.g. US trades with china ( dollar yuan ), , US trades with india ( dollar rupee ), us trades with Australia ( dollar Aus dollar ). Yuan Trades with Rupee, Yuan Trades with US Dollar and with the Australian Dollar.

8. Note: Currency markets are controlled only a few Entities/Traders are do not actually reflect the real exchange rates. Moreover the trade occurs in Very Thin Volumes, whichis not enough to gauge the Right Exchange rate RATIO. Additionally Exchange rate valuation are evaluated, at a country level - at the FED/Central banks level, where their capacity to buy back dollars in lieu of IC14, is a major factor in fixing of Exchange rate ratios.

9. Illegal Currency CASH Flows between entities in International trade can be traced easily.

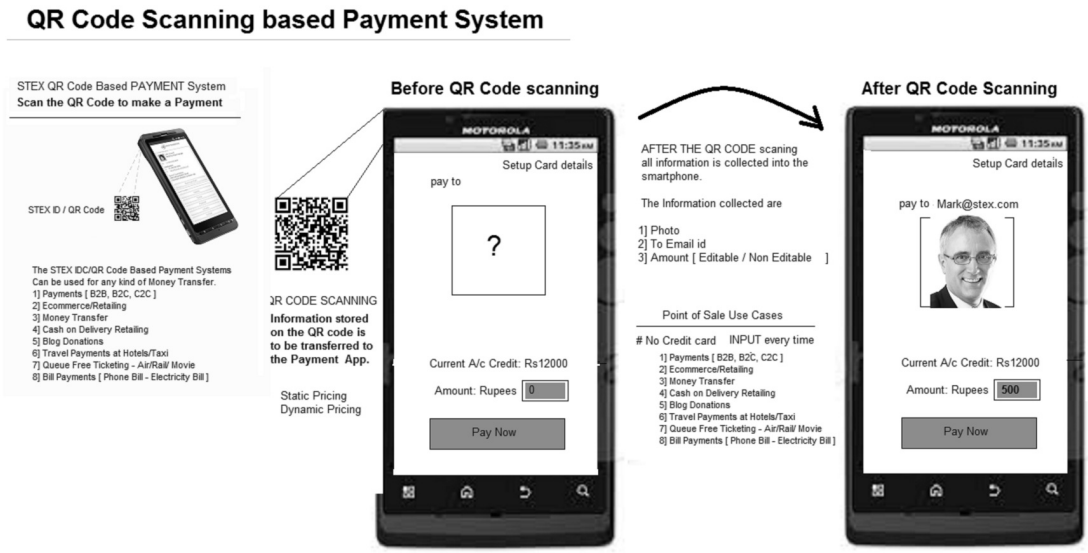
10. No Printing of Cash/Notes is required to use this currency.

# 4) QR Payment technology/QR Code Based Payment System

A novel new payment method that facilitates payments on the web and real world, through QR Codes. It works by a smart phone scanning a QR Code and facilitating a "High Speed Secured Financial Transaction". The QR Code that is scanned could be a QR Code that resides on a:

- Computer/web screen
- News Paper Ad
- A Business card
- A Printed Paper Bill
- An E-commerce Cart

The following is a QR Code-based payment system diagram - as it appears on a smartphone app. The QR Code contains all vital elements necessary for a financial transaction..




**Highlights and Features of QR Payment**

1: No filling web forms with credit card details to make payments on websites.
2: can be used to transfer cash on delivery.
3: The system introduces the concept of the one world integrated single "International currency", where people convert their local currency to an International currency to carry out trade and reconvert it back to local currencies when required.
4: Currency Risk are Mitigated by a Non Fluctuating Currency system.
5: Credit card risks are reduced from Billion of dollars to near Zero.
6: payments on the web involve 3- 4 Hops on servers. This system reduces payment system hops to 1 server Transaction.

# 5) Insurance TAX Project [ Tax reforms ].

**Out of a Total 35% Taxes collected -** 20% Workforce/Industry Taxes could go towards Insurance TAX & 15% Workforce/Business tax could go towards govt taxes, For Defence,Infrastructure.

Insurance TAX is collected in the following categories. ( Advanced Taxation Filing Format )

- LIVLIHOOD INSURANCE - X%/100%
- RETIREMENT LUMPSUM INSURANCE - X%/100% - E.g. 20%/100% Allocated to Retirement
- PROJECT INSURANCE - X%/100%
- INSURED LOANS INSURANCE - X%/100% - E.g. 20%/100% Allocated to Loan Insurance
- EDUCATION INSURANCE- X%/100% - E.g. 20%/100% Allocated to Education
- LIFE INSURANCE - X%/100%
- HEALTHCARE - X%/100%

Contributions By Individuals may factorized when Filing Insurance Tax. For Example, With a 5000$ Tax Return, an individual or Business could choose the % allocation for each Insurance sector. E.g. 1000$ tax Out of 5000$ tax Could be allocated to Retirement Lumpsum. Another 1000$ could be allocated to Education.

The Scheme works in such that all entities - Individuals, Enterprises and Universities and govts pay taxes towards insurance so as to benefit from Insurance Claims ( Incase they Incur Risk or loss of Livlihood, Retire, or Complete Educatation milestones, Loan repayment failure etc.). This allows Govts to collect money from Workforce and Industries for appropriate use.

The Higher the insurance an individual or enterprise pays, the higher is the proportional return in case of a claim, which is called a prorata Insurance. This instigates all Types of Entities to Contribute to Insurance Taxes, which gives them "Insurance protection" in the above categories.

Insurance Firms, further Insure The Workforce/Industry Insurance corpus, at RE Insurance firms. ( E.g Swiss Re ) to mitigate/Share risks.

Insurance firms may be PPP Styled ( A Partnership between Private and Public ). Reinsurer's could also mitigate or share risks by going to [ Re Re ] Insurer Which is a "Reinsurance firm" for Insurer's and Reinsurers.

A certain portion of The insurer's corpus and reinsurer's corpus (E.G. 2%-3%) goes to the [ RE RE Insurance companies ] which insures insurance & Re insurance companies. RE RE Companies could be a division within the Reinsurer like MUNICH RE/SWISS RE.

Note: Different countries Insurer's pay different premiums to reinsurer's based on their respective Risk Profiles. Reinsurance players collect data & insurance premium from Insurer's via insurance companies ( which collect Insurance Tax).

2 Insurance and Reinsurance - Philosophies exist from the corpus point of View -

- INSURANCE COMPANIES BUILD CORPUS AND DONT INVEST to grow the corpus
- INSURANCE COMPANIES INVEST IN PROJECTS to grow the corpus.

Insurance Tax is linked to your bank account and every time there is a claim bank acccounts are checked to ascertain livlihood loss/Retirement or to transfer Insurance claim amounts.

Chapter 6

# Industrial Logic & Industrial Design

# Introduction to Industrial Logic & Industrial Design

STEX BUREAU is an Advanced Enterprise Engineering lab and an Advanced Technology system Design Bureau. STEX stands for Scalable Techniques for Engineered applications using Advanced Design.

Industrial Logic & Industrial Design is no longer about beautification of Systems. It is about creating strategic Intelligent foundations, New Design Directions, New User Experiences (UX), to drive growth and change across industries.

There's an endless market for Industrial Design & Industrial Logic - Driving product Design/Innovation, Product User Interface Design Experience, Innovative & Scalable services and finally systems that need simplification.

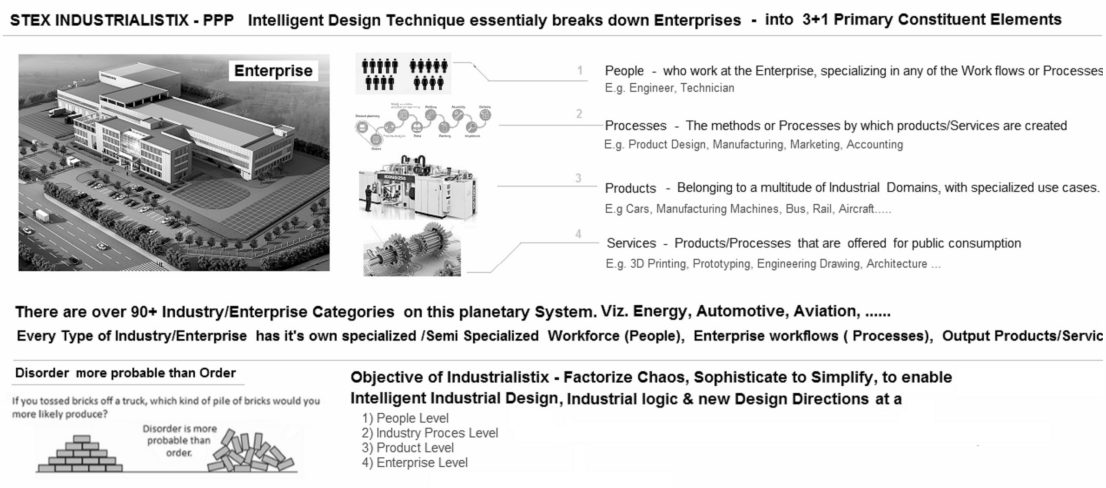
That is the reason, why Industrial design today is so influential and strategic. This is where STEX comes in. STEX Advanced Design Bureau helps you create, Brand new ( Unique ) Systematic Industrial Design & Industrial Logic Based, Product Systems & Product User Experiences (UX's).

# STEX's Research,Design & Development activities are conducted on "The Field", which Comprises of the Following

1. Visits, Observations & Studies of Industrial Factory Sites, Factory Processes and Factory Machines
2. Visits, Observations & Studies of Agricultural/Nursery and Cultivation Sites.
3. Experience in Technology Projects, in ICT Companies.
4. Experiments & Experience in Public Systems like Stock Exchanges & Financial Services.
5. Experiments and Experience in Public information Search Engines like Google.
6. Experiments and Experience in Educational Institutions for Higher Education ( Subject: Computer Science, Science & Commercial Aviation ).
7. Exposure to Informative Documentaries, through TV Channels like - DW (Deutsche Welle) TV, NHK Japan, National Geographic & Discovery.
8. Observations & Studies of Films/Movies like - E.g. Contact, Tron, Matrix, IRobot, StarTrek.....
9. Travel Abroad & Observations/Studies of, modern public systems in - Europe, Japan, Hong Kong, Middle East.
10. Fundamental Research, Breakthroughs and Documentation in the Form of Books like Intelligent Design, 97 Things Every Software Architect must Know.
11. Experiments and Experience in Game play ( Software Games ).
12. Exposure and Visits to Industrial Exhibitions.
13. Study of Complex/Semi Complex Software Systems, Industrial Ecosystems and Economics Systems.
14. Exposure to Technical Blogs on the Internet.

# INDUSTRIAL LOGIC ( Also called - Industrialistix )

## PPP- Intelligent Design Of Enterprises.




## Some of the Notable Industrial Domains on this Planet are listed below

Agriculture
Arts & Crafts
Advertising
Academic [ University ]
Acamedic [ Schools ]
Animal Dairy Farms
Animals/Poultry/Fisheries/Meat
Aviation
Automotive
Bikes Two Wheelers
Bio Technology
BPO-KPO
Books, Periodicals & Magazines
Banks
Boats & Trawlers
CropScience
Consumer Electronics
Computing Hardware[Enterprise]
Computing Hardware[Consumer]
Data center- Cloud
Chemical Industrial
Chemical Others[Agri & Others]
Defence and Military
Drugs and Pharmaceuticals
Engineering
Electrical
Electronics
Energy [Gas-BioMass]
Energy [Coal]
Energy [Water]
Energy [Hydrogen]

Energy [NUclear]
Energy [Wind]
Energy [Clean Tech/Solar]
Energy [Battery/Others]
Earth Moving
Environmental
Education & Learning
Foundry & Forge
Financial services
Financial Payment Gateways
Food Beverages
Fast Moving Consumer Goods
Fashion and jewellery
Fashion and Lifestyle
Furnishing and Interiors
Geography GIS Mapping
Government
Healthcare and medical
Hotels
Heavy Engineering
Insurance
Internet
Industrial Materials
Insurance and Reinsurance
Infrastructure and Construction[Public]
Infra/Construction[Enterprise/Factory]
Infra/Construction[Housing]
Leather and Synthetic Leather
Logistics and Transporation
Lighting & Energy Audit
Mobile & Smartphone
Metals
Mining
Music & Radio
Media & News
Movies & Films
Networking and Wireless
Oil and Gas
Alternative Fuels
Petro Chemicals
Plastics
Paper
Paints, Coatings and Inks, Dyes
Printing
Postal & Parcel
Police, Traffic & Security
Refrigeration, Air conditioning
Railway
Rubber and Tyre
Retail
Restaurants
Software
Intelligent Futuristic Systems
Complex-Exotic-Software
Game Development/Simulations-software
SemiConductor Processors,Chips & Chipsets
Sensors & Measurement Systems
Stock Market Broking Trading
Shipping & Marine
Satellite and Space
Science
Stationery
Sports
Telecommunication
Television, Radio & Entertainment
Trucks Buses
Textiles
Travel & Tourism

Weapon systems
Washing machines & Dish Washing

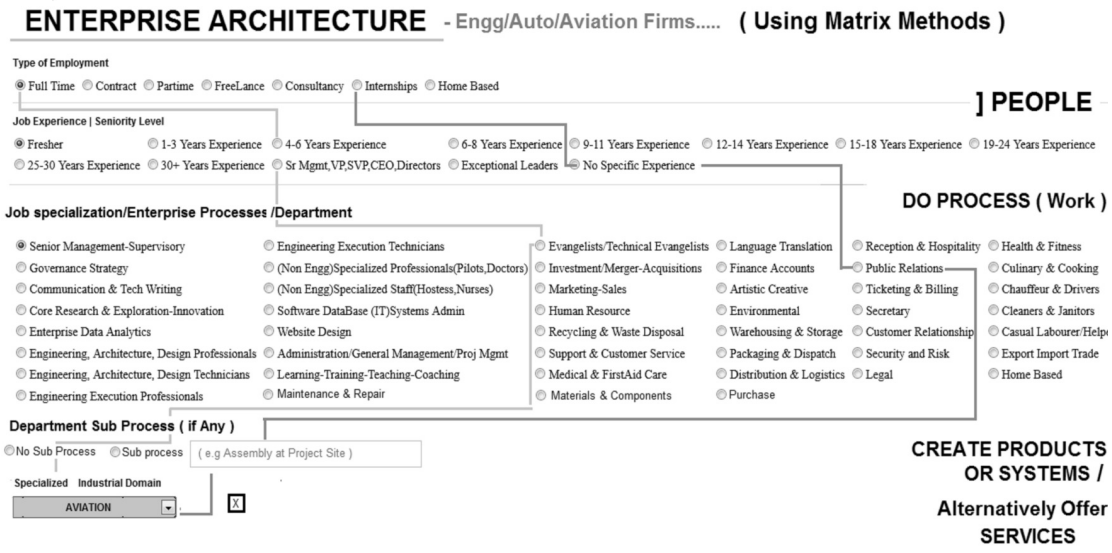
# MATRIX METHODS

Matrix Methods are a Technique by which An Entire Complex system can be Defined USING - Structured Objective Variables. This approch involves joining the Objective Variables in a sequence or in any Permutation/combinations to create a Large, Objective, View or Picture of the System.

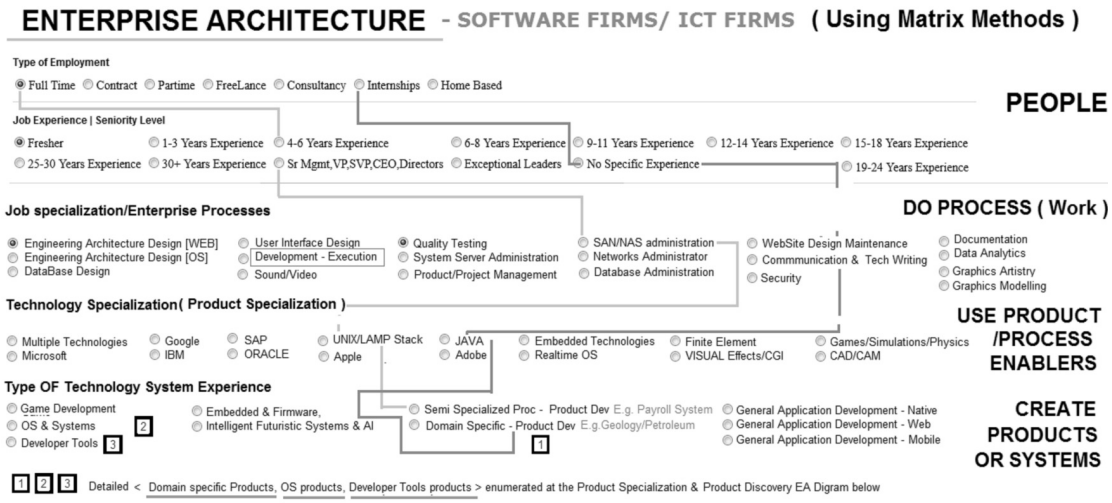
Another important Sub Technique under Matrix Methods is - Reduction of the Total Number of Variables that define a Complex System. This is Called Reduced Variable Thinking/ Reduced Instruction Set Thinking. This is, analogically Similar to the RISC technologu in Computer processors called Reduced Instruction Set Computing.

# Organizational Structuring & Architecture

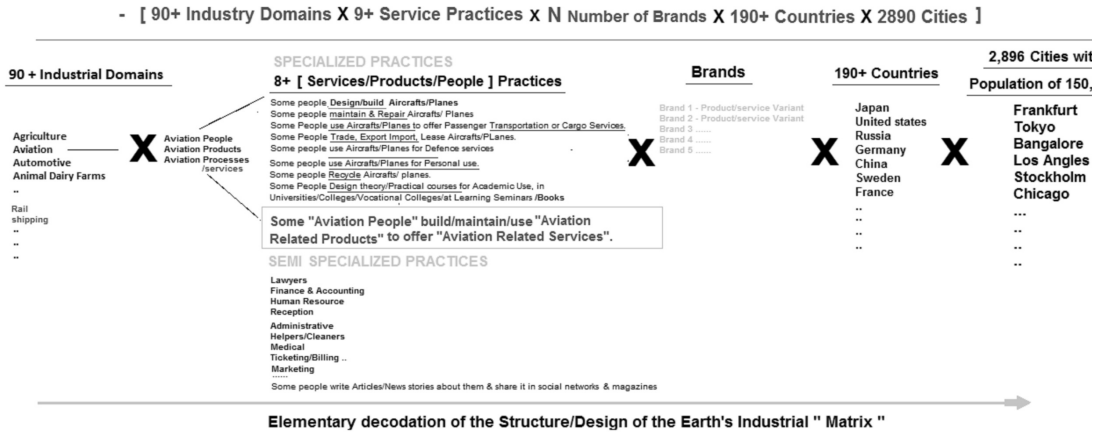
## 1) Enterprise Architecture of Industrial Enterprises or Industrial Systems using Matrix Methods.



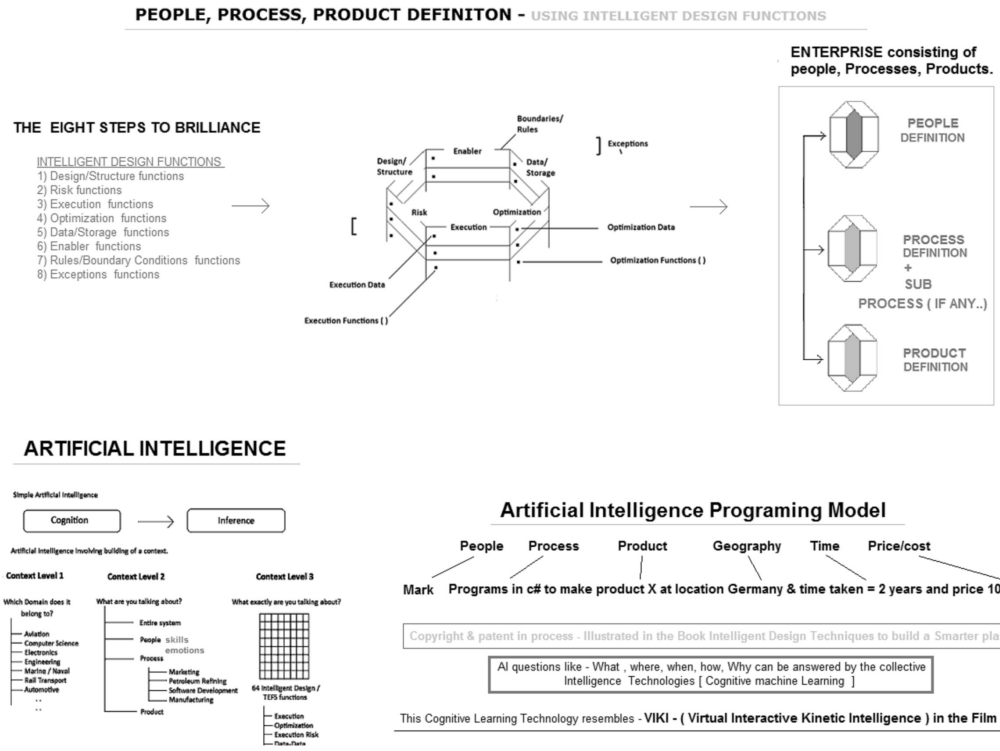
## 2) Enterprise Architecture of Software Enterprises or ICT Enterprise Systems. ( using Matrix Methods )



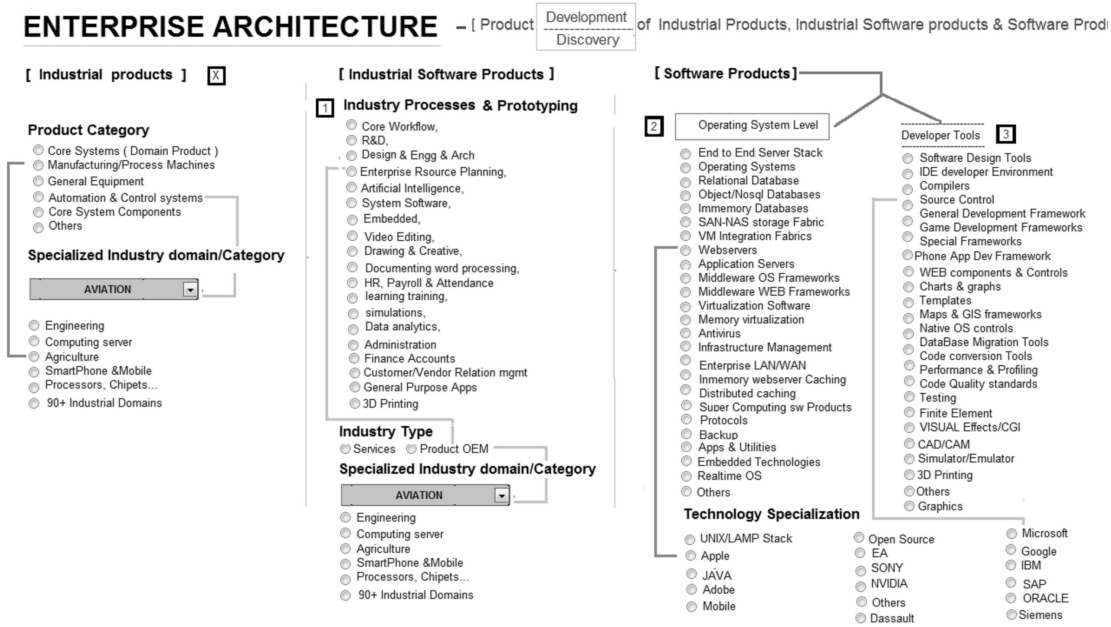

## 3) Enterprise Architecture - The Big Picture, a 20000 Feet View.



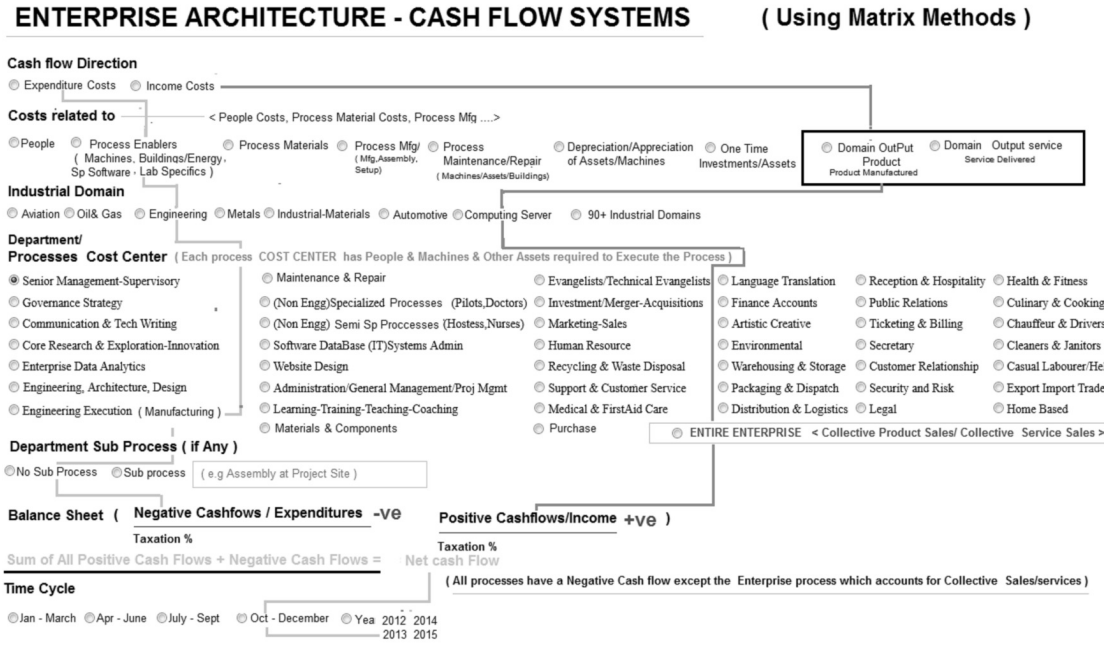
# 4) Enterprise Architecture - Micro Definition of People, Processes, Products - a 50 Feet View.




# 5) Enterprise Architecture - Specialized Product Development & Product Discovery.( using Matrix Methods )

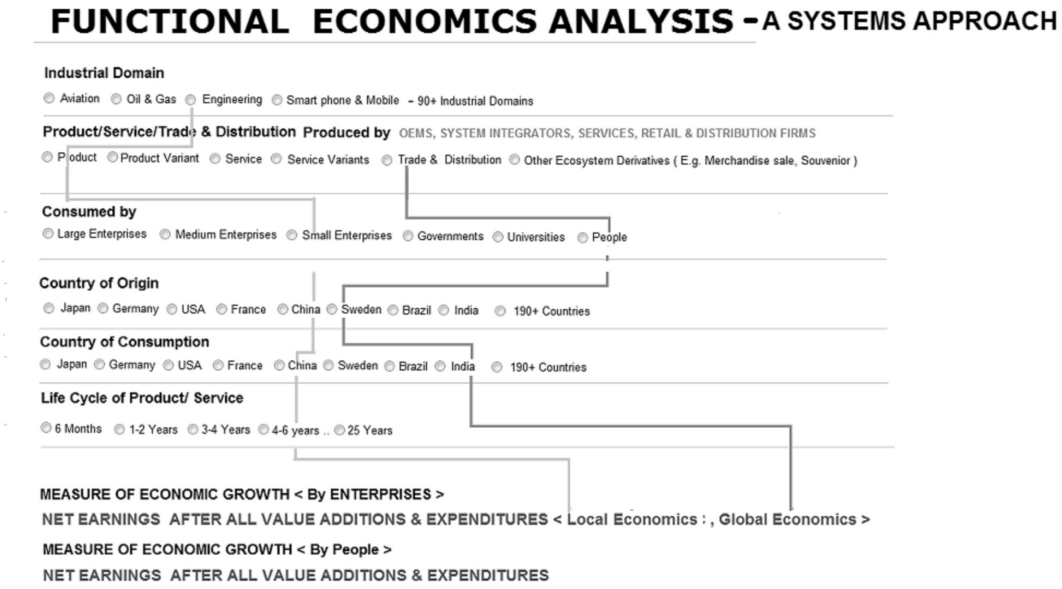


## 6) Enterprise Cash Flow Systems( using Matrix Methods )

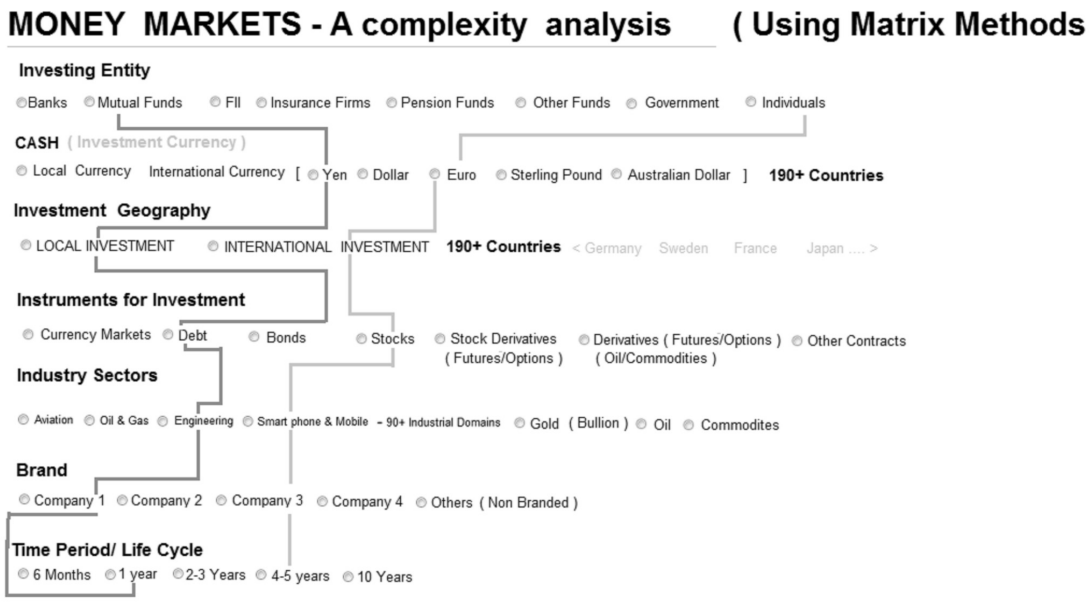



# Economics, Financial Markets & Stock Markets

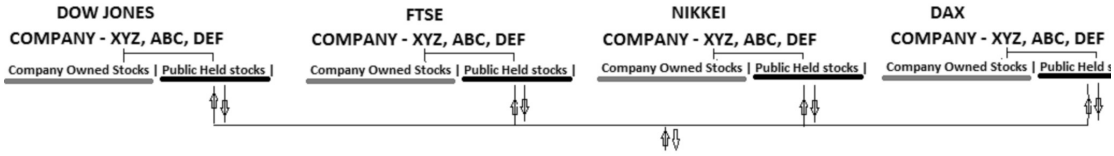
## 1) Functional Economics - A Complexity Analysis ( using Matrix Methods )



## 2) Complexity analysis - money ecosystem & markets ( using Matrix Methods )




## HOW STOCK MARKET INDICES WORK




Note: Oversupply of Money to any Particular sector or Commodity breaks down the Thin Volume Demand Supply equations and leads to Thick Volume Transactions. An Analogy can be drawn to Group Buying. Further When Traditional deman supply equations break down by the supply of Money( Liquidity) , New Triple Equations are formed - [ Real demand supply - Equations ] + [ Thick Volume Demand Supply - Equations ] + [ Actual Value of Money - Equations ].

Note x: Oil Mechanics: If 109 $ is the cost of Oil and consitutes 2 trillion Dollars of a Budget, Economics tells us that value needs to be created at a rate greater than 109$ by the economy, by using the Oil for work, travel, or Factory Production.

## STOCK MARKET STRATEGIES

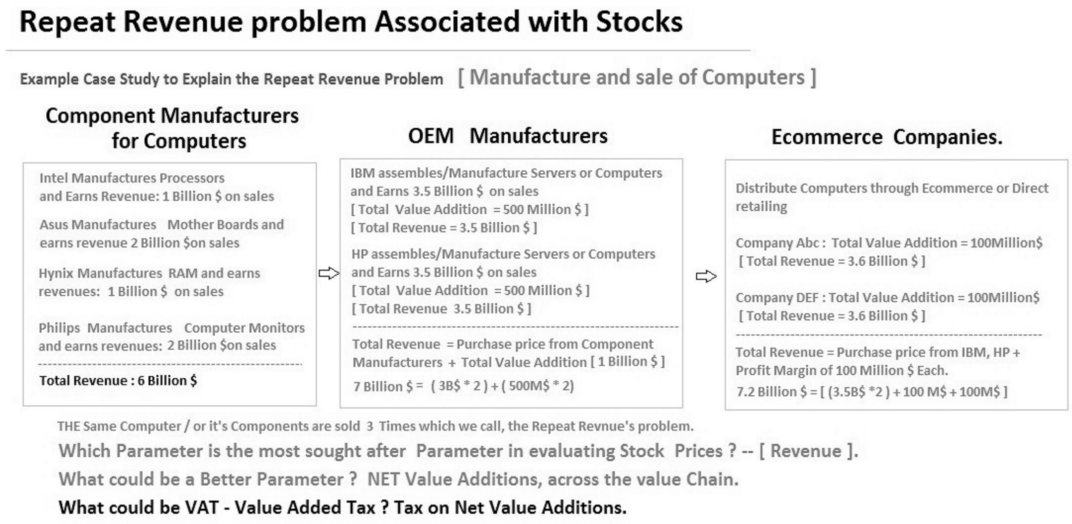
Sample strategy practiced in stock markets = Dump underlying --> Buy futures --> Pump Underlying --> Sell futures --> Generate Cash ---->Rotate sector/Stock - Repeat .

Other Strategies include - Rotating sectors [ Vertical cycle, Horizontal cycle, Vertical Horizontal cycle, Horizontal Vertical cycle ] - cycles rotating Clockwise, Anticlockwise, Randomly, by Price range.

Buying Selling Strategies: Slow Buying ( Pumping ) , Fast Buying ( Pumping ) , Slow selling ( Dumping ) , Fast selling ( Dumping ) .
Other Strategies: False ( Fake) Pumping, False ( Fake ) Dumping.

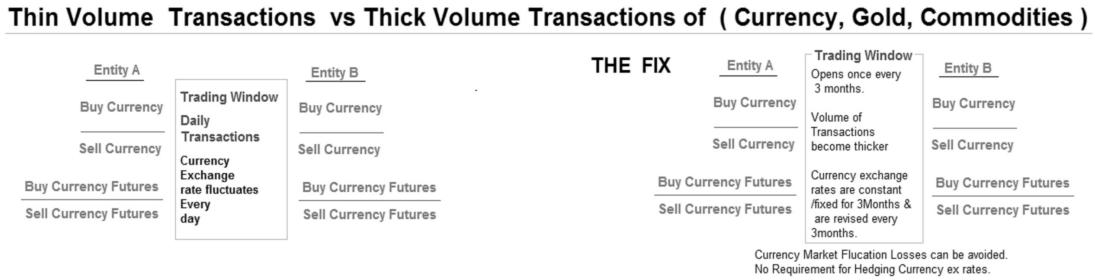
Note: Pumping and Dumping are represented by Distinct [ up & down ] graph Movements.

## 3) The Repeat Revenue Problem - Failure of the economic parameter - "Revenue" as a Major Parameter of growth, in analysisof Stocks or Industry Growth ]....



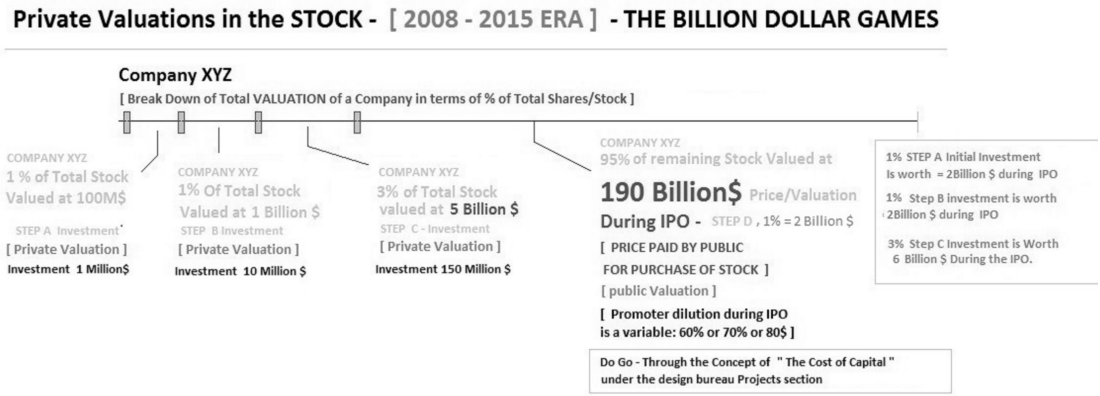

## 4) Complexities,Volatility and Risks associated with Derivatives, Futures, Options and it's Fix.....


## 5)Thick Volume Transactions vs Thin Volume Transactions ...




Currency & Commodities Fluctuate Every Day - as they are Bought and sold on Exchanges. However the VOLUMES at which they are traded, do not reveal the Actual values. These types of transactions are called thin Volume transactions. The Thin volume transaction of currencies or Commodities can be made into thick or high volume transactions by a Technique called Periodical Trade Window. The Periodic trade window opens once every 3/6 Months and a LARGE volume of Transactions are carried out in a short span of time. This technique helps in ACCURATE revealing of actual Vales or price of Currencies / Commodities, acceptable by a large number of players.

## 6) Private valuations of companies and stocks.

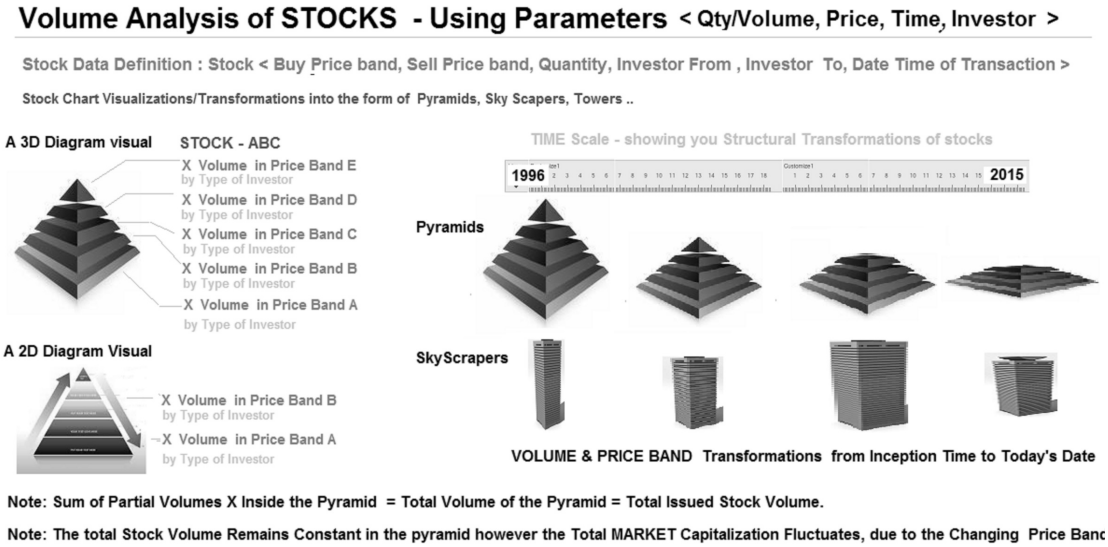



## 7) Volume/Volumetric analysis of Stocks [ Pyramids, Sky Scrapers, [ group by volume, time, Price, Investor TYPE ] ....

Stock are traded on stock Exchanges and have a Life Cycle from Inception date to the Current Date.

Analysis of the Stock Trading patterns from inception - with the perspectives of Volume and Priceband revals multitude of Graphical Structures like Pyramids or Sky Scrapers.

A pictorial Diagram Describing it is shown Below.




**Note: Sum of Partial Volumes X Inside the Pyramid = Total Volume of the Pyramid = Total Issued Stock Volume.**

**Note: The total Stock Volume Remains Constant in the pyramid however the Total MARKET Capitalization Fluctuates, due to the Changing Price Banc**

IF PRICE Volume Data is available only for 5 years, then the Transformational Structure is revealed only for the last 5 years. The Last 5 years of Avg Price Band and Avg Volumes, are used to draw out the structure. The structure then consists of 2 Parts -

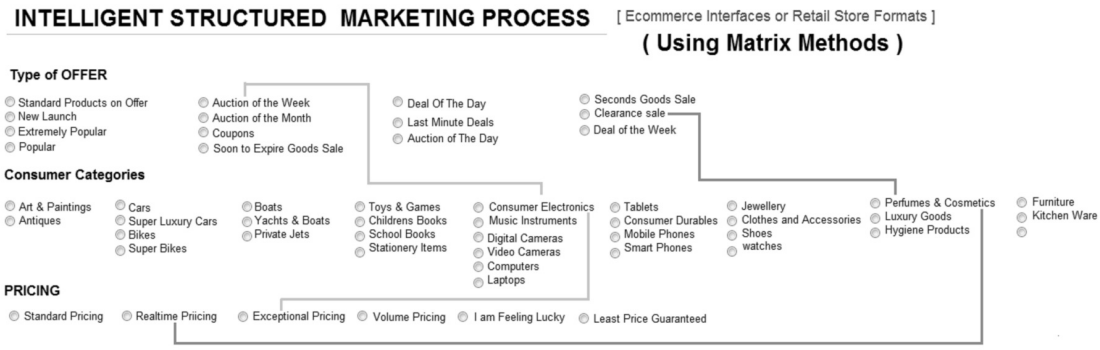
a) The Known Buying Price BAND and Volumes

b) The Unknown Hypothetical Average Buying Price and Dormant Stock Volume.( Representing a HYPOTHETICAL structure pre 5 yr era. )

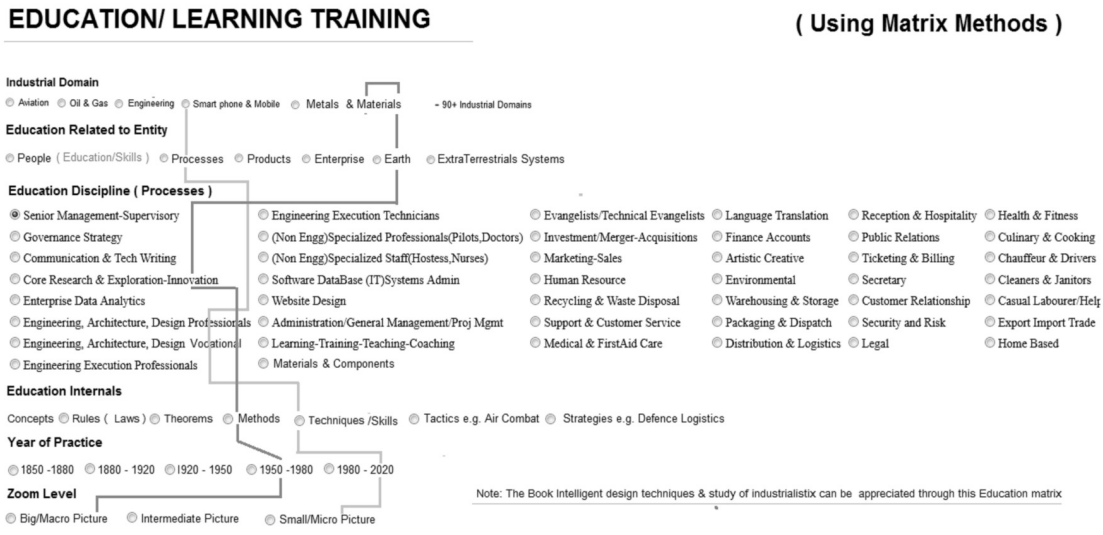
Dormant stock Volume = Total Issued Stock Volume Minus (-) The Actively Traded AVG Volume,( Which could be Thin Volume Transactions, with low volume or Thick Volume Transactions, with Large Volumes )

# Innovative Practices in - HR / SALES / MARKETING / Learning & Training/ Group Buying & Bulk Buying.

## 1) Intelligent Structured Marketing




## 2) Structured Learning & Training




## 3) Group Buying & Bulk Buying - through a Passports Interface

Step 1. A central Authority issues a Physical Passport with Detachable Pages.

Step 2.The Detachable Pages can be exchanged for items like Food Parcels, Fuel, Insurance, Air Tickets though any Shop or Commercial Outlet or Any Counter, Accepting the Passport Schemes.

Step 3.The Detachable Pages have a Commercial Value and can be exchanged for Cash Through the Central Authority.


# 4) Food Factory Concept ( Bulk Food Retailing concept or Bulk Food Parcel Concept ).


Step 1: Fullfilment [ Shops, Factories ] procure Food Items/Groceries ( Rice, Pulses, Fruits, Potatoes, Onion, Soup, Noodles, Chips, Soaps, Tooth Paste, Packed Foods, Corn, Biscuits, Cooking Oil, Coke, Juice, 60-75 Items .... ) from the cheapest & Quality driven Food Source.

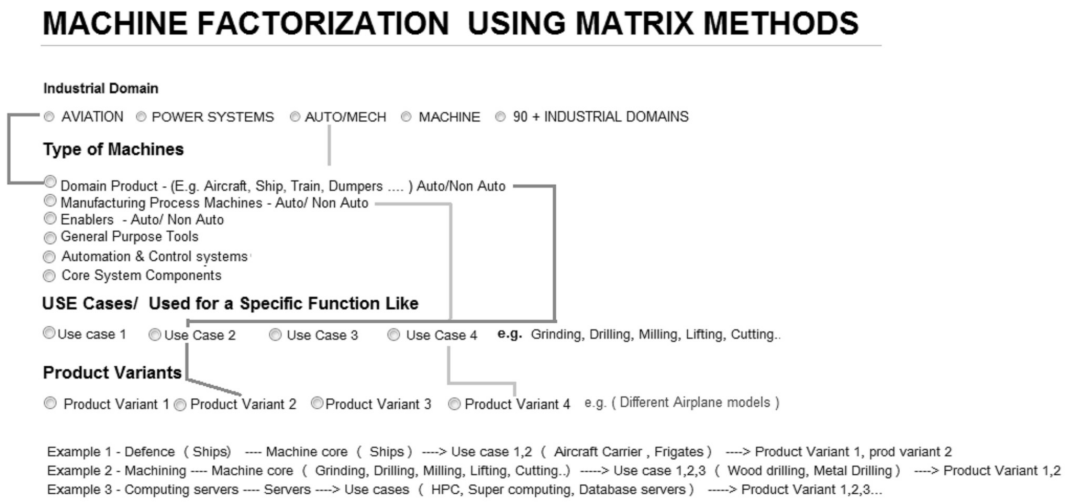
Step 2: Package into Large Boxes or Cartons ( Total Food Quantity = 1 month Consumption ).

step 3 Mrp of Food Food Articles: 1500 ( In Local Market stores ), Selling price: 1100, Procurement Cost 750 ( Food Procured at Scale or in bulk/Large Quanities ).

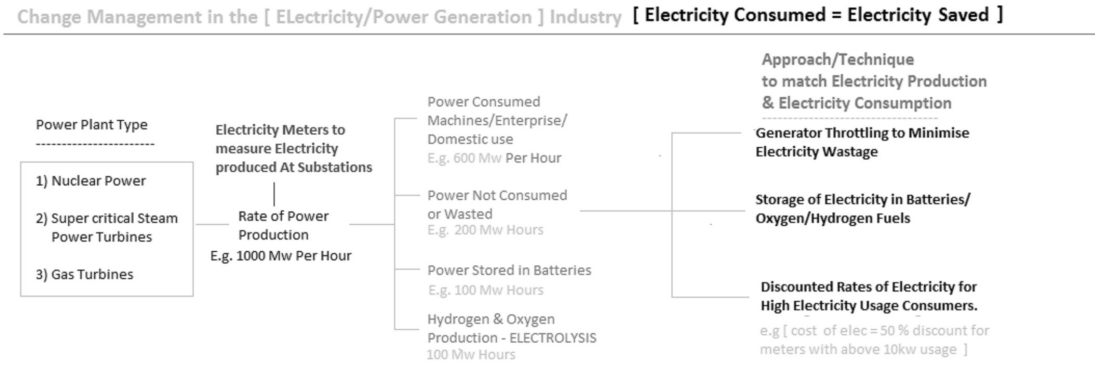
Step 4: Supply the Food Boxes/Food Parcels at House Doorstep, or sell at a Retail Outlet, or Export to any other Country. Bulk Retailing Prices helps: Individuals save Food consumption Costs by 20% to 35%, if they procure food through the food Parcel Scheme. Food Items have a Constant standard price ( without Market Fluctuations ).

# Machines/ Power Systems/ Healthcare

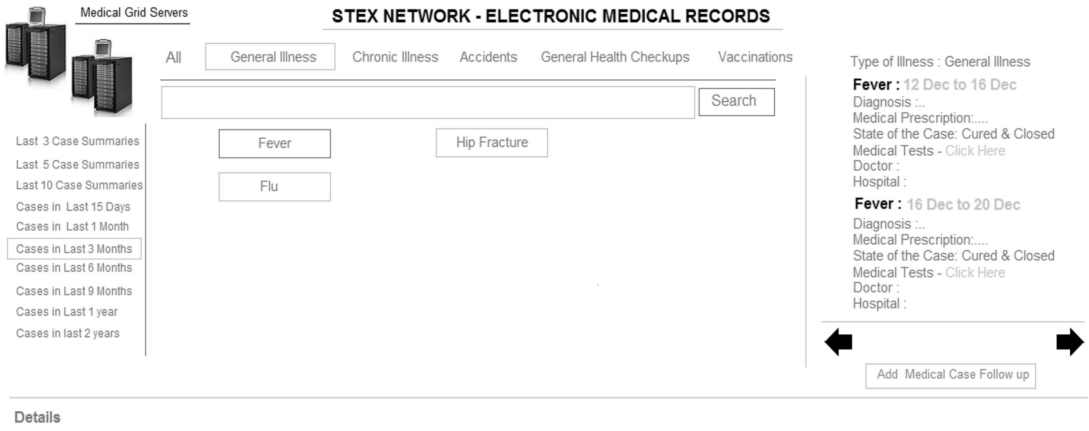
## 1) Machine Product Use Case Analysis




## 2) Building Electricity meters at Substations and Generators to measure electricity produced, to facilitate Generator throttling and usage of unlimited electricity usage slabs by Power Firms.( "Electricity Consumed = Electricity saved" )..



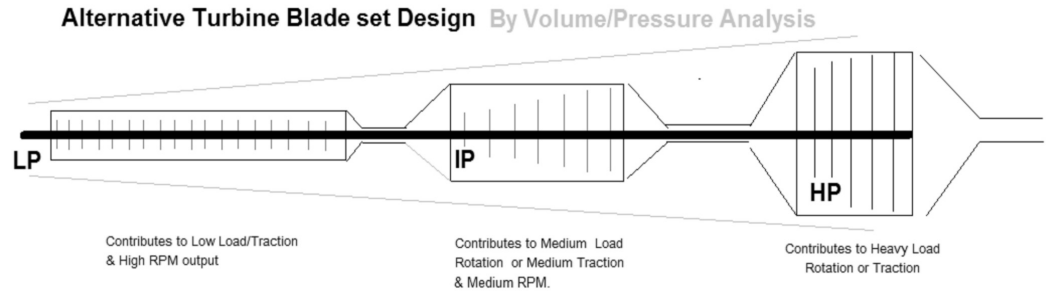
## 3) Electronic Medical records :( HealthCare )




## 4) Volume/pressure analysis and surface area analysis of turbine blades, for solving the WorkLoad Balancing Problem in Super citical Turbine Power plants.

## WORK load = Freescale Function of

## < Traction/Load, High speed Rpm >




The Cone Design helps in keeping the High Energy steam Volume Low ( Constant-ish = gradual expansion ) and Pressure High ( High -ish = gradual reduction ) from the HP to the LP Turbine segments.

The Potential Energy in the Steam is gradually converted to Kinetic Work, under Gradually Lower Volume Container casing conditions , so as to disallow immediate convertion of Super heated steam to Lower superheated steam and steam, by volume expansion of super heated steam.

Chapter 7

# Elementary Research Direction: Physics & Patterns of Colours / Colour Combinations

# Elementary Research Direction : Physics & Patterns, of Colours and Colour Combinations

Colours have a Direct Asociation with Light & Energy

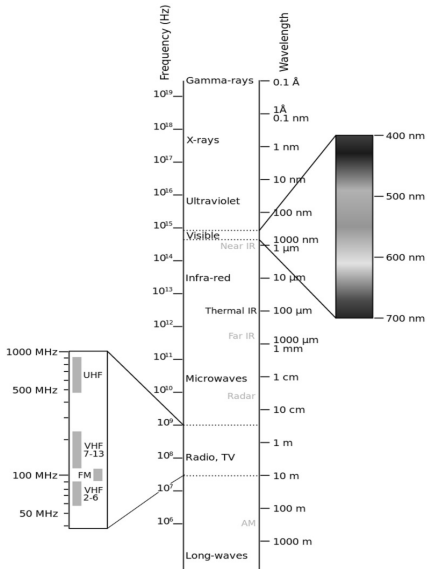



Fig: Electro Magnetic Spectrum

The Occurance of Colours and their associated energies - reveal BEHAVIOUR & ATTRIBUTES associated with the respective colours.

Colours occur in Singularity & in Combination ( Patterns ) in

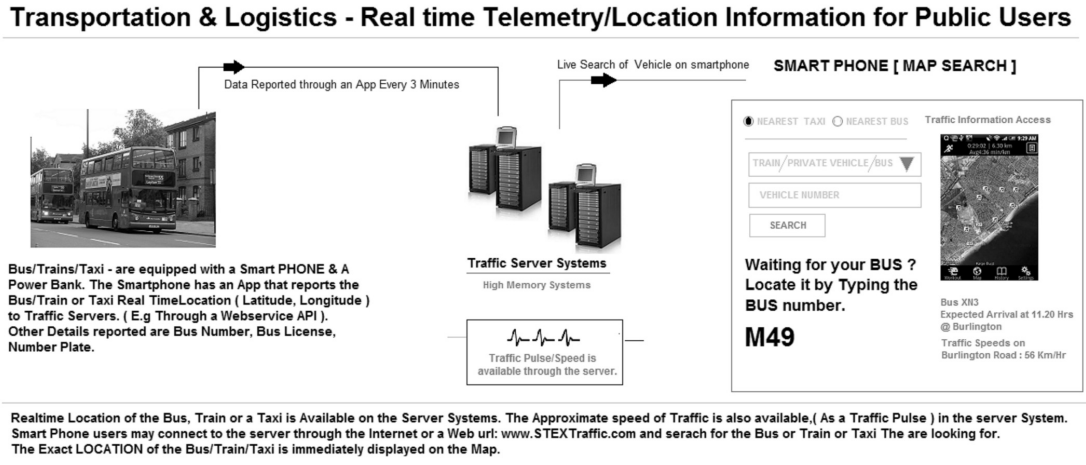
- Biological Life ( Humans, Birds, Animals, Other Biological Organisms )
- Machine Systems
- General Products and Lifestyle Products

Objective is to reveal THE MEANING ( Behaviour & Attributes ) OF COLOURS associated with E.g. A Machine, Design or Structure or Efficiency.

# Chapter 8

# Transportation & Logistics - Real time Telemetry/Location Information to Public Users through Smart Phone Maps

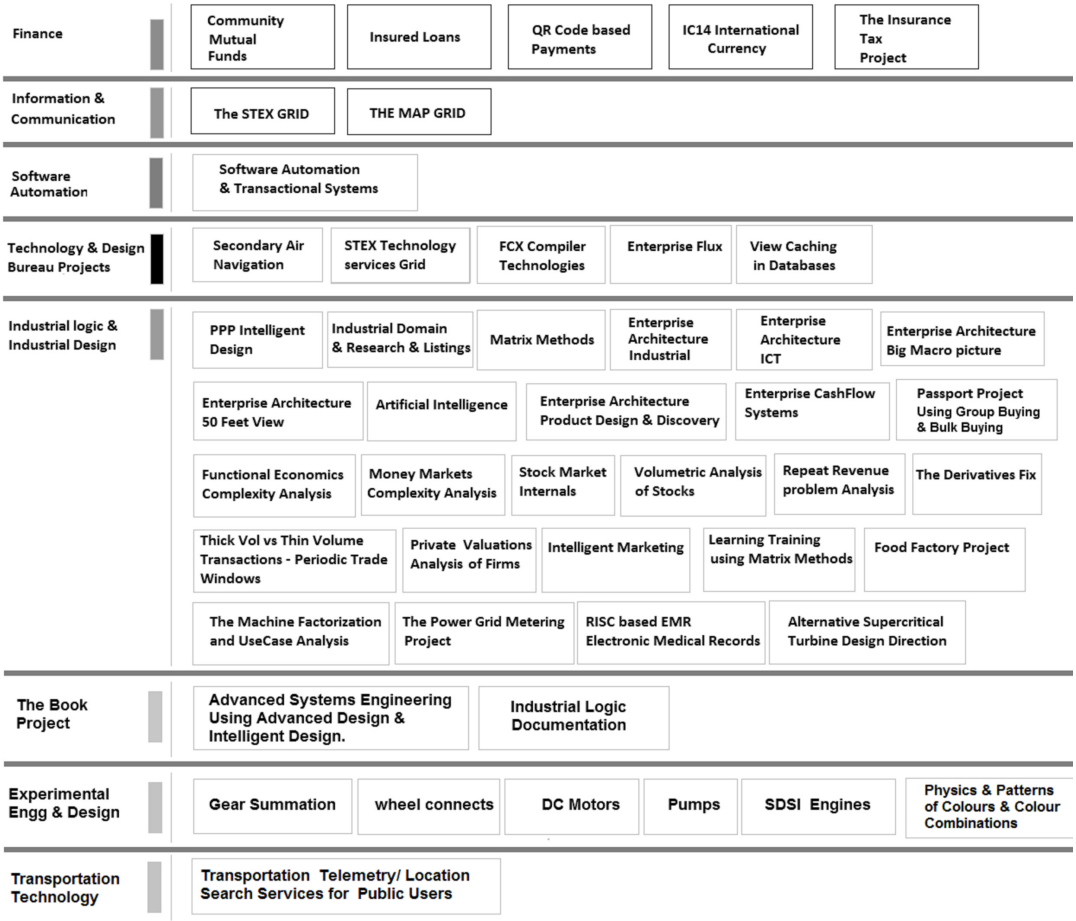
# Transportation & Logistics - Real time Telemetry/Location Information to Public Users through Smart Phone Maps



Chapter 9

# STEX Projects & Concepts - The Complete Big Picture.

## THE STEX PROJECT & CONCEPT PORTFOLIO - THE COMPLETE BIG PICTURE




Note: The Techniques/Concepts under the Section Experimental Engg & Design are not Discussed in the Book. They can be referenced at the website: www.stexinternational.com.

# ADDENDUM 2:

# Revision Date: 11 December 2015

# Extended Chapters and Work - Using Multi Disciplinary Perspectives.

**Chapter 1:**

**Advanced Enterprise Architecture - A 50 Feet View**

**Chapter 2**

**Super Critical Power systems - a Multi Disciplinary Perspective**

**Chapter 3**

**Electrical Meters - at Substations**

**Chapter 4)**

**Advanced Marine Propulsion Systems**

**Chapter 5**

**VRVR Techniques in Automotive**

**Chapter 6**

**Technique of Force Balancing with an Example**

**Chapter 7**

**VRVR Technique - application in Helicopter systems**

**Chapter 8**

**VRVR Technique - application in Super Critical Power Systems &**

**Marine Propulsion.**

**Chapter 9**

**VRVR Technique in Hand Drills**

**Chapter 10**

**VRVR Technique in  Aircraft Jet Engines**

**Chapter 11**

**Industrial Low Cost Housing - Outer City Architecture**

**Chapter 12**

**Concept of Time - Time Evolution and Time Fracture**

**Chapter 13**

**Concept of Strata Evolution & Strata Fracture**

**Chapter 14**

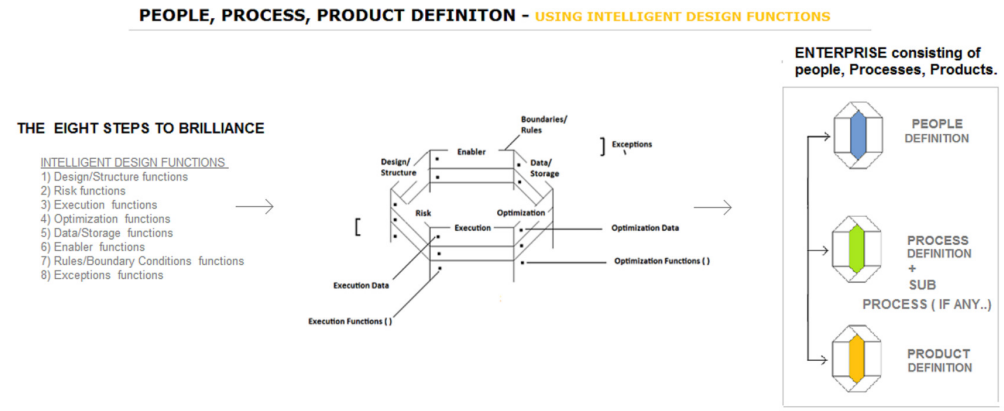
**Decoding the Universe - The Constructs that make up the Universe.**

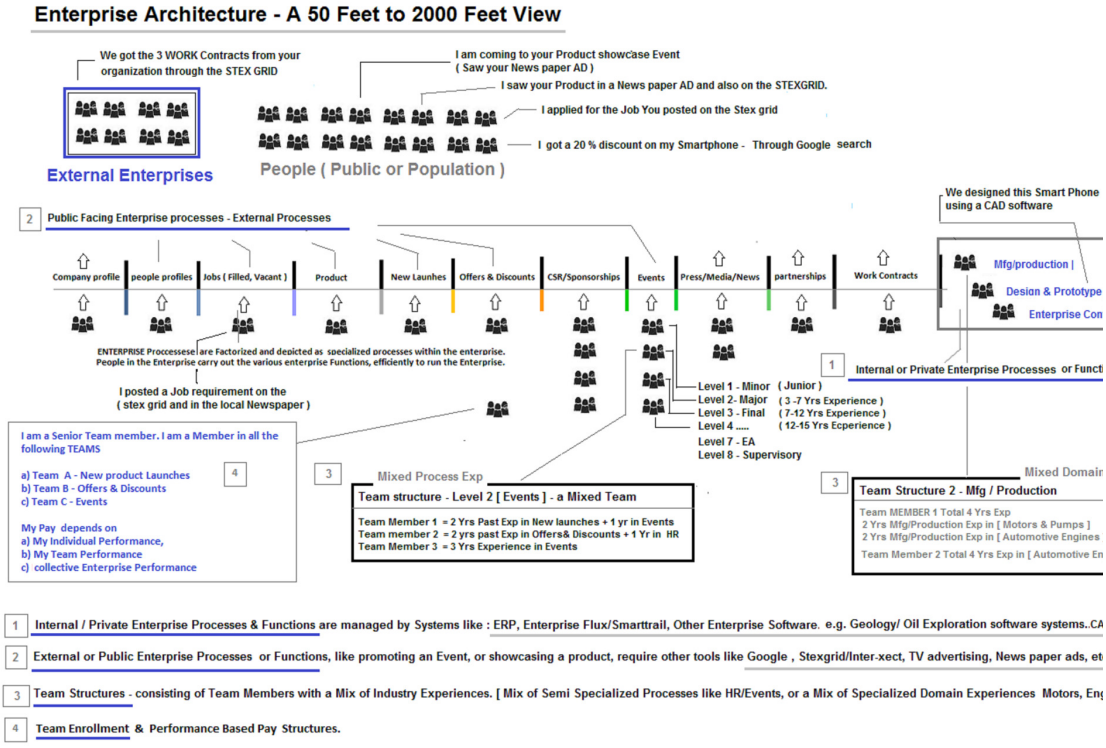
**Chapter 15**

**Visions of the Cosmos - The ArchoSphere / The Sub World / Sub Sub World**

## Chapter 1:

# Advanced Enterprise Architecture - A 50 Feet View



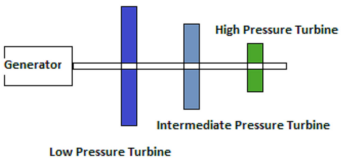



Chapter 2

# Super Critical Power systems - a Multi Disciplinary Perspective

## Alternative Turbine Blade set Design By Volume/Pressure Analysis

### Conventional Power system Design upto - Upto year 2015 - 2017 ERA [ A comparison ]




The TOTAL POTENTIAL Energy of Coal = 100 Units of Power. The Super critical steam gets converted to Kinetic Work at the Turbines stages A. B, C. The Overall efficiency of Power Turbines is roughly 40+ %.

That means that 100 Units of Coal Steam power leads to about 40 Units of Electrical Energy. Lets Break down the work load of the Turbine set into the two simple parameters < Load Bearing and RPM >.

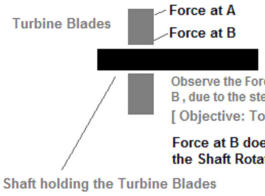
The Generator creates Electro magnetic resistance, which is called the load. This load must be borne by the Turbine Set. The Load consists of two Factors - Rotational Load ( how much load is rotating ) and RPM ( how fast it rotates ).

An analysis reveals that the [ large ] Low Pressure turbine completes one Rotation and contributtes to only 1 Rotation of the Turbine shaft. ( 1 Rpm ). Similarly the HP Turbines which are much smaller contributes to High Load ( but With inefficiency, because of the Force distance - E.g. Force B as shown Below ).

In the proposed next generation Systems - The HP Blades are made larger, contributing to Higher Load bearing Capactity, and the LP Turbines are made smaller contributing to RPM of the power system. The LP Small Turbines are enclosed with low volume casings to achive maximum conversion of Potential Energy to Kinetic Energy.

### Study of Forces Contributing to rotation of the Turbine Shafts

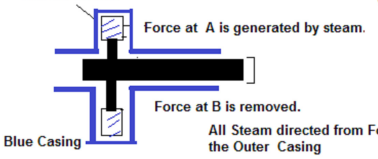
**Existing Power Systems Tech**



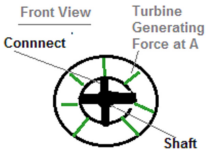

Observe the Force generated by the Turbine Blades at A & B , due to the steam hitting the Blades.
[ Objective: To rotate the Turbine Shaft ]

Force at B does not Contribute heavily to the Shaft Rotation. Force A contributes majorly.

**Other Alternative Designs**

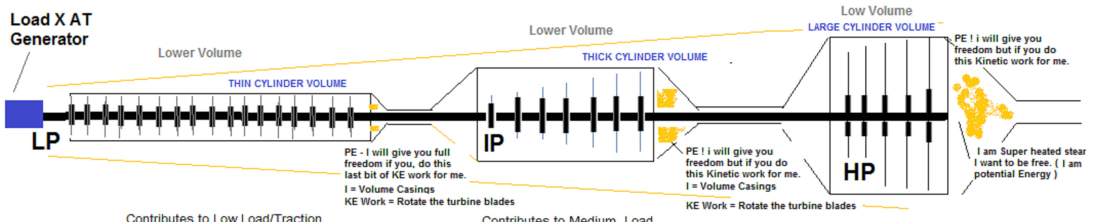



All Steam directed from Force at B to Force at A by the Outer Casing




Hot Steam Converts all the force it was supposed to create at B towards force at A. The Connect of the Turbine near Force A Contributes to easier Rotation of the shaft.

The Proposed Solution or an Alternative design leads us to building Larger HP Turbines, to contribute mainly to the LOAD/TRACTION and lesser to the RPM. Similarly the LP blades are made smaller to contribute to RPM. ( 1 RPM of low Pressure blades contributes to 1 Turbine Shaft RPM. )




Contributes to Low Load/Traction & High RPM output

Contributes to Medium Load Rotation or Medium Traction & Medium RPM.

Contributes to Heavy Load Rotation or Traction

**Turbine Set 3: Takes the Turbine shaft from 2800 Rpm. to 4000 RPM**
**[ Speed Work: Momentum available ]**

[ Large Blades at this stage is not suitable for speed Work ]

**Turbine Set 2: Takes the Turbine shaft speeds from 1800 Rpm. to 2800 Rpm. [ Load + Speed ]**

**Turbine Set 1 : Takes the Turbibe shaft speeds from 0 RPM to 1800 Rpm.**
**[ Load Bearing work ]**

**Turbine 1: I can take this load X to 1800 RPM - My bades are designed for taking load from 0 RPM.**
**Turbine 2: I can take Ths load to X 2800 RPM because my friend Turbine 1 is helping me by providing me momentum and my blades are designed for Speed + Load**
**Turbine 3 : I Can take this Load X to 4000 Rpm because my Friend Turbine1, Turbine 2 are providing me momentum and my blades are designed for speed.**
The Load X can be converted into 2 Loads Q AND Q#, if the Load bearing/Speed capacity of Turbines is high. [ see Power Systems design VRVR Systems ]
**In such a scenario - we can run two Generator Rods/Spindles at 2000 Rpm Each, instead of 1 Generator Rods/Spindles at 4000 Rpm. Note: Theere are High Voltage Engg Limitations per Generator.** [ X= Electro Magnetic Resistance for/Per 1 RPM of the Generator ]

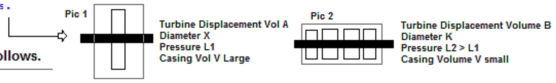
The Cone Design helps in keeping the High Energy steam, Volume, Low ( Constant-ish = gradual expansion ) and Pressure High ( High -ish = gradual reduction ) from the HP to the LP Turbine segments.

**The Potential Energy Drop is controlled using gradually lower Volume casings. PE drop is gradual and not too, SHARP.**

The Potential Energy in the Steam is gradually converted to Kinetic Work, under Gradually Lower Volume Container casing conditions , so as to disallow immediate convertion of super heated steam to Lower subheated steam By controlling the Volume Expansion of Super heated Steam, using, low Volume Container Casings.

THE PE Conversion to KE work is done in stages because - the Entire PE cannot be directly Converted to maximum KE in 1 stage . [ The ultimate goal is to maximise KE from the PE with some minimal losses ]. The Entire system has to be balanced, not to waste PE and convert most of it to KE.

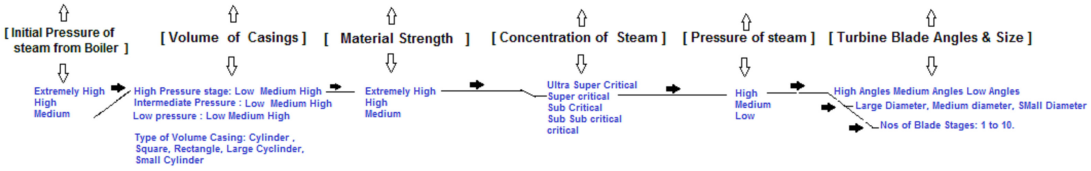
**The Concept of Gradual volume reductions also leads us to a Very Important Concept - [ Volume Shape ]. This Means that a 100 CM Cube volume can be a Rectangular type Volume, A cyclinder type or even a Square type of Volume.** Note the Diagram Above : with Thin , Thick Volume Cylinders .




**The Multi Disciplinary Perspectives required to Design Power System are as follows.**

- **Physics/maths - Volume and Pressure Perspectives. In conventional power systems Boilers Converted Water to Super critical Steam [ Under High Pressure and Closed Volume Conditions ] This Potential Energy is converted to Kinetic Energy through Turbines. [ The Concept of Volume shapes is also an Important Factor in Turbine Size ].**
- **Electrical Engg: Turbine RPM is equated to Electrical HERTZ. + High Voltage Engineering using Copper coils.**
- **Mechanical Engg & Turbine Systems: Turbine Blade Design , Shaft Design and Load, Gears and RPM methods for Multi Generator systems.**
- **Marterials Engg - for Super strong Volume Casings usign strong Materials [ Volume is a factor or parameter used during boiler Mechanics , but is ignored a Little during Turbine Design ].**
- **Architecture - For Big Picture Load Balancing and Micro Picture: Force Balancing [ Force balancing is described in sections below ]**

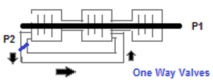
**The Power system design involves the Freescale of 6 Core Variables**




**The Fresscales are applied to design Every Stage of the Power System. Stage [ HP, IP, LP ]**
**E.g. Medium Boiler Pressure - Medium Volume Casings - Material Strength Very High - Super critical - Pressure High - Blades Low Angles -10 Blades in a Stage. -[ Medium Diameter ]**

**Other Advanced Design Strategies – [ Advanced Engineering ]** Experimental & Hypothetical Only !

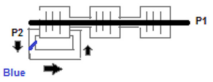
Cyclic Systems




One Way Valves - Marked in Blue

The PE steam is futher recycled to provide Additional Kinetic energy at the Turbines

Cyclic Systems




**The Goal**

The Goal or Concept is to not allow Super critical Steam, to get converted to Sub critical steam, very easily. The Outlet steam at every stage is further recycled to extract, maximum Kinetic Energy at every stage.

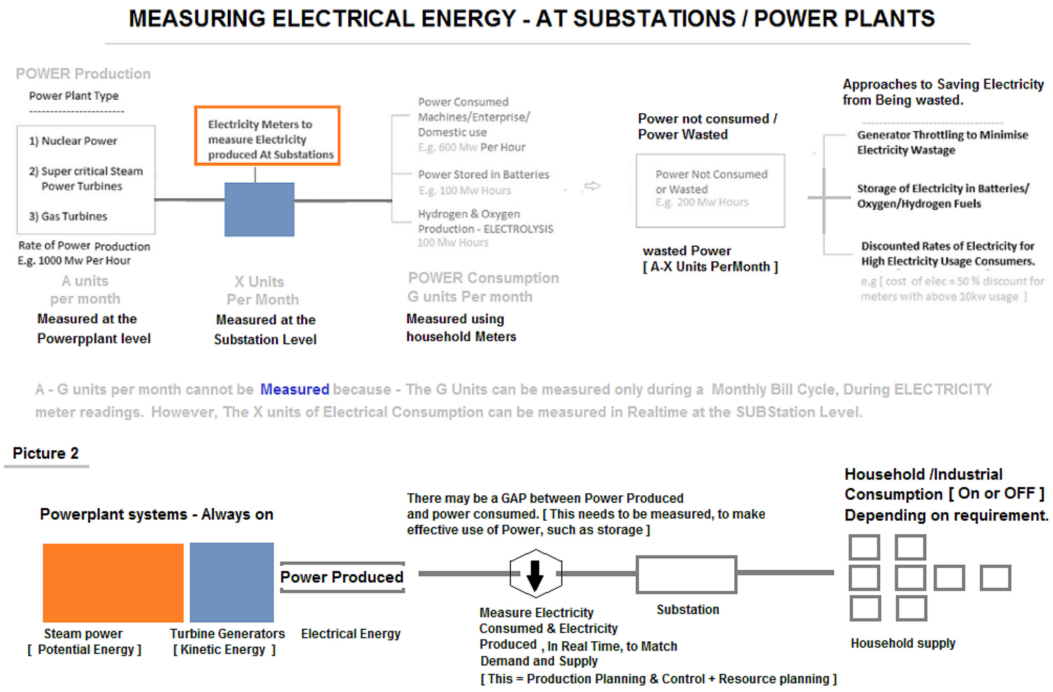
One Way Valves allow Steam to be flow in only the required direction. [ to prevent Back Flow ]

P1 - Pressure at Boiler
P2 - Pressure towards the Outlet
[ P1 Much Greater than P2 ]

A question that's raised is Why Gradually Lower Volumes in Turbine Stages ? Ans: To Not alllow, Super critical Steam to be converted to Sub Critic steam by volume expansion. The Moment Volume is increased The Steam expands losing it's PE. The same steam was created under tight volume conditions at the Boiler End. This same principle is applied at the Turbine end with the Objective of controlling the Steam Expansion. The Super c steam is always looking to find space to expand !. The Pressure difference from the Boiler End and the Low Pressure Turbine end drives the steam HP to LP - BUT IT is done gradually. [ First Heavy Lifting --> Gradually leading to Speed ].

## Chapter 3

# Electrical Meters - at Substations




A - G units per month cannot be **Measured** because - The G Units can be measured only during a Monthly Bill Cycle, During ELECTRICITY meter readings. However, The X units of Electrical Consumption can be measured in Realtime at the SUBStation Level.

**Picture 2**

## Title: Building Next Generation Electricity Meters - at the Substation Level / Power plant level ?

a) SUB STATION LEVEL

b) At THE POWER PLANT Level.

**The Fundamental Question that we are working on is, what happens to Electrical Energy, Electricity, when Electric Energy is Generated at the Power Plants, but is unutilised, at the End level ( Enterprises/Datacenters/Residential Consumers )?**

The Obvious Answer is that - Either Electricity is Stored using, Large scale Industry storage solutions, or is wasted.

**How do we know how much of Electricity is Wasted, and how much is Consumed ?** Lets draw a scenario where - the Electricity Produced at the Electrical Powerplant = 1000 Units Per day. Total Electricity consumed at the End user level ( Residential/Factory/Datacenter ) is 700 Units per day.

That leads to a wastage of 300 Units per day.

**What could we have done with 300 Units of Wasted ELECTRICITY ?** Either throttle the Generators to produce less Electricity, or Store the Excess Electricity in using Fuel Cells, Batteries, Electrolysis of Water ( H2,O2 production ).

**How do we know, exactly How much electricity has been Consumed at the Residential Level, Real Time ?**

Can it be done by Analysing the electricity meters at the Residential level ?

**NO ! .**

There are way too many meters and the End User level ( Residential/Factory/Datacenter ) to be measured in realtime. End user demand is at present measured monthly - when Consumers p their electricity bills. ( In Electrical units ).

### The Proposed Solution by STEX

The STEX solution is to Measure Electricity Consumed at Sub station Level and at the Power Plant Level.

The Power plant level Meters help measure the Electricity generated by the powerplant.

Sub station level ELECTRICITY MEASUREMENT Meters - help Measure Electrical Demand in realtime at the substation level ( That Includes Residential / Factory / Datacenter level / Electricit Storage systems ).

Lets Assume A units of electricity is produced at the powerplant level and lets Assume X Units of electricity is consumed at the Substation Level.

[ Total power station Units A - Total Substation Units X ] = Total Electrical Energy/Power Wasted.

This allows us to, either throttle the Power Generators/Plants to produce less Electricity, or Store the Excess Electricity produced in using Fuel Cells, Batteries, Electrolysis of Water ( H2,O2 production ).

### ELECTRICITY UTILISED = ELECTRICITY SAVED.

As stated electricity stored in Energy Cells or Storage Systems is, one of the ways of saving electricity. The other alternative is to Throttle power generators, to produce less power, when electricity demand is low.

## Commercial Case for Electricity meters ?

Number of Countries = 100+ large market

Number of States = 1 to 50.

Number of Substations and Number of Powerplants = 10 to 50

Market Scale = 100+ countries * Avg 25 states * Avg 25 substations/Powerplants = Minimum 62500 Large Electricity meters. Electricity meters can be used in Tandem at every substation, as a Twin Backup system for [ Measurement & Control of electricity ].

Amount of Electricity that can be saved per Powerplant = Possibly a Very large Amount ( Range MW to GW units per month ).

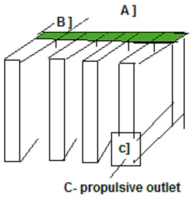
Amount of Electrical energy that can be stored in storage systems, instead of being wasted, creates a potentially new market for Electricity storage systems.

## Chapter 4)

# Advanced Marine Propulsion Systems

### Experimental Engg - Design Directions - IN Marine Propulsion

**Part 1 - Marine propulsion through Air Propulsion**




A] Air Propulsion Systems 4 /5/6 Tubes joined together a the Top.

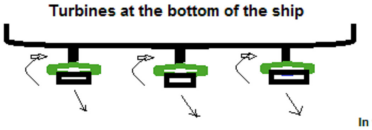
B ] One End of the tube is Air In and

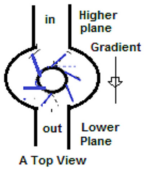
C ] other end is air out. ( propulsion )

water is not allowed to climb through all the 4 tubes simultaneously, therefore tubes 1,2,3 are predominantely dry.

**Part 2 - Turbo Propulsion**

Turbines at the bottom of the ship






In water Based Systems
Water flows in & Out at Turbine speeds

In Air Based Systems - Air Flows in and out at turbine speeds

**Part 3 -Propulsion for Large Ships and Merchant ships**

Conventional Technology ( Proven )


Engine Drive doing X Units of Work - Transfers power to a Propeller to displace Q amounts of Water with a ship load Z . ( Water Displaced = Q Units and helps the ship move at A knots speed .)

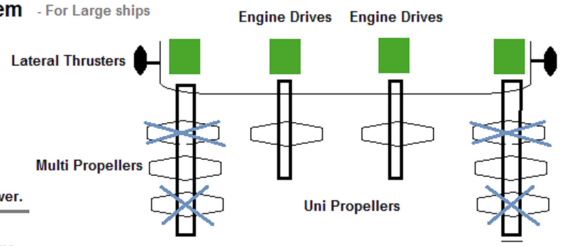
Efficiency of Engine drive converting [ Potential Energy Power ] to do mechanical drive work roughly 25% to 50%.

Efficiency of propeller to Do work - in terms of displacing water, to do work = 15% to 25%

Total Efficiency :
100 Units of Potential Energy drops to 50 Units of Kinetic Energy, drops to 12.5 units of Propulsive Power.( At 25% Prop Efficiency )

### Experimental Technology/Design Multi Prop System - For Large ships

A top View of the Propulsion System [ Multi Prop System ]




**Efficiency of Marine propulsion could possibly be increased by Adding Multiple propellers on the same prop shafts.**

Total Efficiency : 100 Units of PE converts to 50 Units of KE, drops to ( 3 * 12.5 Units ) of Propulsive power. = 30 TO 37.5 Units of Propulsive Power.

The goal is to Increase Propulsive Efficiency through Multi Propeller Systems.

The Load Bearing capacity or propuulsive Speed of Marine systems could be increased by this technique/design.

Most Engine drives operate at a Range of Load or a Range of Work Output : E.g 40 Units of work output to 100 Units of Work Out put.
Most Marine propulsion propeller systems are Fixed - So they operate with a Fixed average Efficiency for the Entire Engine workload range.
( Some propellers are designed to give maximum efficiency at 70 Units Engine Load. They may not be very efficient at 30 Units load or 100 Units load. which is mostly a design to balance speed vs load bearing capacity. A 3 prop System combined may cover the 30 Units to 100 Units Range more easily. )

Finally the Propeller blade angles from prop 1 to prop 3 may vary - 1 works well at high Load bearing capacity ( heavy lifting at low Rpm ) and prop 3 works wel high speed, prop 2 works well at mid weight / mid speed conditions.

Analogy with Oars - 1 Person Rowing a Boat

[ 1 person with 2 oars , 1 person with 4 oars, 1 Person with 6 oars ] . 1 person with 1 oar may not be very efficient. 1 Person with 8 Oars may be too tough.

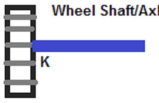
In a Ship - Under Constant Load - 1 Rpm of a Single prop system may be equivalent of 0.6 to 0.7 Rpm of a 3 Prop Systems. Total Volume of water displaced could be 3 x .65 ( 1.95 ) compared to 1 x 1 ( 1 ).

## Chapter 5

# VRVR Techniques in Automotives, Trains, Other Transport Systems

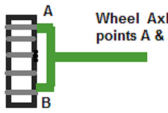
**[ VRVR ] Wheel Connects in Automotive or Other Transportation Systems** [ Analysis of Force TRANSMISSION to the Wheel ]

Existing Systems - Proven Tech




Transmission of Power from the Wheel Axle to the Wheel
Technique A

Experimental Design

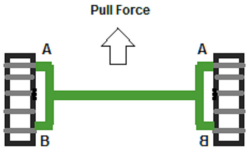



Wheel Axle/Shaft Bolted to the Wheel ( Tyre) at points A & B

Transmission of Power from the Wheel Axle to the Wheel
Technique B [ VRVR Technique ]

**Application areas - Train wheels, Trucks, Earth Movers, Buses, Cars and other Automotives, Bikes.**
**A possible Expected Result: Faster Transport, Larger Load Bearing Capacity.**

E.g. Train Bogie and Train Engines - Transmission of forces [ VRVR Technique ]



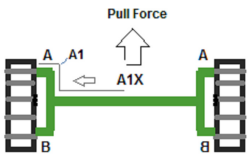

For E.g. A Train Bogie - experiences a Pull Force. The Pull force gets converted to a rotational Force through the wheels. If there w
no conversion to a rotational force - the Train Bogie would skid forward.

The Pull Force is channelised through the Axle to the Wheel Joint where the rotational force is applied. If wheel Connects were designed as per Technique B - The rotational force is transferred to the wheel at points A & B, Rather than at the Center.
[ VRVR Technique ]

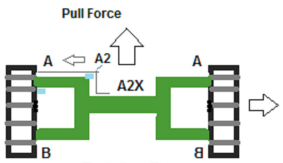
A Test to test this Technique B.

A Threshold Force is USED to pull a Bogie with Regular wheels. It is measured as X. A threshold Force Q is used to pull a similar Bogie with the technique B Wheel Connects.
The Difference between Q and X would explain the Efficiency Gained by Technique B.

**Advanced Approaches - Technique B advanced Approaches** [ VRVR Technique ]




Technique B
[ VRVR Technique ]



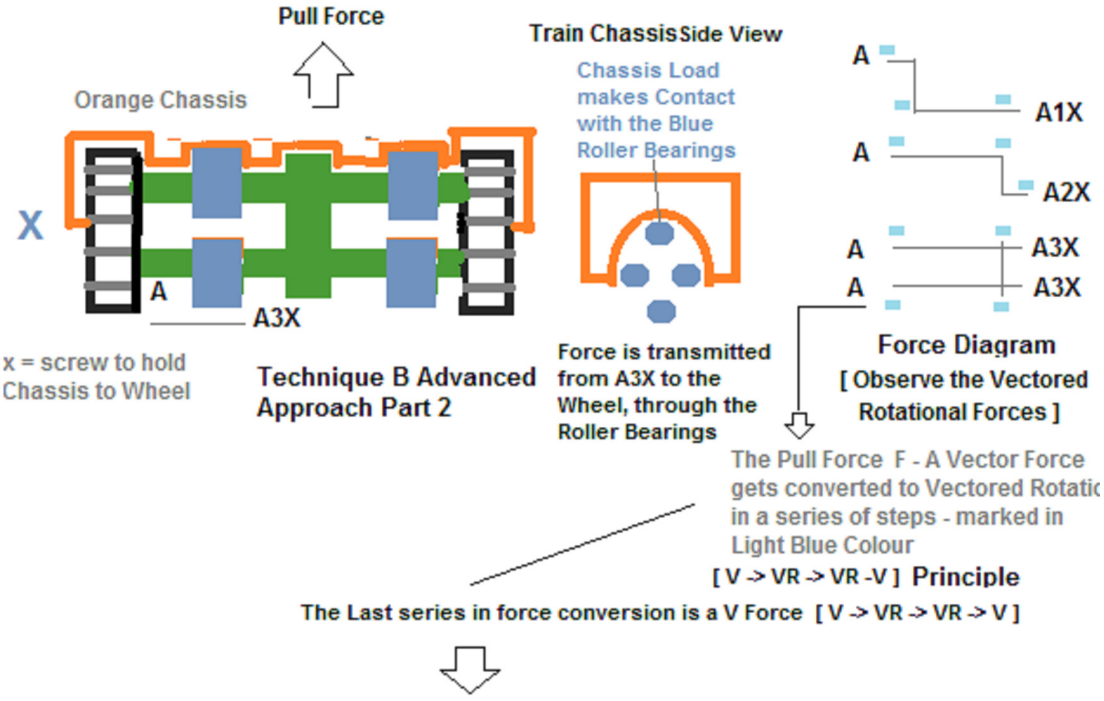

Technique B
Advanced Approach
[ VRVR Technique ]

Observe the Gray Arrow lines depicting the Way Force transmission takes place through Points A1 to A or A2 to

Compare Force generated at A1, with Force Generated a
A2. ( Think of the Distance Parameter /Factor ).

# Advanced Approach Part 2 - Extended tech [ VRVR Technique ]

## TRAIN Bogie




This leads us to to the concept of Force Distribution or Force Balancing.
( Not Load Balancing )

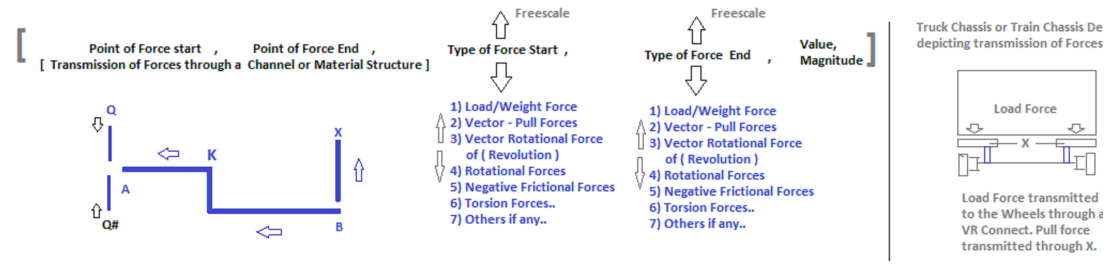
**The Rotational & Torsional Load bearing capacity is an associated to Material Strength.**
**Load is also a Force but is depicted as a Bigger Picture, Below.**
Load is also sometimes total Work load or Work Out Put.

Pure Rotational forces are not very Efficient in Transmission of Forces. [ R -> R ] is not as efficient as [ R - > VR -> VR -> V ] Sequence of Force Transmis

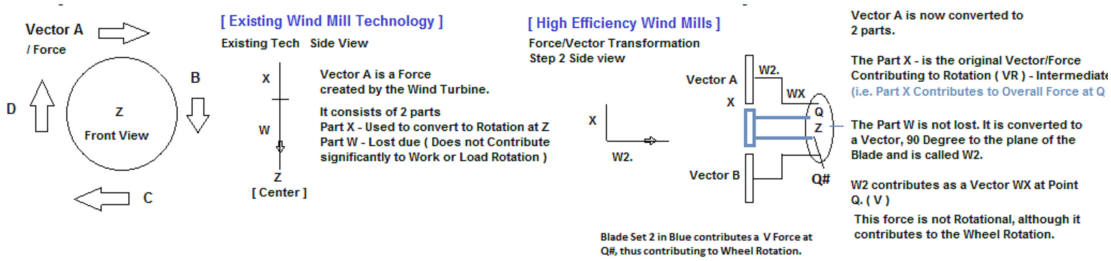
Chapter 6

# Technique of Force Balancing with an Example - Wind Mill Systems & Rail Systems

**The Technique of Force Balancing can be Enumerated or depicted through the following Function Set**

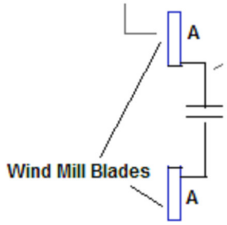



**Analysis of VECTOR Force Transformations in a Wind Mill/Wind Turbine System**



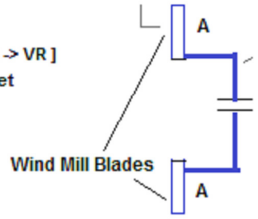
# Application of Force Balancing and Vectored Rotation Forces in Wind Mill/Wind Turbine Technology [ VRVR Principle ]

Wind Creates Force at A
TRANSFORMED TO Rotation.



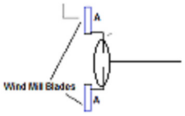

Wind Creates Force at A
TRANSFORMED TO Rotation.

[ V -> VR -> VR ]
Force Set




**The VRVR is principle is applied to Wind Mills as shown.**

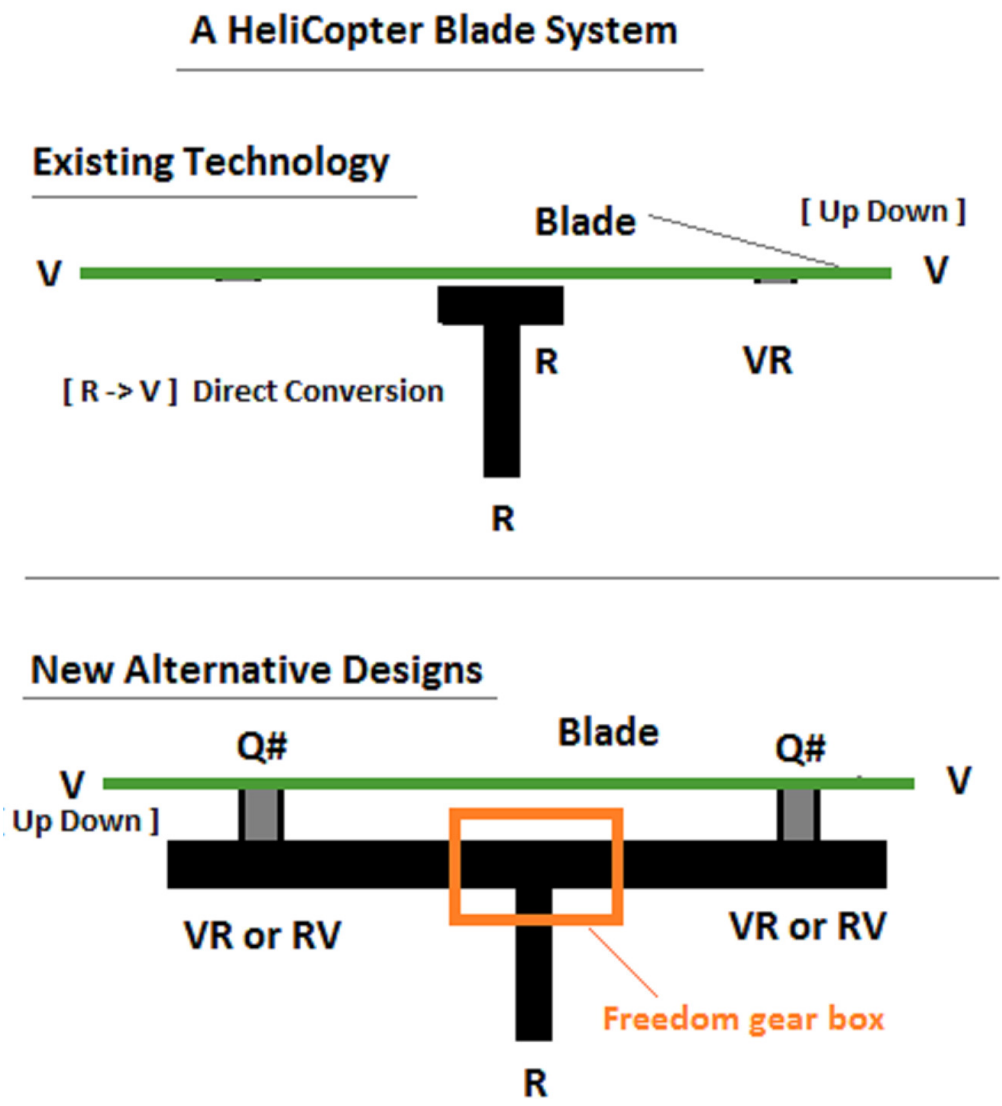
VRVR technology - The Wind Turbine Blade creates a Vectored Rotation Force which is again converted to a Vectored rotation Force before Conversion to Rotational Force at the Shaft. In traditional Tech The Force created by the Wind Turbine blade is vectored directly to a Rotational Force which can be made more Efficient by the VRVR principle .




Note:
This is an E.g. of a Wind Mill with [ V -VR - V ] Force Set finally leading to Rotation of Turbine Shaft.

Chapter 7

# VRVR Technique - application in Helicopter systems



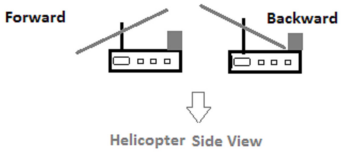

Rotation Force --> Converted to a vector Transmission Force RV/VR applied at Q#. The Force applied at Q# is not rotational, but contributes to the Rotation of Blade ( Marked in Green ).

## Helicopter Systems Design a - Picture / Drawing explaining the Freedom Gears

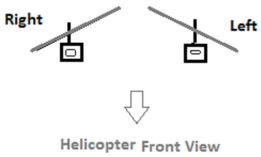
[ High Angle of ATTACK Systems ].

The Freedom Gears system allows the Helicopter to achieve High BLADE system Angle of attack and blade angle of attack , for faster agile syste

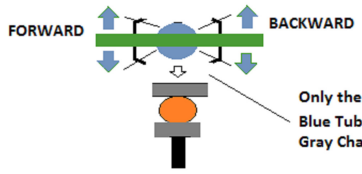
PITCH 1) FORWARD 2) BACK




ROLL 1) ROLL LEFT 2) ROLL RIGHT



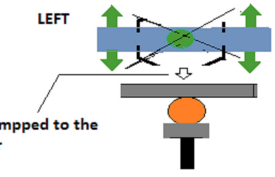

Freedom gear box




Green Tube Goes Through the Blue Tube

Blue Tube - rotates about its axis to make the helicopter move Forward or Backward.

Freedom gear box




Green Tube Goes Through the Blue Tube

Green Tube - Rotates about it's axis to make the Helicopter Move Left or Right .

! The Blue and Green Tubes are Rigid - I.e. If the Blue Tube rotates the Green tube moves Up -Down and Vice Versa.

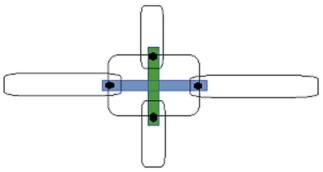
The Pitch of the Blades can be a Constant Pitc system or a Variable Pitch System.

The Chassis System designed to hold the blue ar Green tubes
- allows rotation of the blue as well as Green tubes.

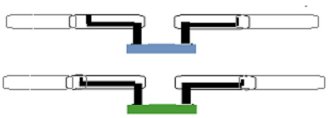
( Through a Gear or Pulley Syste

The Chassis System design, is Shown in the Bottom section - consisting of the ball and Socket Joint. [ orange Ball ]

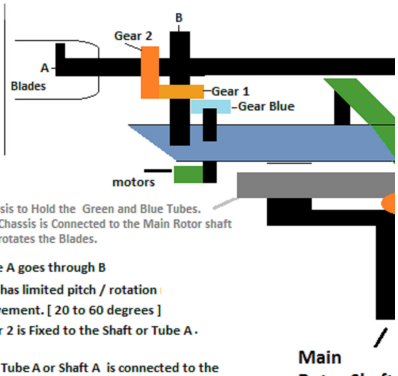
### Blades Top View



### Blades Side View's



Blue and Green Tubes form a 90 degree X - Rigid System




Chassis to Hold the Green and Blue Tubes. The Chassis is Connected to the Main Rotor shaft and rotates the Blades.

Tube A goes through B but has limited pitch / rotation movement. [ 20 to 60 degrees ]

Gear 2 is Fixed to the Shaft or Tube A .

The Tube A or Shaft A is connected to the Blades.

Gear Blue Rotates Gear 1. Gear 1 Rotates Gear 2. Gear 2 Controls the Pitch of the Blades.

Gear 1 Rotates Freely and is not fixed to shaft B or Tube B

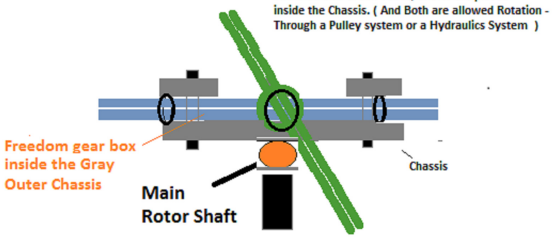
Motors [ Electric - Dual Side Spin ] turn the blue gear, changing the pitch, of the blades.

Hydraulic Systems can also be designed to change the Pitch instead of the Gear Technique.

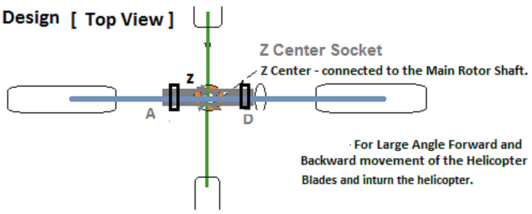
Main Rotor Shaft

### A VIEW of the Gray Chassis from the Side

Gray Chassis is Connected to the Main Rotor Shaft, at the Bottom. The Blue and Green Tubes, Shafts are placed inside the Chassis. ( And Both are allowed Rotation - Through a Pulley system or a Hydraulics System )



**Chassis Design [ Top View ]**



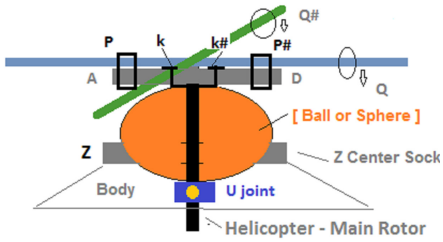

[ High Angle of Attack systems ]

p p# - Are Two Clamp Points where the Blue Tube is Clampped to the Gray Chassis Bar.

The P, P# Clamps allow the Blue Tube to rotate Inside.

K, K#- NUT Bolt system to fasten the Main Rotor to The Gray Chassis.

**Z Center Design - central Joint [ Side View ]**



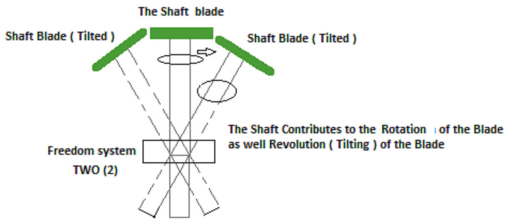

Large Orange Ball placed on a smaller Diameter Socket.
Main rotor shaft - Held by a Screw/Nut joint on TOP.

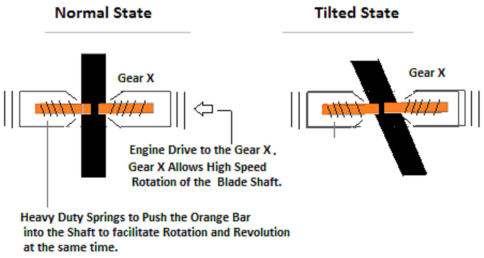
When Blue Tube Rotates I.e Q, the Ball joint moves Forward/Backward. When the Green Tube Rotates, i.e Q#, the orange ball Joint moves left/Right.

U Joint to Allow - Spherical Rotation

AD chassis is always at a 50 -90 Degree angle to the Main Rotor Shaft.

## An Alternative - Simple Helicopter System

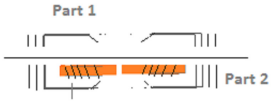



**Gear X - Top View**

**Manufacturing of the Gears**

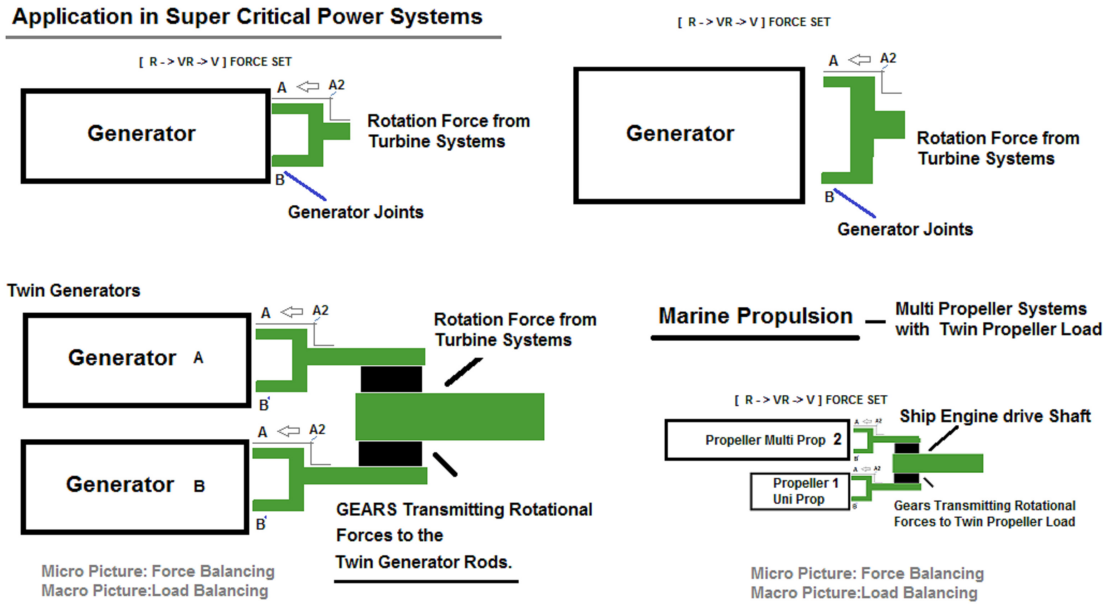
Cut through the X Axis - at the center plane.
Join part 1 and Part 2.




The Spring system

Chapter 8

# VRVR Technique - application in Super Critical Power Systems & Marine Propulsion.




Pure Rotational forces are not very Efficient in Transmission of Forces. [ R -> R ] is not as efficient as [ R - > VR -> VR -> V ] Sequence of Force Transmissi

**The Next Question ? Can The Gears be designed to Increase or Decrease RPM - Rotations per Minute on the Sub shafts or, Maintain Main Shaft RPM. Higher RPM or Lower RPM in the sub shafts, Depends on the LOAD at the Generator A & Generator B or Propeller 1 or Propeller 2 .**

[ Generator Load = Elecromagnetic Resistance load ]

[ Propeller Load - Water Displacement Volue and speed of Displacement ]

Chapter 9

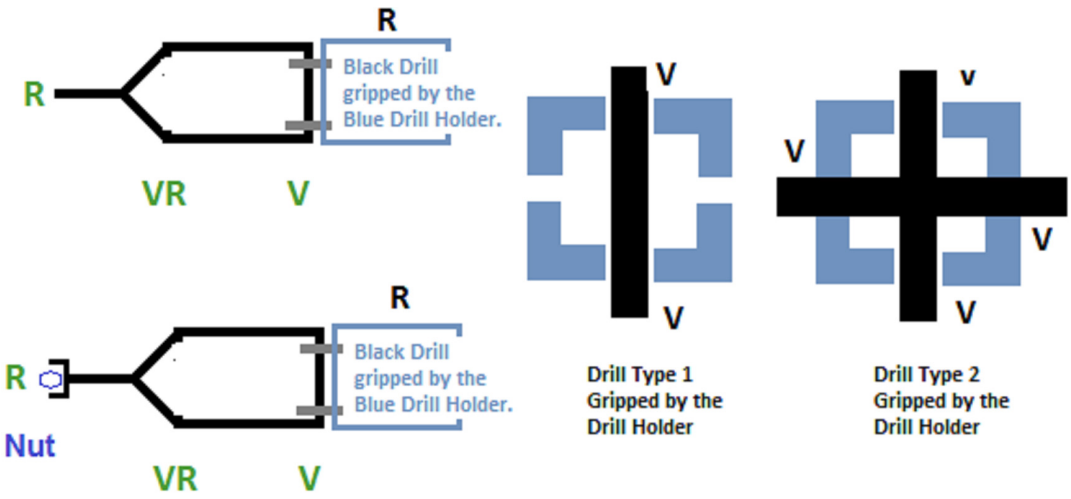
# VRVR Technique in Hand Drills

**THE CONCEPT OF Force Transmission, [ Vectored Rotation ] can be used in Fans, Motor Pumps, Fan Blowers, Rotary Systems, Helicopter blade Systems , Ground Drilling Machines, Screw Drills & Other Drilling Machines.**
Formula 1 Cars too !

---

Pure Rotational forces are not very Efficient in Transmission of Forces. [ R -> R ] is not as efficient as [ R - > VR -> VR -> V ] Sequence of Force Transmissio

## An Example Hand Drill




Drill Type 1 Gripped by the Drill Holder

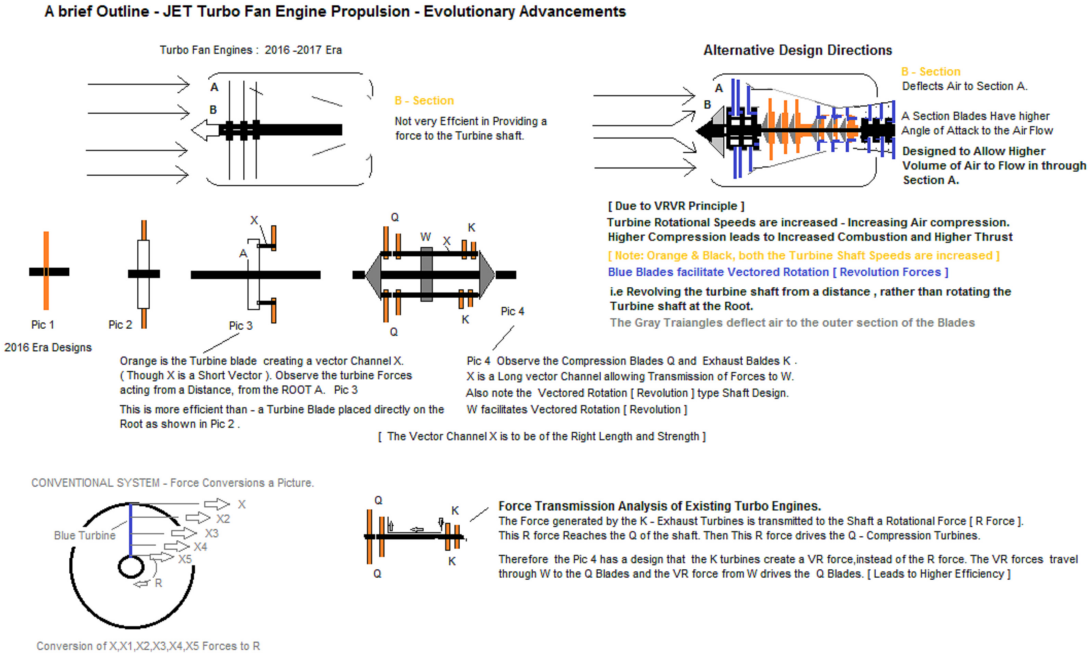
Drill Type 2 Gripped by the Drill Holder

### Front View of a Machine/Motorized Drill

**Transmission of Forces - [ R Drill motor --> V Drill --> VR Drill - R Drill ]**

## Chapter 10

# VRVR Technique Aplications in  Aircraft Jet Engines



Chapter 11

# Industrial Low Cost Housing - Outer City Architecture

**THE HOUSING FACTORY solution - Solving the HOUSING PROBLEM in Emerging Nations, Where slums are a plenty and Standardised housing is available only to a few. ( An Engineers Perspective )**

20 Feet/40 Foot shipping containers can be modified into - Houses or Homes fulfilling the Rural housing Needs.

Each 20 Feet Container Home could be built with a cost of 4000$ - 5000$.

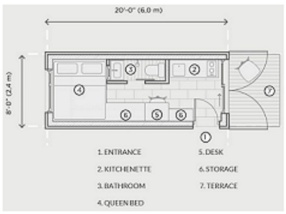
A 1 Billion$ investment by a Government or NGO could provide housing to - 200,000 Families.

Standardised Housing leads to revival of Metal Industries and housing interiors industries

Rural Housing/Slum Landscapes can be changed dramatically without significant investments.

Countries like - China / India have massive populations which are underprivelaged and can therefore create massive Markets.

This architecture/Format could be used in Immigrant territories, Eastern Europe, China, India, Underprevileged territories /Places.




A 20 Feet Shipping container modified to a Standardised House/Home

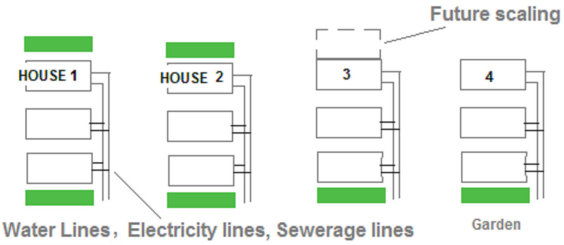
Twin Container Design : 320 Sq Feet


Design 1

Approx Cost: Twin container solution: [ 10-11k $ ] Inclusive of all Home Accesories - Bed, Modular Kitchen, 2 sofas, Dining, Bathroom accesories.

Approx Cost: Single container solution: [ 6K-7k $ ] Inclusive of all Home Accesories - Bed, Modular Kitchen, sofas, Dining, Bathroom accesories.

Smart Outer City or Smart Village



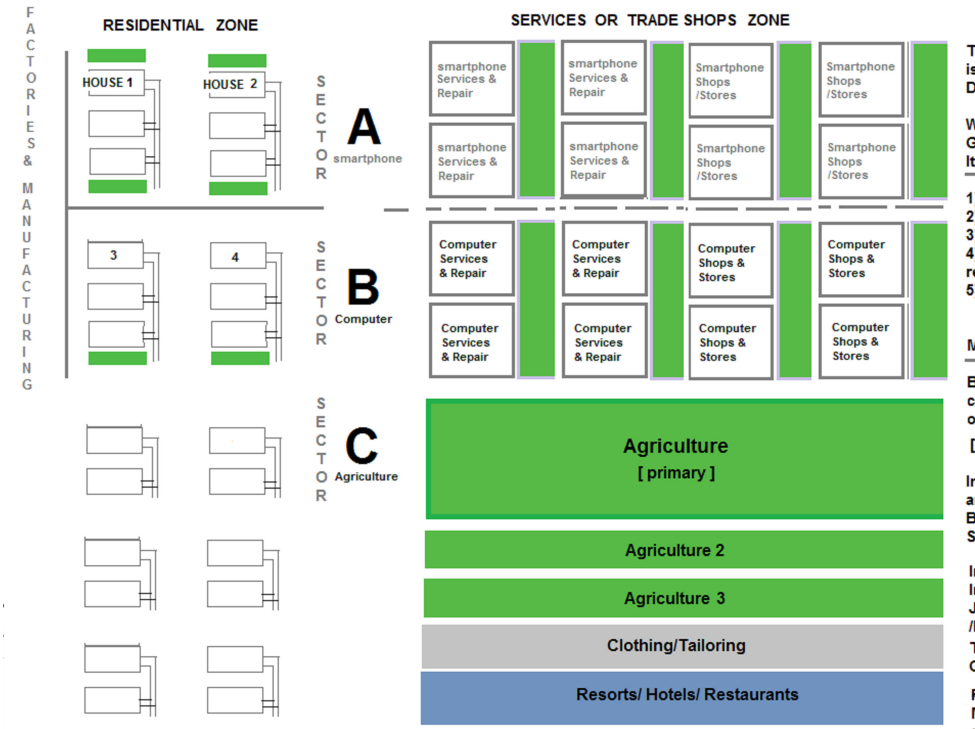

A realistic Picture

Twin Container Design : 320 Sq Feet


Design 2

Outer City Architecture - Home Spaces & Commercial Workspaces - segregated according to SECTORS - Electronics / Airconditioning / Electricals / Computers / Smartphone ... Note: There are 100+ Industrial Domains in the system.

## SECTORAL ARCHITECTURE - SPECIAL ECONOMIC ZONES - MICRO/MINI ECOYSTEM




Time to Build ecosy is minimal - Factory Driven.

Wide range of Geographical scope It could include

1) Outer cities
2) villages/Rural Set
3) war stricken areas
4) Under Privelaged regions
5) Tourism - Resorts

Market economics

Building an Ecosyst could cost : in the R of 250K$ - 300K$

[ 50 Building Blocks

Introduces the conc an individual or tean Building, Owning & Sustaining an ecosy

Investors could be
Individuals
Joint individuals /Partners
Traditional Investor Governments

Fractional Ownersh Micro Ecosystems i a possibility with investments rangin 10K$ - 20K$

Micro Ecosystems be rented out to pe - on a rental basis.

[ Partially or Comple

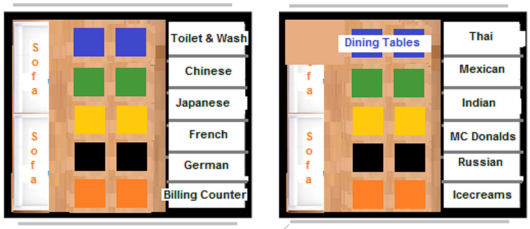
Employment

E.G. People agree contractually to wor a Micro Ecosystem, their livlihood, Pay on their Home.

They Are a part of t Consumption cycle

people already having existing establishments within the city - may have extended ownerships or twin ownership in these micro systems. [ without much expenses ].

### MODEL - World Cuisine Restaurants

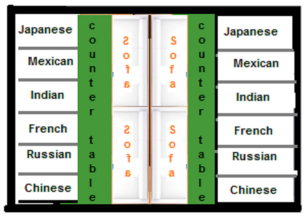



Floor 1 - 960 sq Feet  Glass outer  Floor 2 - 960 sq feet

6 * 20 Feet Containers modified into 960 Feet Rooms

Parent Restaurants or Brands may open Counters under one roof

### LIFESTYLE PRODUCTS - E.G FABRICS & TEXTILES




E.g Japanese Fabrics, Textiles & Patterns/Prints

Floor 1 - 960 sq Feet

Chapter 12

# Concept of Time - Time Evolution, Time Relativity and Time Fracture

## THE Concept of Time & Time Fracture

The Concept of System Efficiency extrapolates from people, Processes, Products, all the way to the Quantum System Efficiency of a Planet System. Under Favourable Conditions -System Efficiency increases as we go up in time. E.g. New Machines are Invented, New Information Systems spread information with ease , New Techniques & Procesess ease the way work is done and the speed at which useful work can be done.

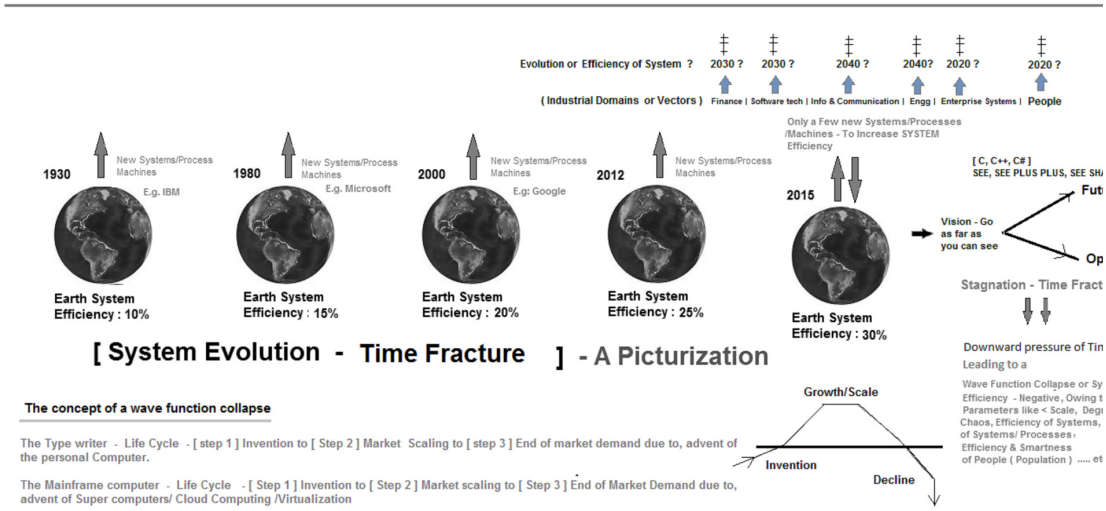
The Earths Systems efficiency can be defined by parameters like

**< Industry Domain, Speed of Doing USEFUL Work, Efficiency of Work & Processes, Scale of work - Number of Users using the Systems , Degree of Chaos/Entropy. >**

**An Example of System Efficiency :**

Max Speed of Cars in 1950 Era / Compared to Maximum Speed of Cars in 2015 Era
Maximum Mileage of Cars Engines : 1950 Era to Maximum Mileage of Cars in 2015 Era
Ease of Access to Information pre Google ( Pre 1996 Era ), compared to Ease of Access of Useful Information ( post 1996 Era ).

Each Time Era is attributed to it's Own Environmental conditions and Each Time Era requires it's own set of Systems Efficiency. Systems constituting People, processes, products, therefore are classified as being, In time, Before time or Out of Time. For a System to climb up in Time - constituent People ( Population ), Processes, products are of the System. are required to Evolve or become more efficient. [ i.e. Be In Time or Be Before Time - Adage -> Time and Tide Wait for no man ]. Time Fractures occur when Planetary Systems and their constituent systems like machines/Information, People systems are out of Time. This is analogically similar to a Quantum Wave Function Collapse . This leads to A State of system where - Over all System Efficiency deteriorates, and as a further consequence, puts systems under the DOWNWARD pressure of time. DPT may lead to low system efficiency, higher entropy, Chaos or Fragmentation.

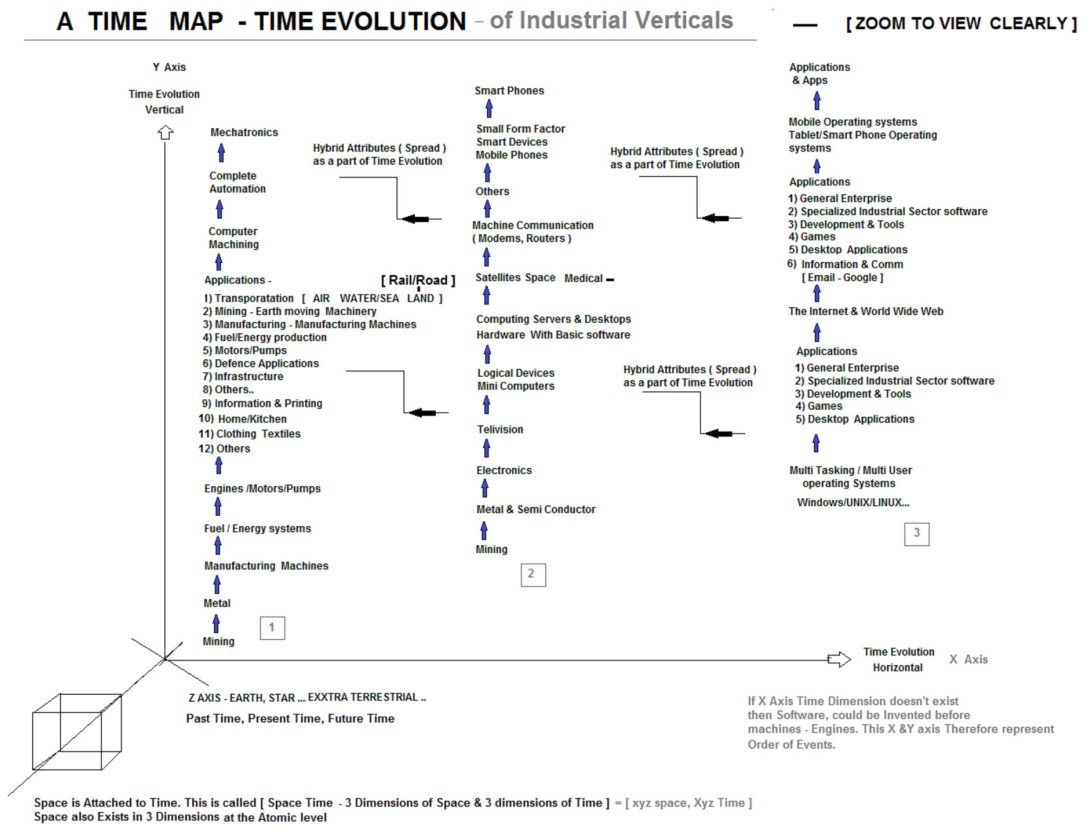


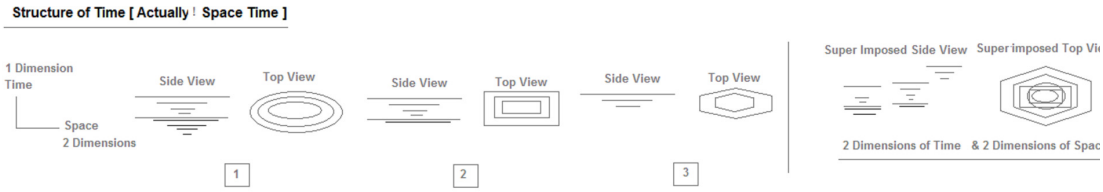

**The concept of a wave function collapse**

The Type writer - Life Cycle - [ step 1 ] Invention to [ Step 2 ] Market Scaling to [ step 3 ] End of market demand due to, advent of the personal Computer.

The Mainframe computer - Life Cycle - [ Step 1 ] Invention to [ Step 2 ] Market scaling to [ Step 3 ] End of Market Demand due to, advent of Super computers/ Cloud Computing /Virtualization

# Time Evolution



# Time Relativity

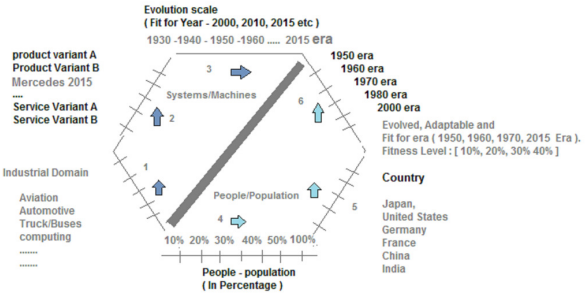
## Time Relativity

**Time RELATIVITY : The relative speed between the system speed and the speed of the individuals/Players of the Constituent system.**

| | Analogy 1 | Analogy 2 | Analogy 3 |
|---|---|---|---|
| **Average System speed of Earth Program [ System Earth Environment Changing ]** | **Average speed of a Class room Course** | **Average speed of a computer program/ [ A game ]** | **Avg speed of a book/Film** |
| Average speed of the Players/Agents or constituent systems, Doing Work in Time. ( sustaining the system ) | Average speed of the Learner / Player /Agent | Average speed of the Player relative to the Game program | Average speed of the perception/Understanding |

---

### THE SYSTEM - FITNESS TEST




An Example :

10% of Population of Germany Adaptable, Fit & Evolved ( Fitness level: 15%) for Automotive Domain, Product Variant ( Mercedes abc ..) built for era ( 2015 ).

90% of Population may be possibly out of TIME for evolution. ( ie Evolution strength required to adapt to the 2015 time line machines/Systems, may be out control from a population of Users Perspective ).

Population of users must try to be in time, for the present / Future Systems built and fit for 2015-2020 ERA, to qualify as being efficient. Systems built for the population, must also find users for the system and qualify to be in the Right Tim Era.

No or Few users for a system/machine, implies a time or evolution Gap, betweer machines/systems and People ( Users ) - further implying a sectoral Time Fracture

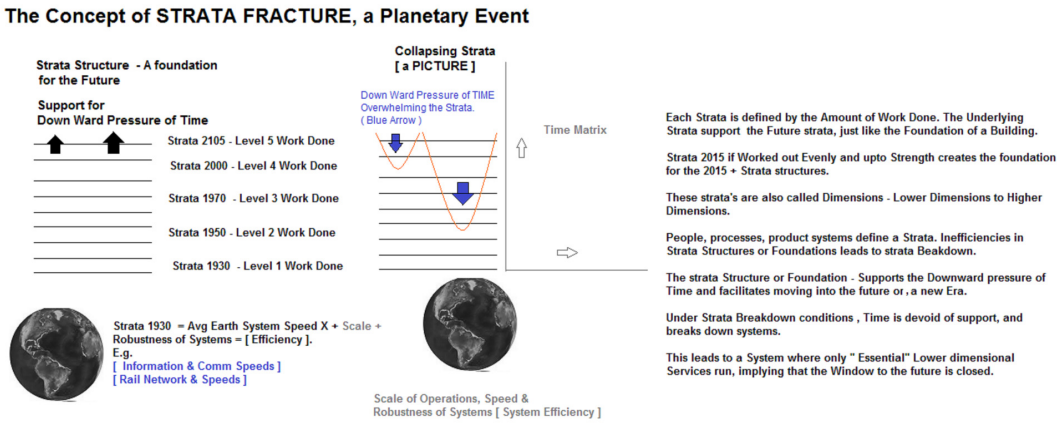
Similarly a Machine system designed for 1960 Era may not find users in 2015, du to the Time Gap between machines/People ( Population ). Time Gaps between systems and system users lead to system fractures.

Multiple sectoral Time Fractures, leads to, larger Systemic Time Fractures. ( Quantum Wave Function collapses ).

[ The representation is for an example purpose only. ]

# Chapter 13

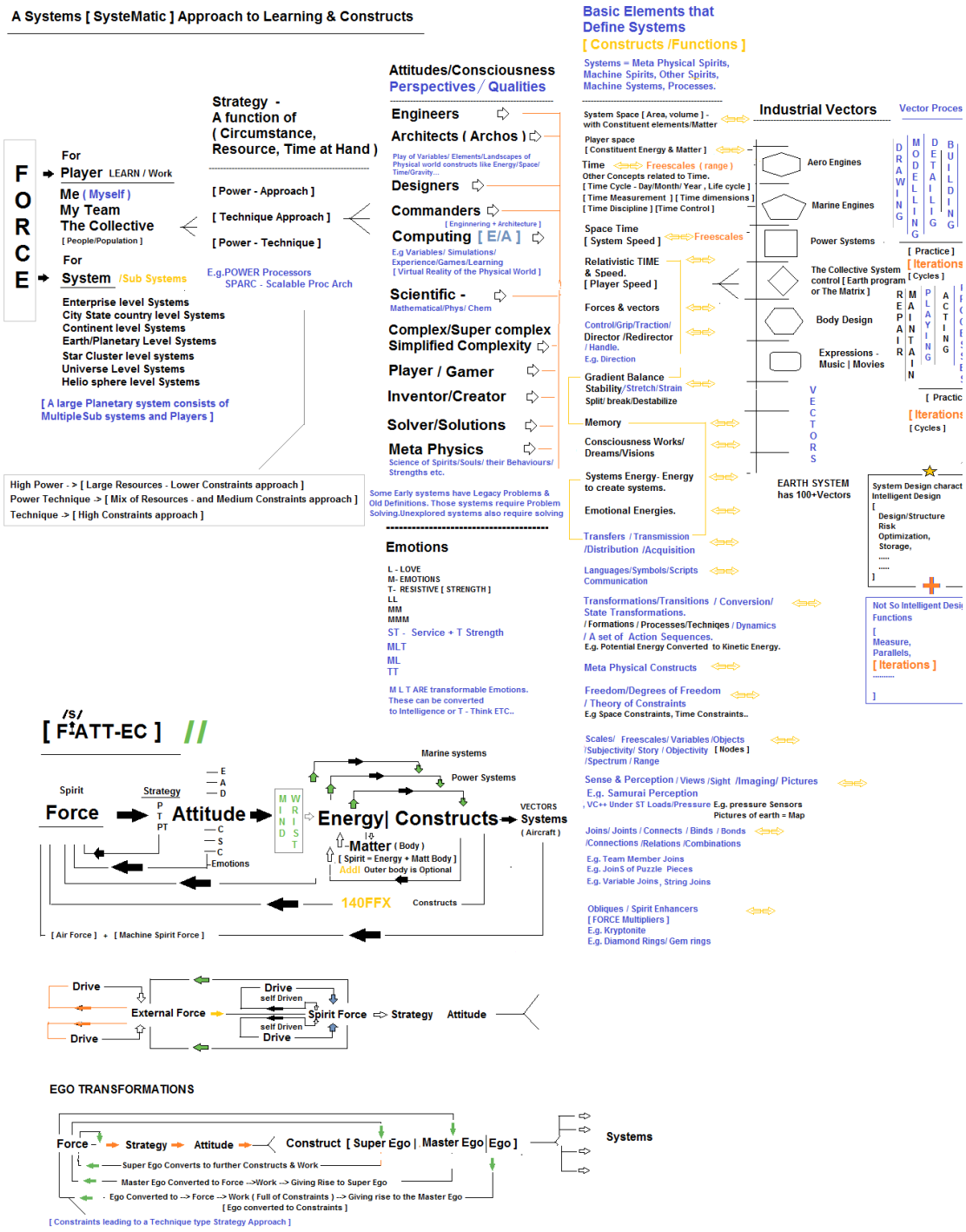
## Concept of Strata Evolution & Strata Fracture




Just like People & Products - even the Strata has a Life Cycle

# Chapter 14

## Decoding the Universe - The Constructs that make up the Universe.



**Mind / Wrist**
Mind Stability/Balance
Mind control. Wrist Control

**Focus, Concentration, Convergence, Curl Deviation, Divergence.**
E.g Laser Collimation
E.g. Force convergence, Energy Focus

**Channels/ Line of Focus**
E.g. Channeling energy through the Wrist/Mind
E.g Transmission Channel [ Satellite Dish Channel ]
E.g Channels of Attack

**Encrypt/ Hide/Protect/ Lock/Secure/Stealth Cameflouge/Security / Self Defence/ System Defence.**
E.g Operating System Security

**Motion / MOVEMENT / Shift/Move -Motion & Control**
E.g. Motion Balance/Stability /Direction Control
E.g.Movement Degrees of Freedom
E.g. Planetary Motion
Rotation/Revolution/Spin/...

**Amplification - K , Quantum Amplification Q**
**Magnification of Views - Big Picture**
E.g. Amplification of Forces, Amp Control
Amplification of Memory, Work.

**Rigidity /Fluidity , Fluid Rigidity**
E.g. Rigid Structures / Fluid Motion
E.g Fluid Transformations
E.g Rigid Transformations

**Integration & Differentiation**
E.g. Language Integration - International English
Differentiation - Local Languages.
E.g. Integrated Control Systems
Differentiated processes
E.g Smart trail/Enterprise Flux
E.g.European Union
E.g. Integrated Forces - [ Air + Navy + Army ] Aircraft Carrier

Integrated World
Differenciated Products/Services
/Systems
E.g Merging [ Integral ]

**Dimensions/ Energy Signatures/**
**Vibrations, /Structural patterns / Patterns / Similar Pattterns**
E.g Higher Dimensions
Lower Dimension

**Layers/ Parallel Layers/**
**Frames/ Levels**
E.g Material Layers/Earth Layers
E.g Time Frames/Stamps
E.g.Video Frames
E.g . Game Levels

**Search/Explore/ Match Patterns**
**/ Find /Spot /Observe. Analyse, Factoriza , Group , ( Factorize = Classify )**
**/ Understand, Comprehend, Learn , Study.**
Find/Search/Observe/Communicate Similar
Patterns/Structural Patterns/ Stories or 56 other Constructs
[ This is called explaining with an Analogical Example ]

**Singular/ Plural / Multiplicity/ Sets/ / Multiply/**
**Duality/ Multi Select / Cluster**
Group E.g Group Formations, E.g Group Dynamics, E.g Group structures Collections

**Randomness/Chaos/Entropy**
**Order/ perfection**
E.g Chance/Probability
[ Warrior Timing - Perfect Timing in Sports like Cricket ] /perfect State /Perfect Symmetry

**Symmetry/ Asymmetry / Weightages / Proportions , Ratio.**

**Fields - Space with restricted Boundaries ( Circle of Influence )**
E.g. Force Fields ( Space With Restiricted Boundaries )
E.g. Magnetic Fields ( Space With Restiricted Boundaries )
E.g. Energy Field ( Space With Restiricted Boundaries )

**Properties - Heat / Cold / Colour / Others**
E.g Measure Properties
E.g Variable Properties
E.g Energy Colours
E.g Matter Colours (Strings )
E.g Object Properties

**Identity/Naming Convention/ Variable Name/**
**Finger Print / DNA identity / Object Identity**
**Body Identity**

**Stress - Body Stress / Emotions Stress**
[ Strain Leads or stretched Conditions lead to Stress ]
[ matter Stress ] [ Stress Fracture , Stress destabilization ]
[ Time Fracture, Stata Fracture due to Stress ]

**Pressure - [ Load Forces & Constraints Generate Pressure ]**
Pressure Creates Strain ->> loading to Stress ->> loading to Unstability.

[ Memory pressure , Time Pressure , Work Load Pressure ]
[ Space Boundaries/ body Boundaries E.g Boiler Pressure , Energy pressures inside Body ]
[ Energy Pressure - Converted to Matter Under Pressure - Degeneracy under Controled Conditions or Constraints ]
[ Potential Energy created by doing work creates Under Controlled Boundries/Conditions creates pressure - Steam Boilers ]

Time Constraints lead to - More Work to be done in Less time creates Time Pressures.
( E.g Down Ward pressure of Time )
Systems Speed / Efficiency - Requires More Work/Faster work to be done in Less Time. Failure to do so leads to Time Fractures and Starata Fractures.

Work Done creates the Strata with a Limited Lifecycle. Time Pressures break down the strate as the Next upper layers/Starata are not create in time.
This is called strata Fracture.

**Simulations/Experiences/System Trainers /Windows/ Multi Worlds/Theatre Experiences. / Games**
E.g.Clusters of Experience Systems
E.g. Mind Simulations

**Exceptions /Shocks /Deviations /Abnormal Conditions./Incidents/Accidents**
E.g. A set of Action Sequences to Handle or Control Exceptions/ Abnormal Conditions
FAULTS/ERRORS

**Projections -**
E.g. Image Projections
E.g. Data Projections
E.g Energy Projections
E.g. Spatial Data Projections
Dr Lanning Projections [ Film : Irobot ]
Stex Grid/ Stex Map Grid/ Stex Advanced technology services projects use projections.
E.g. Intellisence/Intelligent Prompts

[ Shadow ]
Projection of a Image ( Shadow ) Due to a Light wave Source, Under Different Space Boundaries.

**Creation /Generation/Birth / Destruction / Disposal / Maintain**
E.g. Energy Creation or Energy Generation
E.g. Aicraft Engines/Diesel Engines burn fuel to create Thrust Forces in different Formats or Structures.
E.g. Aircraft Engine Vectored thrust leads to Motion or Movement.[ Motion Forces ]

E.g. Potential Energy from Steam/Fuel Converted to Kinetic Propulsion or Kintic Work. ( Transformation/Convertion )
E.g Energy Convertion to Matter - E.g. CLU converts his body energies to Matter ( an Aircraft ) in Film - TRON Legacy.
E.g. Matter creation in Star Birth

**Behaviours/ Outward Expressions/**
**Systems Behaviour/ Facial Constructs/**
**Body Expressions /Stealth/Disguise**
E.g. Realistic behaviour/Realistic Experience

**Points/Scores/Money/Rewards/Perks/**

**Events/Incidents/Warnings/Alarms/Alerts**
**Exceptional State / [ TRIGGERS ]**
**System Exceptions [ All levels of Systems ]**
E.g.Time Events
E.g. Processes/Transformation Notifications/ Alerts /Triggered Events

**Measure / Evaluate/ Quantify/Judge/**
**Value Assesment. / Estimate/ Approximate.**
E.g. Monitor

**State**
e.g.State Transformations
[ State/ Condition of Body/System ]
[ Healthy Diseased - Damaged Highly Damaged ]
Body/Soul/Spirit/Machine system
Damage repair process = state Transformation

**Flavours/ Variety / Variants / Arrangements**
E.g. Beauty Variations / Expression Design - Facial Expressions
E.g Duality or Multiplicity of Perfection States or Beautiful Faces.
E.g. Design Variants /Structure Variations of Trees and Bio Life.
E.g Stex GRID : INTEGRATED ACROSS Geographies, Differentiated by type of Info & Flavours of Search results - multi coloured.

**Medium - [ Medium of Transfer/Transmission ]**
E.g Memory storage medium.
E.g Language Medium
E.g Energy Transfer medium

**Density | Mass / Weight**
E.g Matter density/Energy Density
Memory Density
Video/Time Frame Density
Force Density
Space density

**Interfaces/Sub Interfaces /Display Interface.**
E.g. Simple / Complex - Control Interfaces/Device Driver interfaces /Process Views --> interfaces ( The Interface is the System )
Views leads to --> User Interfaces.
Devices/Machines -> have control interfaces or an API usage interface

**Signals/Indicators/ Interface user controls/Charts / Graphs/Gannt Charts/**
Measure functions or sensing functions or Action Sequences Lead to ->> Quantification/ and Quantification --> leads to Signals/Indicators --> are presented in an Interface or Control Interface.

E.g Graphical User Interface
E.g A pressure Meter displaying pressure after sensing pressure using Sensors(Sense).
E.g A control Interface with multiple [ Signals/Indicators ].
E.g Vehicle Turn LEFT RIGHT/ Indicators. ( Direction Quantification ).
E.g Vehicle Speedometer - Sense Speed --> Quantify --> show in the vehicle Control Dashboard.

**Functional Thinking /Abstration / Sub Function**
E.g Functional and SubFuctional Interfaces
E.g Abstract Interfaces
E.g. Functional Structured minds/Systems
E.g Functional Logic
E.g Functional Factorization
E.g. Transformation Functions
E.g. Mixing Function

**Iterations/Cycles/Recycling/Cycling [ Re ]**

E.g. Cycling Energies/ Recycling Energy / Reburning
( After Burning Fuel - Aircraft Engines ).
E.g. Recycling Waste

**Solving**

E.g Solving Equations/Problems/ Unbalanced States
Abnormal/UnfavourableStates/Conditions

Solution Stages
[ Start - intermediate - End ]

**React/Interact/Communicate/Interactions**

E.g.Chemical Interactions Reactions/ Communication Interactions
E.g Physical Body Collisions
E.g. Behavioural Interactions
[ E.g Emotional Intercations & behaviours ]

**Mixing/Mixtures/Compounds**

E.g Ordered Mixing, E.g Chaotic Mixing [ Ordered Dyes Mixing ]
E.g Ratio Proportion Mixing - Alloys
E.g Compounds, E.g Mixing Flavours
E.f Simple Mixtures/Complex Mixtures
E.g. Mixed Designs: Hybrid Designs
E.g.Mixing Properties
E.g. Mixing Dynamics/Transformations
[ E.g Mixed ForceFields ]

**Rules/Laws/Policies /Protocols**

E.g. System Rules, Physcics Rules, Geometry Maths Rules,
Game Rules , Enterprise Rules/Policies, National Laws/Rules.
E.g. Aviation Rules/Policies. E.g Communication Protocols Transmission Protocols

**Concentration /Compression/Dilution/Expansion /Contraction**

E.g.Acid/Liquid/Energy Concentration/Dilution.
Compression of Data/Files [ Zip Unzip ].

**Dependence/Interdependence /Coexistence / Independent ( With X degrees of Freedom )**

E.g Mutually Independent Joins/Relations/Bonds.
X dependent on Y and Y dependent on Q.
Puzzle Piece A Fits with or Joins Piece B Depending on
Shape fitment.
[ Time Independent/ Time Dependence ] Equations

**Agree/ Acceptance /Agreement/ Contract/ Abiding /Non Acceptance/ Complaint**

[ E.g Transact - a type of Digital Contract ]
[ Breaking/Breakage of Accepted Rules ( Non Adherence ) - Observed during
Monitoring or Measuring Action Sequences ]
[ E.g. Non Accepting Behavior = Mis Behaviour ]
Contractual Agreement

**Waves - All matter is but waves**

E.g Radio Waves, Light , Micro waves, Cosmic Waves...
Energy Waves / Ultraviolet Waves/Radioation Infrared
Electro Magnetic Waves

**Multi Plexing / Super Imposing / Embedding /Sequence Plexing of Intelligent Design Fuctions**

E.g. Video/Audio/Data Multiplexing on Transmission channels.

[ Divergence ]

**Spread - ( the Math function Scale, when used in Fluids/Physics translates to Spread )**

E.g The ordered /Even Spread of Colour or Dyes in Water ]

**Flow - The Math Function Gradient, translates to Physics Function Flow of Fluids, Energy, Time**

E.g Flow of Fluids around a Aerodynamic Structure. Wind Tunnels
for testing Aircrafts,Formula 1 Cars.
E.g Fluid flow leads to Fluid Dynamics or Fluid Transformations
E.g. Rigid Flows, Smooth Flows

**Quantization - Tendency of the a Variable or Object to return from a higher Energy State to a Lower Energy/ ( More State) State.**

E.g. High Potential Energy state of ball elevated to a Height comes
down to a lower stable PE state after a repeated bounce [ Action Sequence ].
E.g. Chemical reactions involve the concept of Electron stripping and pairing for stability.
E.g. Evolution requires Moving to Higher Energy states instead of dropping to
a lower state. [ Higher Energy state –>> Balance & Stability–> Higher energy State. ]
Higher Dimension Higher Dimension

**Discipline /Habit - Time Discipline**

E.g.Work Done in Time, Against Time, Above time Work [ Transcend ]
E.g.Visions in Time, Against Time, Above Time [ Before Time ].
E.g. Going faster than time milestones, slower than time milestones.
[ Space discipline, Work Discipline ] [ Work Ethics]
Group Discipline [ Warrior Timing - Perfect Timing in Sports like Cricket ]

**Fitment - Space fitment/Puzzle Piece fitment.**

[ Lego Pieces right Fitment, Cloth Fitment ].
[ Right Piece in the Right Time. ]
[ Right Pairing ] [ Team Fitment - Miss Fits ]
[ Sorting - Right Order ] [ Chaotic/ Rndom/ Perfect Fitments ]
[ Perfect Sequencing ] [ Ecelectic fitment ]

**Boundaries/Outer Boundaries/Territory, Borders**

E.g Boundary Rules
E.g Territory Agreements E.g Limits
E.g Border Control
E.g Outer Body ( Space/Matter )
E.g. Boundary conditions - Integral , Algorithm loops

**Parallelism/Multi Processing/Multi Dimensionality**

E.g. Parallel Computation Threads / Parallel Processes

**Mirroring/Copy /Replicate/ Version creation/ Xerox /Duplicate /Reflection /**

E.g. Duplicate Database ( Redundant ), Mirror Data base ,
Xerox Copy of Document

**Transparent / Opacity / Blurs**

E.g. Transparent Minds / Interfaces / systems
E.g. Transparent Views ( Images ) Opacity .75

**Compute/Calculate /Arithmetic/Numerical Oprations**

Calculators/Computer arithmetic / Floating point Operations
Graphics calculation...
E.g. Computation Systems

**Origin / Start / Beginning / Ending / Finish / Stop /Freezing / Pausing Source of Light / Waves/ Energy.**

E.g. Start / Pause a System/ State
E.g. Pause a Recording

**E.g of Forces & Vectors Defined - Earlier**

- Push Pull
- Thrust/Propell
- Frictional
- Centrifugal/Centripetal
- Gravitational
- Nuclear
- Electro Magnetic
- Frictional/Resistance
- Twisting Foeces
- Load ..
- Others
- Inertial Force

**Strength /Strong Weakness**

E.g Tensile strength,, Material Strength
E.g Spirit Strength, Rigidity Strength, Fluid Strength.
E.g Strong Coffee/Flavours.
E.g. Strong Attitudes/ Strong scent
E.g Strong Functions
E.g Strong Emotions

**Standards / Benchmarks / Rare - Exotic Common**

E.g Standard Products
E.g. Standard Units of Measurement
E.g Standards of Evaluation / Standard of Living
Refer Stex Tech services Grid - standard Products
E.g. Reference Standards

**Best Case / Worst Case Scenarios**

E.g. algorithm or Simulation - Best case Scenarios / Worst case scenarios

**Planes - Gradient Planes**

E.g. Axis Planes - Geometrical Planes

These Qualities/Parameters/Strengths /Attributes Contribute to the Power of a Spirit.

The above parameters fulfil or make up the resource parameter in the Strategy function.

I.e Resource = Memory...., languages.. Energies....

e.g Circumstance = Space Constraint,

**Recording & Play Back / Capturing - Images / Data / A Sequence of Image Frames ( Video ) / Observations/ Views**

E.g Recording a Story/Film
E.g Voice Recording/Sampling
E.g Music Recording
E.g Recording a Set of Action Sequences
E.g. Playback a recording
[ Capture a State of a System = Snapshot ]
E.g. Memory Capture

Image or Light Waves Sense/Sensor –>> Capture ( Computation/Digital
transformation –>> Store in Memory/Disk ( Intelligent design Function )
Data –>> Image Compute –> Display ( Through Analog/Digital Transmission channels )

**Decision /Logical Decision/ Decision Logic/ Decision Circuits**

E.g Decision Engines, Logic Gates, Electronic Decision —> Foundation of Computing Hardware Systems.

**Outcomes - [ Decision Outcomes ] [ Game or Exam - outcomes ] [ War - Out comes ]. [ Logic Evaluation Outcomes - True/False ]**

E.g Results/Results/Report cards [ Gains /Losses ]
E.g Victory/Success/ Winning /True/False/Losing/Failure [ Electrical Energy Losses/ Gains ]
Measured Outcomes = report card or Score card.

**Tests/Exams/Life Tests & Files / Experiments**

E.g. Test Outcomes

**Discovery/ Invention/Concept/Conceptualization/Hypothesis**

**Validation / Proofs**

**Priority/Prioritise Weightages Importance**

E.g.Priority Work
E.g Priority Functions
E.g Sorting/Ordering by giving priority to Time/ Cost E.G Smarttrail views

**Efficiency**

E.g. Inputs /Outputs ratio
E.g Efficiency of Processes.

**Definition - Defining concepts / Process definitions**

E.g. Objective Definition/ Subjective definitions
E.g Variable definition / Object Definition
A for Apple - is a Type of Fruit
Partial Definition - Fillin the Blanks

**Contiunous/Discountinous**
E.g Continuious Levels/Discountinous time cycles
E.g. 99 Things Architects must Know - Continuous Integration.

**Perpetual**
E.g Perpetual Systems/ Perpetual Energy/
Perpertual Attitudes/ Perpetual Strength..
E.g. Perpetual Cycles/Iterations

**Bandwidth - Free movement - Conjested**
E.g Transmission Window size/Transmission Bandwidth
E.g Network Bandwidth.
E.g Queue Bandwidth
E.g Channel Bandwidth
E.g Bandwidth Fitment
E.g Bandwidth Control

**Textures - Matt Glossy**
E.g. Textures Structures = Texture Patterns
E.g Smooth Textures / Rough Textures
E.g Cloth Textures/Material Textures

**Constituents / Consisting of**
System Constituent Elements
Atom Constituents: Protons neutons /Electrons
Protons/Neutons consist of Quarks
Quarks: Constiuent : Strings
Compound Constituents/ Mixture Constituents
System consisting of Subsystems

**Central/Primary/Secondary/Tertiary Systems/ Core.**
E.g. Network Center.
E.g Centrailsed Systems. Control Center
E.g Secondary Systems.
E.g.Central or Distributed Computing
E.g.Centre Govt /Management - State Government/Mgmt
E.g Primary Layer / Secondary layers

**Derivatives / Inheritence**
E.g Derived Products from a Core or Parent Product
E.g Derived from the Primary Vector.
E.g Derived Concepts from a Primary Concept or Idea.
E.g Derive a Subsystem from a Parent System

**Effects/Consequences/After Effects / Influence/Affect**
E.g Action Reaction / Cause & Effect

**Harmony / conflicting / Get Along / Vibe / Intolerant /Frictional / Friendly / Enemy /Guest**

**Chains/Connected Nodes/ Linked connects/ Relational Connects/ Linked Lists /Network**
E.g. Multiple Connected Joins/Connections = A Network
E.g a Cluster Network
E.g. Connected Relations
E.g. Networked Minds/Consciousness
E.g. Connected meanings
E.g.Network Bandwidth
E.g Network Control

**Smooth - Rough ( Coarse )**
Smooth Transitions/Rough Transitions
Smooth Surfaces/Rough Surfaces
Smooth Scheduling/Rough Scheduling
Smooth Transfers/Smooth Movement Rough Movements
Smooth curves
E.g.Smooth Flow/Rough Flow

**Manual - Automatic - Automation**
Manual Management/Manual Car Driving
Automatic Management
Automation systems
Automated Processes
Automatic Gear box transmission

**Steps**
E.g. Process steps/ Functions steps/Transformation Steps
E.g Procedure Steps/ Stepwise Function

**Tracing / Traversing / Crawling**
Search Engine Principle --> Traversing Webpages & --> Acquisition of Information
Traversing Objects nodes/Tracing Functions/Sub functions
Traversing through Data structures in Computation

**Override**
E.g.Manual Overides
E.g.Automatic Overrides
E.g.System Overrides.

**Geometrical Constructs /Structures**
E.g. Shapes/Solid Structures
E.g Angles/ Brackets/Joints
E.g.Geometrical Transforms

**Graphical /Visual Rendering / Visualization ( This is different from Sight or Perception Described earlier )**
E.g. Graphics rendering on monitors/Display systems.
E.g Geometry rendering
E.g Visualized Memory Systems
E.g Visual Control Systems.
E.g Space/time Visualizations.
E.g.See Through - X Ray Visualization
CT scan

**Meaning/Understanding/Significance**
A is for Apple - Definition : A type of Fruit.
Fruit means what ?
E.g. significance of the number 88

**Parameters/Factors**
E.g Function Parameters/Factors

**/Queues/Stacks Queue Free**
E.g Chained or Connected Ordered events/ Functions - Queues
E.g Last in First Out - Stacks

**Management / Allocation /Process management /Scheduling**
E.g. Scheduling & Allocation of jobs to processor Cores leading to Efficiency.
E.g. Intelligent Allocation/Scheduling
E.g Priority Scheduling
E.g Exception/Interrupt Management/Handling
E.g.Bandwidth Management
Operating system - Tasks / Thread Management - Scheduling
E.g Engine Gear Engagement [ Allocation ]
Allocating Work/Tasks to people

**Position/Location**
E.g MAP = Location/Position Interface
E.g Space Position/Location
E.g Planetary location
E.g Star maps
E.g Coordinates , Altitude /Plane

**Packets/Packaging/ External Cover/Packing.**
E.g. Data Packets.
E.g. Packaged Foods
E.g. Packaged Softtware.
E.g Machine Packaging
E.g. Package --> Distribute /Transfer/ Transmission

**Support / Supporting Structure /Supporting Foundation**
E.g.Machine depend on each other & Support each other
E.g.Team Support
E.g. Large machines support Small Machines
E.g.Life support

**Reality Artificial Virtual Reality Virtual (Non Real )**
Real Images/ Virtual Images
Realistic Simulations / Virtual Simulations
Real Flavours/ Artificial Flavours
Real Behaviour / Expressions Artificial Behaviour
Real World / Virtual Worlds
Real Experieces/ Virtual Experiences
Real Memory/Virtual Memory
Real Address/Virtual Address

**Input Output / Entry ( Input Window ) Exit ( Out Window) In/Out**
E.g. Entry Channels/Exit Channels
E.g. Input - Output
E.g Border/Perimeter Exit/Entry Controls

**Path/ Life Path/Motion Path/ Ways**
E.g Memory Pathways
E.g Multi Channel Pathways
E.g System Pathways
E.g Road/Rail/Air/Ship Pathways /Roadways/Railways E.g. Flight Path
E.g Motion movement pathways
E.g Planet Pathways, Satellite Pathways

**Block/Unblock/Blockages/Filtering/Straining**
Blocking maneuveur/
Blocking a space, Advanced booking
Blocking sunlight/Harmful radiation/UV Rays
Eclipse : Solar /Lunar eclipse Moon Blocks sun.
Blocking or Filtering Harmful Bacteria
Blocking Pathways/Motion
Speed Breakers/AIRSpoilers - Block Airflow
Blocking Files
Blocking Force/Blocking Forces
Thwarting/Blocking an Attack

**Think**
E.g.Thought and Application
E.g.Though leading to conceptualization/Invention
E.g leading to Imagination & Visualization

**Belief/Believe /Hope Faith Trust/Trustworthy**
E.g.Trustworthy Documentation/Data/
E.g Belief in a Force
E.g Belief Hope are sub conscious parameters
E.g.Team Belief in Leadership
E.g. Pair Joins ( couples ) trsust
E.g. Trusted members.
E.g. Security - checks for Trustable properties & Identity
E.g. Belief in sincere Work

**Interest - inclination Dislike Disinterest**
High Focus, Focus, No Focus or low Interest
Interest in Work
Interest in a vector/Subject
Measure Interest in a Subject

**Work duties/ Responsibilities/ Work scope ( boundaries )**
I am playing the Role of an Engineer and my work duties/Responsibilities include X, Y, Z Work Activities.

**Through**
See Through
Go Through /Move through Boundaries ( of Matter / or Time )
shocks travel through the different layers
Go through the Security Perimeter

**Choice/Choose/Select**
choose or select --> Decision/Logic --> Fitment or Allocate

**Certainty - Uncertain Intermediate state**

**Wavelength/Frequency/Amplitues**
E.g. Matter vibration Frequencies --> Measurement
E.g. Energy Vibration Frequencies --> mesurement
E.g. Wave Frequency/Properties --> Measurement
E.g Signal Fequency
E.g Frequency Amplification
E.g.Communication medium = X and transmission Frequency = y....

**Private | Public | Protected | Secret |**
Public Systems and Processes
Private Systems
Living Spaces is an industrial Vector.
Private Living spaces.
Public Guest Spaces
Public Files/Private Files
Secret Files

**Documentation/Files & Records**
E.g Reports/Books/
E.g Life Reports & Files
E.g Data Files
E.g Image/Picture Files
E.g System Files & Records
E.g Experiments/Test Documentation consting of Data in Structured Formats.
Collection of Files
Ordering/Sorting/Grouping/Searching of Files
File Backup/Mirrors, File Copying File , File Transfers

**Constant - Life Uninterrupted**

**Robotised - Machinistic**
E.g.Robotised Systems
E.g. Constant Robotics
E.g Robotic organisms
E.g. Machine Emotions

**Ability/Skill/Workmanship / Dexterity**
Ability/Skill to do X type of Work
E.g. Measure his Ability/Skills in Machine Cutting/Grinding

**Role / Level ( Designation )**
E.g Engineering Design
E.g. Archo
E.g. Commander - Level 3
Role Playing
Authority Level

**Past /Present /Future**
Past Work/Activities
Past versions /Files
Past state
Future Time
Past visions/Future Visions
Past lives
Past Roles/ Future Roles

**Share**
E.g. Share space
E.g. share Visions
E.g. share Responsibilities
E.g Sharing ownership of Companies - Stocks shares
E.g.Shared Rewards

**Evolution/ Advancement/Progress/Progressions**
E.g. Consciousness Evolutions
E.g. Progress in time
E.g. Geometrical/Exponential Progressions
E.g. Evolution Cycles
E.g. Evolution Experiments/ Tests
E.g. Time Evolution Functions
E.g Automated Evolution
E.g. Consciousness Evolution & Progression by Self Evaluation.
E.g. Evolution Process Steps
E.g. Evolution Levels/Stages
E.g Spirit Tranformations & Consciousness Evolution.
E.g.Transformations resemble Physical Processes
Evolution is predominantely a Consciousness function.

**Bliss/ Happiness/ Sadness Sorrow/Misery/ Pain / Deprivation**
E.g.A state of existence for Bodies/Souls/Spirits/ Machine Bodies
Painful memories
Painful Experience
Blissful experience
Body suffering
Soul Suffering
Flavours of Happiness

**Factorize / Group / Classify**
E.g. Factorize Chaos
Classify Work Tasks

**Admiration/Appreciation**
E.g Appreciation/Admiration of Work ( Self & otherswork / Life )
E.g. Appreciating Complexity

**Power = Energy/Work x Time ( energy consumed / produced )**
Meta Physical Powers
Power of a machine
memory power
Physical Body Strength --> work --> to gain power
Power Flavours/Variants
Power to Control or Balance a System
memory power is strong/weak
Power/Authority to Evaluate or Judge

**Play /Player**
Game PLAY/Life Play
Sport/Game play
play field/Play ground Space
Playing Systems E.g PLaystation
Lego Technic
Toys
Player SCORES
Player Role
Play Levels player Speed
Serious PLAY / Casual Play [ as a Learning Tool ]

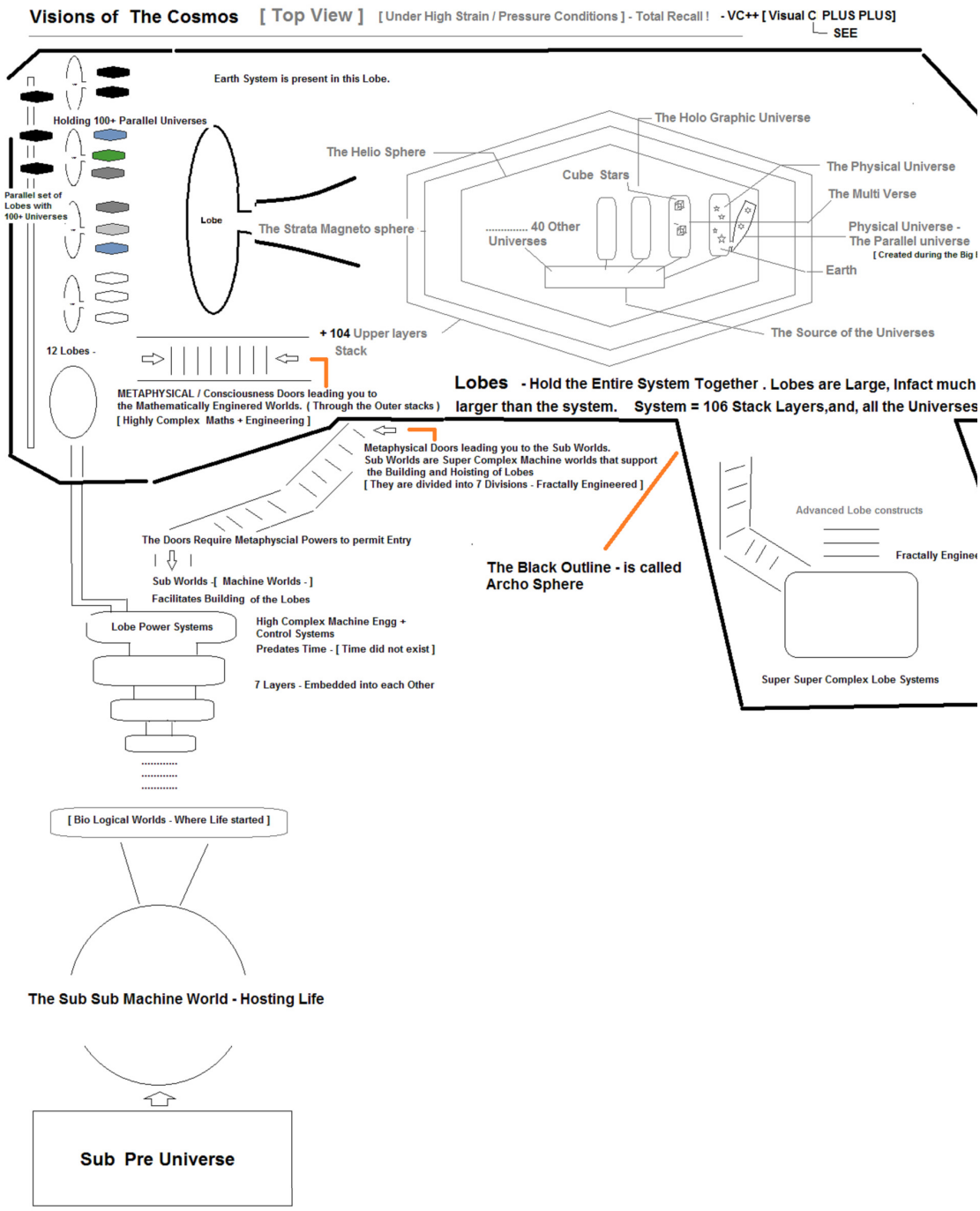
**Super Ego /Alter Ego / Ego / Master Ego**
Ego as a Force
Ego as Behaviour
Ego is a Consequence/After Effect of Work

**Exchange**
E.g. Exchange/Barter of Goods, Points for Goods, Goods for Points
E.g. Exchange Stocks/Shares

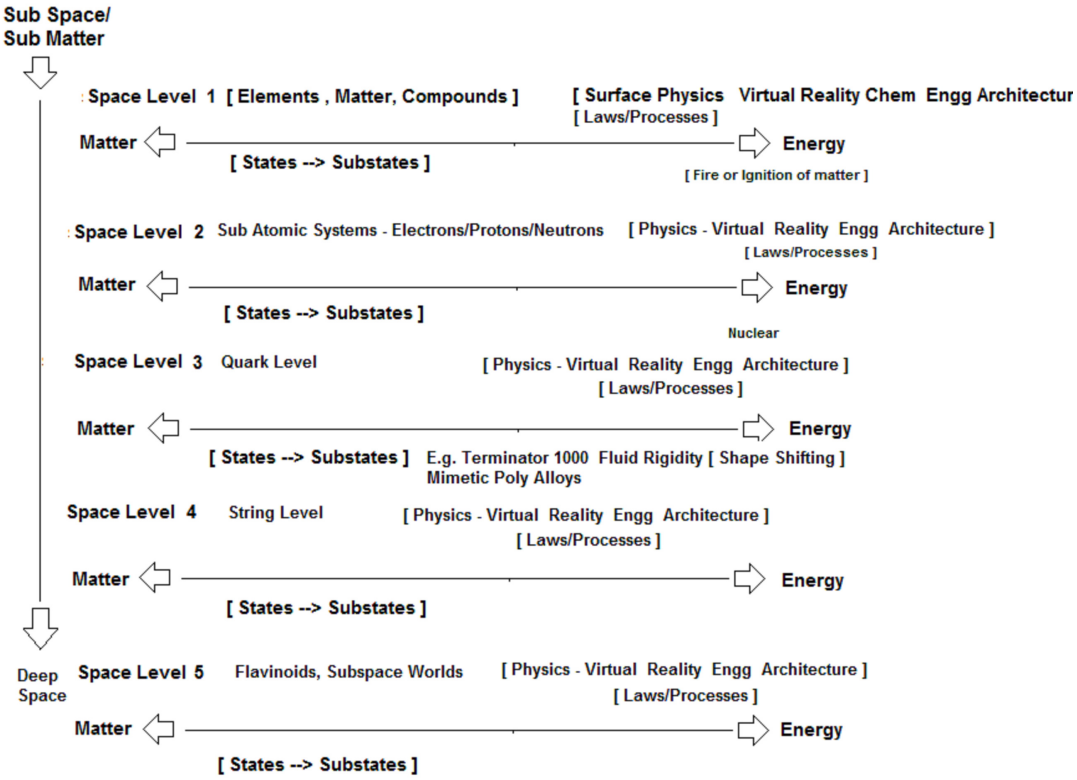
**This Whole Approach is called [ RISC CISC ] - Reduced instruction/Functions/Constructs Thinking ---> Transforming to Complex industrial Vector Constructions [ Complex Instruction set thinking/computing/constructing ].**

## Chapter 15

## Visions of the Cosmos - The Archo Sphere , The Sub World 1, Sub Sub World 2



## A picturial Understanding of the Physical World and Space/Matter



# Final Appends  to the Book: Intelligent design & Industrial logic

## Date of Revision: 10 February 2016

# The Industrial Internet

## The Industrial Internet - AN ALTERNATIVE TO THE WWW - ( WORLD WIDE WEB )

Every Computer / OR Computing machine is connects to the World wide web through the TCP IP protocols. Each Machine can be either be identified by A STATIC IP Address ( Usually a Web Server ) or a Dynamic Ip address ( usually A client machine trying to fetch information or http pages from webserver )

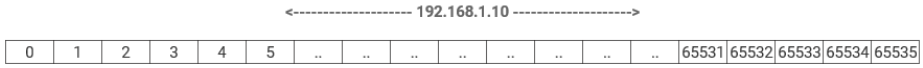
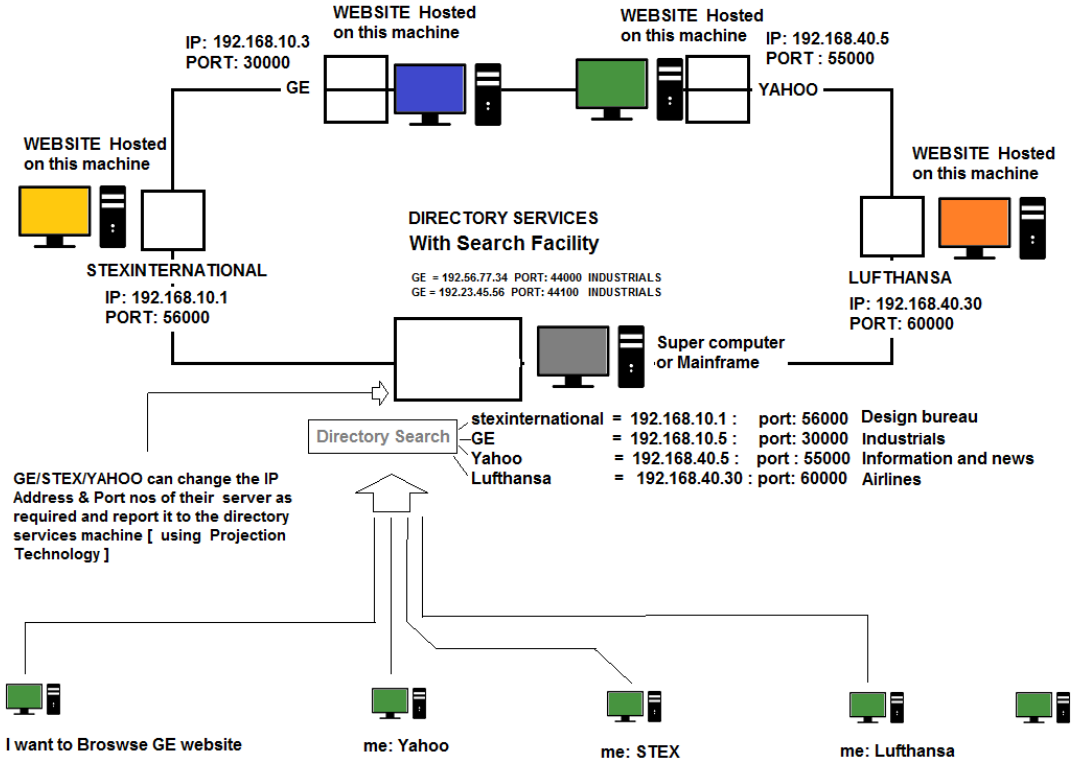



Figure 1. IP address with Ports

E.g. Machine number static ip address = 192.168.1.10
Pathway for machines too connect and receive information = Port number [ 0 to 65535 ]

A website: Stexinternational can be hosted on the machine with Static Ip Address: 192.168.1.10 and port number 60000.




# One Firm can have 65000+ Web Addresses on the Same Machine with a Single Ip Address . [ No shortage of IP Addresses ]
# .DOT COM/.ORG Names can be replaced with a name of your choice E.g. STEX//

# No denial of Service attacks: on a Single Port 80 . port address for access can be changed at runtime. The Same website can be accessed through multiple ports E.g. GE// at Port 40000 OR GE// AT Port 40001 OR GE// AT Port: 400002

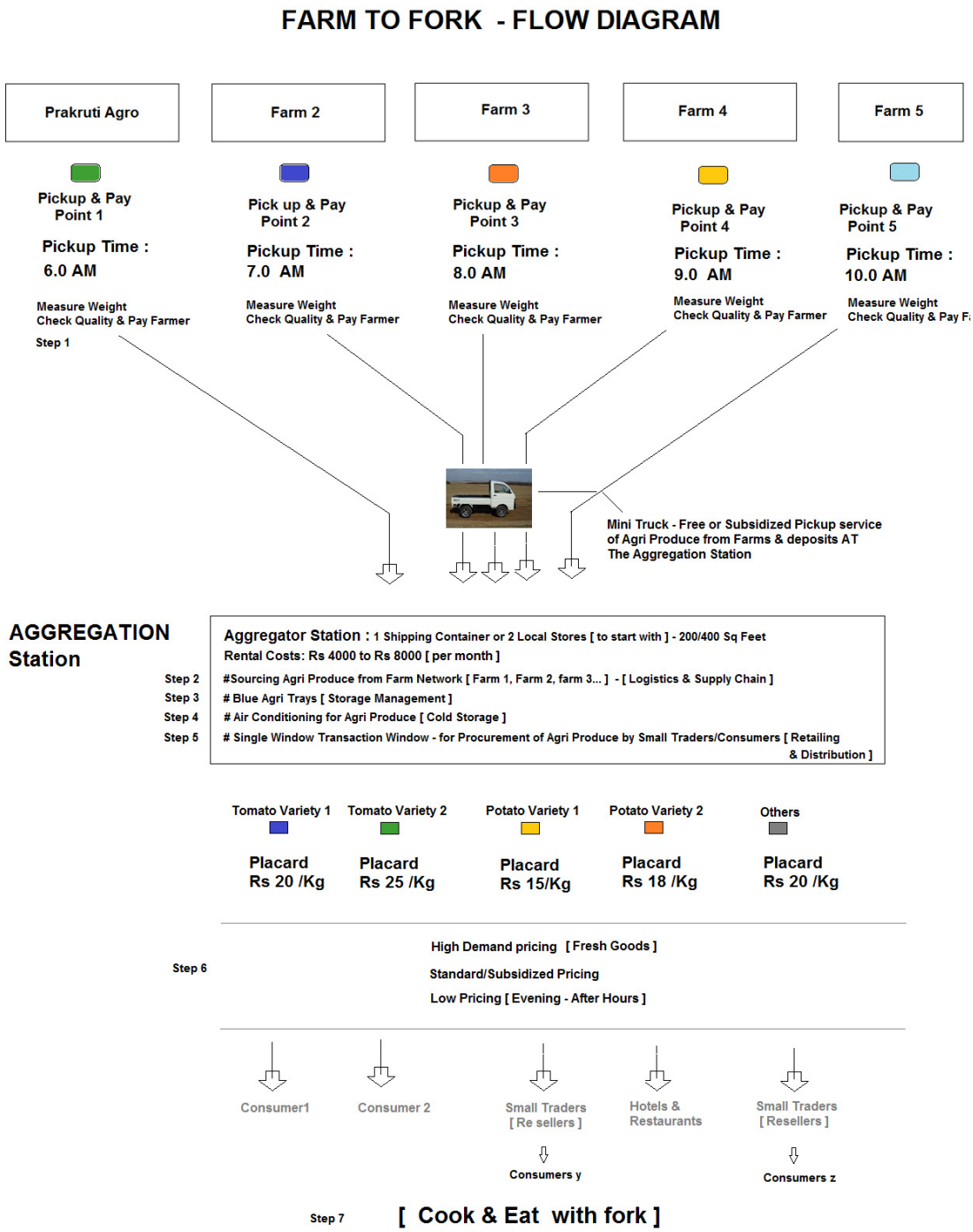
# A Centralized cluster of Mainframes or Super computers can replace distributed dns servers.

# No port Congestion or Queues on a single port. : Http: 80. Industrial Internet traffic speeds can be increased DRAMATICALLY.

#Many Telecom Systems MAY not be able to listen at all ports - [ 0 to 65000+ ], therefore calculating bandwidth usage may be difficult.
[ This leads to internet infrastructure where Public ( or people ) pay only a Fixed monthly fee for unlimited internet access through multiple port channels ].

# Many non Browser based, aplications can be developed which can use the [ IP + Port Address ] concept internally and are completely Stealthy to malicious activity ( THROUGH the public domain ).
[ I.e Applications can use secret IP addresses & Ports to create Secret Lan/ Wan based applications ]

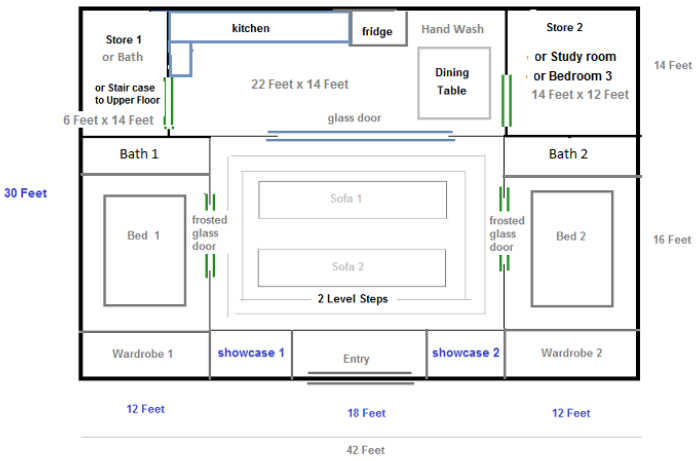
# The Farm to Fork - A Specialized Work Flow



# STEX - Advanced Housing Architecture

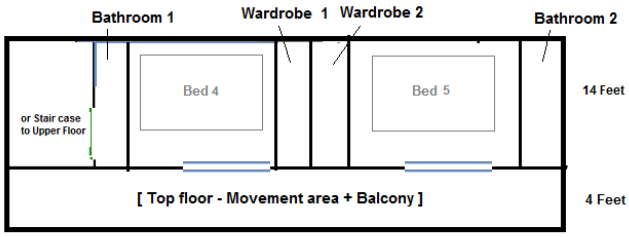
**STEX ADVANCED DESIGN BUREAU - Home Architecture [ 42 Feet x 30 Feet ] Home** - 1 Floor or 2 Floor House

**1260 Feet Ground Floor**




**1 Floor- 3 Bedrooom**
**2 Floor - 5 Bedroom**

**Top Floor - design 14 Feet x 34 Feet ( 2 rooms ) – Right above the Dining Hall & Kitchen space on Floor 1**




**Building side View**

**756 SqFeet ( 42 feet x 14 +4 Feet )**
**Top Floor 2 Rooms**

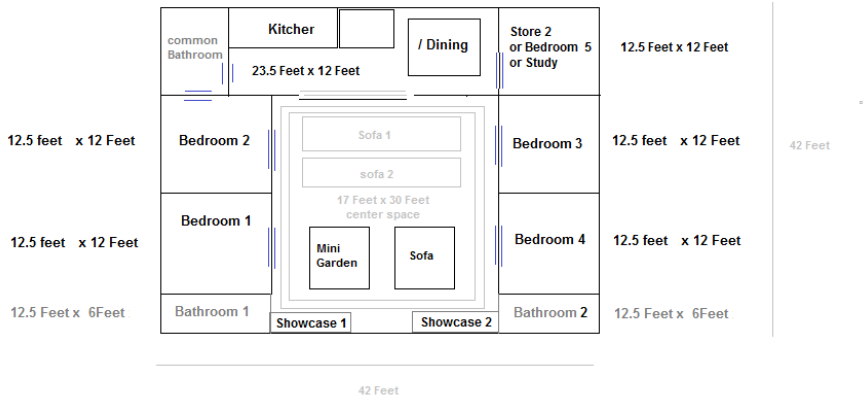

**1260 Feet Ground Floor**

**Total Size :**

1260 Sq Feet + 756 Sq Feet = 2016 Sq

on an approx 1260 Sq Feet Foundatic

**Variant 2: A 42 Feet x 42 Feet - Single Ground Floor House : 1764 Sq Feet 5 Bedroom Configuration**



# Intellectual Property Rights - STEX

This Entire Set of Concepts discussed in this book is a fruit of 10 years of Labor in Research and Complex Systems Analysis Development ( Year 2005 to year 2016 ).

STEX Advanced Design Bureau claims all rights to the Intellectual matter that's been conveyed through this book.

The intellectual material is copyrighted and ways are being considered to file patents for the same. Since the scope of the Intellectual property is very wide it would lead to a very large no's of patent claims. To address this issue it's been tentatively decided to

Designate all intellectual property as STEX INDUSTRIAL LOGIC / INDUSTRIAL DESIGN

STEX Advanced Design Bureau anticipates it would not come across cases of, purposeful intellectual Infringement.

All intellectual material may be used for self-development purposes within the enterprise space or the private space only.

Any application of the Intellectual property in development of a product for commercialization would require garnering the rights for the same, from STEX

## Exercise space/ Notes

## Exercise space/ Notes

## Exercise space/ Notes

## Exercise space/ Notes

## Exercise space/ Notes