An "Engineering Systems" View of "Systems Engineering" and "Systems Architecture"

by C.L. Magee

Engineering Systems Doctoral Seminar

ESD.84 – Fall 2002

Session Number 9

October 30, 2002

Seminar Co-Leads: Chris Magee and Joel Cutcher-Gershenfeld

Systems Engineering

- Tools, methods and processes (enablers) for dealing with complexity (and uncertainty) in engineering design projects.
  - The integrative aspect of systems engineering is problem and domain dependent but USUALLY involves some TOP-DOWN analysis, decomposition efforts, requirements analysis, etc.
  - Systems engineering attempts to enable COMMUNICATION among specialties necessary for the effective design of complex systems

Elements of Systems Engineering 1

- Specifications, Function and Design are represented at several linked levels of abstraction (multi-level SE)
- Linking and traceability of detailed requirements before design begins
- Functional analysis follow specifications and also precedes design
- Quantitative requirements available at component level
- Integrated analytical results often guide subsystem and component requirements (functional analysis precedes embodiment)

Elements of Systems Engineering 2

- Integrative analytical simulations help understand emergent behavior and help determine verification (and validation) procedure
- Systems engineering has an essential management or leadership aspect.
- Systems engineering emphasizes process discipline
- A disciplined process is used for doing trade studies/decisions thus achieving balanced system solutions
- Status of design vs. requirements first tracked and reported by analysis and other tools rather than just “testing”

- When is engineering not "Systems Engineering"?
- What is the opposite of "Systems Engineering"?
- Is there really an "Opposite"?

Systems Engineering "Definition"

- Systems Engineering Opposites
  - Component
  - Bottom-up
  - F, R, A; F, A, R
- Hybrids (It All Depends)
- Thoughtful Analysis/Synthesis vs. Fast Integrated Physical Prototype

Factors Influencing Application

- Scale
- Number of attributes / “ilities”
- Attribute refinement
- complex interactions and attribute tradeoffs
- User understanding and predictability
- Lack of radical technology
- Partitioning clarity – known interactions
- Operand and process mix
- Integration simulation capability
- Prototype and testing cost and timing
- Re-Use
- Long use life
- Focused or single customer
- Commercial vs. Government customer
- System component supplier strength

Factors - 1

- Scale
- Number of attributes / “illities”
- Attribute refinement
- Complex interactions
- Attribute tradeoffs

All of these increase complexity as they increase in “value” or number and SE application becomes essential

Cases:

– Vary significantly in these measures, but all benefited by “SE thinking and methods.” Need for linked representation methods at different levels of abstraction most important at highest levels of complexity

Factors - 2

- Lack of radical technology
- User understanding and predictability

Both of these factors must be at “high values” to facilitate efficient “requirements before design” approach. SE application where user needs are not deeply understood or where technology is uncertain is extremely ineffective

- Cases:
  - New materials processing
  - Surgeon “feel” (human interfaces in general)
  - Aesthetic appeal (all consumer goods)

Factors - 3

- Operand /process mix
- Partitioning clarity
- Re-Use
- Long life cycle
- Cases:
- Energy vs. information fundamental differences, SE most valuable at high information flows in system
- Re-use increases need for SE and drives need for “system of systems” analysis.
- Possible interactions should be monitored.
- Separate “modules” according to “clockspeed” using standards/protocols

Factors - 4

- Integration simulation capability
- Prototype and testing cost and timing
- As these increase, SE becomes possible and then necessary

Cases/Lessons:

- Simulation must be integrated fully into process (and organization)
- Consideration of validation before design can increase testing effectiveness significantly
- Distinguish between overall customer needs and technical requirements

Factors - 5

- Focused or single customer
- Governmental vs. commercial customer
- System/component supplier strength
- All of these move towards requiring SE

Cases/Lessons

- Breadth of appeal
- Government implies political process
- Clear target understanding necessary with multiple strong suppliers

Elements "Extending" Systems Engineering

- Fundamental concepts apply broadly (TUV, BSI,SEI,FDA)
- Variable terminology and organization
- Early, rapid and multiple prototypes (before specifications) for human perception factors
- Multiple loosely linked "V's" early

- Decouple Technology Development from Systems Engineering, OR keep the other complexity drivers small

Systems Architecture

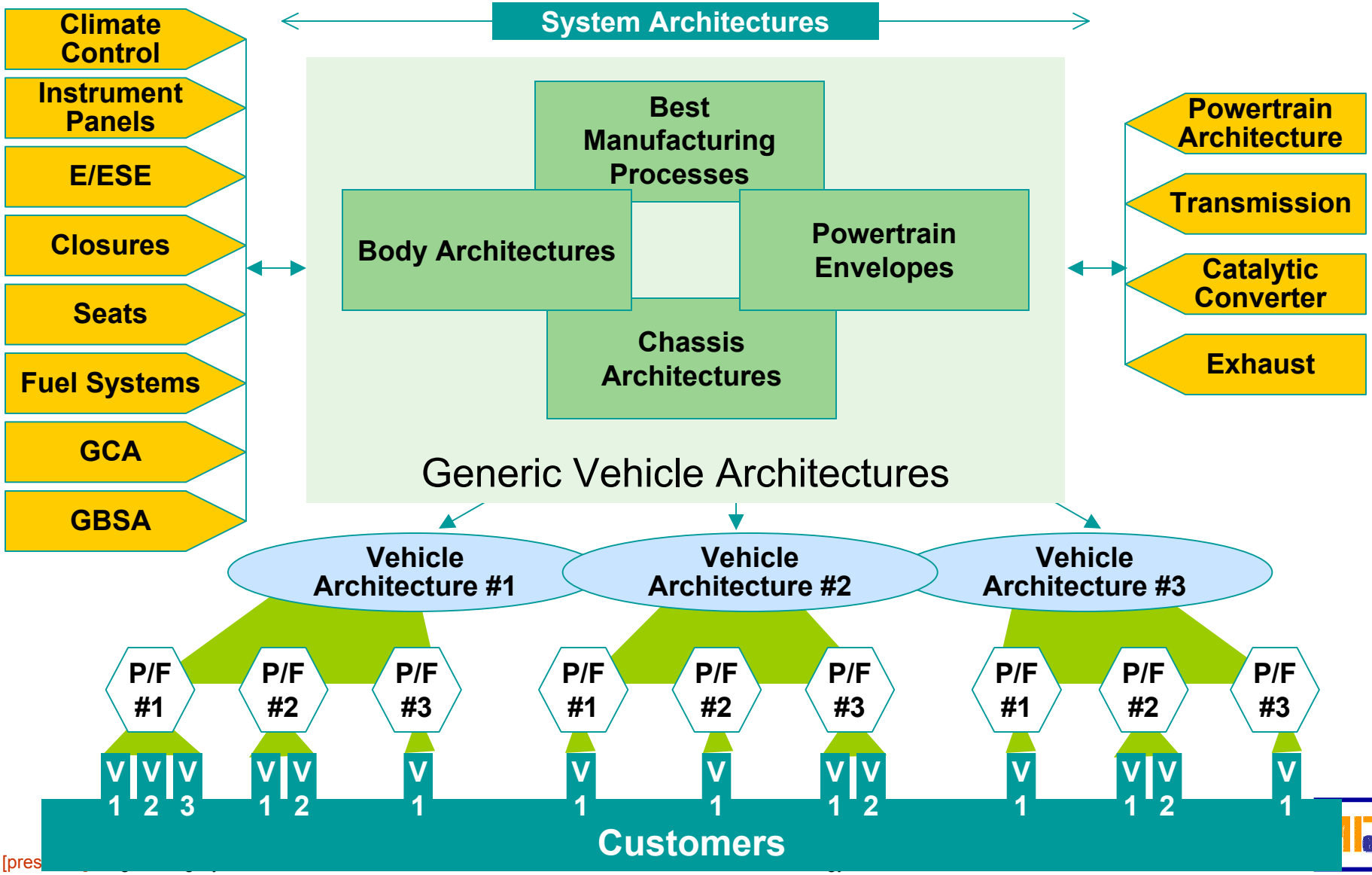
- SA is Systems Design at the next higher level of abstraction and is performed before detailed design at each level of abstraction in formal SE.
- SA for level n is SE/SD for level n+1
- At the level where major societal impact occurs, SA usually involves design of standards/codes/protocols/laws

GAP Goal

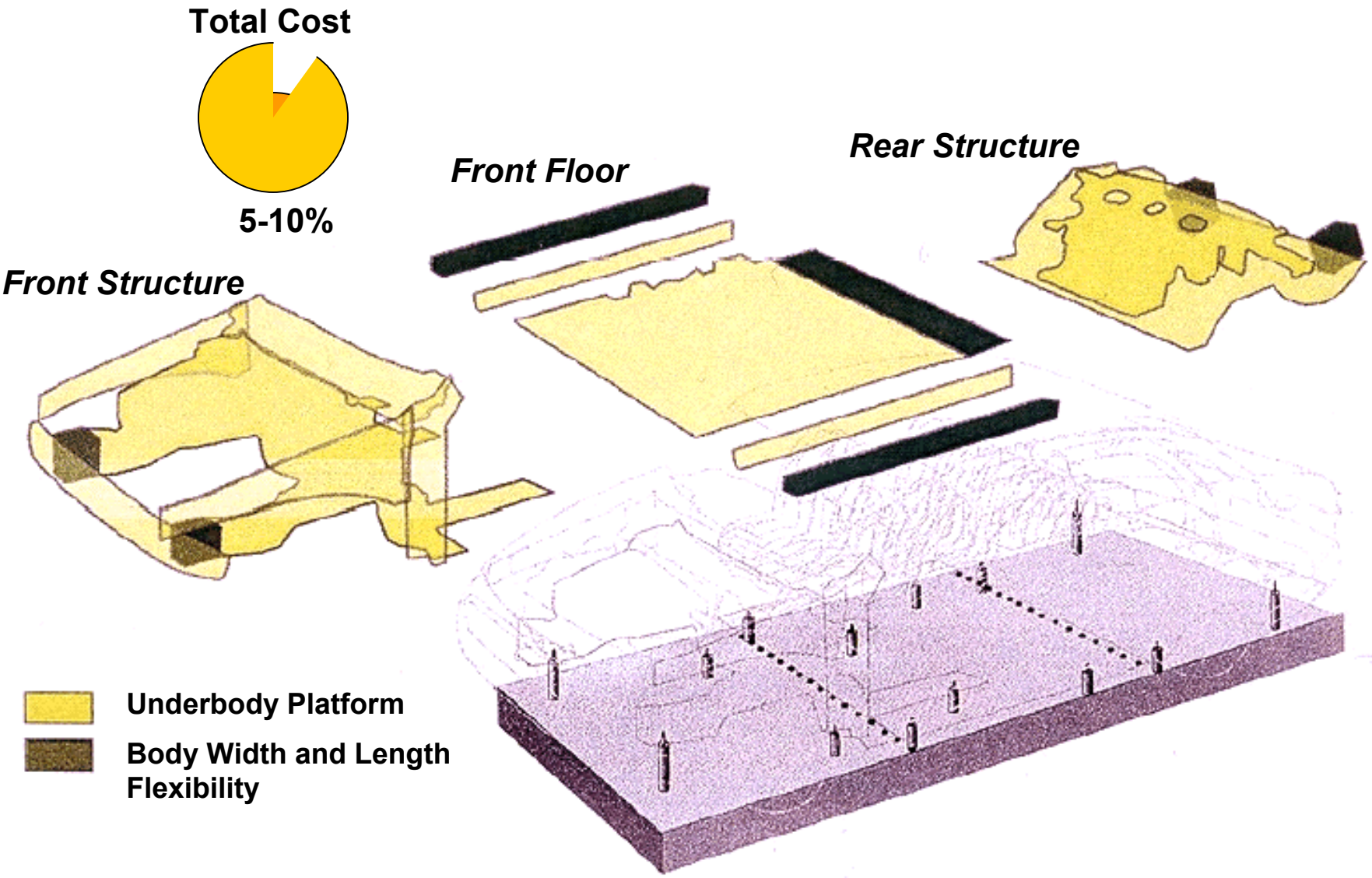
Global Architecture Process Charter

Ford will lead the industry in balancing the achievement of product integrity and customer satisfaction while maximizing the economies of complexity reduction and shared components.

Vehicle Architecture Process

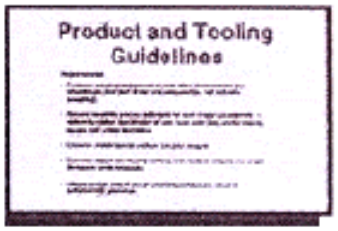


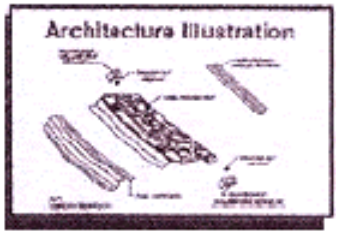
Underbody Platform

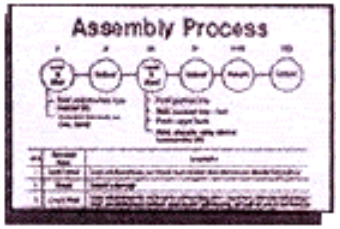


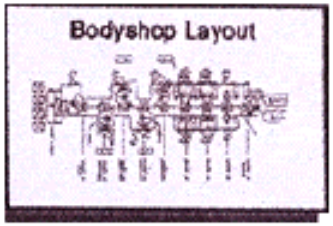
The Generic Body Architecture is an optimized design based upon DFA, NVH, crash, durability and Best Practices for Manufacturing and Engineering

Generic Body Architecture (GBA)









- Locations
- Sealing
- Joints
- Sections
- Closures Hanging
- Closures
- Framing
- Bodysides
- Underbody
- Front Structure
- Definitions

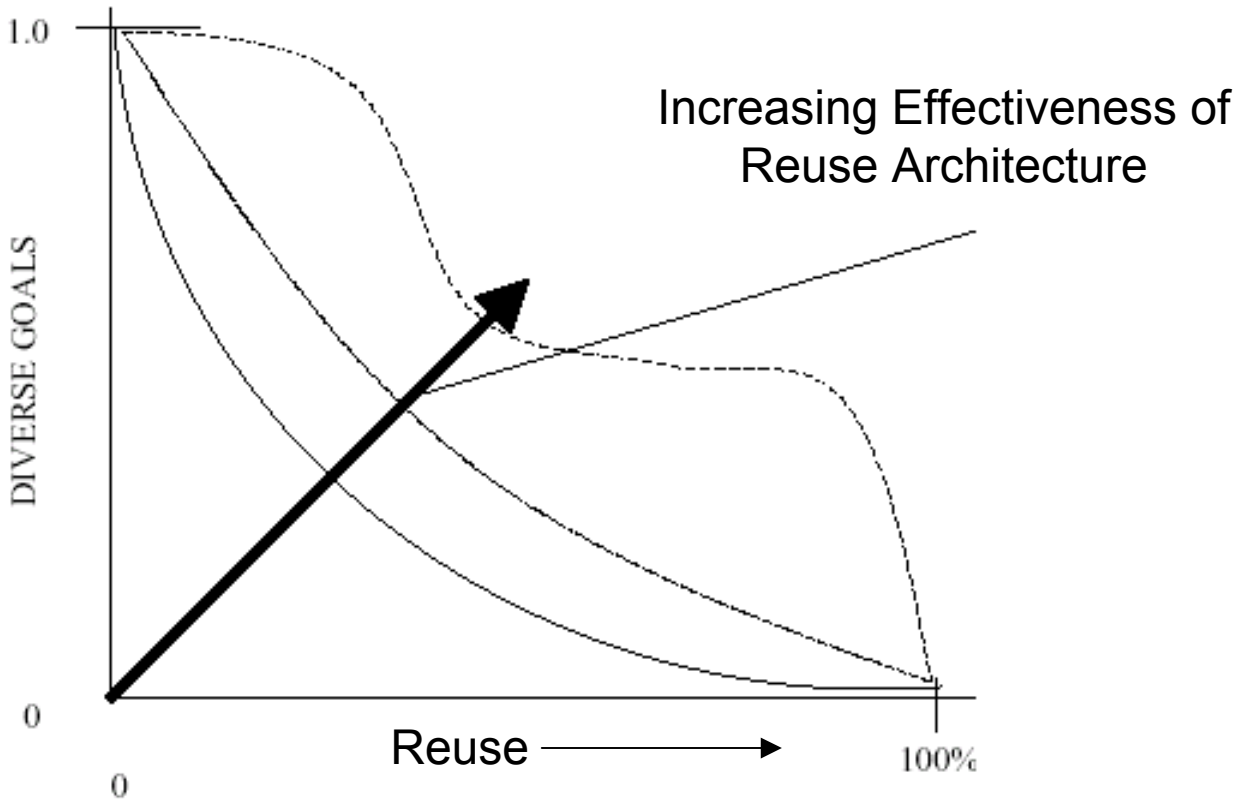
Reuse Principles/ Heuristics - 1

- Reuse occurs on many “layers” and reuse in each layer brings significant benefits and risks
- From a corporate standpoint, reuse decisions attempt to maximize long-term profitability (SHV)
- Optimizing program level reuse decisions can be negative for overall corporate optimization
- Programs should first explore broad design options and then attempt to achieve as much as possible with the highest possible reuse

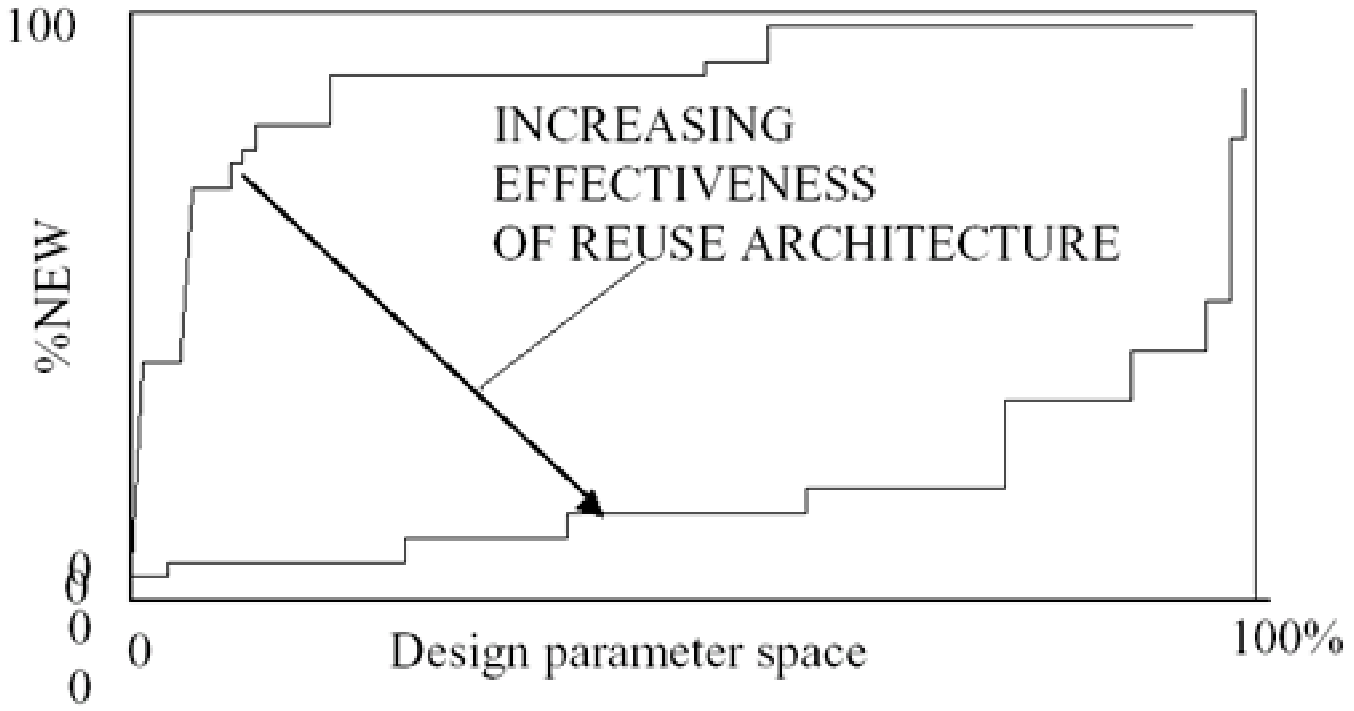
Reuse Principles/Heuristics - 2

- Reuse decisions (as most design decisions) are not usually optimizing but satisficing (H. A. Simon)
- Effective reuse architectures are fundamental to better profitability for companies/industries
- System interaction structures (DSM) determine breakpoints

Reuse/flexibility Tradeoff



Breakpoint Distributions



Reuse Principles/Heuristics - 3

- Effective reuse architectures require developing the right standards at multiple levels
- The right standards involve minimal standards on only the most highly interactive design parameters (DSM)
- Disciplined adherence to rapidly evolving standards is essential to effective reuse process/practice

Reuse Principles/Heuristics -4

- Implementation of a more effective reuse architecture must occur incrementally from existing architectures (for mechanical systems)
- Development of a more effective reuse architecture starts with in-depth analysis of the existing/implicit reuse architecture
- Study of existing “reuse standards” at multiple levels and their evolution are the critical tasks for the “reuse systems architect”