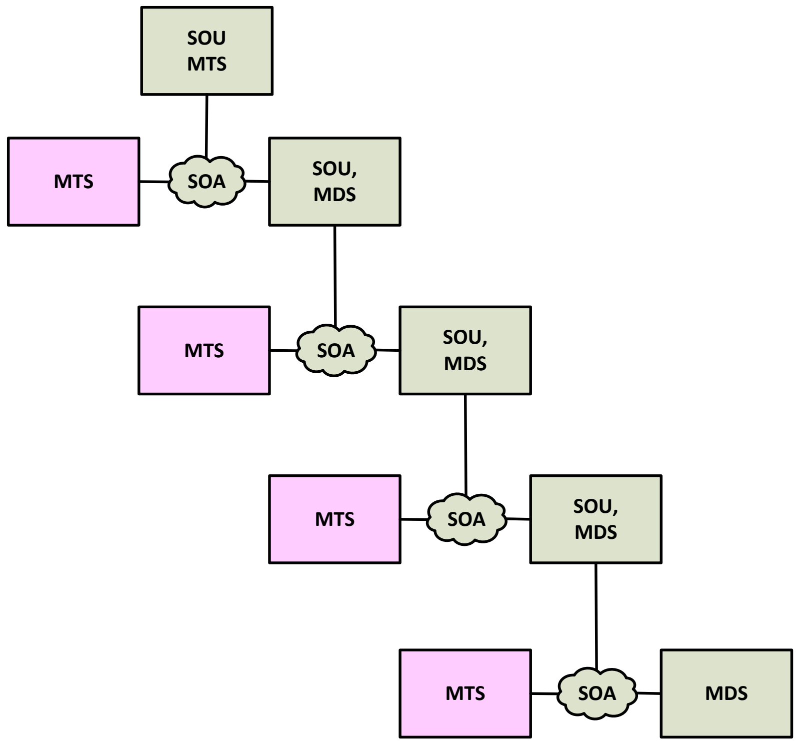
Part II (Afternoon)

- **The Embedded Intelligence (EI) Pattern**: For any embedding of intelligence, in the form of automation, human operators, or other systems of management, feedback, regulation.
- **The Smart Manufacturing Pattern, for the IoT Age**: For any manufacturing process, and with varied forms of instrumentation and management.
- **Capitalization of MBSE Patterns as Financial Assets**: How to shift the burden of model cost to the time of use and benefit.
- Exercises

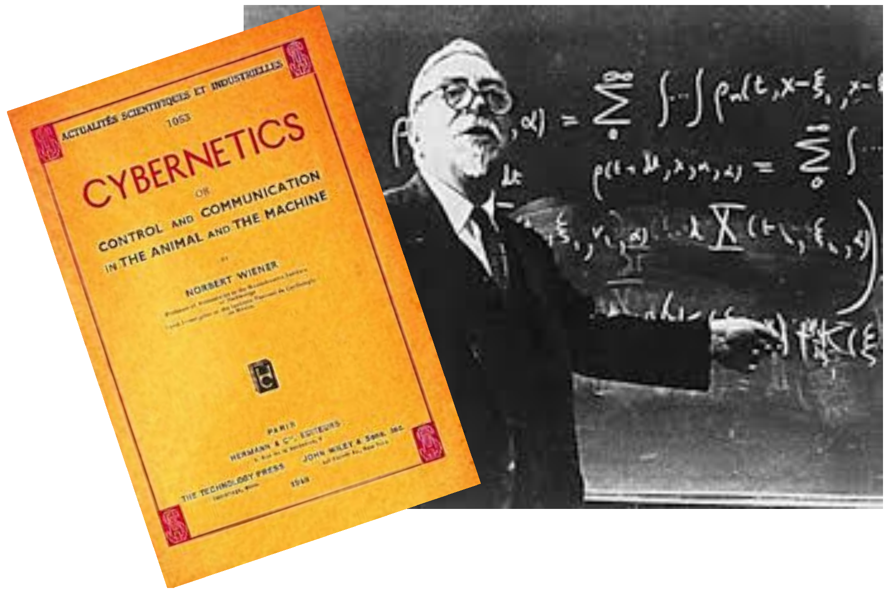
The Embedded Intelligence (EI) Pattern

- For any embedding of intelligence, in the form of automation, human operators, or other systems of management, feedback, regulation.



Embedded Intelligence (EI) Pattern

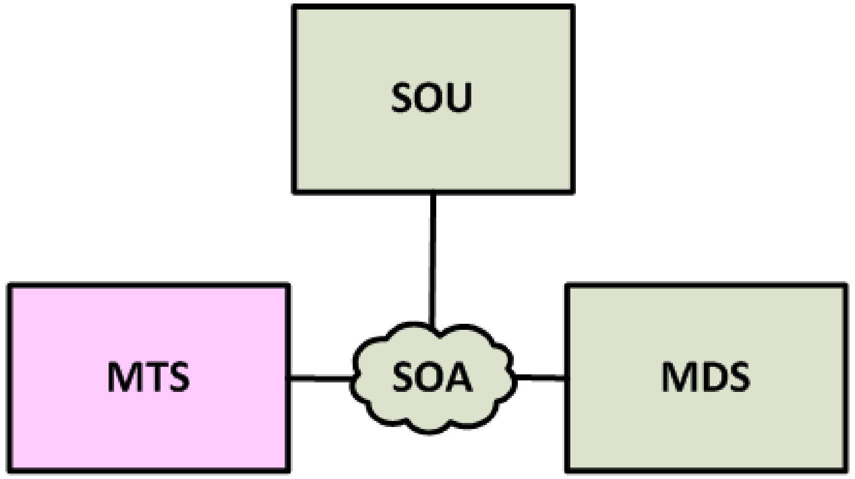
- The EI Pattern returns to the perspective of Norbert Wiener, who first coined the term “cybernetics” to refer to the study of communication and control in living and human-engineered systems:

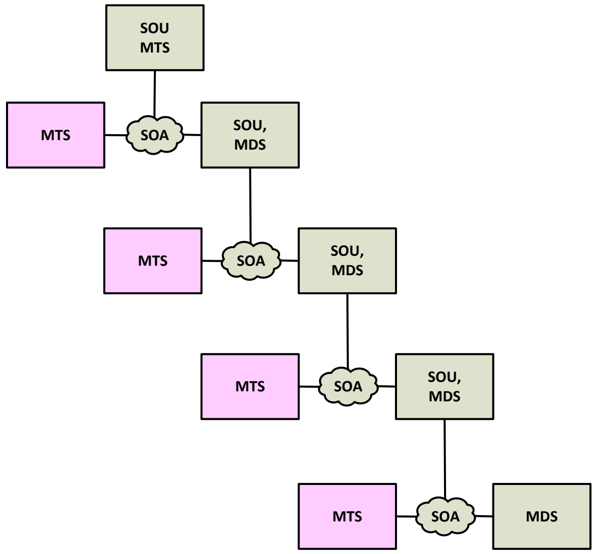

ACTUALITÉS SCIENTIFIQUES ET INDUSTRIELLES
1053
CYBERNETICS
OR
CONTROL AND COMMUNICATION IN THE ANIMAL AND THE MACHINE
NORBERT WIENER

- Especially appropriate if we are interested in Cyber-Physical Systems – but now we are interested in more than just feedback and control performance (studied by Wiener) . . .

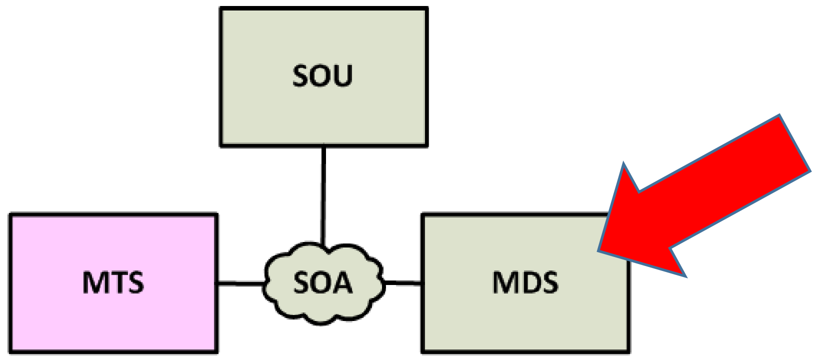
Embedded Intelligence (EI) Pattern

- The EI Pattern is an S*Pattern that emerges to describe intelligence in explicit models of evolving systems in the natural and man-made world:
  - Also referred to as the Management System Pattern.
  - Concerned with the emergence of four roles, emergent at multiple hierarchical levels:



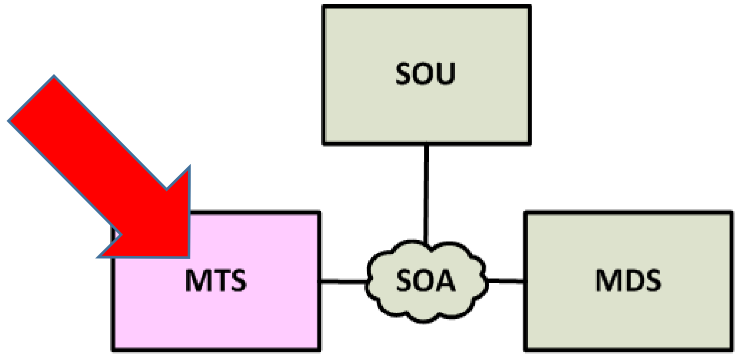


Embedded Intelligence (EI) Pattern



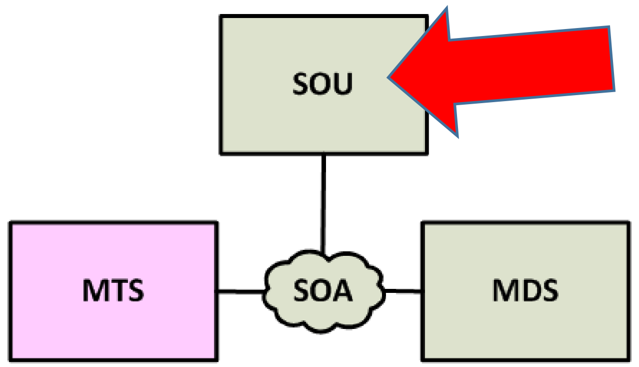
- Managed System (MDS): Any system behavior whose performance, configuration, faults, security, or accounting are to be managed--referred to as System Management Functional Areas (SMFAs) or in ISO terminology fault, configuration, accounting, performance, security (FCAPS).
- These are the roles played by the so-called “physical systems” in a cyber-physical system, providing physical services such as energy conversion, transport, transformation, or otherwise.

Embedded Intelligence (EI) Pattern



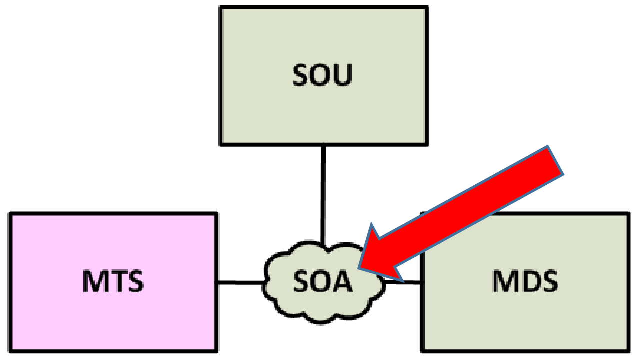
- Management System (MTS): The roles of performing management (active or passive) of any of the SMFAs of the managed system.
- These are so-called “cyber” roles in a cyber-physical system, and may be played by automation technology, human beings, or hybrids thereof, to accomplish regulatory or other management purposes.

Embedded Intelligence (EI) Pattern



- System of Users (SOU): The roles played by a system which consumes the services of an managed system and/or management system, including human system users or other service-consuming systems at higher levels.

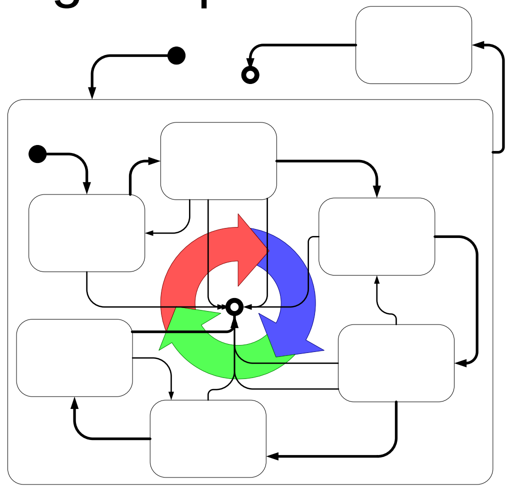
Embedded Intelligence (EI) Pattern



- System of Access (SOA): The roles providing a means of interaction between the other EI roles.
- Engineered sensors, actuators, the Internet, and human-machine interfaces have contributed greatly to the emergence of the “Internet of Things”..

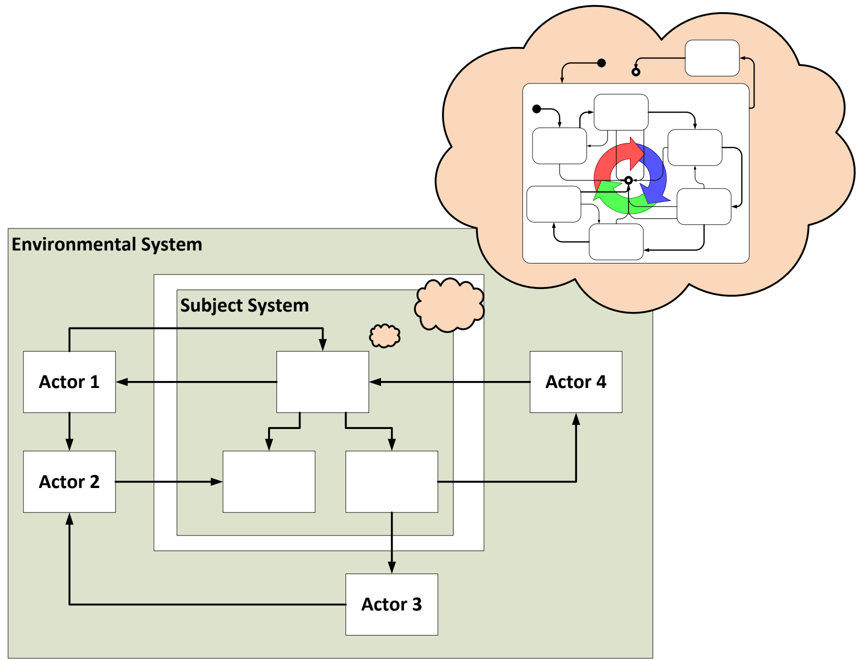
Embedded Intelligence (EI) Pattern

- The State Model portion of the EI Pattern provides insight into the nature of the "regulatory" role of embedded intelligence.
- These show numerous "situation resolution cycles" that drive the managed system to nominal states, when various situations are encountered:
  - Major mission cycles, from mission start to completion
  - Fault resolution cycles, other lesser or minor situation resolution cycles
  - Configuration change cycles, including adaptations
  - Fulfillment of requests for services
  - Security condition resolution cycles
  - Other situation resolution cycles
- Specific or general situations


Sample EI Situation Resolution Cycle

Embedded Intelligence (EI) Pattern

- A system that is capable of not only traversing a situation resolution cycle, but also <u>recognizing</u> that a triggering situation has arisen in the first place is said to be "Situationally Aware":
  - If a human operator control panel has a "mode switch", the system relies on the human to be aware of situations, launching the appropriate cycles
  - More advanced systems recognize these situations autonomously—also leading to EI Attention Model recognition of finite system resources.

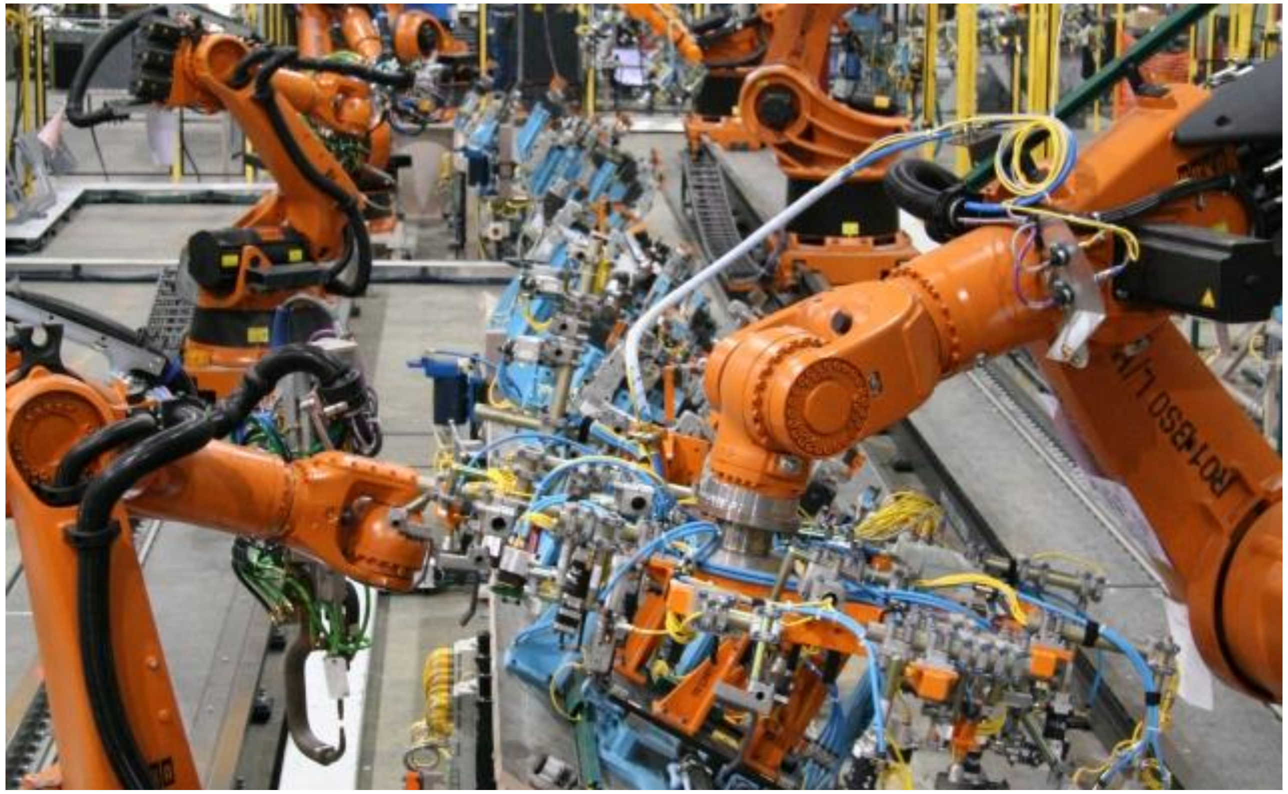


Exercise 4: Applying the Management System Pattern (EI Pattern)

1. Identify a possible Management Systems application for the EI Pattern, for some system of interest. What is the Managed System?
2. Are there multiple levels of control for your example? Draw a multi-level EI Hierarchy and identify the levels.
3. Are there human-filled Management System roles? Automation-filled Management System roles?
4. Which of the five System Management Functional Areas (SMFAs) are involved?
5. What types of Management Situations would occur, for resolution by the Management System?

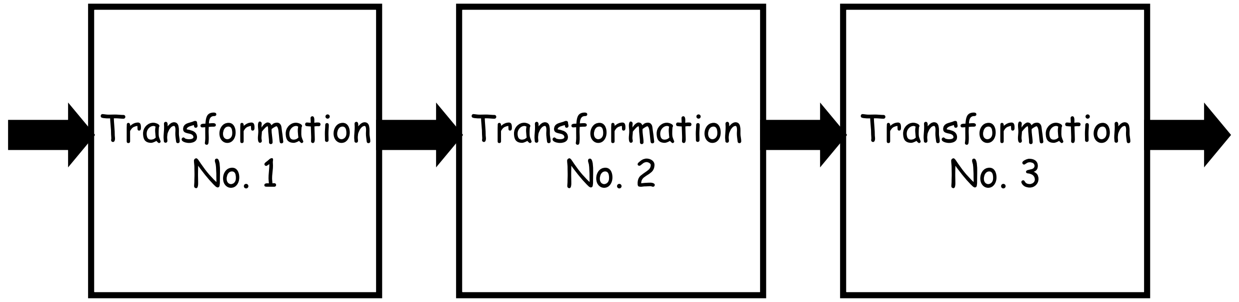
The Smart Manufacturing Pattern, for the IoT Age

- For any manufacturing process, and with varied forms of instrumentation and management.



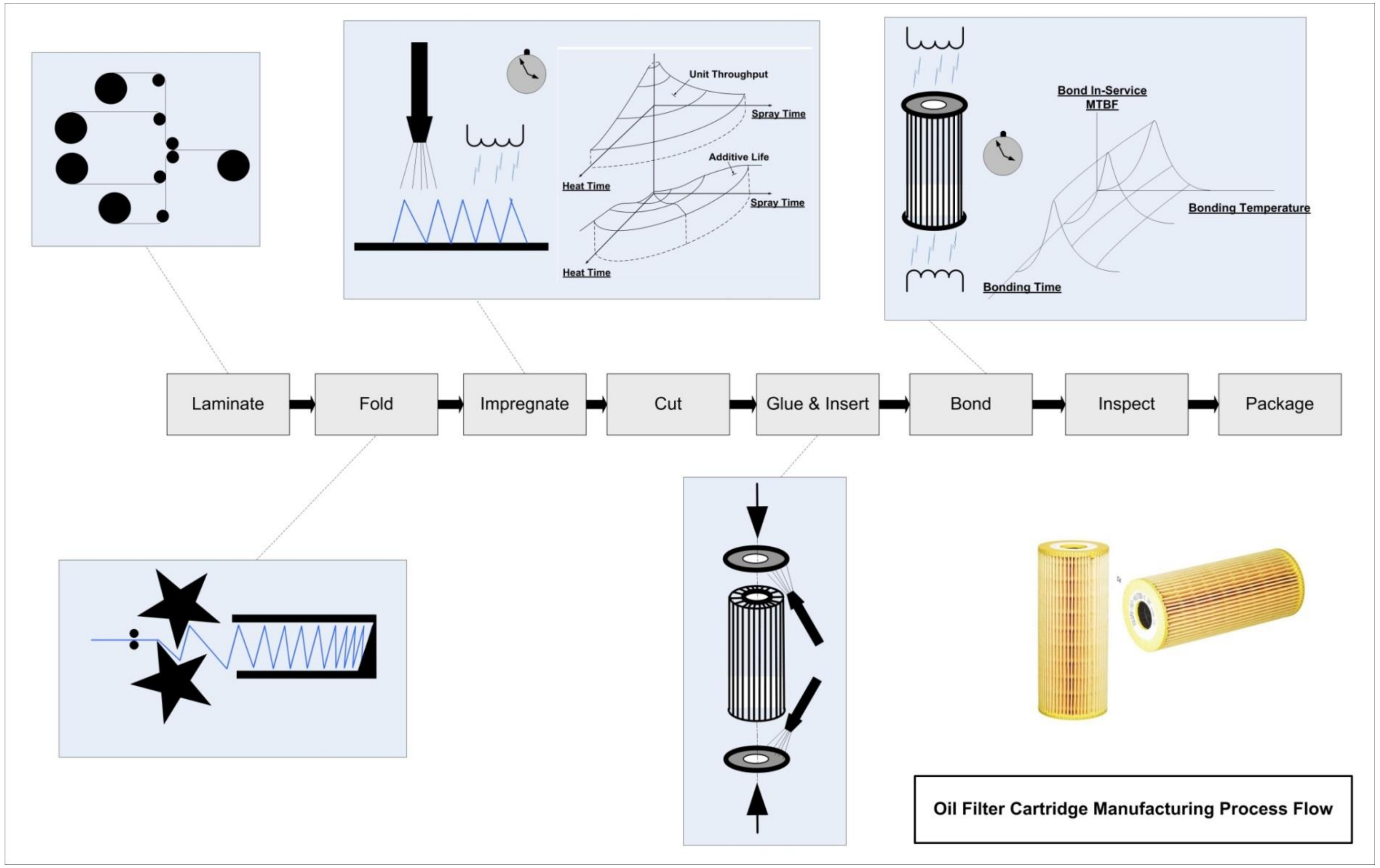
The Process Engineer's Perspective

- Process Engineers are trained to visualize manufacturing as transformations of material (or of information).
- This is frequently represented graphically using Process Flow Diagrams (PFDs):

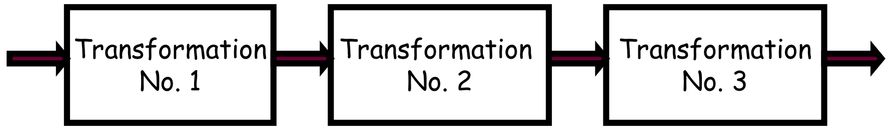


- The material flowing out is different than the material flowing in--it is "transformed" chemically, structurally, thermodynamically, as information, visually, etc.

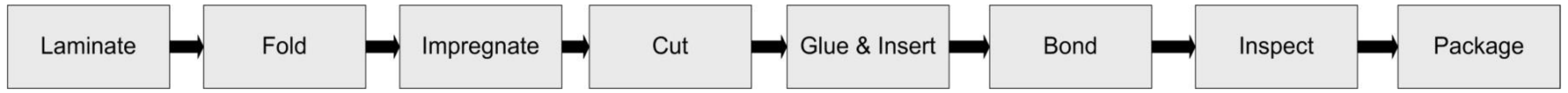
A Simple Example: Manufacturing Oil Filter Cartridges



Process Engineering vs. Equipment Design



- By omitting equipment-specific design, the PFD perspective has the advantage of emphasizing what is required to be changed (transformed) about the material, without describing how manufacturing equipment, tools, people, or control systems will accomplish those transformations:



- Since it describes the required transformations, it is a form of partial requirements on a manufacturing system.

Process Engineering Challenges

- Process Engineering and Process Flow Diagrams provide powerful tools for conceptualizing manufacturing processes.
- However, the fact they use a perspective or language separate from design of equipment requires that the enterprise bridge a gap when integrating PE into the larger engineering context.
- For example, not all requirements on a manufacturing system are requirements of the process itself—they may even conflict.
- Various enterprises and trade groups have wrestled with the question of integrating the larger engineering process for manufacturing systems . . .

Integration with the larger engineering context: Challenges

1. How can the language and perspective of <u>process engineers</u> be more effectively coupled to those of <u>equipment designers</u>?
2. How do <u>process</u> requirements fit into <u>overall</u> manufacturing system requirements, which have larger scope?
3. What is the relationship of physical <u>equipment design</u> to these requirements?
4. How can process requirements for <u>new or modified</u> products be incorporated <u>early enough</u> in the equipment design cycle?
5. How are manufacturing system requirements that are <u>not</u> transformation of materials related to this?
6. How can we conceive new manufacturing solutions without being mentally trapped in assuming constraints of past designs?
7. How can candidate manufacturing designs, design changes, or design risks be evaluated in light of process engineering needs?
8. How are industry reference models of manufacturing (e.g., ISA, ISPE, etc.) related to these issues?
9. More generally, how can increasingly complex advanced manufacturing systems best be engineered, over their life cycles?

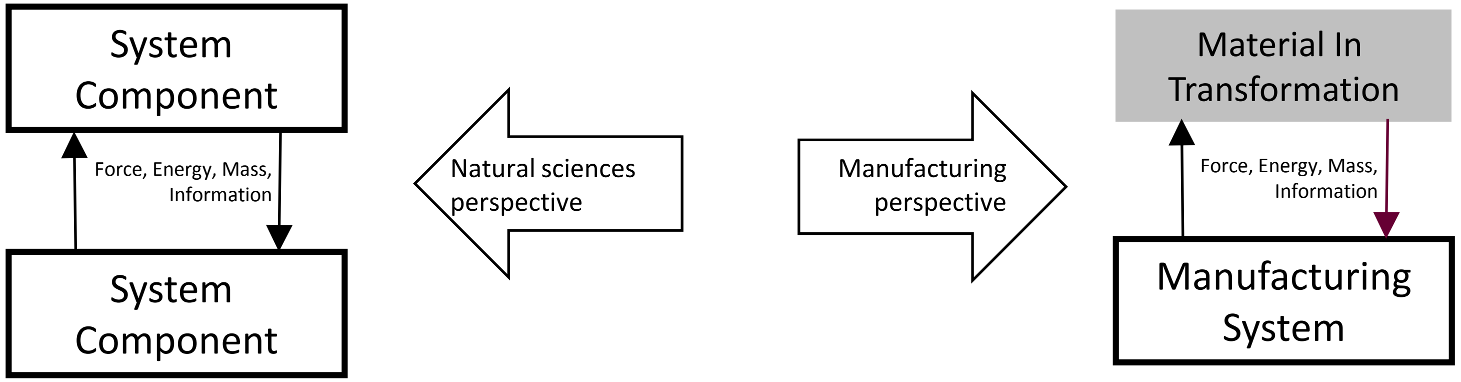
The need for a Science-based Understanding

- Industry trends increasingly emphasize science-based understanding of manufacturing processes:
  - Unit operations: key parametric relationships—materials science, chemistry, physics, etc.
  - First principle and empirical characterizations;
  - Mathematics of production flow;
  - Process capabilities and control laws;
  - Regulatory (e.g., FDA) pressures for a more science-based approach.

- How do we fit science-based understanding into an integrated framework of process and equipment engineering?

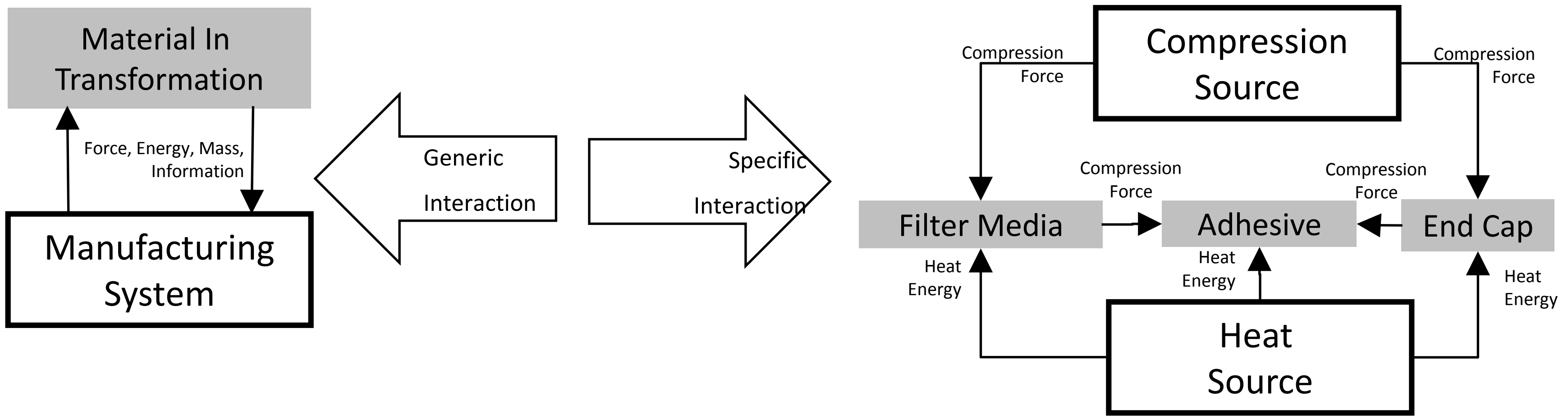
The need for a Science-based Understanding

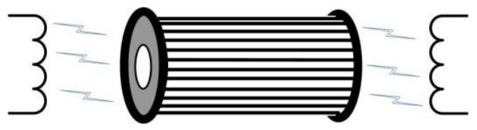
- *Literally everything we know* from the physical sciences is about the *behavior of interacting system components*—whether in chemical reactions, electromagnetics, acoustics, mechanics, thermodynamics, or other discipline-specific interactions:


System Component
Force, Energy, Mass, Information
System Component
Natural sciences perspective
Manufacturing perspective
Material In Transformation
Force, Energy, Mass, Information
Manufacturing System

- Accordingly, the *interactions* of Materials In Transformation with the Manufacturing System assign "roles" to the *Manufacturing System* and the *Materials*, which are required to be met by what we have learned from sciences and by the results we want.

An example Interaction





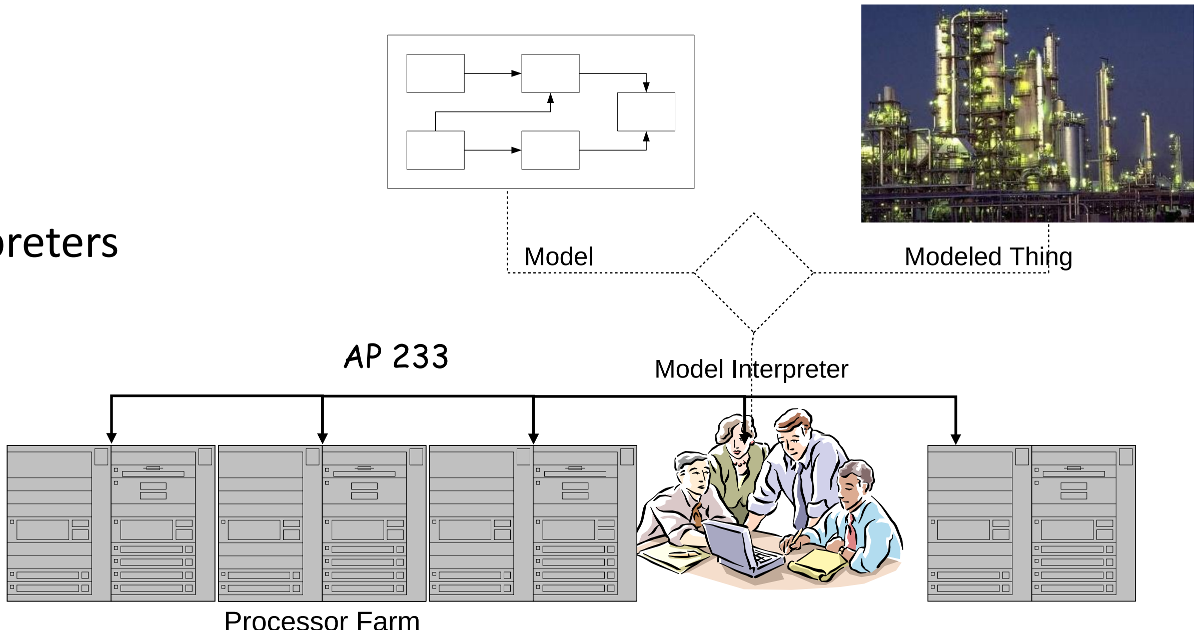
- Interaction = “Bond Filter Media to End Cap”
- Functional Roles (of materials and equipment):
  - Filter Media
  - End Cap
  - Adhesive
  - Heat Source
  - Compression Source

- Each of these “Roles” includes specific Required Behavior in order to meet expectations for the overall Interaction.
- The Physical Component to which the Role is allocated must meet those requirements—whether Equipment, Materials, or People

Model-Based Systems Engineering (MBSE)

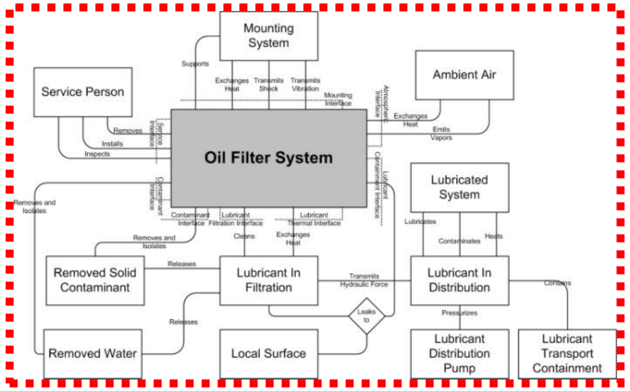
- Model-based systems engineering is an emerging approach to systems engineering:
  - See www.incose.org
- Uses explicit models where previously informal, intuitive, natural language prose (e.g., English) of documents was used

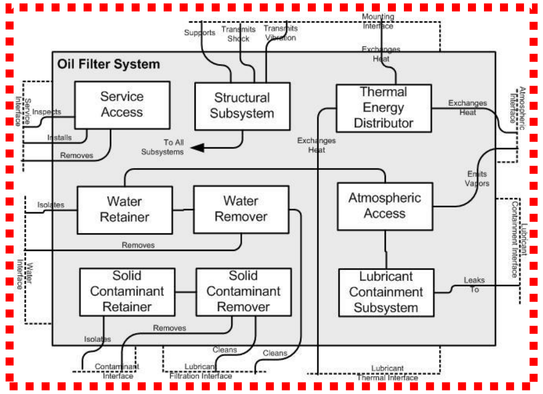
- Not all model interpreters need be human

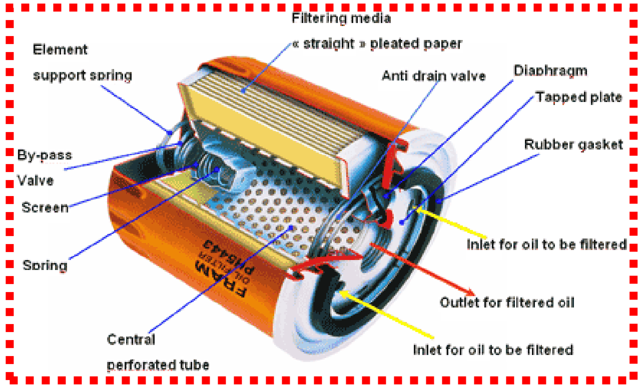


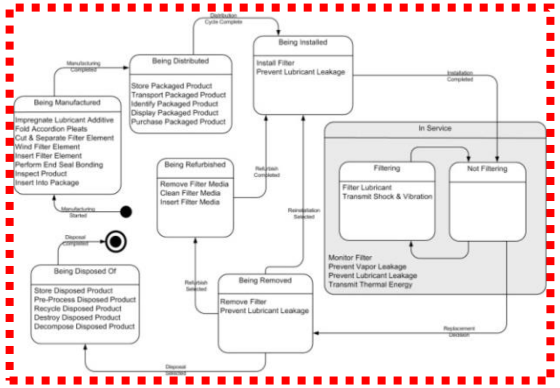
Assumed MBSE background we'll need

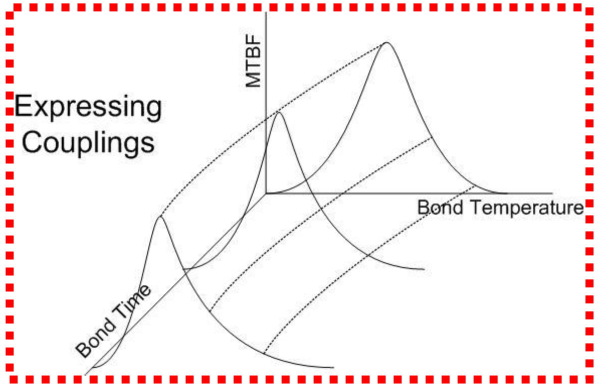
There is a growing practice and literature on Model-Based Systems Engineering.







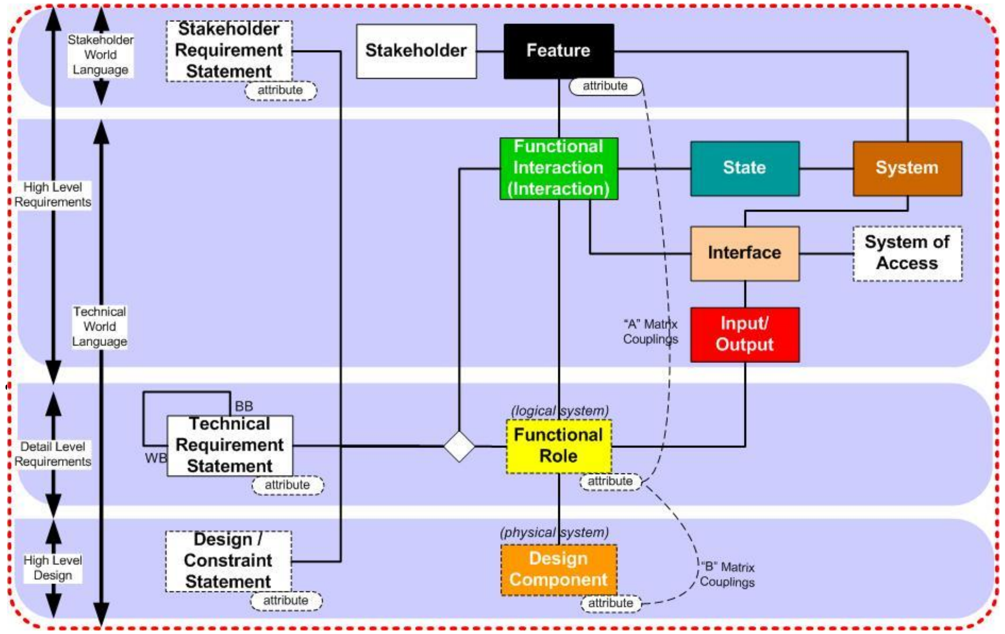




Systematica approach to MBSE

- Uses models ("blueprints") instead of prose, to specify requirements and design of complex systems (product systems, manufacturing systems, operations processes, the engineering process, etc.).
- Increases understanding while lowering costs.

- Establishes a common language and data model for all systems engineering , across people, tools, information systems—for leadership as well as technologists.
- Expresses model-based formal industry standard (e.g., ISA) descriptions of systems.
- Uses S*Metamodel to express underlying concepts.



Model-based systems engineering (MBSE)

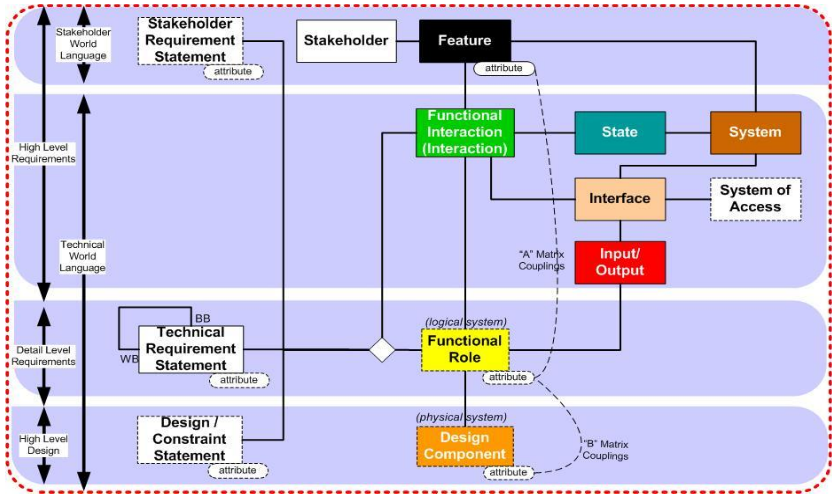
What does Systematica mean by “Metamodel”?

- The framework in which all models are described
- *The minimum set of ideas necessary to express all concepts of system requirements and design, independent of technology*
- The overall model to which any system model must conform
- Constrains community to a common framework, across technologies and functions

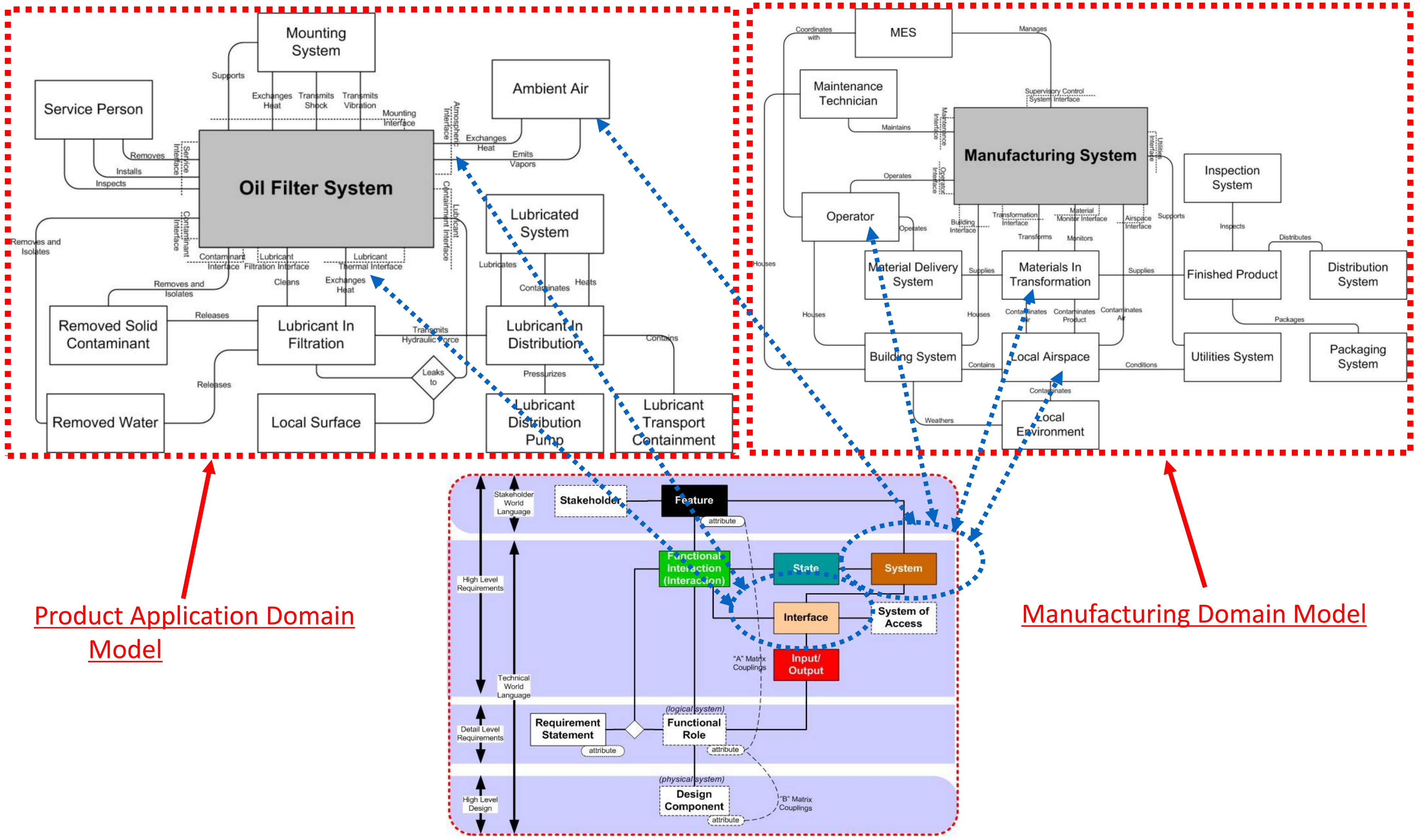
– Within this framework, we create an Enterprise or Industry Language for Shared Patterns, to consistently express system requirements, designs, validations and verifications, FMEAs, etc.

– Incorporating industry, enterprise, governmental standards as needed

Summary of S*Metamodel

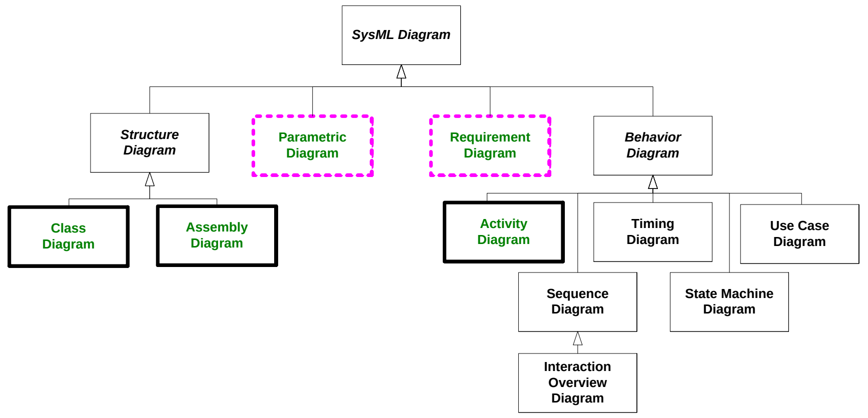


Models can describe Manufacturing Systems, as well as Manufactured Products.



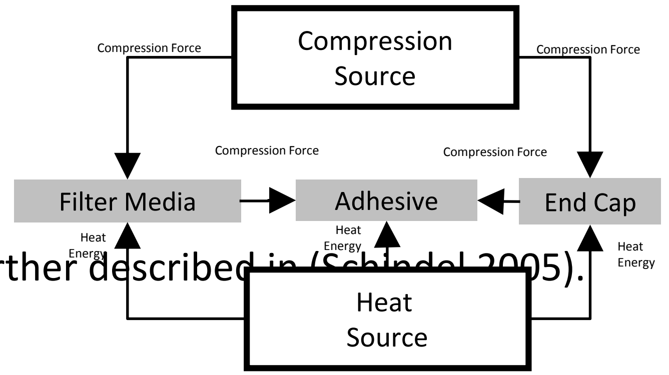
Assumed MBSE background we'll need

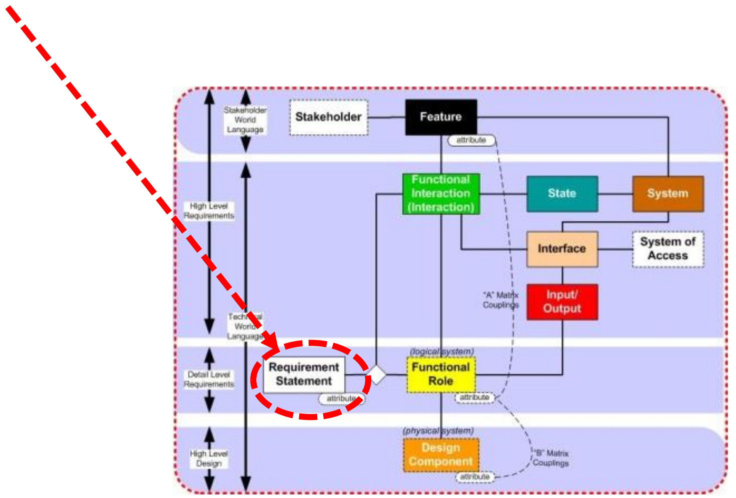
- Model-based methods supplement the use of natural language prose in traditional engineering documents with the use of "models" which are explicit data structures (typically relational tables and formal diagrams).
- The structure of these models can be exploited to create analyses and checks that would be much more difficult and subjective to perform using purely prose-based methods.
- When applied well, they can also more effectively convey shared meaning to human readers.
- We will focus here on how Manufacturing Transformations can be more deeply integrated as a part of such MBSE models.
- See the attached example for other aspects.



Modeling transformation behavior

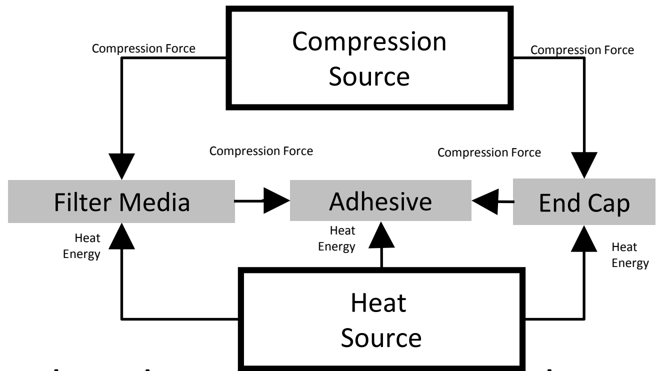
- This Metamodel re-positions prose functional “Requirements Statements”:
  - These textual statements become a formal part of the model.
  - All functional requirements are modeled as external interaction behaviors.
  - They become input-output relationships describing external system “black box” behavior during Interactions with external actors—a “prose transfer function”:
    - *“The Manufacturing System shall deliver to the Materials In Process a <u>Compression Force</u> of [Min Bond Force] for a period of [Min Bond Time]”.*
    - *“The Manufacturing System shall deliver to the Materials in Process <u>Heat Energy</u> sufficient to maintain a bond temperature of [Min Bond Temperature] for a period of [Min Bond Time].”*
- Further described in (Schindel 2005).

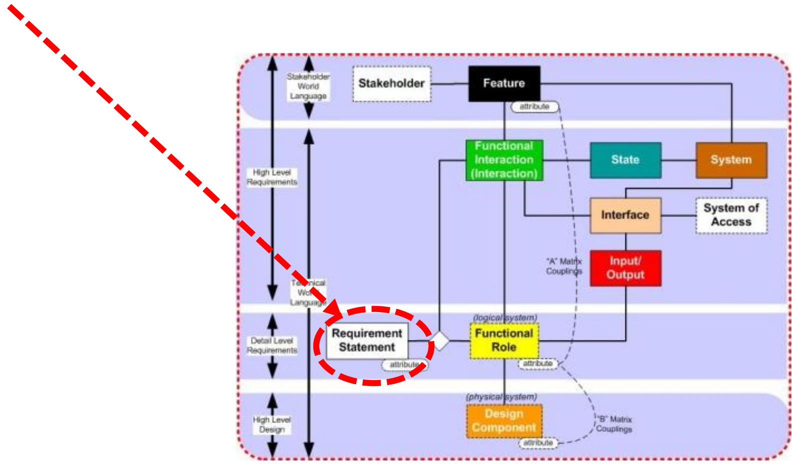




It works for the Materials in Process, as well as the Manufacturing System

- In the same way, in the same model we can describe the required behavior of the Materials in Process:
    - *“The Adhesive, Filter Media, and End Cap shall bond upon input of a Compression Force of [Min Bond Force] for a period of [Min Bond Time], accompanied by input of Heat Energy sufficient to maintain a bond temperature of [Min Bond Temperature] for a period of [Min Bond Time].”*
    - *“The Oil Filter shall operate in service at Lubricant Pressure of [Max Lubricant Pressure] with bond or other structural failure rates less than [Max Structural Failure Rate] over an in-service life of [Min Service Life].”*

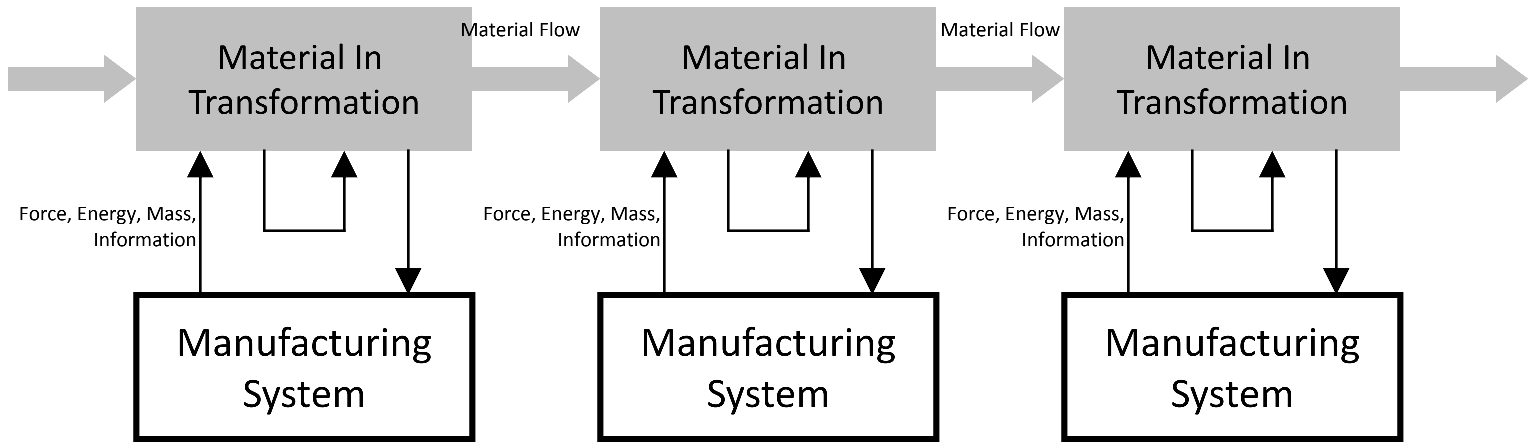




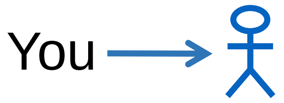
- Further described in (Schindel 2005).

Applying the concepts to manufacturing processes

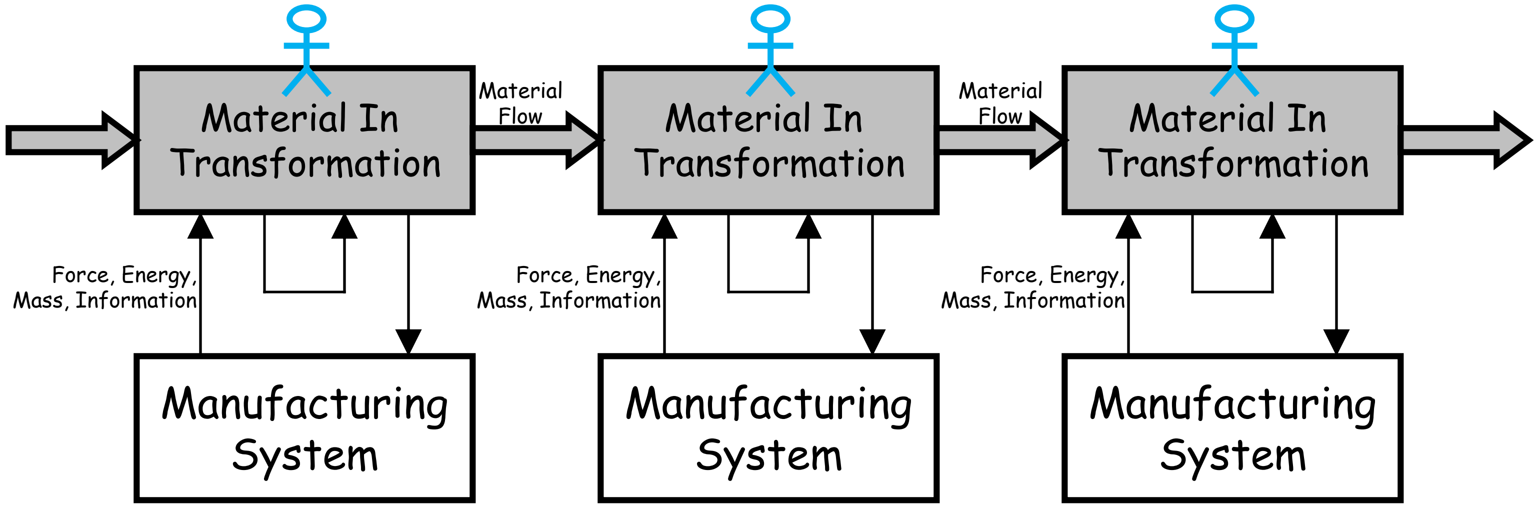
- For some process engineering specialists, material scientists, or other disciplines, an understanding of the behavior of the material during transformations is essential:
  - bending, forming, structural deformations, cutting, milling, extruding, compression
  - chemical, biochemical, electrochemical reactions, distillation, fermentation, etc.
  - heating, cooling, bonding, welding, fastening, mixing, blending
  - other transformations
- These specialists think about the "Material In Transformation":
  - how the material behaves <u>during</u> each of a series of sequential unit operation transformations;
  - During each transformation, the Material will exchange <u>energy</u>, <u>force</u>, <u>mass</u>, or <u>information</u> with the Manufacturing System, as well as with itself--


Material In Transformation
Material Flow
Material In Transformation
Material Flow
Material In Transformation
Force, Energy, Mass, Information
Force, Energy, Mass, Information
Force, Energy, Mass, Information
Manufacturing System
Manufacturing System
Manufacturing System

Process: What the Material "Sees"

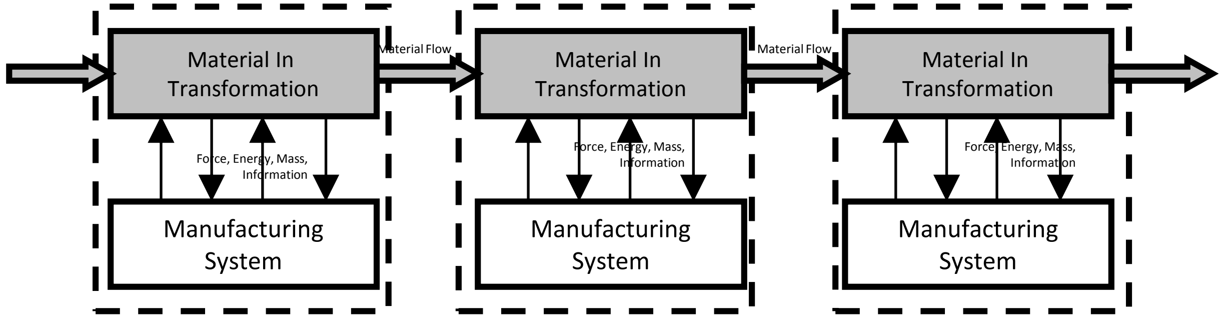


- Imagine that you could "ride through the process with the material".
- Imagine that you could "see what the material sees" (forces, temperatures, etc.).
- This is the "process view" of the process engineer, materials scientist, chemist, metallurgist, or other process-related specialist:

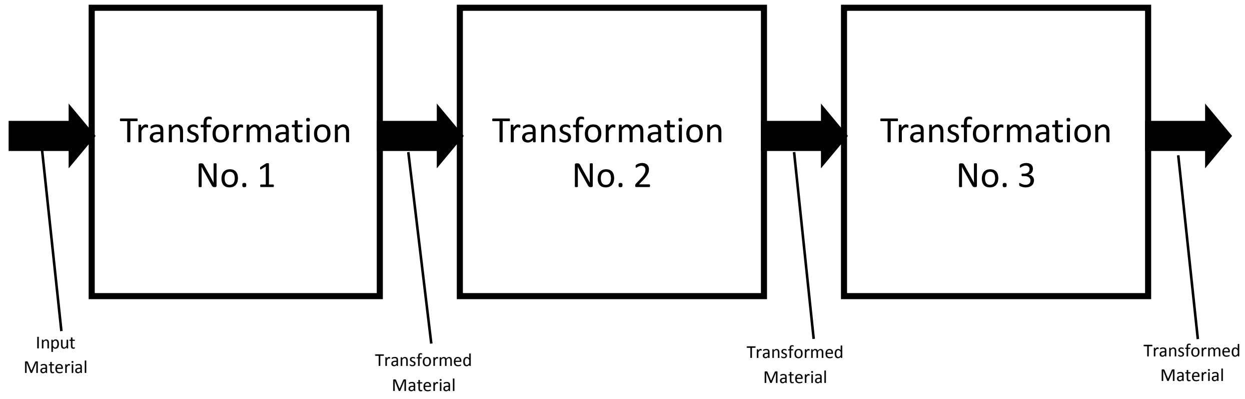

Material In Transformation
Material Flow
Material In Transformation
Material Flow
Material In Transformation
Force, Energy, Mass, Information
Force, Energy, Mass, Information
Force, Energy, Mass, Information
Manufacturing System
Manufacturing System
Manufacturing System

Less detailed PFD views

- Others people's jobs don't need that much detail, so they think of the transformations as "black boxes"; so that . . . .

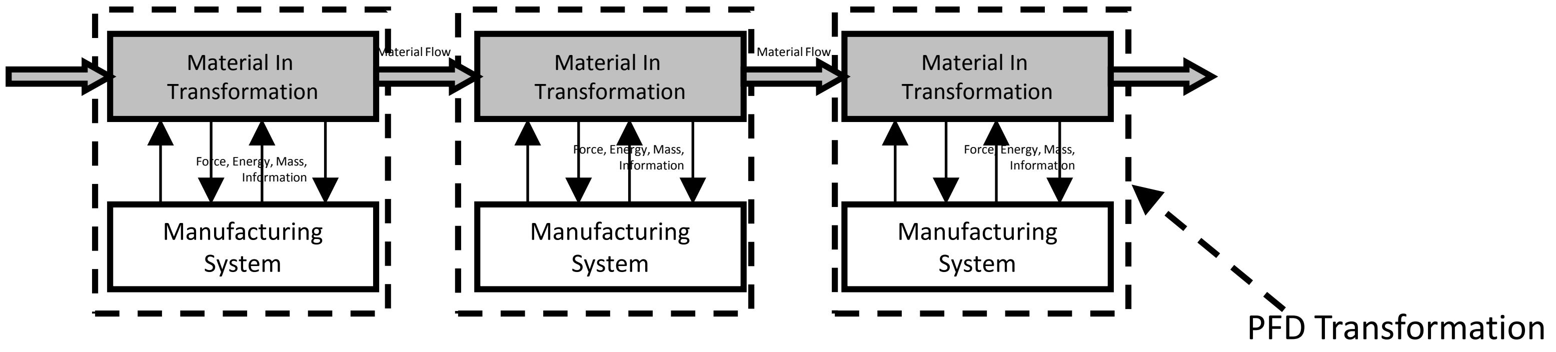

Material In Transformation
Material Flow
Material In Transformation
Material Flow
Material In Transformation
Force, Energy, Mass, Information
Force, Energy, Mass, Information
Force, Energy, Mass, Information
Manufacturing System
Manufacturing System
Manufacturing System

becomes a Process Flow Diagram (PFD):



Material In Transformation can be modeled as “logically outside” the equipment’s transformation role

- Difference between these two representations:
  - the Material In Transformation is “logically outside” the Manufacturing System, but . . .
  - that Material In Transformation is “logically inside” the PFD Transformations:


Material In Transformation
Material Flow
Material In Transformation
Material Flow
Material In Transformation
Force, Energy, Mass, Information
Force, Energy, Mass, Information
Force, Energy, Mass, Information
Manufacturing System
Manufacturing System
Manufacturing System
PFD Transformation

- After all, the Material In Transformation is not a part of the BOM of the Manufacturing System!
- The advantage of this approach is that it allows us to use the MBSE technique that *all the functional requirements on the manufacturing system are found at the points of input-output boundary crossings of that system*

“Registered Process” As Requirements

- Many manufacturing “processes” have a kind of managed existence separate from their specific implementation with equipment:
  - When a PFD describes a process before there is equipment design;
  - When a “registered process” has been approved by a regulator, and a factory is constructed to implement that specific process;
  - When a low-volume process has come out of a laboratory to a pilot production line, but not yet been scaled up to production volume.

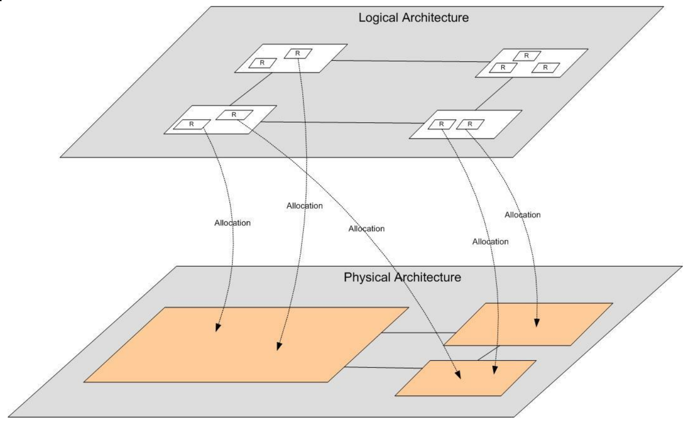
- This reflects the idea that the requirements of a manufacturing system are something more than producing the end outputs from the initial inputs—it is also expected to embody a specific targeted manufacturing process.

- This is why we model the “Materials In Process” as an external actor interacting with the equipment.

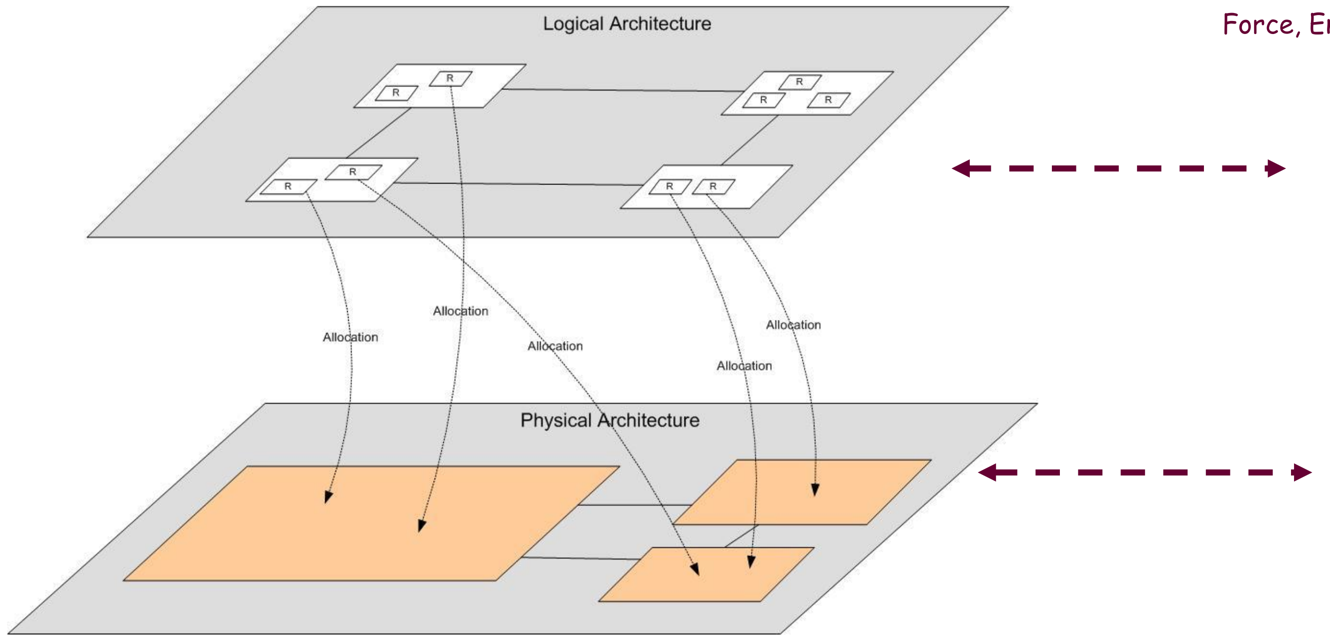
Logical Systems vs. Physical Systems

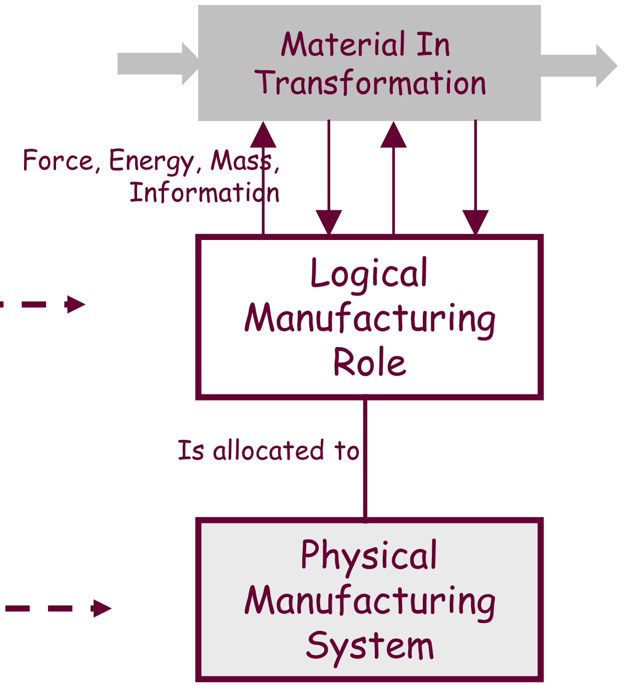
- MBSE expresses what the Manufacturing System contributes to the process, using Logical Systems:
  - Logical systems are defined by their required externally visible behavior, as seen by the other interacting actors, without regard to the physical design used to accomplish that behavior.

- Logical System Roles:
  - represent transformation or other behavior of the manufacturing system, without regard to its design.
  - Certain Logical Manufacturing Roles must produce (or consume) certain forces, energy, or information, exchanged with the Material In Transformation.
- Physical Manufacturing Systems:
  - Are defined by their physical identity, not their behavior.
  - Logical behaviors are then allocated to physical equipment.
- Logical Roles are allocated to Physical Systems



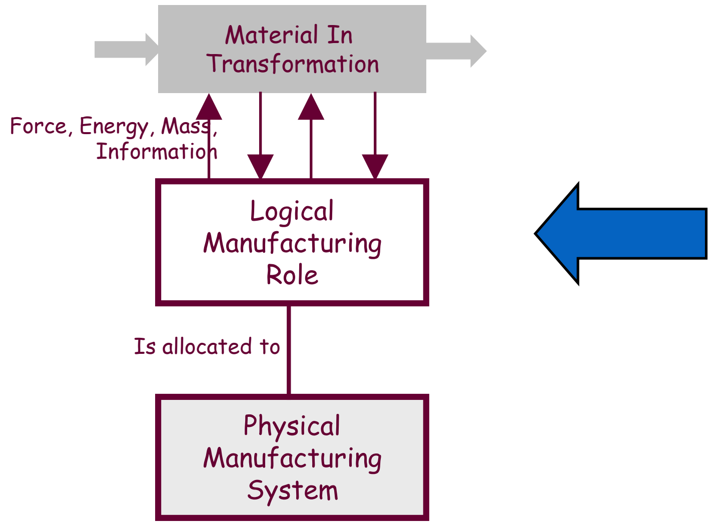
Logical Systems vs. Physical Systems




Material In Transformation
Force, Energy, Mass, Information
Logical Manufacturing Role
Is allocated to
Physical Manufacturing System

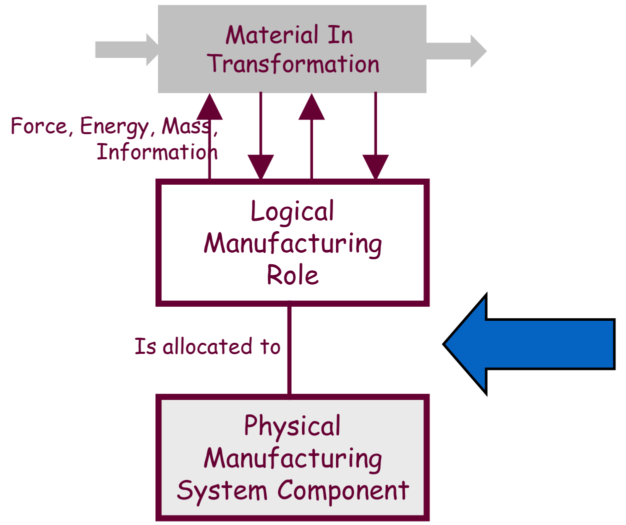
Manufacturing system requirements

- The input-output relationships (relationships between input-output Forces, Energies, Masses, Information that are exchanged with the Material In Transformation) of the Logical Manufacturing Roles turn out to express the requirements allocated to the Manufacturing System to accomplish the transformation:


Material In Transformation
Force, Energy, Mass, Information
Logical Manufacturing Role
Is allocated to
Physical Manufacturing System

Manufacturing equipment design

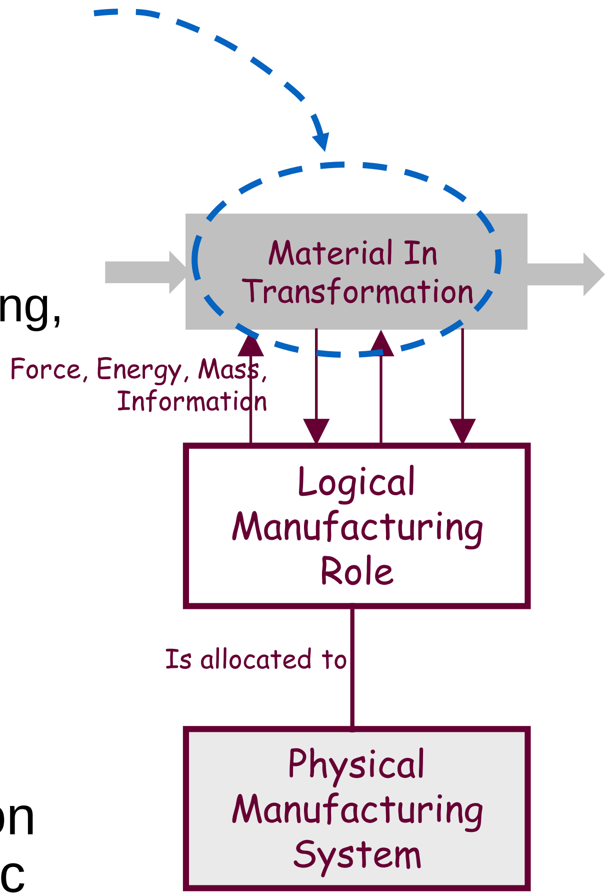
- The allocation of logical manufacturing roles to physical equipment components describes the high level design of the manufacturing system:


Material In Transformation
Force, Energy, Mass, Information
Logical Manufacturing Role
Is allocated to
Physical Manufacturing System Component

➢ This begins the embedding of process requirements into an integrated framework of system requirements.

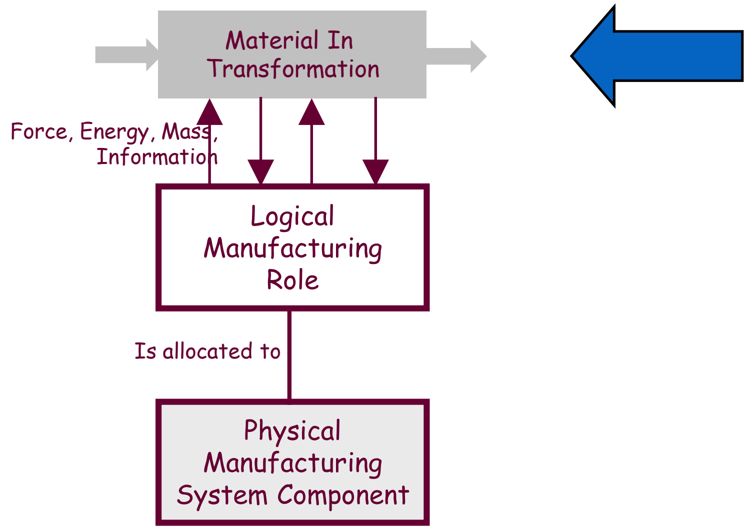
Materials roles

- For materials scientists, chemists, metallurgists, and other specialists in materials . . .
- These specialists seek out materials that have properties desirable for transformations:
  - bending, forming, structural deformations, cutting, milling, extruding, compression
  - chemical, biochemical, electrochemical reactions, distillation, fermentation, etc.
  - heating, cooling
  - bonding, welding, fastening
  - mixing, blending
  - other transformations
- The logical transformation model facilitates description of those properties, somewhat independent of specific materials:
  - Encourages understanding of materials requirements and opens thinking to new materials solutions.

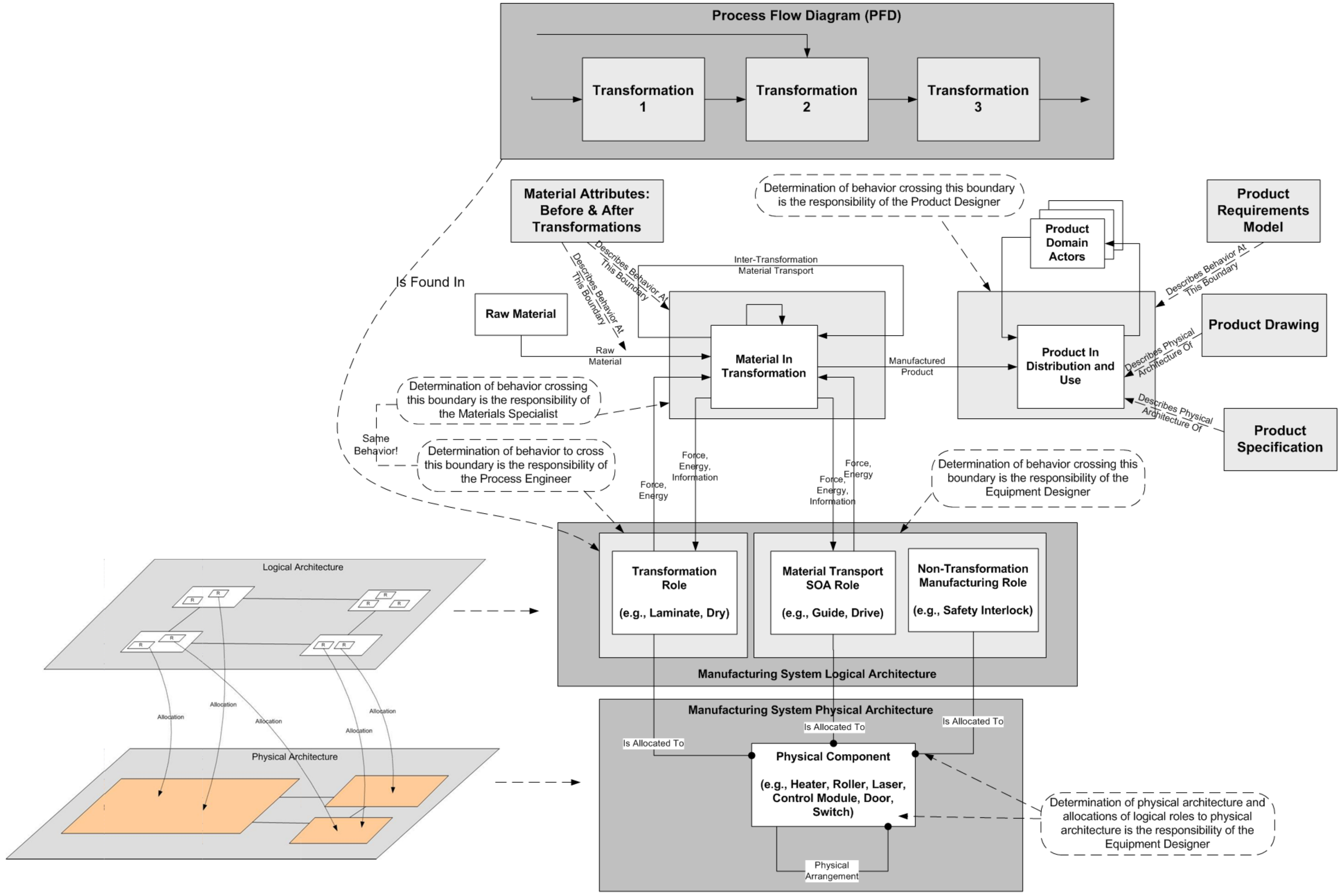

Material In Transformation
Force, Energy, Mass, Information
Logical Manufacturing Role
Is allocated to
Physical Manufacturing System

Materials roles

- Just like the equipment, logical roles are allocated to the Materials In Transformation, which they must satisfy in order for the transformation (or transport) to succeed:


Material In Transformation
Force, Energy, Mass, Information
Logical Manufacturing Role
Is allocated to
Physical Manufacturing System Component

- This means that we can create an integrated model that couples the roles of interest to the process engineer and equipment design with those of interest to the materials specialist . . . .


Process Flow Diagram (PFD)
Transformation 1
Transformation 2
Transformation 3
Is Found In
Material Attributes: Before & After Transformations
Describes Behavior At This Boundary
Describes Behavior At This Boundary
Determination of behavior crossing this boundary is the responsibility of the Product Designer
Product Domain Actors
Product Requirements Model
Describes Behavior At This Boundary
Inter-Transformation Material Transport
Raw Material
Raw Material
Material In Transformation
Manufactured Product
Product In Distribution and Use
Product Drawing
Describes Physical Architecture Of
Describes Physical Architecture Of
Product Specification
Determination of behavior crossing this boundary is the responsibility of the Materials Specialist
Same Behavior!
Determination of behavior to cross this boundary is the responsibility of the Process Engineer
Force, Energy
Force, Energy, Information
Force, Energy, Information
Force, Energy
Determination of behavior crossing this boundary is the responsibility of the Equipment Designer
Logical Architecture
Transformation Role (e.g., Laminate, Dry)
Material Transport SOA Role (e.g., Guide, Drive)
Non-Transformation Manufacturing Role (e.g., Safety Interlock)
Manufacturing System Logical Architecture
Allocation
Allocation
Allocation
Allocation
Allocation
Manufacturing System Physical Architecture
Is Allocated To
Is Allocated To
Is Allocated To
Physical Architecture
Physical Component (e.g., Heater, Roller, Laser, Control Module, Door, Switch)
Determination of physical architecture and allocations of logical roles to physical architecture is the responsibility of the Equipment Designer
Physical Arrangement

Logical and Physical Architecture, Cross-Domain Development & Engineering Roles

Conclusions

Applying this PBSE approach to manufacturing systems helps:

1. Integrate science-based understanding of processes, materials, and transformations into the life cycle engineering of manufacturing systems.
2. Improve integration of Process Engineering with other engineering disciplines.
3. Improve manufacturing process IP capture—particularly using PBSE.
4. Improve teams' and individuals' abilities to "think outside the box".
5. Speed discovery of new product and process implications for equipment design.
6. Improve understanding of newer references and standards for describing manufacturing processes that use the language of "models".
7. Improve the ability to perform long-range planning and portfolio management of manufacturing technologies, along with related product science and technologies.
8. Organize patterns of re-usable IP for processes, materials, technology, and design.

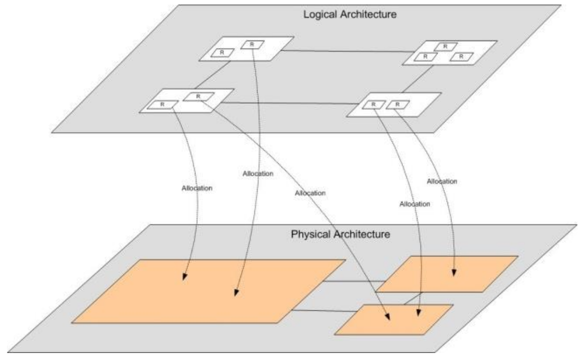
Additional information

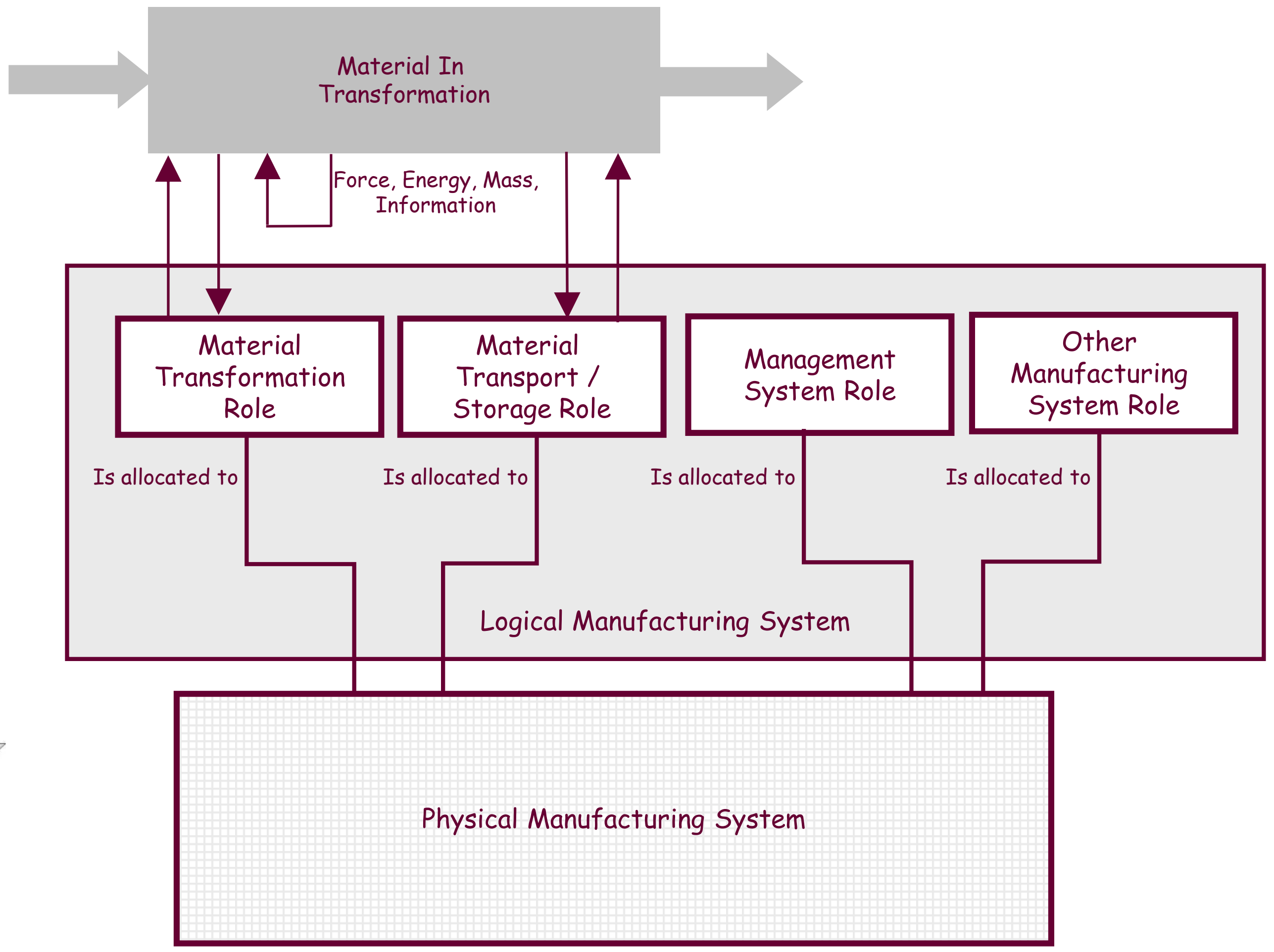
- Non-transformation manufacturing roles
- Manufacturing patterns, parameterized recipes
- Unit operations vs. higher level systems
- Portfolio management
- An extended example

Non-transformation manufacturing roles

- There are additional logical roles that the Manufacturing System must perform, beyond physical transformations.
- For example:
  - Transport and storage roles;
  - Material systems of access (interface) roles;
  - Infrastructure roles (utilities, etc.);
  - Management: Operations, maintenance, configuration, security, accounting roles

Non-transformation roles





Material In Transformation
Force, Energy, Mass, Information
Material Transformation Role
Material Transport / Storage Role
Management System Role
Other Manufacturing System Role
Is allocated to
Is allocated to
Is allocated to
Is allocated to
Logical Manufacturing System
Physical Manufacturing System

Transport and storage roles

- Requirements on the manufacturing system for:
  - Transport (movement of material in process)
    - Liquid transport
    - Web transport
    - Powder, solid materials, gaseous transport
    - Logistics considerations, carriers, space, etc.
  - Storage
    - Roles typically filled by tanks, warehouses, shelves, etc.

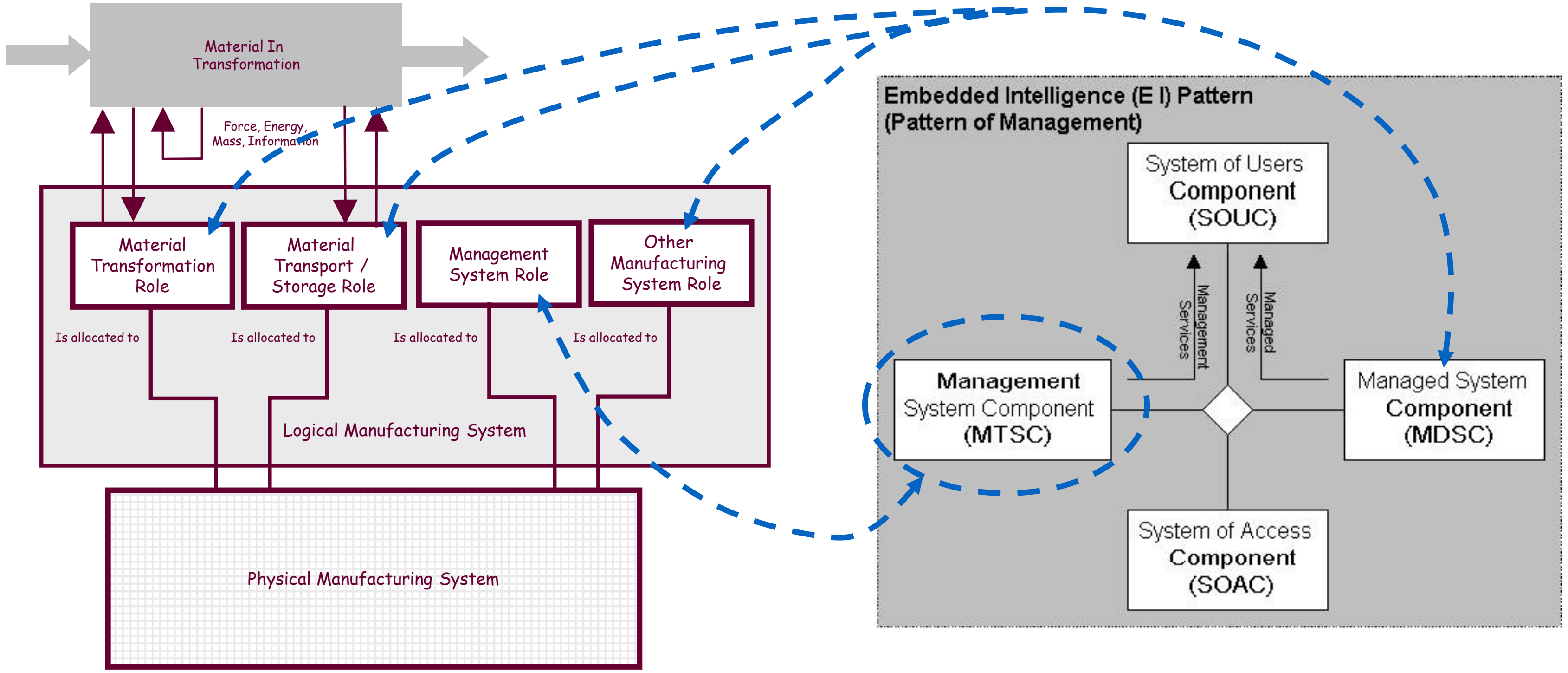
Material Systems of Access (SOA) Roles

- A System Of Access is part of an Interface Model—the system that enables physical interaction between two other systems.
- SOAs are important “glue” for practical engineering as well as scientific understanding of system interactions.
- Two SOA classes important to Process Engineering models:
  - Transformation Systems of Access--
    - Example: the logical roles played during material transformations, by heated tank jackets (heat transfer) or bubbling gas through liquids (maximize contact area), etc.
  - Transport Systems of Access—
    - Example: the logical roles played during material transport, by slurry pumps, conveyer belts, augers, rolling bins, etc.
- Separating SOAs in the model improves the ability of the underlying transformation and transport processes to be modeled independent of technology.

Infrastructure (utilities, infrastructure, etc.) roles

- Regular utilities (electrical & pneumatic power, heating & cooling media, etc.)
- HVAC
- Clean or specialized utilities
- Consumables treated as utilities
- Waste disposal, treatment, co-generation, or recovery streams
- Plant space, structural resources
- Site resources

Management roles: Operations, maintenance, configuration, security, accounting


Material In Transformation
Force, Energy, Mass, Information
Material Transformation Role
Material Transport / Storage Role
Management System Role
Other Manufacturing System Role
Is allocated to
Is allocated to
Is allocated to
Is allocated to
Logical Manufacturing System
Physical Manufacturing System
Embedded Intelligence (E I) Pattern (Pattern of Management)
System of Users Component (SOUC)
Management Services
Managed Services
Management System Component (MTSC)
Managed System Component (MDSC)
System of Access Component (SOAC)

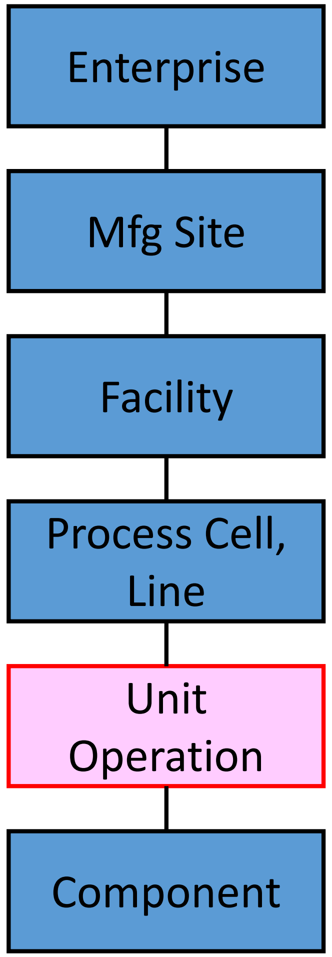
- Electronic controls and automation are “management system roles” that are part of the model.
- These roles are also played by humans (operators, etc.).
- They are usually organized into hierarchical controls patterns:
- For more on this, consult the Systematica materials on Embedded Intelligence (EI) Pattern of Intelligence-Based Systems Engineering (IBSE).

Manufacturing patterns, parameterized recipes

- MBSE “models” describe both requirements and design, for both equipment and materials;
- PBSE “patterns” are re-usable Models, requiring less effort to use than creating Models from scratch;
- Patterns can be configured for different needs and uses:
  - One reason to configure a general pattern is to describe a site specific system (e.g., a manufacturing system installed at a site).
    - A single configured system of this type might still be capable of carrying out many different recipes.
    - This type of configuration is “configuration at design time”.
  - Another reason to configure a pattern is to express a specific recipe:
    - This has the effect of configuring a site specific system for a single recipe, and is a “configuration at run time”
  - For more on this, see the Pattern Configuration Process, ISA S.88, etc.

Unit ops vs. higher level aggregations

- The "materials in transformation" approach to modeling particularly applies to the Unit Operation level, where the transformation occurs;
- There are many other requirements not about transformations, and other hierarchy levels, as well;
- This is all very typical SE hierarchy of decomposable requirements;
- Frequently addressed by multiple disciplines or specialties, and integrated together by SE;
- As usual, it also means that there are attributes (parameters) that are characteristics of the different levels—some are lower level process attributes, but couple to higher level product Quality, Capacity, Yield, Cost, or other critical attributes;
- MBSE attribute coupling models help to make the relationships between these attributes more evident—typically these couplings are characterized by DOE studies, first principles, process characterization, and other sources.



Exercise 5: Applying the Manufacturing System Pattern

1. What is new, changing, or challenging that might drive a need to more effectively model production/manufacturing systems in your or some other enterprise?
2. What types of production material transformations may need more attention? What interactions are involved (equipment-material, material-material, management-equipment-material)?
3. Draw the related Process Flow (transformations) Diagram and its underlying Interaction Diagram.
4. What additional instrumentation or embedding of networking or intelligence in the production process may be occurring, and what challenges to planning and representing this are expected?
5. What are the challenges to the organization or individuals to make this transition?

Capitalization of MBSE Patterns as Financial Assets: How to shift the burden of model cost to the time of use and benefit

- Cost of innovation (development or (otherwise) is a major concern in the strategy and execution of R&D or other advancement.
  - These costs have most frequently been expressed as an expense, subtracting from the current bottom line.
  - The benefits (e.g., increased revenue, etc.) gained from this investment sometimes will not occur until somewhere in the future, making the investment harder.

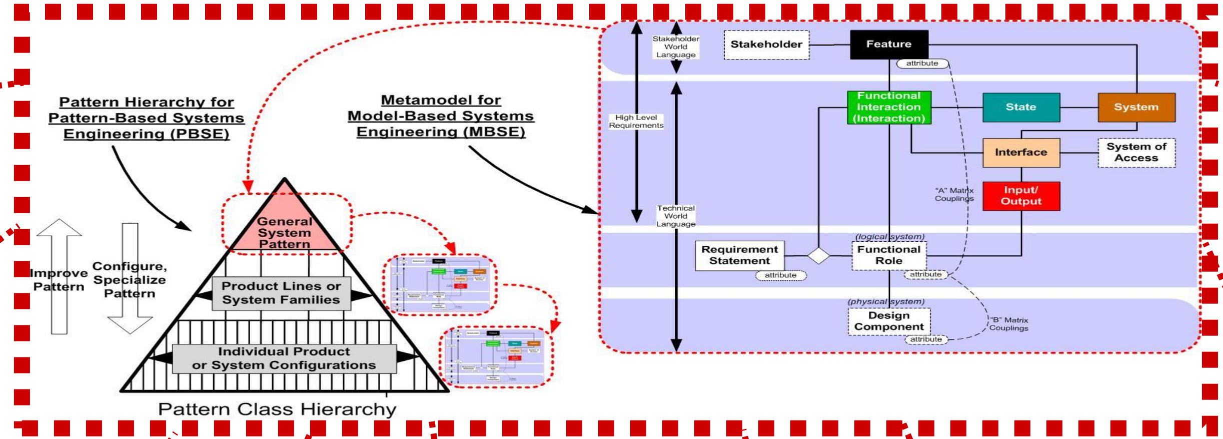
Capitalization of MBSE Patterns as Financial Assets: How to shift the burden of model cost to the time of use and benefit

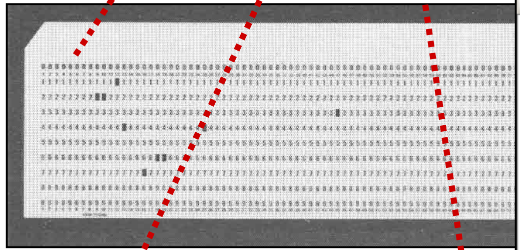
- In the Construction and Capital Equipment businesses, this situation was addressed many decades ago, through capitalization of assets:
  - Construction or fabrication costs are shown on the balance sheet as creating new (tangible) financial assets—buildings or equipment
  - Those assets are then “expensed” (amortized) over future times, with the incremental amortization generating modest annual expenses, during the years of productive life of the (building or equipment) asset.
  - Those are the years that the asset is producing value (revenue or other benefits)
  - It is a little bit like renting an asset instead of buying it, but all carried out within the same financial statements.

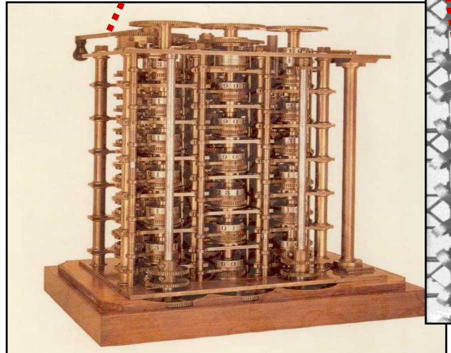
Capitalization of MBSE Patterns as Financial Assets: How to shift the burden of model cost to the time of use and benefit

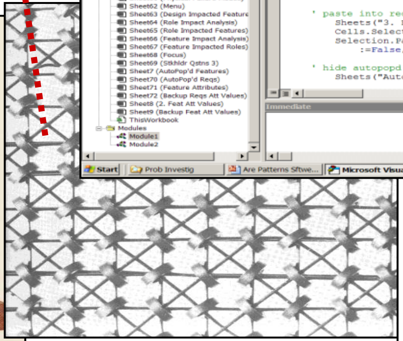
- Over the decades, capital investment in tangible (e.g., bricks and mortar) assets has been outpaced by investment in intangible (e.g., intellectual) assets.
- In the 1980's, this led to the adoption, by the Financial Accounting Standards Board (FASB) of accepted accounting standards for capitalization of computer software. (See FASB86)
- How are MBSE Patterns similar to, or different than, computer software?

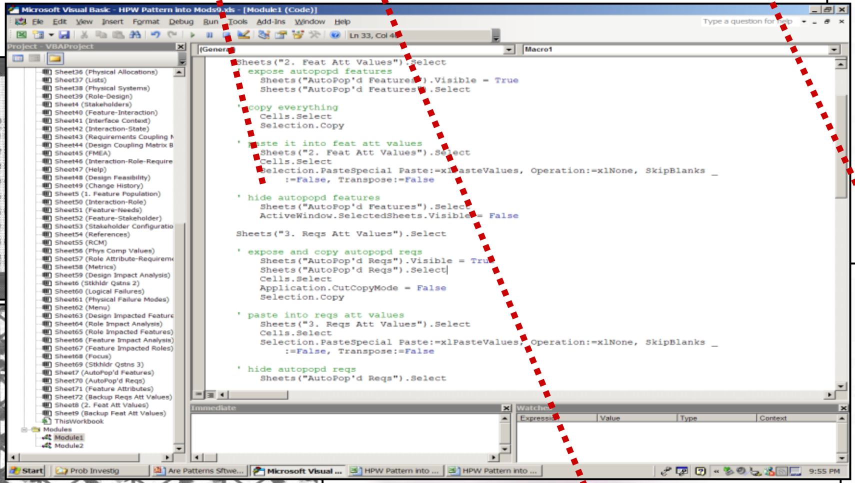
The Question: Are MBSE Patterns Software?



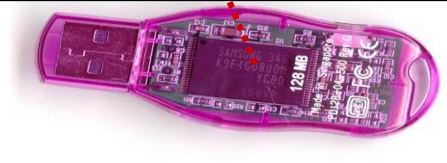


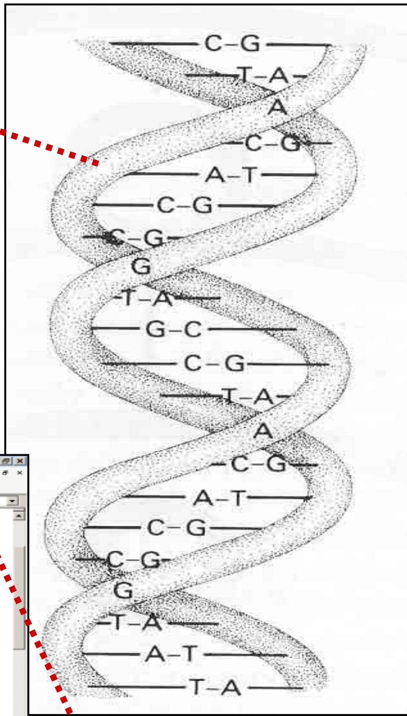


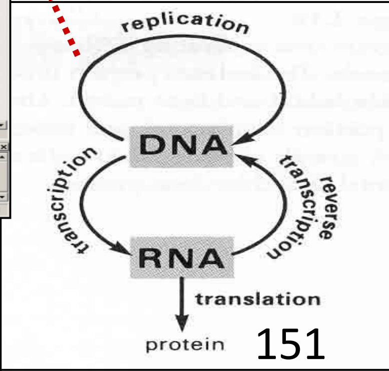




```
Sheets("2. Feat Att Values").Select
' expose autopopd features
    Sheets("AutoPop'd Features").Visible = True
    Sheets("AutoPop'd Features").Select

' copy everything
    Cells.Select
    Selection.Copy

' paste it into feat att values
    Sheets("2. Feat Att Values").Select
    Cells.Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False

' hide autopopd features
    Sheets("AutoPop'd Features").Select
    ActiveWindow.SelectedSheets.Visible = False

Sheets("3. Reqs Att Values").Select

' expose and copy autopopd reqs
    Sheets("AutoPop'd Reqs").Visible = True
    Sheets("AutoPop'd Reqs").Select
    Cells.Select
    Application.CutCopyMode = False
    Selection.Copy

' paste into reqs att values
    Sheets("3. Reqs Att Values").Select
    Cells.Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False

' hide autopopd reqs
    Sheets("AutoPop'd Reqs").Select
```

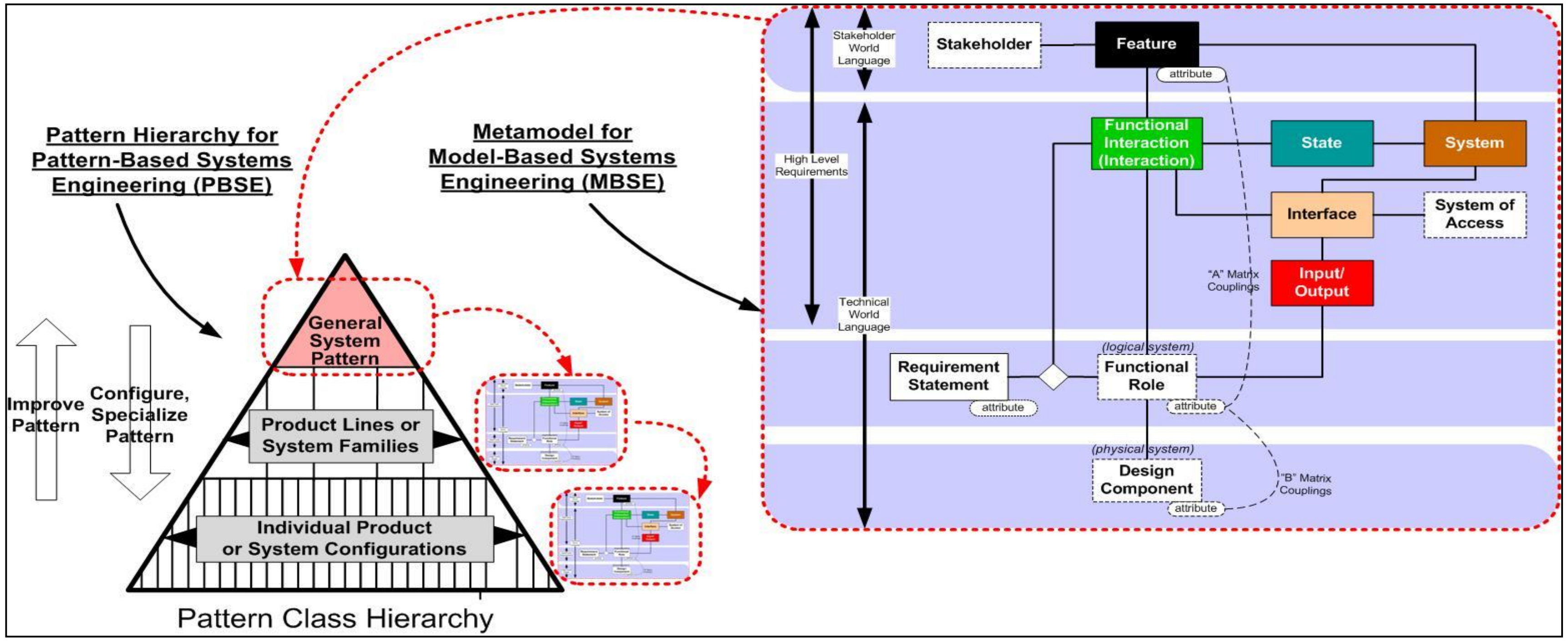






What are MBSE Patterns?

- S*Models are *explicit* descriptions of systems:
  - Their Requirements, Design, and other aspects
  - Using data structures as models.
- S*Patterns are re-usable, configurable Models.



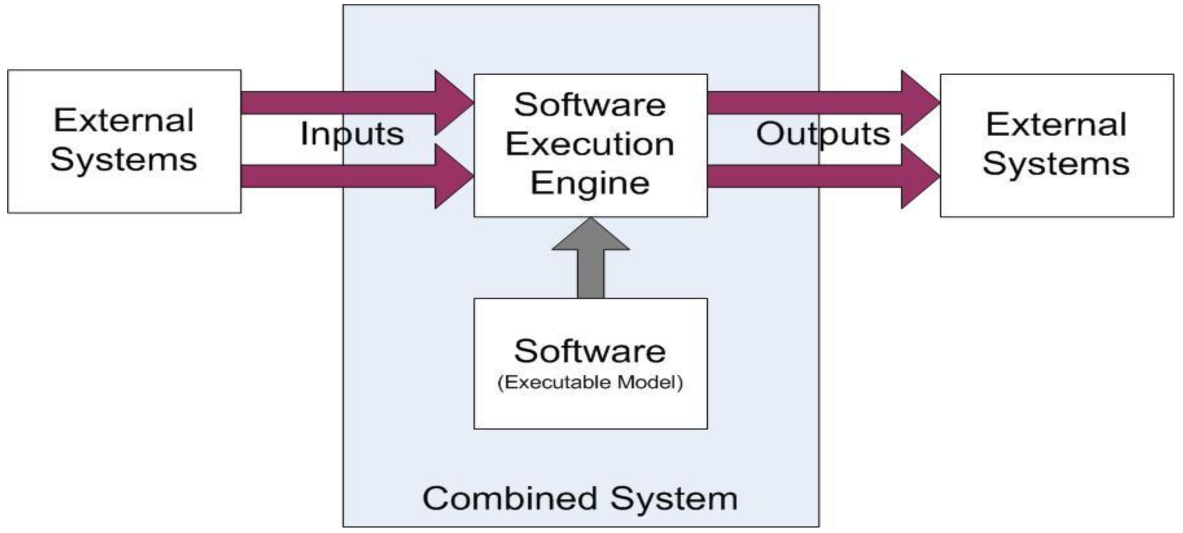
What is Software?

"It cannot be software unless it is written by a computer programmer in ALGOL 68 . . . "

Let's step back and gain a better perspective . . .

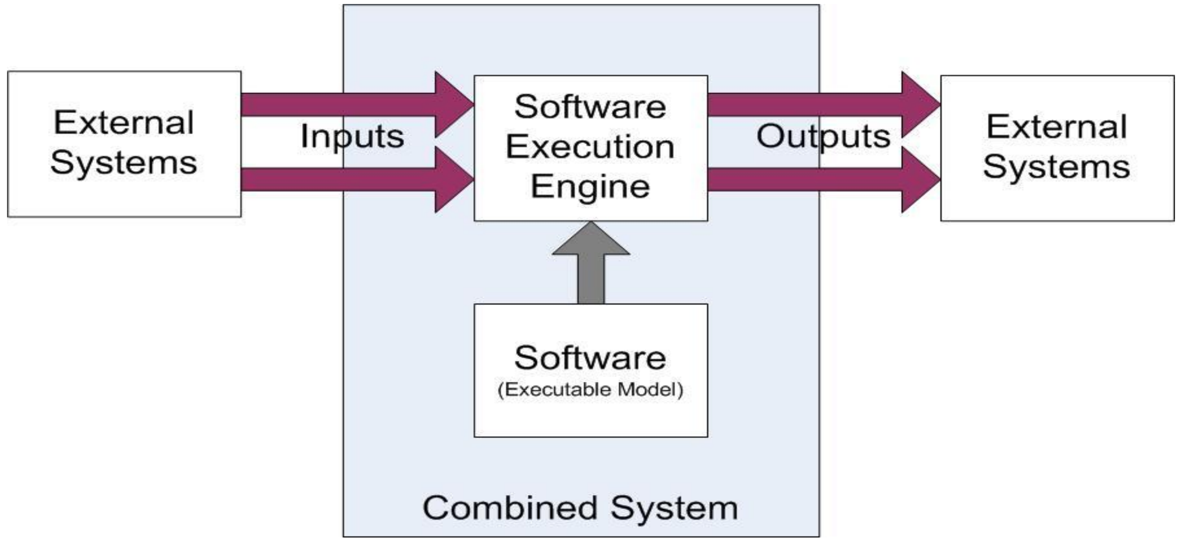
What is Software?

- Software is a special type of information:
  - Software unambiguously specifies the behavior, structure, and other aspects of certain types of systems.
  - Software is always “paired” with something that can interpret, or “execute”, the software.
    - Most typically -- a “Computer”
  - So, software is an executable (interpretable) model.



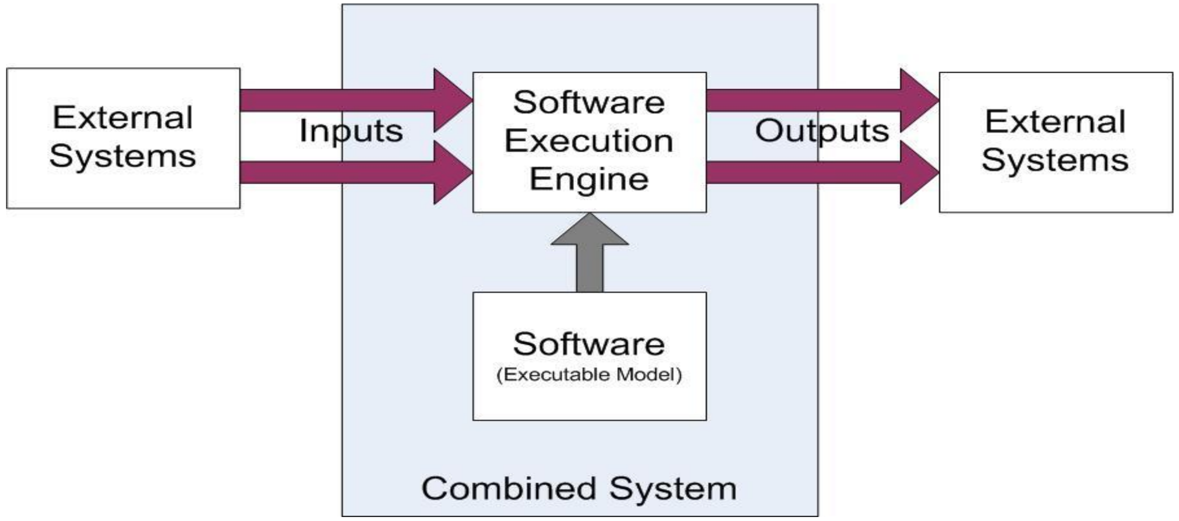
What is Software?

- The “execution engines” that interpret software transform Inputs into Outputs, under control of the Software:
  - These Inputs and Outputs can be Information, Mass, Force, or Energy.



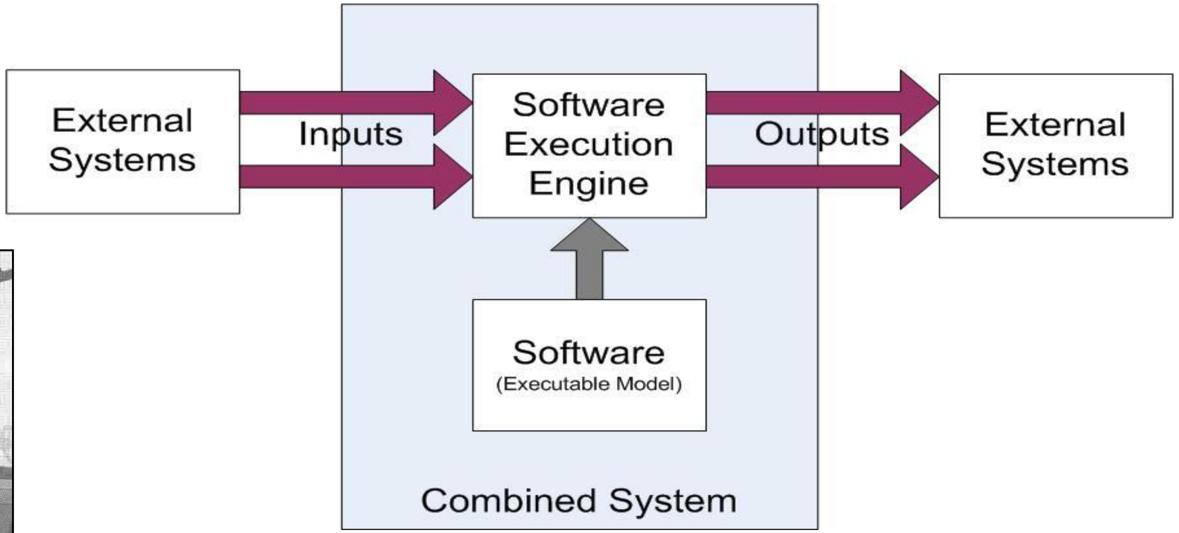
What is Software?

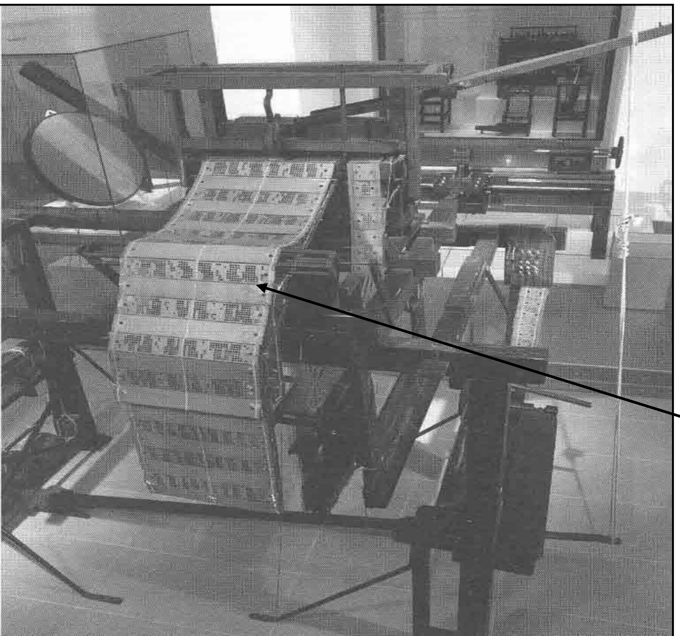
- The most familiar thing that can execute software is a “General Purpose Computer”.
  - But, it is not the only thing that can execute a model
  - And, there are many “data structures” that can represent the model . . . .



History of the technology tells us

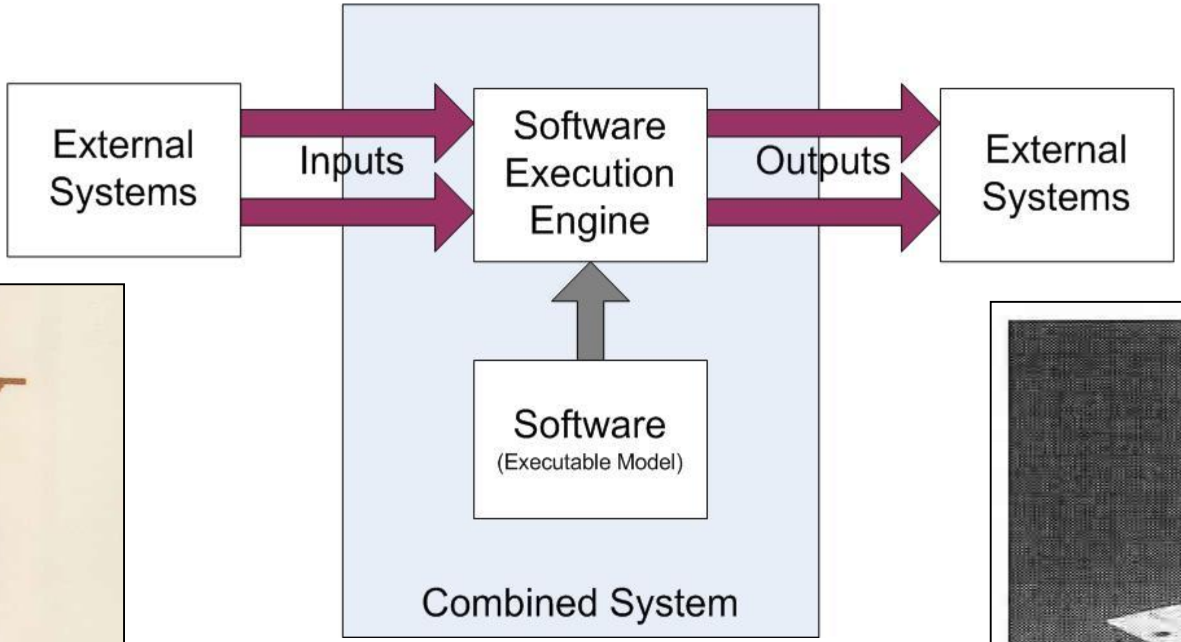
- The Jacquard Loom was programmed with an early version of punched cards to drive its weaving of textile patterns—a revolution in textiles. (1804)

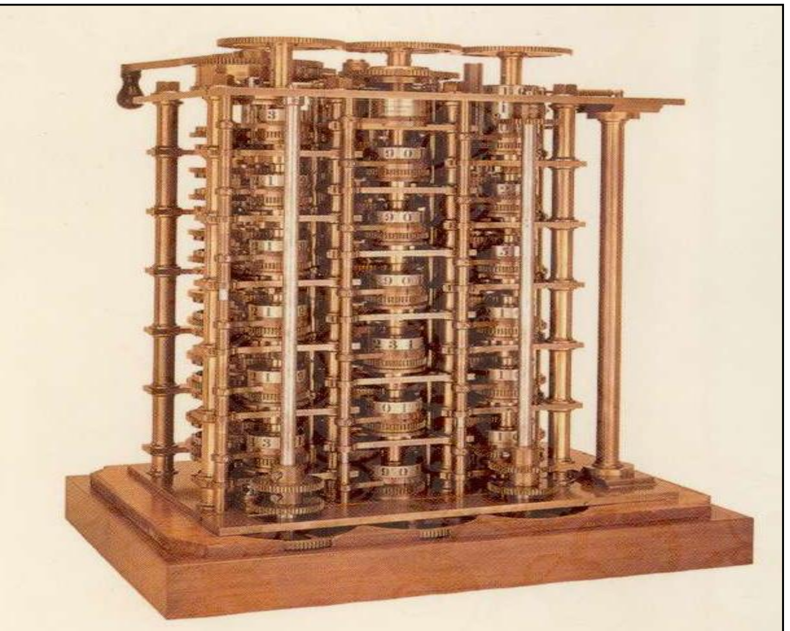


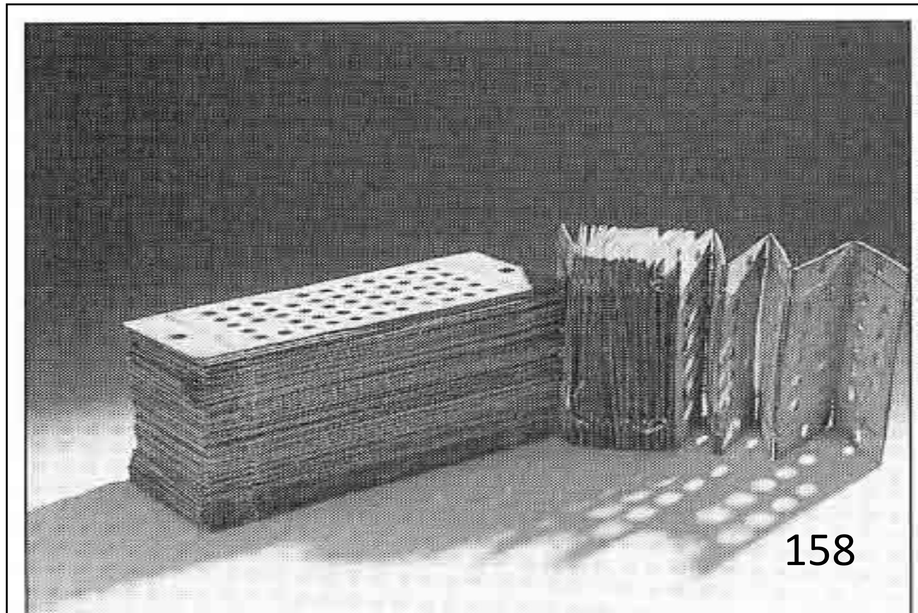


History of the technology tells us

- Charles Babbage designed the Difference Engine and the Analytical Engine, programmed by another form of punched cards to drive arithmetic calculations. (1821)

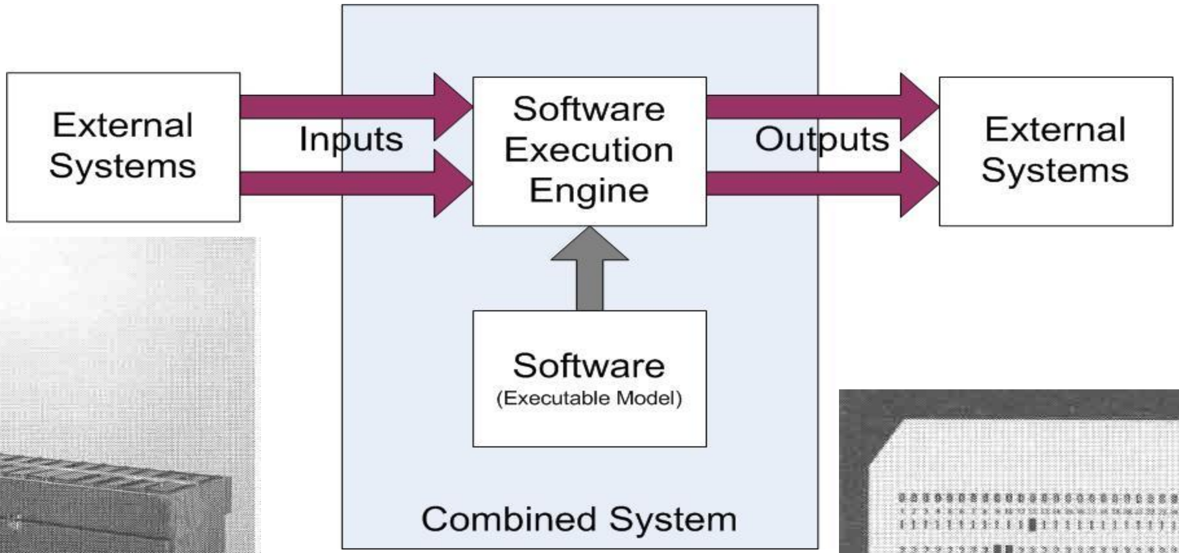


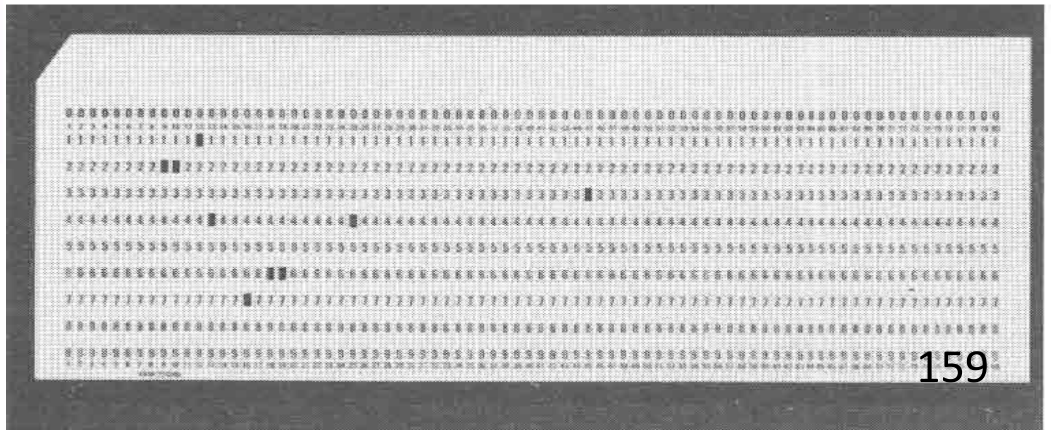




History of the technology tells us

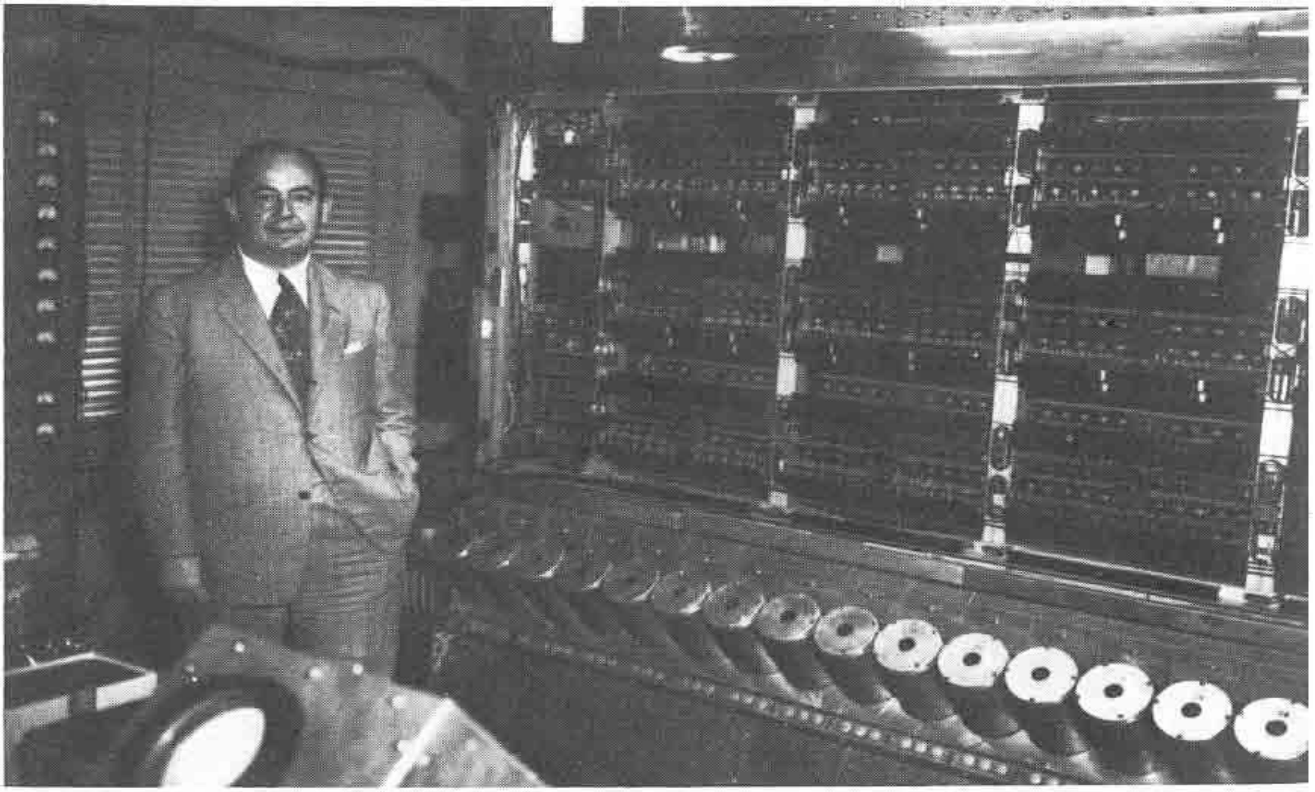
- Herman Hollerith "re-invented" the punched card to develop mechanical and electrical sorters, tabulators, counters for statistical counting, leading to IBM and others. (1900)
- The "programs" for these machines were typically in wired plugboard information, with the cards used for inputs and outputs.

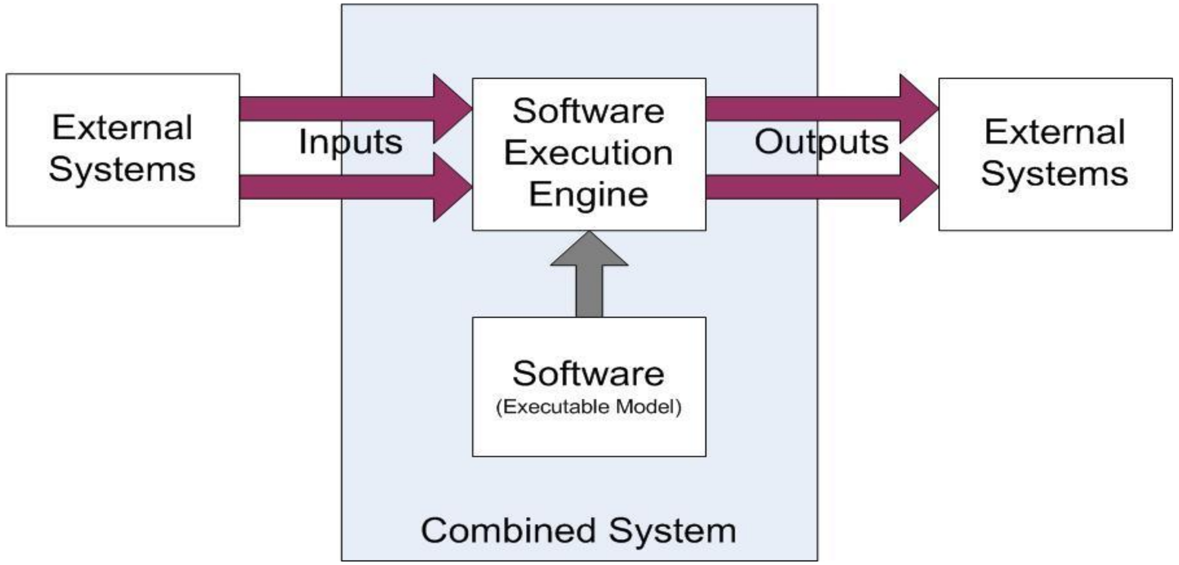




History of the technology tells us

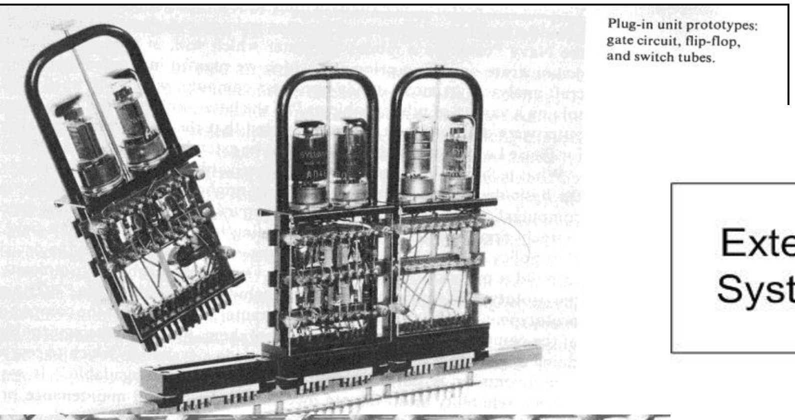
- John von Neumann and others developed the idea of storing the program information as part of the machine's other data—but did not invent the idea of program information, which was much older. (1940s)

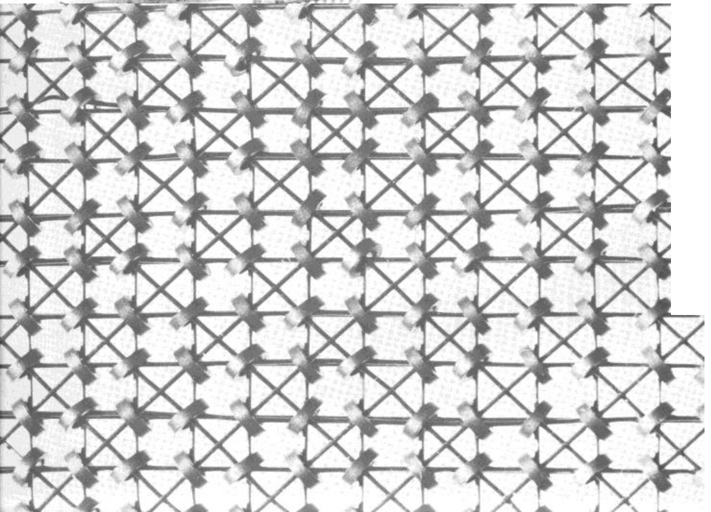


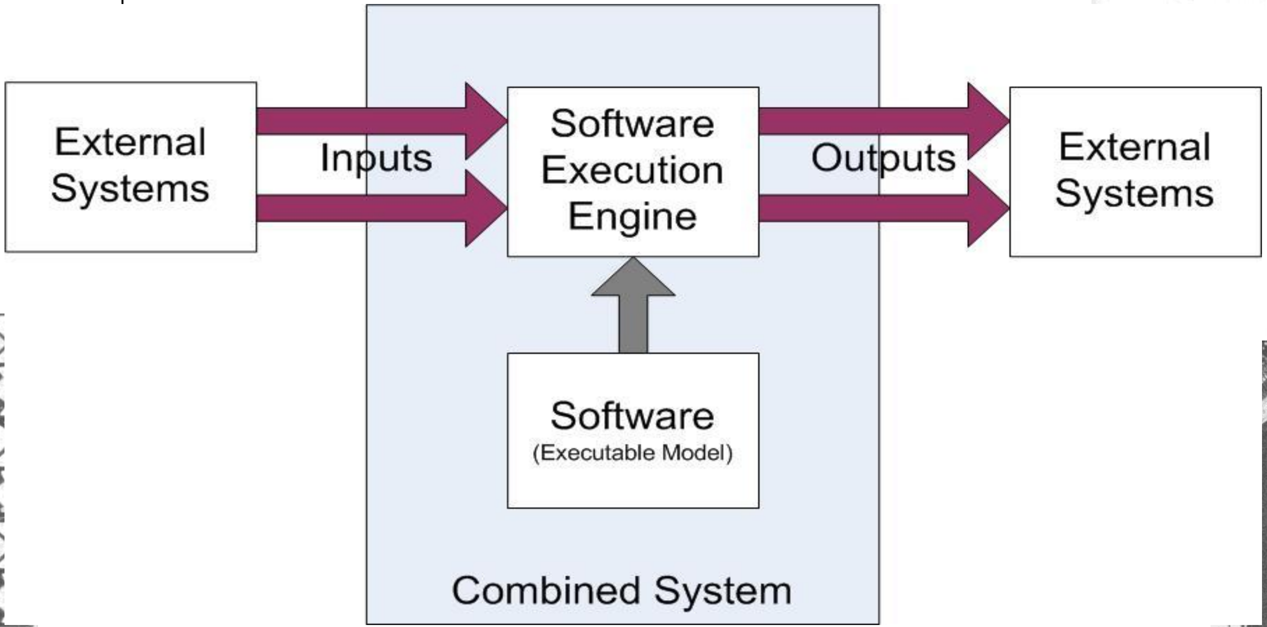


What history of technology tells us

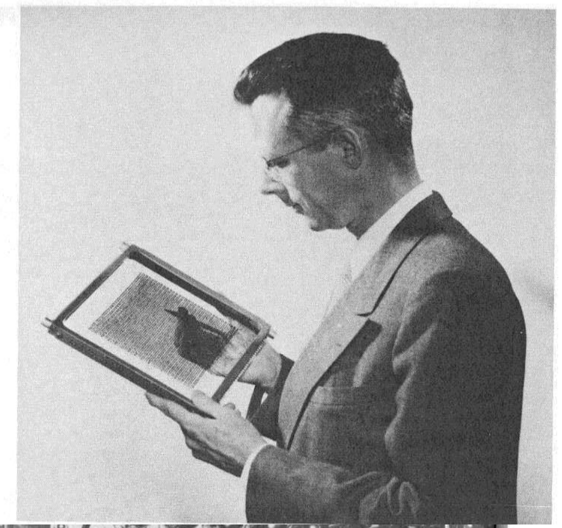
- Jay Forrester moved program data into magnetic core storage. (1950s)

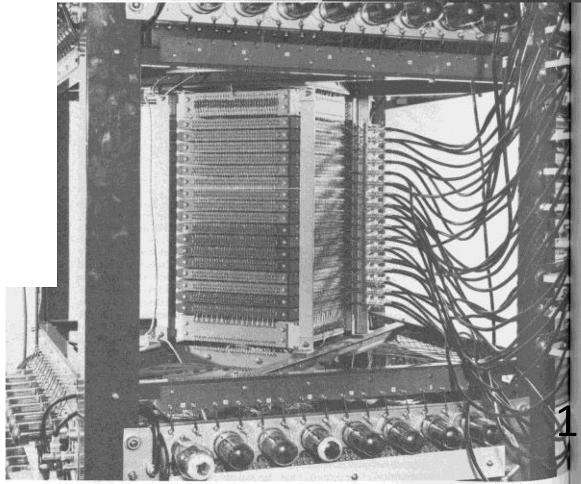

Plug-in unit prototypes: gate circuit, flip-flop, and switch tubes.





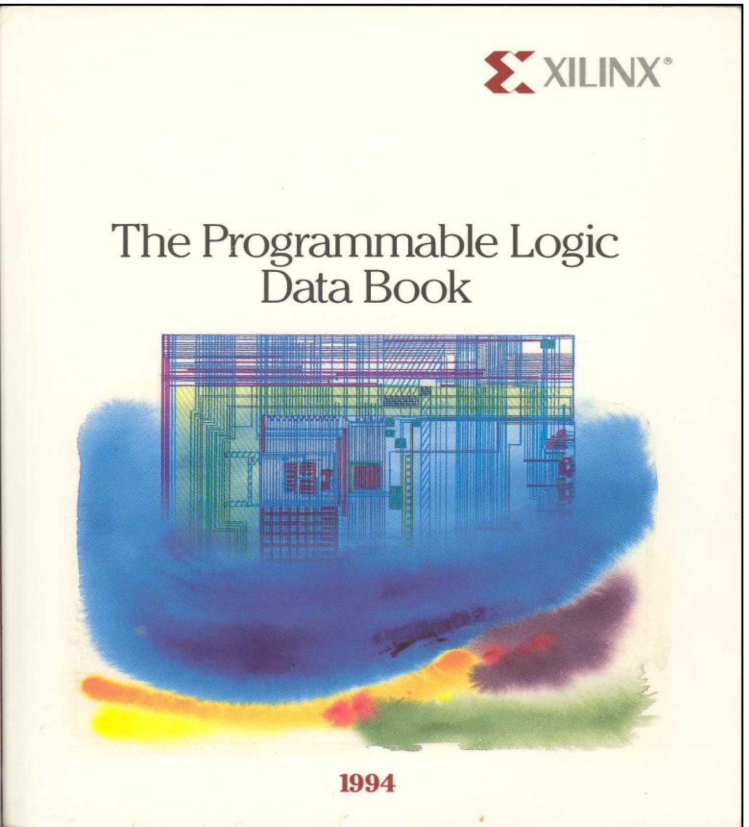
Forrester with a 64 by 64 core memory plane.

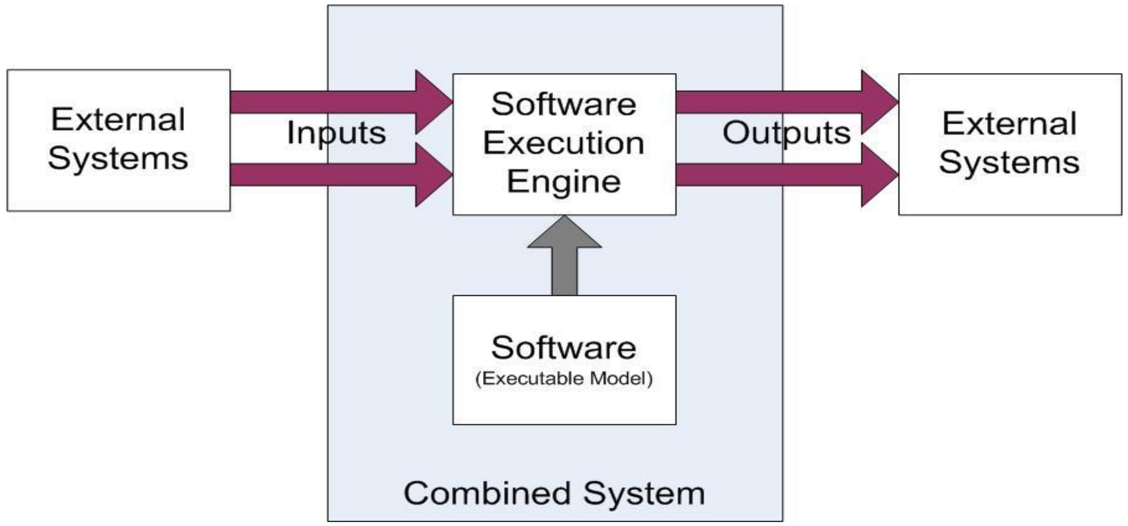



Close-up of a magnetic memory bank. Two memory banks were installed in Whirlwind in 1953, increasing its storage to 2,048 words of 16 bits each.

History of the technology tells us

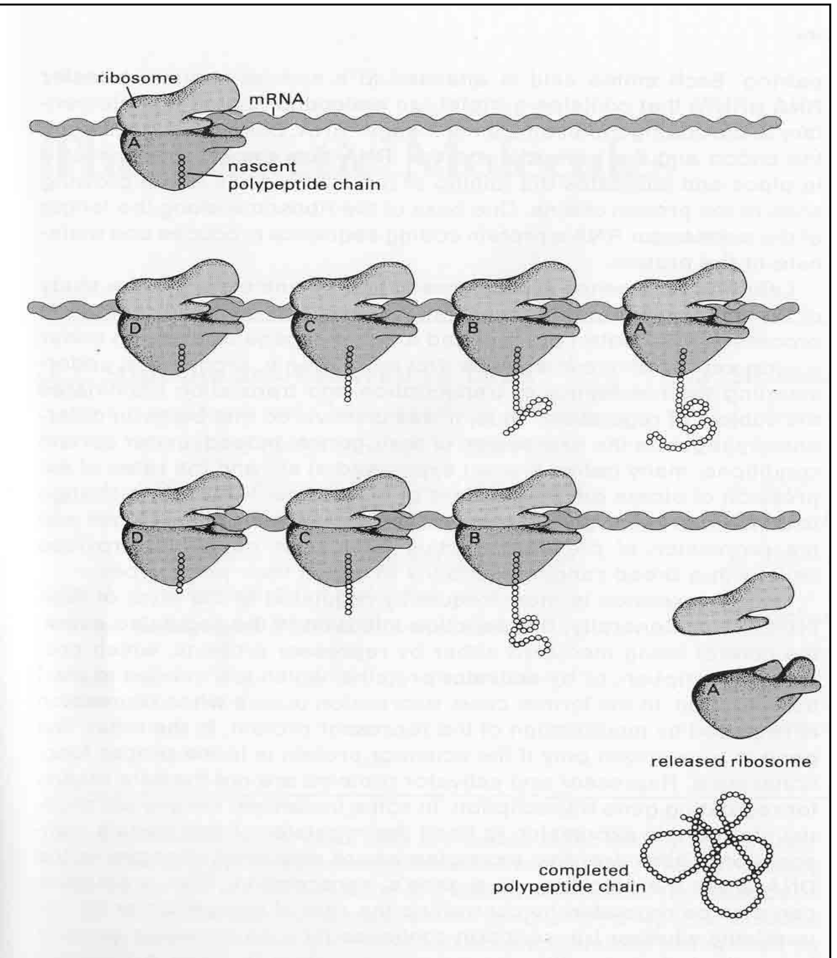
- Xilinx and other electronic hardware vendors develop “configurable hardware”: the idea of storing information as hardware, in very large scale high speed processors—ASICs and FPGAs.
- Other vendors developed VHDL, HDL, and RTL languages to define and test high complexity chip hardware.

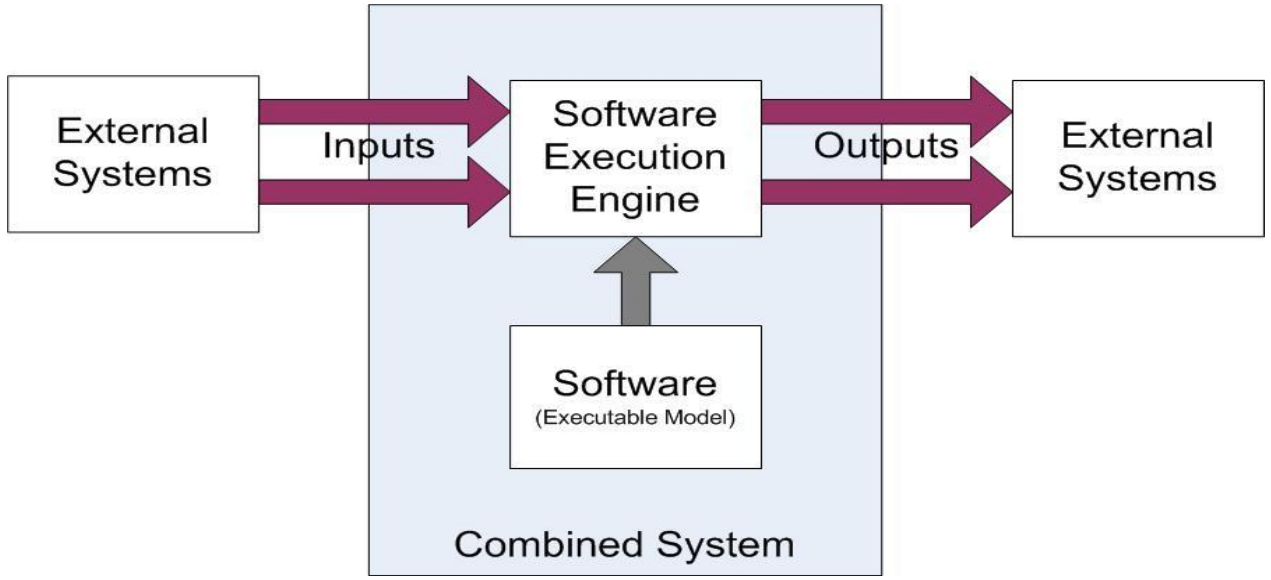


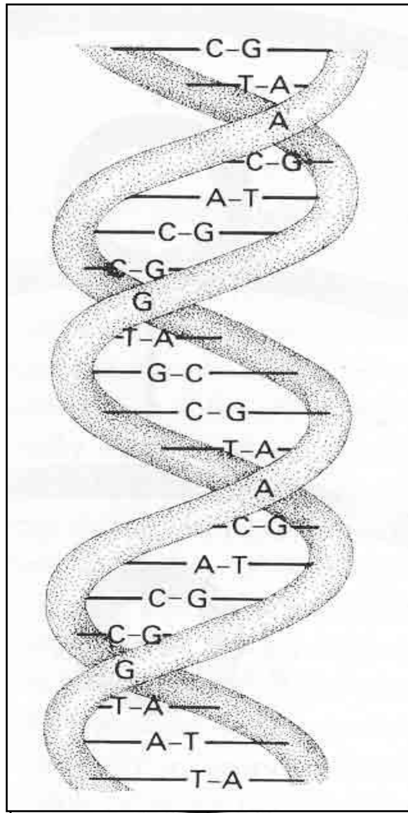


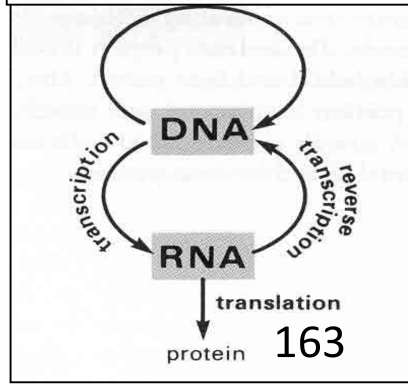
What the natural world tells us

- In nature, information stored in DNA is replicated, transcribed, and then used by ribosomes to generate protein molecular structures in living “epigenetic” systems.



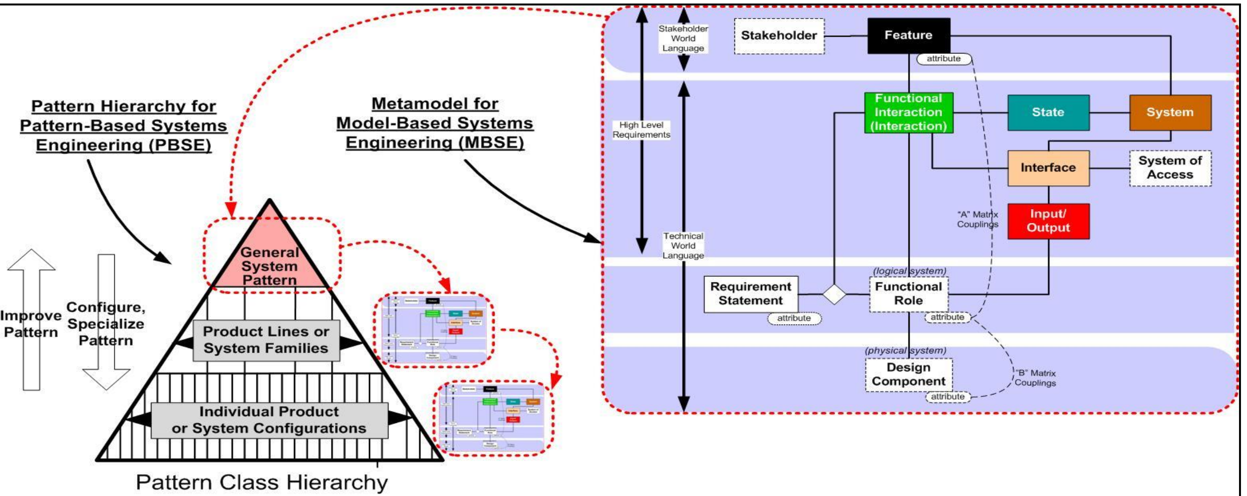


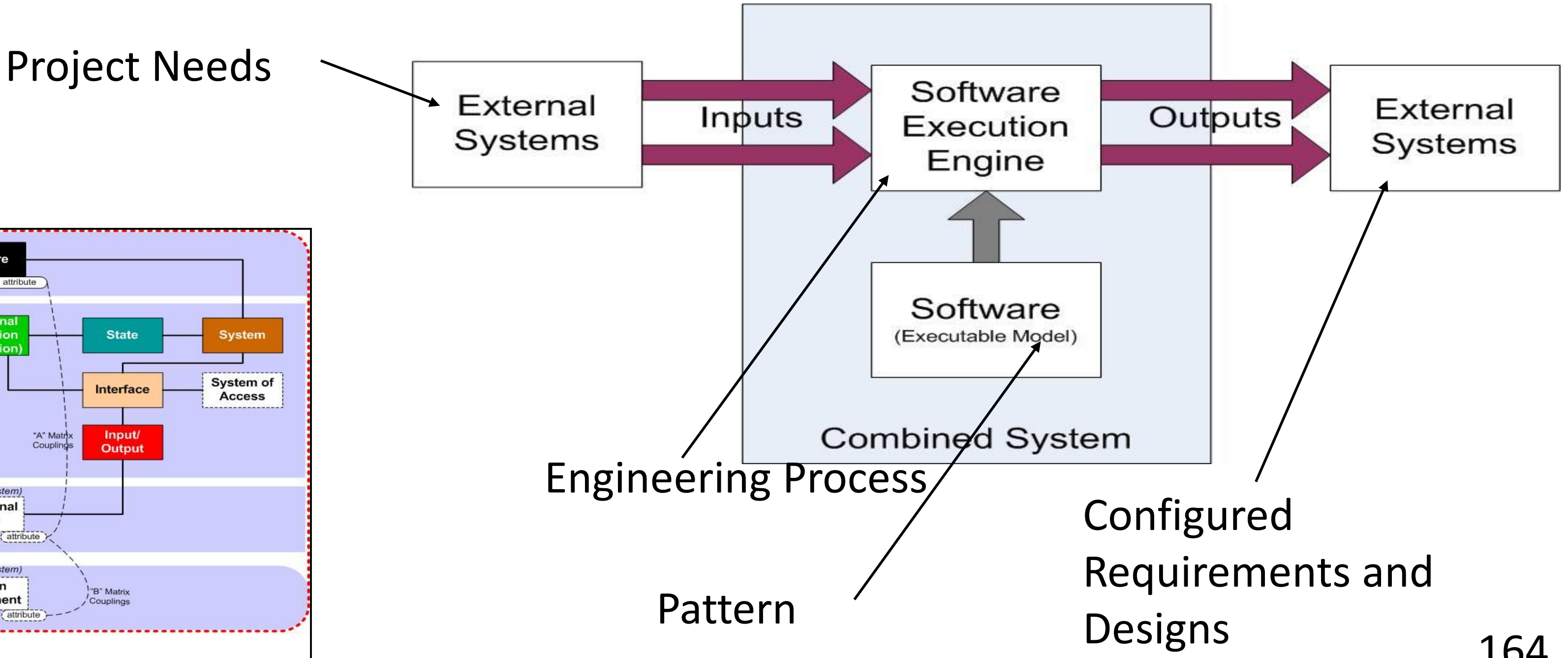




An engineering process application

- Even the engineering process itself (along with its internal tools) is such an engine—using configured models to produce requirements and designs of new systems, in a never-ending cycle.

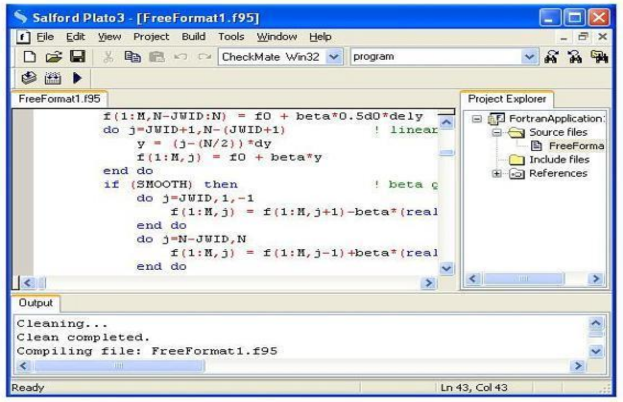




All these programming languages are themselves data structures – compiled or even run time interpreted by other programs

Software Languages as Data Structures

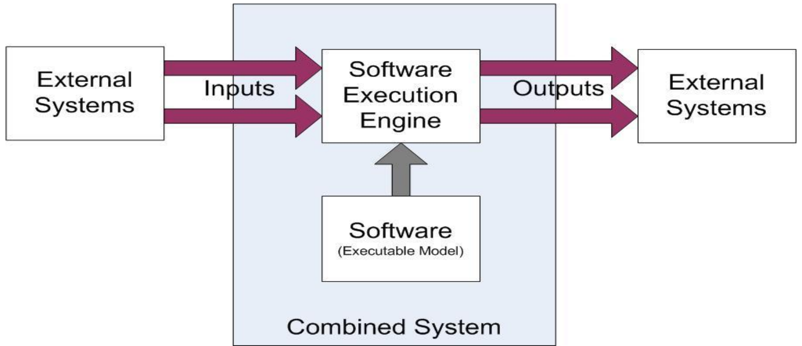
- FORTRAN (Formula Translation) Language:
  - A procedural programming language invented by John Backus to express mathematical formulas. (1950's)

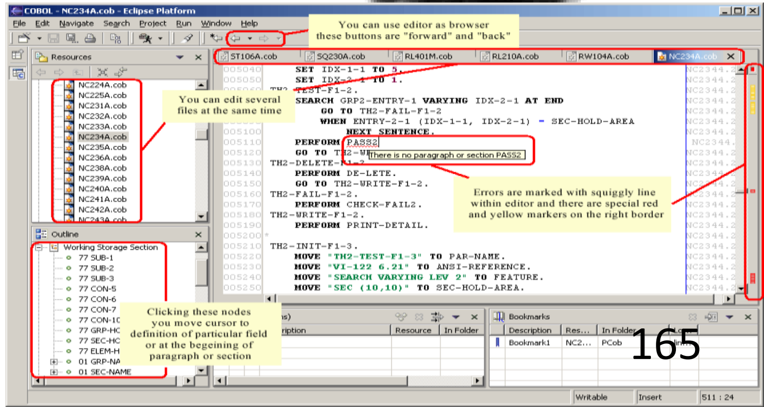


```
        f(1:M,N-JWID:N) = f0 + beta*0.5d0*dely
        do j=JWID+1,N-(JWID+1)                    ! linear
              y = (j-(N/2))*dy
              f(1:M,j) = f0 + beta*y
        end do
        if (SMOOTH) then                          ! beta g
              do j=JWID,1,-1
                    f(1:M,j) = f(1:M,j+1)-beta*(real
              end do
              do j=N-JWID,N
                    f(1:M,j) = f(1:M,j-1)+beta*(real
              end do
```

```
Cleaning...
Clean completed.
Compiling file: FreeFormat1.f95
```

- COBOL (Common Business Oriented) Language:
  - A procedural language invented by Admiral Grace Hopper to express business algorithms. (1950-60's)





You can use editor as browser these buttons are "forward" and "back"

You can edit several files at the same time

Errors are marked with squiggly line within editor and there are special red and yellow markers on the right border

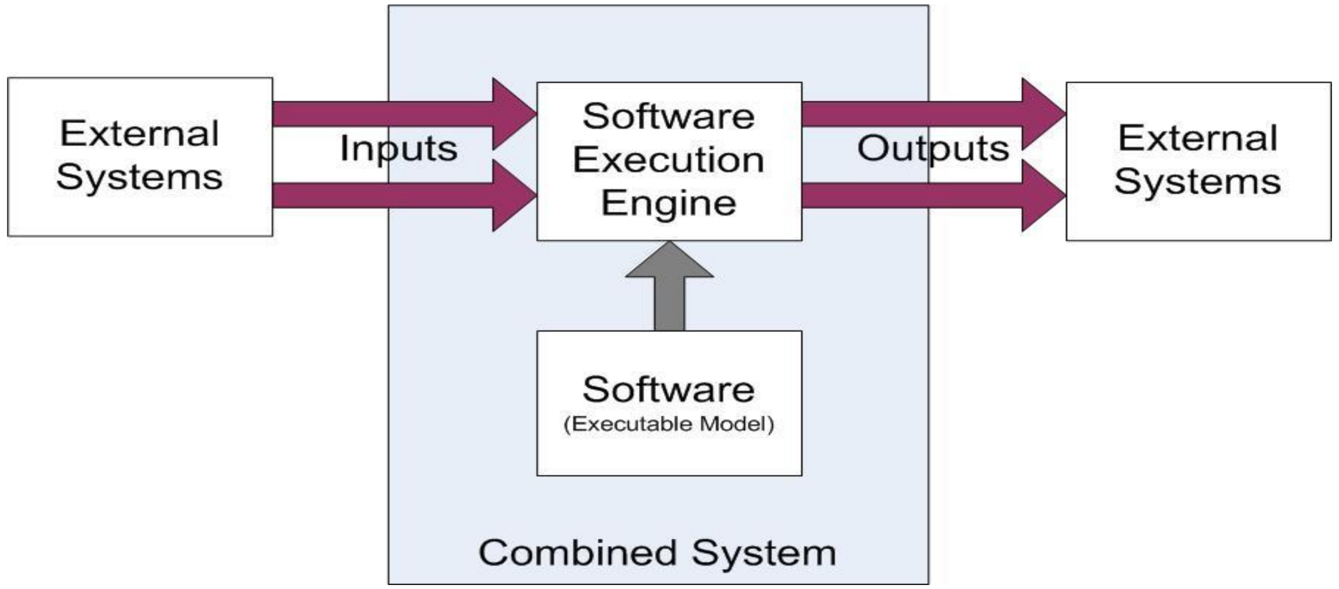
Clicking these nodes you move cursor to definition of particular field or at the begeining of paragraph or section

```
005040      SET IDX-1-1 TO 5.
005050      SET IDX-2-1 TO 1.
       TH2-TEST-F1-2.
            SEARCH GRP2-ENTRY-1 VARYING IDX-2-1 AT END
                GO TO TH2-FAIL-F1-2
                WHEN ENTRY-2-1 (IDX-1-1, IDX-2-1) = SEC-HOLD-AREA
                     NEXT SENTENCE.
005110      PERFORM PASS2
005120      GO TO TH2-WR
005130 TH2-DELETE-F1-2.
005140      PERFORM DE-LETE.
005150      GO TO TH2-WRITE-F1-2.
005160 TH2-FAIL-F1-2.
005170      PERFORM CHECK-FAIL2.
005180 TH2-WRITE-F1-2.
005190      PERFORM PRINT-DETAIL.
005200*
005210 TH2-INIT-F1-3.
005220      MOVE "TH2-TEST-F1-3" TO PAR-NAME.
005230      MOVE "VI-122 6.21" TO ANSI-REFERENCE.
005240      MOVE "SEARCH VARYING LEV 2" TO FEATURE.
005250      MOVE "SEC (10,10)" TO SEC-HOLD-AREA.
```

There is no paragraph or section PASS2

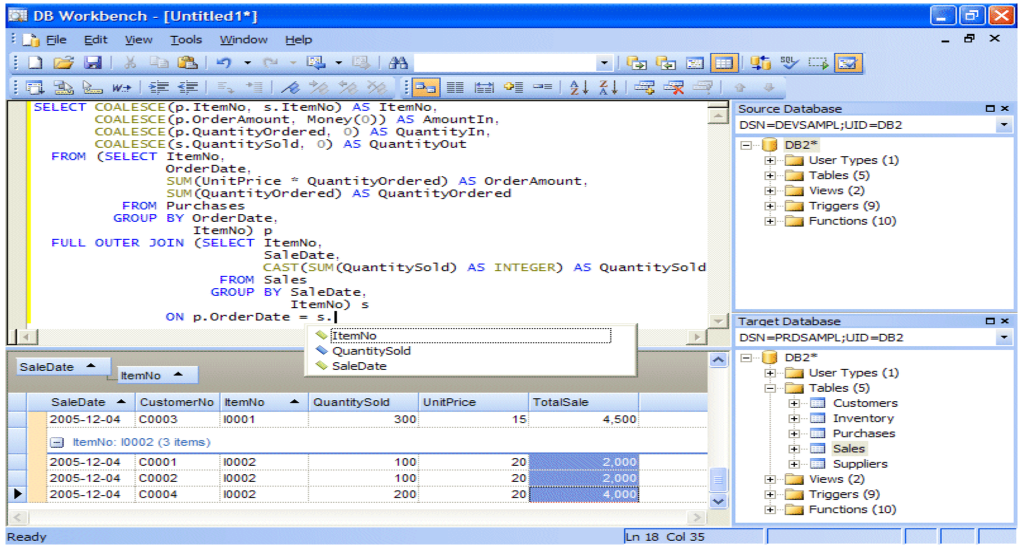
Not all software describes procedure

- It is not even safe to say that "software describes a sequence of operations" –
  - Because all Non-Procedural Languages are precisely <u>not</u> procedures!
  - Examples: SQL, XML, SCHEME, etc.



All these programming languages are themselves data structures – compiled or interpreted by other programs

Non-Procedural Software Languages



DB Workbench - [Untitled1*]

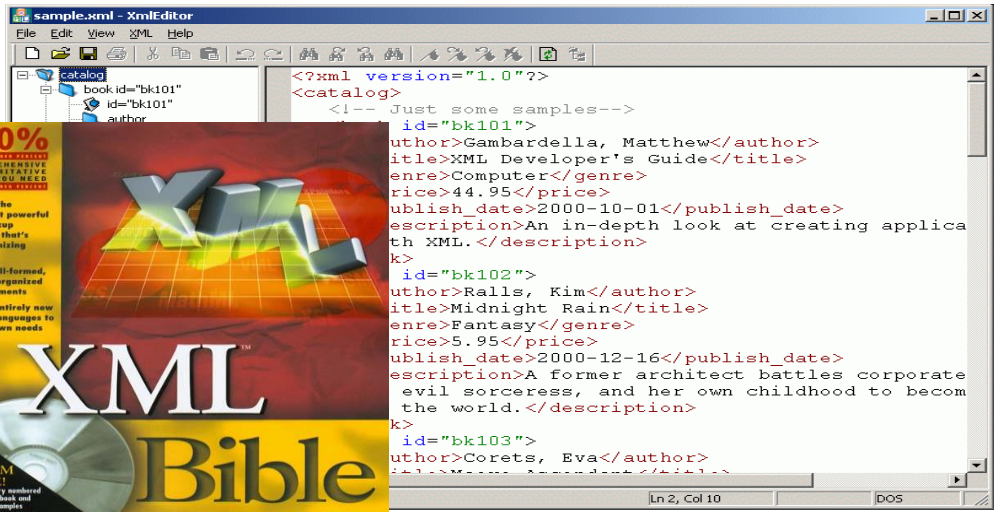
```
SELECT COALESCE(p.ItemNo, s.ItemNo) AS ItemNo,
       COALESCE(p.OrderAmount, Money(0)) AS AmountIn,
       COALESCE(p.QuantityOrdered, 0) AS QuantityIn,
       COALESCE(s.QuantitySold, 0) AS QuantityOut
  FROM (SELECT ItemNo,
               OrderDate,
               SUM(UnitPrice * QuantityOrdered) AS OrderAmount,
               SUM(QuantityOrdered) AS QuantityOrdered
          FROM Purchases
         GROUP BY OrderDate,
                  ItemNo) p
  FULL OUTER JOIN (SELECT ItemNo,
                          SaleDate,
                          CAST(SUM(QuantitySold) AS INTEGER) AS QuantitySold
                     FROM Sales
                    GROUP BY SaleDate,
                             ItemNo) s
               ON p.OrderDate = s.
```

SaleDate	CustomerNo	ItemNo	QuantitySold	UnitPrice	TotalSale
2005-12-04	C0003	I0001	300	15	4,500
ItemNo: I0002 (3 items)					
2005-12-04	C0001	I0002	100	20	2,000
2005-12-04	C0002	I0002	100	20	2,000
2005-12-04	C0004	I0002	200	20	4,000

Source Database: DSN=DEVSAMPL;UID=DB2
DB2*: User Types (1), Tables (5), Views (2), Triggers (9), Functions (10)

Target Database: DSN=PRDSAMPL;UID=DB2
DB2*: User Types (1), Tables (5): Customers, Inventory, Purchases, Sales, Suppliers; Views (2), Triggers (9), Functions (10)

Ready Ln 18 Col 35

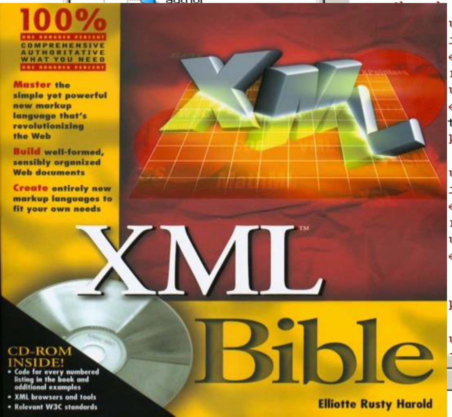


sample.xml - XmlEditor

```
<?xml version="1.0"?>
<catalog>
   <!-- Just some samples-->
 id="bk101">
uthor>Gambardella, Matthew</author>
itle>XML Developer's Guide</title>
enre>Computer</genre>
rice>44.95</price>
ublish_date>2000-10-01</publish_date>
escription>An in-depth look at creating applica
th XML.</description>
k>
 id="bk102">
uthor>Ralls, Kim</author>
itle>Midnight Rain</title>
enre>Fantasy</genre>
rice>5.95</price>
ublish_date>2000-12-16</publish_date>
escription>A former architect battles corporate
 evil sorceress, and her own childhood to becom
 the world.</description>
k>
 id="bk103">
uthor>Corets, Eva</author>
```

Ln 2, Col 10 DOS

File Edit View XML Help
catalog
book id="bk101"
id="bk101"
author



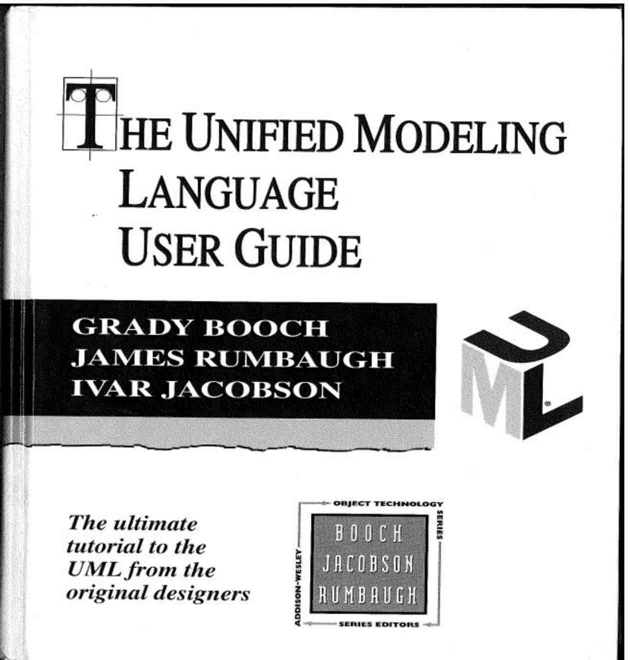
100%
Master the simple yet powerful new markup language that's revolutionizing the Web
Build well-formed, sensibly organized Web documents
Create entirely new markup languages to fit your own needs
XML Bible
CD-ROM INSIDE!
Elliotte Rusty Harold

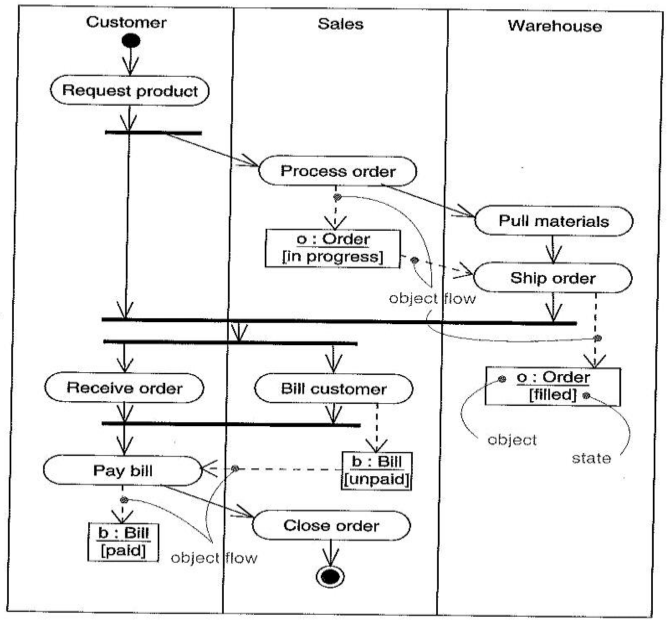
- These programming languages express the relationship of output data to input data without intermediate algorithms (D. Parnas):
  - SQL (Structured Query Language)
    - Invented by Codd and Date to express relational data models and operations on them.   (1970's)
  - XML (Extensible Modeling Language)
    - Invented to express data models and transformations, as an evolution of SGML (1990's)
    - The foundation of many additional languages (e.g., Molecular Modeling Language, etc.)

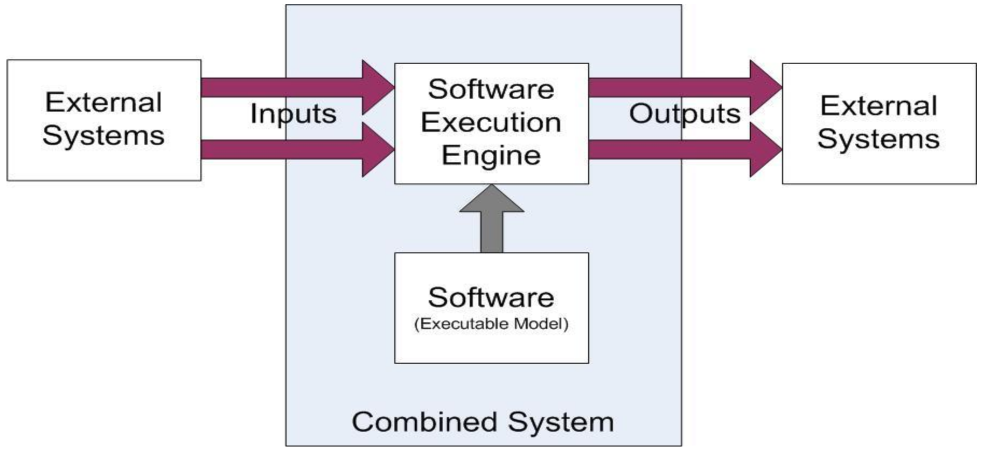
All these programming languages are themselves data structures – compiled or interpreted by other programs

Model-Defined Software

- More and more "traditional" software is now being developed by expressing both requirements and design in graphical data structures called "Models":
  - UML™ (Unified Modeling Language) is the most popular current example (Booch, Rumbaugh, Jacobson; OMG™)



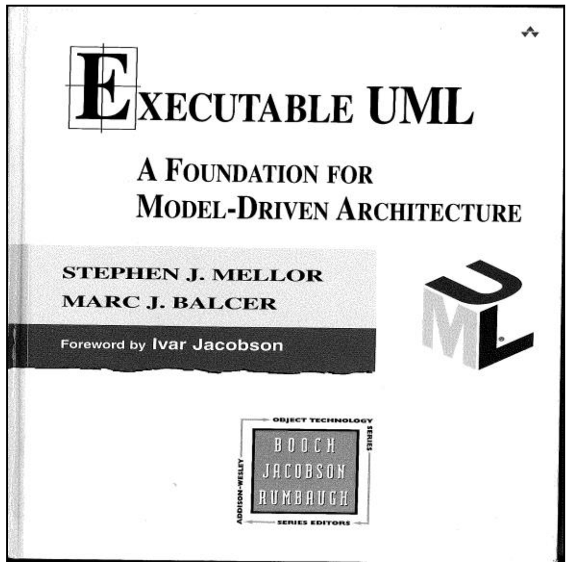


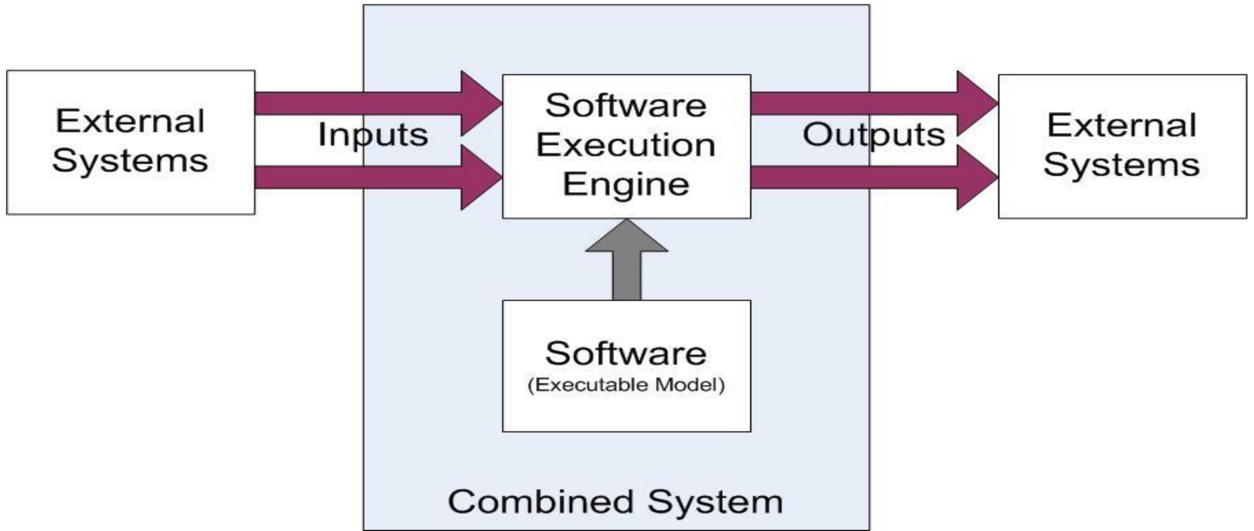


All these languages are themselves data structures – compiled or interpreted by other programs

Executable Models

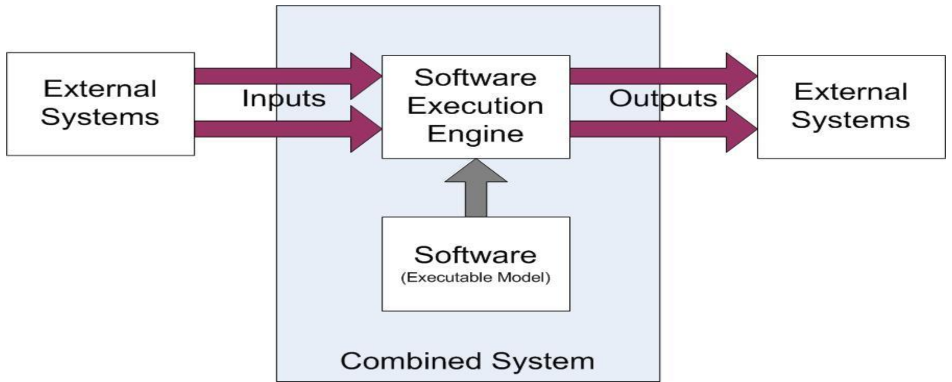
- Many "executable" models are being generated:
  - For traditional simulators (e.g., MATLAB™, etc.)
  - For requirements validation simulations (e.g., STATE MATE™, etc.)
  - For dual use as both source code generation as well as simulation execution (e.g., RHAPSODY™, etc.)





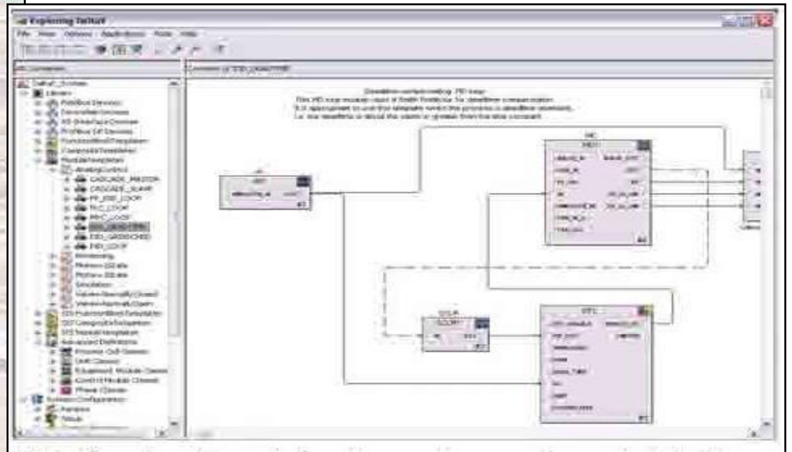
Not all software is “executed by hardware”

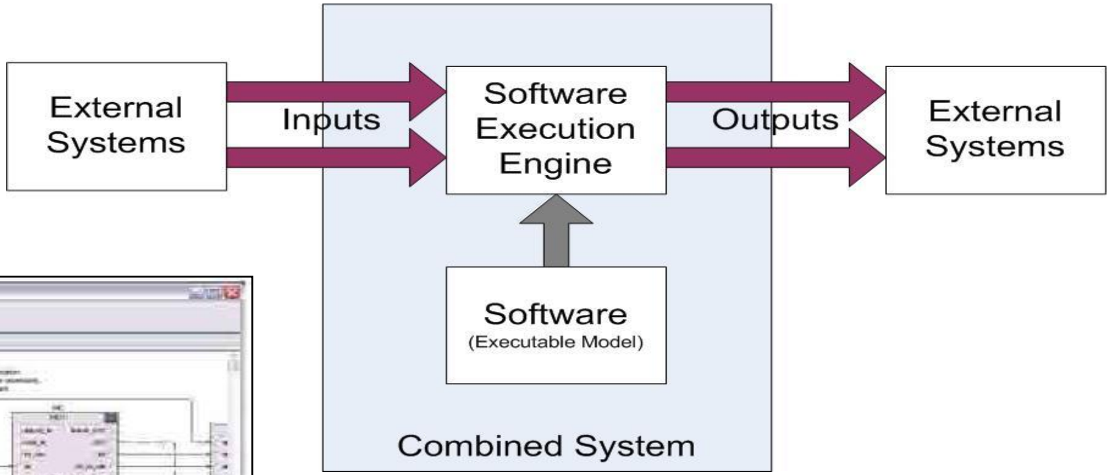
- Interpreted languages are very common:
  - e.g., BASIC is typically interpreted by software interpreters
  - “Virtual machines” are used to “execute” Java, etc.
  - This “code” is “executed” by other programs, not hardware!
- Microcoded emulators in chips:
  - Most modern microprocessors, PCs, servers, mainframes use “microcode” and are really “interpreters”.
- Emulators:
  - Many software and hardware debugging products use emulation by other code to “execute” the code being developed
- Spread sheets allow expression of complex relationships that are executed by software engines (e.g., Excel™, etc.)



An engineering process application

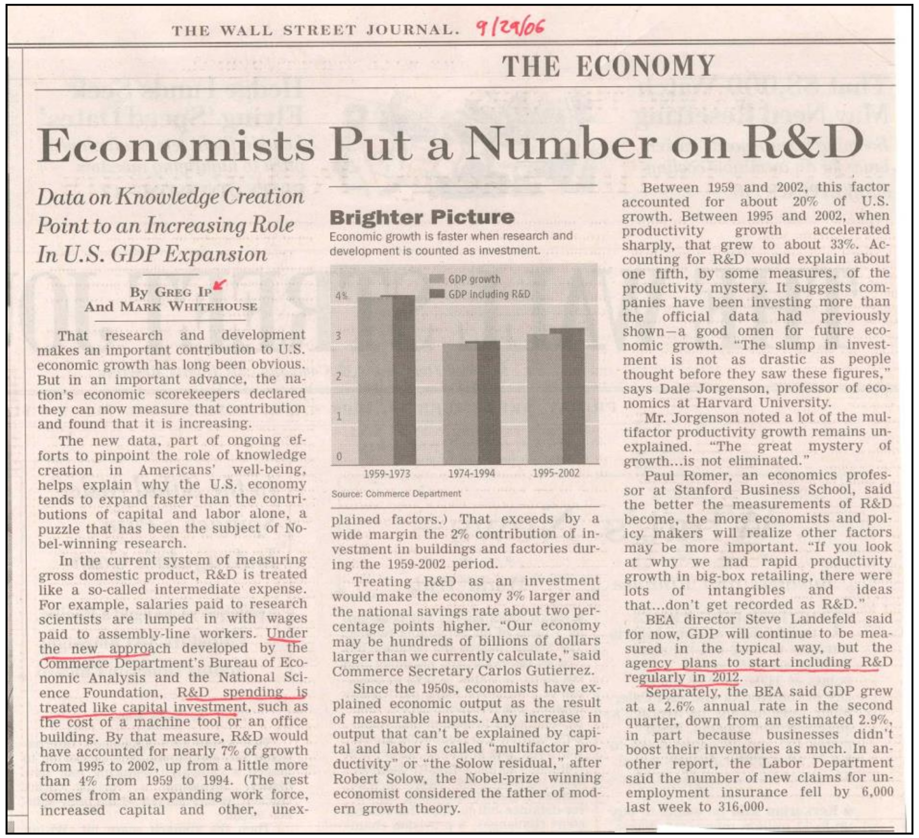
- Control systems suppliers and products (e.g., Emerson DeltaV™) now allow us to “program” our control systems using data structures created by inputting models.
- Many other programs are also created this way.





The economy itself becoming IP based

- US Government and Economists adopt increased capitalization of developed information assets:



THE WALL STREET JOURNAL. 9/29/06

THE ECONOMY

Economists Put a Number on R&D

Data on Knowledge Creation Point to an Increasing Role In U.S. GDP Expansion

By Greg Ip
And Mark Whitehouse

That research and development makes an important contribution to U.S. economic growth has long been obvious. But in an important advance, the nation's economic scorekeepers declared they can now measure that contribution and found that it is increasing.

The new data, part of ongoing efforts to pinpoint the role of knowledge creation in Americans' well-being, helps explain why the U.S. economy tends to expand faster than the contributions of capital and labor alone, a puzzle that has been the subject of Nobel-winning research.

In the current system of measuring gross domestic product, R&D is treated like a so-called intermediate expense. For example, salaries paid to research scientists are lumped in with wages paid to assembly-line workers. Under the new approach developed by the Commerce Department's Bureau of Economic Analysis and the National Science Foundation, R&D spending is treated like capital investment, such as the cost of a machine tool or an office building. By that measure, R&D would have accounted for nearly 7% of growth from 1995 to 2002, up from a little more than 4% from 1959 to 1994. (The rest comes from an expanding work force, increased capital and other, unexplained factors.) That exceeds by a wide margin the 2% contribution of investment in buildings and factories during the 1959-2002 period.

Treating R&D as an investment would make the economy 3% larger and the national savings rate about two percentage points higher. "Our economy may be hundreds of billions of dollars larger than we currently calculate," said Commerce Secretary Carlos Gutierrez.

Since the 1950s, economists have explained economic output as the result of measurable inputs. Any increase in output that can't be explained by capital and labor is called "multifactor productivity" or "the Solow residual," after Robert Solow, the Nobel-prize winning economist considered the father of modern growth theory.

Between 1959 and 2002, this factor accounted for about 20% of U.S. growth. Between 1995 and 2002, when productivity growth accelerated sharply, that grew to about 33%. Accounting for R&D would explain about one fifth, by some measures, of the productivity mystery. It suggests companies have been investing more than the official data had previously shown—a good omen for future economic growth. "The slump in investment is not as drastic as people thought before they saw these figures," says Dale Jorgenson, professor of economics at Harvard University.

Mr. Jorgenson noted a lot of the multifactor productivity growth remains unexplained. "The great mystery of growth...is not eliminated."

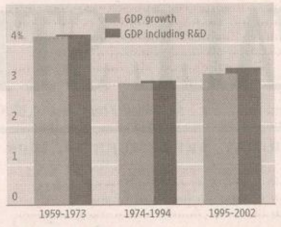
Paul Romer, an economics professor at Stanford Business School, said the better the measurements of R&D become, the more economists and policy makers will realize other factors may be more important. "If you look at why we had rapid productivity growth in big-box retailing, there were lots of intangibles and ideas that...don't get recorded as R&D."

BEA director Steve Landefeld said for now, GDP will continue to be measured in the typical way, but the agency plans to start including R&D regularly in 2012.

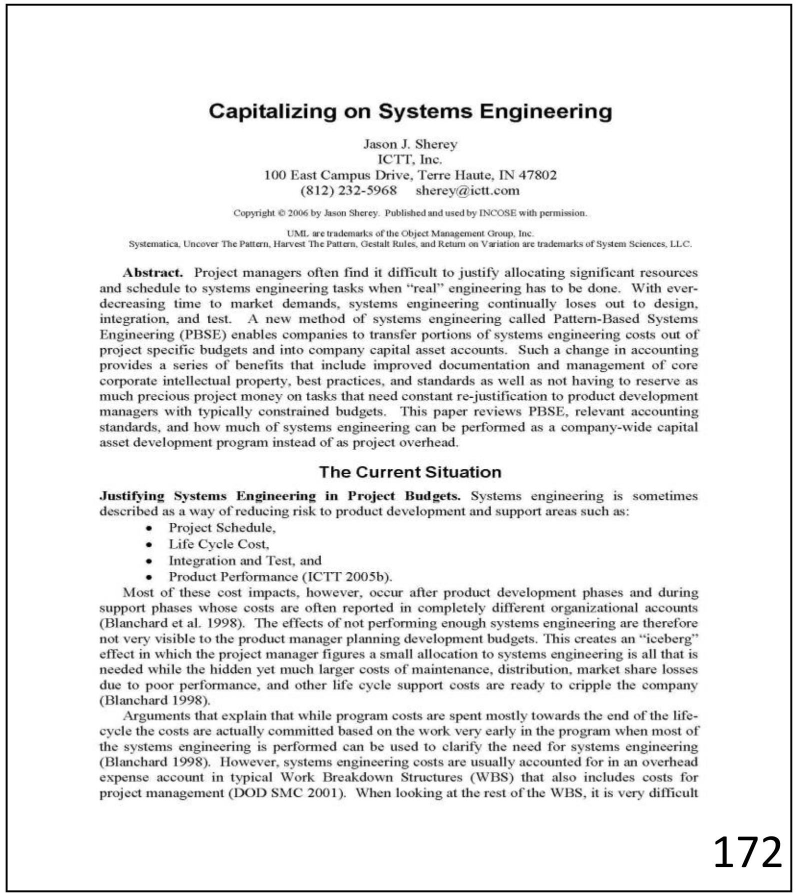
Separately, the BEA said GDP grew at a 2.6% annual rate in the second quarter, down from an estimated 2.9%, in part because businesses didn't boost their inventories as much. In another report, the Labor Department said the number of new claims for unemployment insurance fell by 6,000 last week to 316,000.

Brighter Picture

Economic growth is faster when research and development is counted as investment.



Source: Commerce Department



Capitalizing on Systems Engineering

Jason J. Sherey
ICTT, Inc.
100 East Campus Drive, Terre Haute, IN 47802
(812) 232-5968 sherey@ictt.com

Abstract. Project managers often find it difficult to justify allocating significant resources and schedule to systems engineering tasks when "real" engineering has to be done. With ever-decreasing time to market demands, systems engineering continually loses out to design, integration, and test. A new method of systems engineering called Pattern-Based Systems Engineering (PBSE) enables companies to transfer portions of systems engineering costs out of project specific budgets and into company capital asset accounts. Such a change in accounting provides a series of benefits that include improved documentation and management of core corporate intellectual property, best practices, and standards as well as not having to reserve as much precious project money on tasks that need constant re-justification to product development managers with typically constrained budgets. This paper reviews PBSE, relevant accounting standards, and how much of systems engineering can be performed as a company-wide capital asset development program instead of as project overhead.

The Current Situation

Justifying Systems Engineering in Project Budgets. Systems engineering is sometimes described as a way of reducing risk to product development and support areas such as:

- Project Schedule,
- Life Cycle Cost,
- Integration and Test, and
- Product Performance (ICTT 2005b).

Most of these cost impacts, however, occur after product development phases and during support phases whose costs are often reported in completely different organizational accounts (Blanchard et al. 1998). The effects of not performing enough systems engineering are therefore not very visible to the product manager planning development budgets. This creates an "iceberg" effect in which the project manager figures a small allocation to systems engineering is all that is needed while the hidden yet much larger costs of maintenance, distribution, market share losses due to poor performance, and other life cycle support costs are ready to cripple the company (Blanchard 1998).

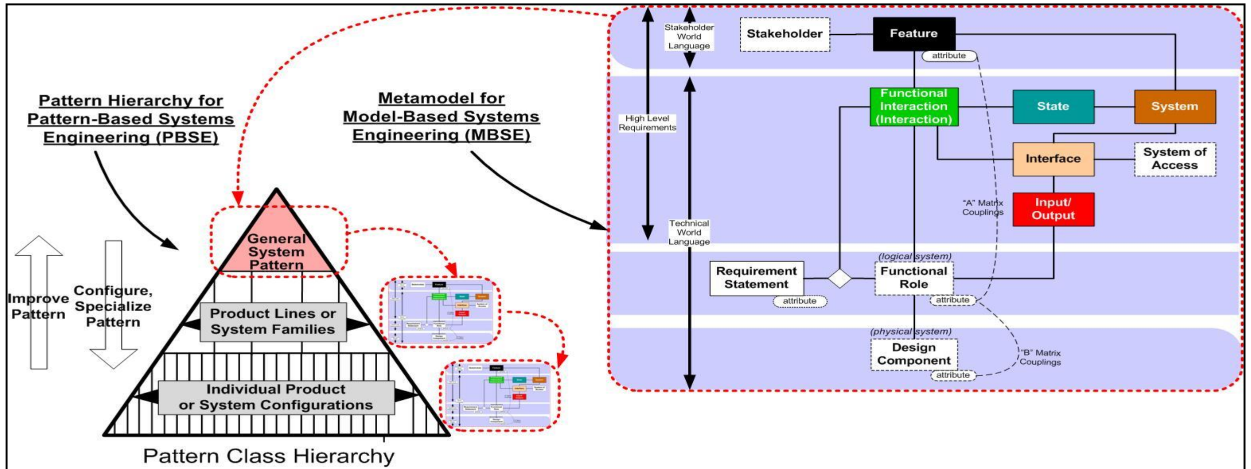
Arguments that explain that while program costs are spent mostly towards the end of the life-cycle the costs are actually committed based on the work very early in the program when most of the systems engineering is performed can be used to clarify the need for systems engineering (Blanchard 1998). However, systems engineering costs are usually accounted for in an overhead expense account in typical Work Breakdown Structures (WBS) that also includes costs for project management (DOD SMC 2001). When looking at the rest of the WBS, it is very difficult

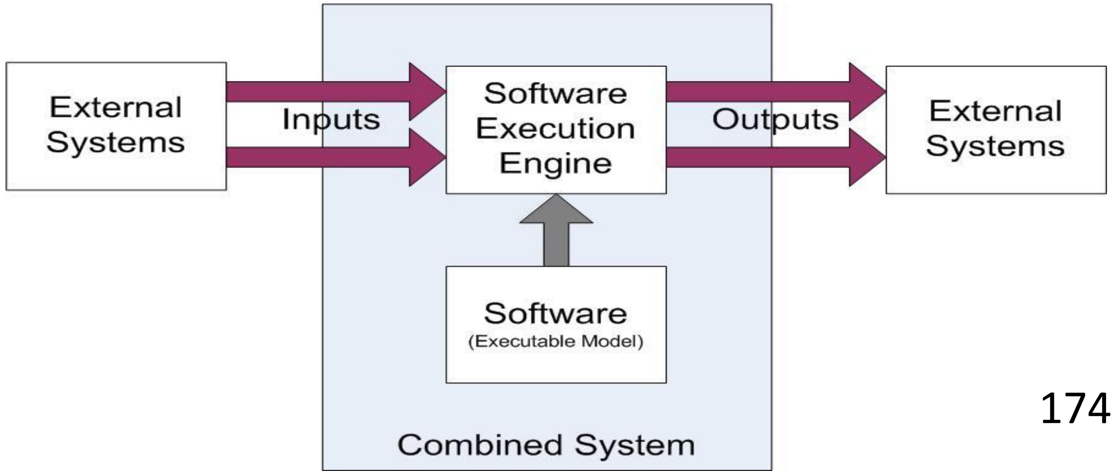
Capitalizing information assets

- Does this mean that all software deserves to be capitalized?
  - Of course not!
- There are many “hurdles” to capitalization, that only <u>some</u> software will clear:
  - For example, $ valuation and life of the asset;
  - And (especially for software) solid life cycle management of the asset: Its requirements, design, verification, maintenance, configuration and version management, support, etc.
  - FASB and other industry or professional criteria;
  - Specific capitalization criteria of the enterprise.

Pattern Capitalization: Implications

1. We are moving toward the Model Based Economy (MBE)—more of our assets are intellectual property (IP)—and many are models.
2. Software is information used as an executable model.
3. This model is interpreted by many types of "execution engines".
4. Patterns are a form of software.
5. Even the engineering process is such an engine—configuring patterns produce requirements and designs of new systems.
6. Software life cycle management informs us about pattern life cycle management.





Exercise 6: Financial Capitalization of System Patterns

1. How much of your systems engineering costs might be dealing with variants around a common core theme?

2. How important would ability to afford more systems engineering cost be in your enterprise, moving cost to time of realizing value?

3. What pattern would you capitalize?

4. Who would care about moving cost of development off P&L and onto balance sheet?

Related INCOSE, ASME communities

- **INCOSE:**
  - Model-Based Engineering Transformation Initiative
  - INCOSE-NAFEMS Joint Working Group on Simulation
  - MBSE Patterns Working Group
  - Agile Systems & Systems Engineering Working Group
  - Tools Interoperability and Model Life Cycle Management Group
  - INCOSE-OMG MBSE Initiative: Challenge Teams, Activity Teams

- **ASME Computational Model V&V Committee / Working Groups:**
  - V&V 10: Verification & Validation in Computational Solid Dynamics
  - V&V20: Verification & Validation in Computational Fluid Dynamics and Heat Transfer
  - V&V 30: Verification and Validation in Computational Simulation of Nuclear System Thermal Fluids Behavior
  - V&V 40: Verification and Validation in Computational Modeling of Medical Devices
  - V&V 50: Verification & Validation of Computational Modeling for Advanced Manufacturing
  - V&V 60: Verification and Validation in Modeling and Simulation in Energy Systems and Applications

Additional Sources of Help:

- **S*Patterns Community:**
  - A member community of people, enterprises, and institutions employing advanced methods and assets for the world's most challenging systems issues—unlocked by Model-Based Patterns using the S*Metamodel
- **Virtual Verification, Validation, and Visualization Institute (V4I):**
  - A member community of people, enterprises, and institutions improving the effectiveness of product development and other life cycle processes, employing model-based verification, validation, and visualization
- **Uncover the Pattern™:**
  - A fast path to creation of the first draft of your organization's fundamental system S*Pattern in 90 days or less

GLRC 11

INCOSE

End of Part II

GLRC 11 INCOSE

Attachments

- Exercise hand-outs
- Pattern extract hand-outs

References

Model Planning and Purpose, Learning, Agility, Adaptation, and PBSE:

1. "INCOSE MBSE Transformation Planning & Assessment Framework: Beta Test": http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:planning_assessment_requirements_for_mbse_model_applications_v1.4.2.pdf
2. Schindel, W. "Pattern-Based Systems Engineering: An Extension of Model-Based SE", INCOSE Tutorial TIES 4. *Proceedings of INCOSE 2005 Symposium*. 2005.
3. Schindel, W. "Requirements Statements Are Transfer Functions: An Insight from Model-Based Systems Engineering", *Proceedings of INCOSE 2005 Symposium*. 2005.
4. Schindel, W., and Smith, V. "Results of Applying a Families-of-Systems Approach to Systems Engineering of Product Line Families", SAE International, Technical Report 2002-01-3086. 2002.
5. Beihoff, B., et al, "A World in Motion: INCOSE Vision 2025", INCOSE.
6. Schindel, W., "What Is the Smallest Model of a System?", Proc. of the INCOSE 2011 International Symposium, International Council on Systems Engineering (2011).
7. Schindel, W., and Dove, R., "Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern", in Proc. of INCOSE 2016 International Symposium, 2016.
8. Schindel, W., "Got Phenomena? Science-Based Disciplines for Emerging Systems Challenges PBSE methodology summary", Proc. of INCOSE IS2017 Symposium, Adelaide, UK, 2017.
9. INCOSE Patterns Working Group, "MBSE Methodology Summary: Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models", V1.5.5A, retrieve from: http://www.omgwiki.org/MBSE/doku.php?id=mbse:pbse

References

Model VVUQ and Credibility

10. Hightower, Joseph, “Establishing Model Credibility Using Verification and Validation”, INCOSE MBSE Workshop, IW2017, Los Angeles, January, 2017. http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2017:models_and_uncertainty_in_decision_making_rev_a.pptx

11. *Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification* ISBN 978-0-309-25634-6 THE NATIONAL ACADEMIES PRESS, http://nap.edu/13395

12. Web site of ASME VV50 https://cstools.asme.org/csconnect/CommitteePages.cfm?Committee=100003367

13. “ASME V&V 10-2006: Guide for Verification and Validation in Computational Solid Mechanics”, ASME, 2006.

14. “ASME V&V 20-2009: Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer”, ASME, 2009.

15. “ASME V&V 10.1-2012: An Illustration of the Concepts of Verification and Validation in Computational Solid Mechanics”, ASME, 2012.

16. *Journal of Verification, Validation, and Uncertainty Quantification*, ASME. https://verification.asmedigitalcollection.asme.org/journal.aspx

17. AIAA (American Institute for Aeronautics and Astronautics). 1998. *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations.* Reston, Va.

18. Box, G., and N. Draper. *Empirical Model Building and Response Surfaces*. New York: Wiley, 1987.

EI Pattern and IOT; Manufacturing Pattern and Digital Plant:

19. Peterson, T., and Schindel, W., “Pattern Based Systems Engineering – Leveraging Model Based Systems Engineering for Cyber-Physical Systems”, Proc. Of NDIA 2014 NDIA GVSETS Symposium on Systems Engineering, Novi, MI, August, 2014.
20. Lu, Y., Morris, K., Frechette, S., “NISTIR 8107: Current Standards Landscape for Smart Manufacturing Systems”, National Institute of Standards and Technology, US Dept. of Commerce, February 2016.
21. “Industry 4.0 / Smart Manufacturing Final Report”, ISO TMB Strategic Advisory Group, September 2016.
22. “Plattform Industrie 4.0 Progress Report”, German BMWi, April 2016, at http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/digitization-of-industrie-plattform-i40.pdf?__blob=publicationFile&v=4
23. ANSI/ISA–88.00.01–2010, “Batch Control--Part 1: Models and Terminology”, RTP, NC: International Society of Automation.
24. IBM Corporation. “The Metamorphosis of Manufacturing: From Art to Science. IBM Global Services, Somers, NY, 2005.
25. Ilegbusi, O., Iguchi, M., Wahnsiedler, W. *Mathematical and Physical Modeling of Materials Processing Operations*, Boca Raton: CRC Press, 2000.
26. U. S. Dept of Health and Human Services, FDA. “Innovation and Continuous Improvement in Pharmaceutical Manufacturing — Pharmaceutical cGMPs for the 21st Century”. 2004.   http://www.fda.gov/cder/gmp/gmp2004/manufSCIWP.pdf
27. U. S. Dept of Health and Human Services, FDA. “Final Report on Pharmaceutical cGMPs for the 21st Century— Risk-based Approach”. 2004, http://www.fda.gov/cder/gmp/gmp2004/GMP_finalreport2004.htm
28. U. S. Dept of Health and Human Services, FDA. “Q8 Pharmaceutical Development ICH Draft: Step 2”. 2004. http://www.fda.gov/cder/guidance/6672.dft.htm
29. “ICH Quality Guidelines 2009, ICHQ1 – ICHQ10”, International Conference on Harmonisation web site, 2009: http://www.ich.org/cache/compo/276-254-1.html
30. U. S. Dept of Health and Human Services, FDA Science Board. Scherzer. R. “Quality by Design: A Challenge to the Pharma Industry”. 2005.   http://www.fda.gov/cder/OPS/Scherzer-Camp/index.htm
31. Gunyon, R., and Schindel, W. “Engineering Global Pharmaceutical Manufacturing Systems in the New Environment”, *Proceedings of the 2010 INCOSE International Symposium*. International Council in Systems Engineering. 2010.
32. Berg, E., “Affordable Systems Engineering: An Application of Model-Based System Patterns To Consumer Packaged Goods Products, Manufacturing, and Distribution”, INCOSE International Workshop, IW2014, Los Angeles, Jan, 2014.
33. Schindel, W. “Systems Engineering for Advanced Manufacturing: Unit Op Insights from Model-Based Methods”. *Proceedings of the INCOSE 2011 International Symposium*. 2011.

References

Capitalization of Patterns

34. W. Aspray, *John von Neumann and the Origins of Modern Computing*, MIT Press, 1990.
35. K. Redmond and T. Smith, *Project Whirlwind: The History of a Pioneering Computer*, ISBN 0-932376-09-6, Digital Press History of Computing Series, 1980.
36. M Singer and P. Berg, *Genes & Genomes: A Changing Perspective*, University Science Books, 1991.
37. J. Essinger, *Jacquard's Web: How A Hand Loom Led to the Birth of the Information Age*, Oxford U. Press, 2004.
38. D. Swade, *The Difference Engine: Charles Babbage and the Quest to Build the First Computer*, Viking Press, 2000.
39. *The Programmable Logic Data Book*, Xilinx, 1994.
40. J. Sherey, "Capitalizing on Systems Engineering", *Proceedings of the INCOSE 2006 Symposium*, July, 2006.
41. "Economists Put a Number on R&D", *Wall Street Journal*, 09/29/06.
42. Gamma, Helm, John, and Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995
43. Financial Accounting Standards Board, *Statement of Financial Accounting Standards No. 2: Accounting for Research and Development Costs*, October 1974.
44. Financial Accounting Standards Board, *Statement of Financial Accounting Standards No. 86: Accounting for the Costs of Computer Software to Be Sold, Leased, Otherwise Marketed*, August 1985.
45. Financial Accounting Standards Board, *Statement of Financial Accounting Standards No. 142: Goodwill and Other Intangible Assets*, June 2001.
46. Schindel, William D., "Requirements Statements Are Transfer Functions: An Insight from Model-Based Systems Engineering", *Proceedings of INCOSE 2005 Symposium*, July, 2005.
47. Sherey, Jason J., "A New Method to Justify Systems Engineering", INCOSE Crossroads of America Chapter Fall Mini-Conference, Fort Wayne, IN, 2004.

Speaker

Bill Schindel chairs the MBSE Patterns Working Group of the INCOSE/OMG MBSE Initiative. He is president of ICTT System Sciences, and has practiced systems engineering for over thirty years, across multiple industry domains. Bill serves as president of the INCOSE Crossroads of America Chapter, and is an INCOSE Fellow and Certified Systems Engineering Professional. An ASME member, he is part of the ASME VV50 standards team's effort to describe the verification, validation, and uncertainty quantification of models.

ICTT System Sciences®
Understand your systems.