Proceedings of DETC'03
ASME 2003 Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
Chicago, Illinois, USA, September 2-6, 2003

DETC2003/DTM-48668

REPRESENTING PRODUCT ARCHITECTURE

Mike J. Van Wie
Department of Mechanical Engineering
The University of Texas at Austin
Austin, TX 78712
mjv@mail.utexas.edu

Palani Rajan
Department of Mechanical Engineering
The University of Texas at Austin
Austin, TX 78712
k_palanirajan@mail.utexas.edu

Matthew I. Campbell
Department of Mechanical Engineering
The University of Texas at Austin
Austin, TX 78712
mc1@mail.utexas.edu

Robert B. Stone
Basic Engineering Department
University of Missouri-Rolla
Rolla, MO 65409-0210
rstone@umr.edu

Kristin L. Wood[1]
Department of Mechanical Engineering
The University of Texas at Austin
Austin, TX 78712
wood@mail.utexas.edu

ABSTRACT

Product architecture is the transformation of function to layout. Like much of conceptual design, it is a highly dynamic process whereby engineers must consider a deluge of information in terms of both function and form. One shortcoming of current engineering practice is the absence of representations or abstractions used to aid in developing, refining, and exploiting alternative layout solutions. The purpose of this paper is to present a representation for product architecture that sufficiently captures the design factors relevant to product architecture design which are not taken into account in current practices. An example is given to illustrate the technique, and results of a validation experiment are shown.

Keywords: conceptual design, representation, product architecture, design layout

1.0 INTRODUCTION

In modern design, at least in conceptual design, engineers still do not yet practice with design languages that facilitate a clear, consistent, and efficient representation of conceptual solutions. The so-called architecture design phase, as part of conceptual design, is the precursor for subsequent stages including embodiment and detailed design. Due to the impact of product architecture on downstream issues, the need for a quality method and representation for architecture design is significant. The research problem here is the development of a representation, and our results are the two fundamental elements of a representation: a *lexicon* of core product architecture terms and a practical *notation* for instantiating these terms in format that is consistent with what an engineering support system should be [1]. Past work has addressed product architecture issues in varying degrees through studies on customer needs based methods [2], function-based methods and modularity [3-7], product families [8-10], form-based methods [11,12], and design synthesis in general [13,14].

This paper presents the development of a new representation that offers reasonably complete content and a format purposely designed to be practical particularly for an audience in an educational setting where low-overhead methods are desirable. Our hypothesis is that an improved representation can be created by developing a suitable lexicon, notation, and decomposition of architecture design.

2.0 BACKGROUND

The concept of architecture, with respect to product design, is synonymous with the layout, configuration, or topology of functions and embodiments. Ullman [1] states "Engineers generally work from the function of a system, to the architecture of an assembly, to the shape of parts." The goal of this architecture design task is ultimately to create a spatial arrangement of components and assemblies although initial steps may include manipulation (chunking / partitioning) of a functional model to impose some desired modularity-integrality early on.

Figure 1 is an example of two different architectures for a screwdriver. The entire screwdriver is reproduced for

---

[1] Corresponding author: ETC 4.146B-C2200, The University of Texas at Austin, Austin TX 78712.

each different bit in the conventional case. By segregating the bit from the handle, only one handle is needed. Recognizing such opportunities is the challenge. The goal of this work is to generate a framework for capturing product layout so that the designer can more effectively maintain awareness and control over the dominant factors affecting the layout.

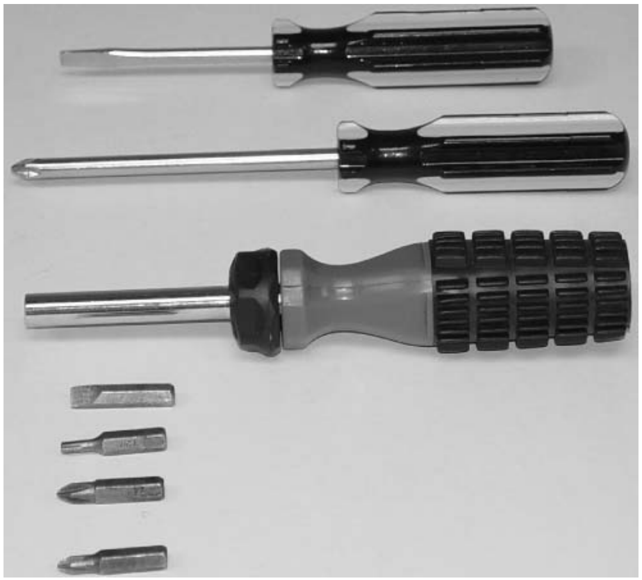


Figure 1. Traditional and exchangeable bit screwdrivers

2.1 Existing methods

The following discussion begins with a few selected examples of representations used mainly in manual design and ends by addressing more computable concepts used mainly in automated design.

The Pahl and Beitz [15] approach shown in Figure 2 utilizes multiple domains from a function structure to working principles, to working structures, then concept variants, and finally embodiment design.

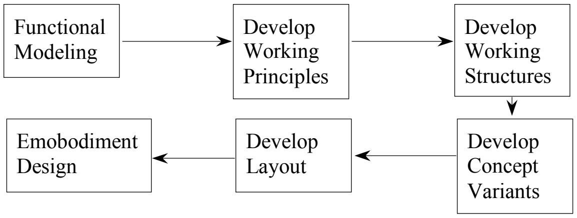


Figure 2. Pahl & Beitz approach [15]

In this method, layout is performed after the concept variants are developed and certainly after several form issues have been introduced. The representations here are defined somewhat loosely as to the details of their structure. This characterization also describes the representations based on domain theory as part of the more general theory of technical systems [16] in addition to other systems engineering approaches such as those design processes given in the IEEE-1220 [17] standard and the NIST 5939 report [18].

Ulrich and Eppinger [19] as well as Cutherell [20] propose a procedure that integrates the concept of “chunking” or “clustering” elements of the design. Table 1 shows the steps involved.

Table 1. Ulrich and Eppinger [19]

1. Create schematic containing elements of both function and form
2. Cluster elements of the schematic
3. Create a rough geometric layout
4. Identify fundamental and incidental interactions

In this approach, the schematic is a type of graph that is combined in the sense that the nodes represent *either* functions *or* physical chunks. In this case, the physical layout of function and form elements take place simultaneously via the abstraction of a graph.

Otto and Wood [21] use a similar procedure but maintain a greater separation of function and form representations. Here they take advantage of this decomposition by using, at the functional level, modular heuristics [4]. This has led to additional modular heuristics from Zamirowski and Otto [7] and has also led to efforts to develop architecture by quantitative means [5,22]. Holtta [22] proposes a dendrogram representation where a distance function is defined for aiding in the judgment and selection of appropriate modules.

While there is no sharp cutoff between manual and automated design representations, a good survey of automated design synthesis work is given in the texts from Antonsson and Cagan [13] and Chakrabarti [14] which both give attention to the representations that support synthesis. Here in a critique of several function-based design methods, Wood and Greer [23] note a common goal among them: “..developing, ultimately, a mathematical, grammar based, or lexicon language for transforming functional representations to physical designs or products.”

Toward this objective, the current status is encouraging, especially in some narrow areas, based on the above two survey pieces. Various incarnations of schematic structures or layout workframes with defined ports (inputs and outputs) are used to address particular synthesis problems such as mechanism design, electromechanical system design, and dynamic system design [24-26]. A common drawback to graph based or schematic based approaches is a lack of spatial information embedded in the representation. Alternatively, shape grammars are well suited to deriving shapes and configurations of a given class of known devices by using a basic set of elements such as lines and rules for transforming those elements. In all cases, the specific problem of representing the mapping of function to form is difficult and largely still outstanding despite notable attempts to decompose the function-form continuum [27] or use a single intermediate domain [28] based on an object oriented scheme.

Given the above examples, many attempts have been made to represent architecture or related design artifacts by means of a range of conceptually different approaches. Briefly, these prior models have included i) sets, eg. the prescription of 'domains' such as the function domain, ii) graphs with assigned meanings to nodes and vertices eg., schematics, bondgraphs, systems engineering flowcharts, etc., and iii) higher level structures, eg. metrics, matrices, algebras, object oriented structures, etc.

The representation structure of this research is based on the concept that a 'mental model' [29] is a suitable representation framework for design problems. A mental model consists of three elements: a set of items (lexicon) representing the real world system, a set of relations among these items (rules for using the representation), and a notation for the lexicon.

Three main observations of prior work and current practices drive this research: 1) The discontinuity between function and form is problematic and some degree of step-wise progression during initial layout synthesis seems appropriate. 2) Much of the formal (computable) design representation work is generally unavailable to end users because it is either not directly suitable for manual implementation or is far from immediate implementation as an automated platform in real world problems. 3) The dominant mode of synthesis today is directed manually in conjunction with CAD support for handling detailed form. These observations lead the authors to seek a representation that is substantive in terms of semantic content (a good lexicon), practical in terms of immediate implementation (a good notation), and supportive of incremental layout synthesis (a good decomposition). This goal is a compromise between developing a representation that will enhance the development of formal methods and provide short term benefits for end users.

2.2 Representation requirements

A discussion of the architecture problem is given next in order to develop requirements for the representation. Upon reflection of the architecture design process, seven major characteristics are found as shown in Table 2. Of course, many other factors such as safety and reliability are relevant but are assumed to be addressed via the set of customer needs developed early in the design.

Table 2. Architecture Design Characteristics

Iterative
Successive approximation
Large number of relevant issues to consider
Strong interdependence on multiple factors
Non-constant starting point
Concurrent engineering
Constraints on the order of operations and scheduling issues

Each of the above characteristics is not surprising given the nature of design. Like other phases of the design process, architecture design exhibits an iterative aspect in the solution search. Secondly, one of the prime difficulties of architecture design is the overwhelming number of issues the designer must consider when transforming from function to form. In addition to the volume of issues to consider, there is generally a high degree of coupling among them and so keeping track of these dependencies is a problem considering short term memory constraints. Not all design projects have the same starting point and many projects involve legacy artifacts from prior work that dictate the departure points for design activities. While often simply referred to as the transformation of function to form, architecture design involves simultaneous activities or concurrent engineering such as developing aesthetics, user interfaces, exploring alternative concepts, exploiting promising concepts, improving efficiency, reducing number of parts, addressing manufacturing concerns, etc.

It is clear that architecture design is a complicated process and that some requirements are needed to develop a successful representation. Ullman [1] presents a thorough treatment of requirements and issues relevant to the design of an ideal engineering support system, and many of them are consistent with those given here. Although the requirements listed in Table 3 are developed for the architecture design phase, they are generally compatible with representations used in other aspects of design.

Table 3. Representation Requirements

Accommodate concurrent engineering
Reconcile short term memory capacity with the huge set of design issues
Do no harm
Support multiple start points and iteration
Support generation of multiple alternatives
Be formally defined
Be practical to implement
Facilitate progressive application of device partitions
Facilitate efficient management of the design

The above requirements are, in some cases, discussed individually below for clarity. Here concurrent engineering issues are noted in order to maintain the possibility of multiple parallel tasks so that, for example, the representation should not preclude concurrent external interface definition and concept development for functional modules. Whether explicitly or implicitly, the representation must give rise to a relatively large set of relevant architecture design issues in a coherent format that keeps these issues at the forefront of the designers thought. The reason for this need is to mitigate the limitations of a designer's short-term memory which hampers full and frequent consideration of multiple design factors.

Do no harm should not be taken literally, of course, but the representation must generally improve the design situation relative to not using any representation at all. The representation must allow for alternative design starting points as well as the revisions that occur during iteration among different phases of the design process. Additionally, the representation must facilitate diverging design activities in which several alternatives are generated. In order to improve operational repeatability and future extensibility, the representation must be well-defined. Low overhead and resources should be required so that the representation is practical to implement. The representation must work despite reasonable variations in designer skill, product domain, project type, and designer resources. A means for establishing product partitions in terms of modules and components should be given in the representation but the representation should not require such a partition or selection of components to be imposed from the starting point. The representation must allow the designer to control the design process from both high and low level perspectives. That is, the representation must include information relevant to both high level product management needs and lower level product development processes. This promotes utility and reuse of the representation across multiple users in the product development cycle.

For this paper, no assumptions are placed on the type of design applicable other than the product is mainly mechanical in nature and the scale is somewhat restricted. While not fundamentally limited to a particular scale of device, the representation is directed mainly toward the small and medium scale of devices like mechanical pencils, power tools, washing machines, etc. This scale excludes the very small such as MEMS devices and the very large such as cars and aircraft.

Two items are needed to establish the representation: a lexicon of objects that are relevant to architecture and a notation for instantiating the lexicon. The next section presents the overall approach for the remainder of the paper.

3.0 APPROACH

The premise is that an architecture representation offers designers benefits if it presents a well-defined domain in which designs can be documented, observed, and manipulated. The primary objective is therefore to generate a formal architecture design representation that fulfills the criteria above. In order to reach this goal the following actions form the research plan:

1. Generate an architecture lexicon.
2. Develop a meaningful notation for the lexicon.
3. Test the representation to evaluate effectiveness.

4.0 LEXICON DEVELOPMENT

The lexicon is generated by first searching for a relatively large set of issues that are relevant to architecture design. Following this search, the list is pruned to yield a final set. It is well known that product architecture is related to a vast number of design issues ranging from aesthetics to manufacturing choices [20]. By performing an individual and group brainstorming search as well as a literature search, a list is generated that contains over 200 generally different issues. Nearly every design issue is directly or indirectly related to some design variable although there are a few exceptions. For example, patent infringement problems are certainly design issues, although one is hard pressed to define a succinct and meaningful set of design variables that reflect patent infringement issues.

The lexicon is refined by considering each design issue and developing a manageable set of relatively important, observable, and controllable artifacts of the design. The rationale is based on the Pareto Principle (the 80/20 rule) and so the goal is to identify those relatively important issues and design variables with the reasonable expectation that this reduced set captures the bulk of what is really important in product architecture. Table 5 presents the derived lexicon, which is small compared to the number of design issues considered. Criteria for selecting this final lexicon are related to two issues. First, the terms in the lexicon are chosen because they are somewhat abstract in the sense that each term generally subsumes a set of terms that are less abstract. Secondly, the lexicon does not include those items that are outliers in terms of relevance with respect to architecture design. One concern about lexicon pruning is about the potential for restricting designer creativity. However, any detrimental impact on creativity should be minimal given that the design space is not restricted although the designer's attention is more in tune with a restricted set of design issues. The following discussion addresses the relation between creativity and restrictions on the terms in the lexicon.

Engineering creativity can be defined by the characteristics of the design solution. A design is creative if it is original. Based on empirical studies of creativity, Finke [30] has found that certain restrictions on the objects used to design will in some cases enhance creativity and in other cases detract from creativity. If the objects or building blocks for a designer restrict the set of solutions relative to the solutions in the design space of interest, then creativity will be potentially hampered. If, however, the restrictions merely force the designer to view the design problem from a perspective outside the norm of his or her view, then creativity will likely be promoted.

The effect in this second case is offering the designer potentially new search directions by avoiding, or restricting, the normal cues that lead a designer to solutions typically associated with a given perspective. The difficulty is in contriving a restriction that forces a designer to think outside the box while not overly restricting the design space. In the case of pruning the design issues to a smaller list as given in Table 4, this restriction forces the designer to focus more intently on a relatively small set of items that normally might get lost in an unrestricted clutter of design issues. Given the set of terms provided by this list, the next section develops an appropriate notation.

Table 4. Architecture Lexicon

TERM	DEFINITION	CONSTITUENTS and STRUCTURE
Function	A form independent operation that a machine imposes on a set of energy, material, and signal flows.	Constituents: Form independent operation: a verb-noun (operation-flow) phrase such as “Dissipate Heat” A set of input flows of energy, material, and signals A set of output flows of energy, material, and signals Structure: The form independent operation acts on the input flows to yield the output flows.
Industrial Design Syntactics	The rules and issues governing the physical embodiment with respect to human perception of the embodiment.	Constituents: A set of gestalts: physical effects and their relation to the human perception of those physical effects Structure: The gestalts are mapped to a physical solution to characterize the physical solution in terms of human perception
Device Operations and User Activities	The operations and activities that are imposed on the device during the device lifetime.	Constituents: A set of user activities associated with product operation. Structure: The user activities are given with respect to the device and the physical flows. The activities are related among themselves temporally.
Customer Needs	Explicit and latent needs and desires of the customer.	Constituents: A set of needs. Structure: The needs are generally ranked in order of importance.
Physical Solution Topology	A physical embodiment of some set of functions. (A concept in the physical or form domain)	Constituents: A set of physical embodiments that satisfy some function and the connectedness of those embodiments. Structure: The physical embodiments are given and related among each other in terms of geometry and material. In architecture design, the physical solutions are generally only roughly sized in terms of relative size, shape, position, and orientation. Physical solutions are partitioned into physical modules and components.
Function to Form Mapping	The relationship between a set of functions and the physical embodiment that instantiates that functionality.	Constituents: A set of relationships that map a set of functions to a set of physical embodiments. Structure: Functionality is correlated with spatial regions of the product. (Similar to the wirk element concept [31])
Manufacturing Choices	The specifications for how a subset of a physical solution is manufactured.	Constituents: A specification that describes the choice of manufacturing for one or more components. This choice generally distinguishes between OEM and custom fabricated parts and can further specify those custom fabrication techniques. For example, injection molding is such a technique. Structure: Specifications are correlated to components, physical modules, and possibly functional modules.
Component	A physical part.	Constituents: Geometry and material specifications that describe the part. Structure: A description of the geometry and material specifications either explicitly or more vaguely by referring to a common part name.
Functional module	A set of related functions.	Constituents: A set of functions. Structure: A functional module may contain both functions and lower level functional modules.
Physical module	A set of components that exist as a stable assembly even without any external	Constituents: A set of components. Structure:

	effort holding the items together. [32]	A collection of components and possibly physical modules in terms of a spatial region.
Flows	A set of energies, materials, and signals that are processed by the product. (Enumerated in [33]).	Constituents: A set of energy, material, and signals. Structure: Flows can be defined both non-spatially with respect to functions and spatially with respect to physical solutions.
Interfaces	The physical regions where physical flows exist (not including the regions internal to component material).	Constituents: A set of spatial regions where energy and / or material flow between components or between a component and the external environment. Structure: Physically, interfaces are the partitions within a physical solution and they are given spatially. Functionally, interfaces are equivalent to the flows between functions.
Functional Solution Topology	A set of functions and flows that satisfies some overall main product function. (A concept in the functional or form-independent domain).	Constituents: A set of functions, flows, and their connected arrangement. Structure: The functions and flows are given so that their arrangement represents the processing of flows from device input to device output.
Product family elements	The family platform, common components, similar components, and distinct variant components.	Constituents: Generally two sets: common and uncommon components. Common components makeup the platform while unshared components support different variants. Structure: Common components are distinguished from unshared components in order to identify the platform.
Relative motion	The degrees of freedom and range of motion for components and physical modules.	Constituents: A degree of freedom and range of motion specification for a component. In cases of compliant mechanisms, this specification may be given relative to itself rather than a fixed frame. Structure: In most cases, the relative motion specification is given relative to some fixed frame although in the case of compliant mechanisms, this specification can be given relative to the component itself. (Refer to R-links and C-links in [32]).

4.1 NOTATION DEVELOPMENT

In a broad view of the architecture problem and of the resulting lexicon, it is clear that a comprehensive notation must facilitate the representation of a wide range of design artifacts as indicated by the lexicon content. An assumption for notation development is that the notation must incorporate all elements of the lexicon either as an input to the notation or as part of the notation itself. Of course, many tools already exist for representing many of these items. This research leverages the utility of these existing notational devices and creates new notational devices where prior work falls short. In addition, Table 5 correlates some existing representations to phases in the design process. While the exact nature of visualization and imagery in conceptual design is not completely understood [34], it is clear that sketching is part of design. The notation adopts a workframe concept that incorporates the use of rudimentary sketches for documenting elements of the lexicon.

Table 5. Current Design Notations

Design Task	Existing Notations
Problem Clarification & Definition	Customer needs analysis results Mission statement Benchmarking QFD
Industrial Design	External illustrations Foam models
Functional Modeling	Black box Function structure with functional models identified
Physical Solution Generation	Morphological matrix of schematic physical solutions Solution description in textual format
Physical Solution Combination – Concept Layout Generation	Rough geometric schematic ***Architecture notation needed here*** Mathematical models Proof of concept physical prototypes
Manufacturing and Assembly Design	Manufacturing data, selected processes, source of material Bill of materials Assembly tree Force flow diagram
Final Form Specification	CAD, solid models with complete specifications on all parts

In order to capture terms of the lexicon in a practical manner, a six member workframe is used as shown in Table 6. Two requirements drive the selection of diagrams for the workframe: all lexicon terms should be accounted for *and* the number of diagrams in short term memory, about seven, should not be exceeded. They are also developed so that the spatial constraints diagram serves as a reusable template for remaining diagrams. These six diagrams generally build upon each other and can be created according to the procedure shown in Figure 3. Note that the input information, shown in gray, to the procedure consists of typical information known at the time when concept generation is in progress. At this stage, a functional specification has been established and early results of form design have been developed such as physical solutions within a morphological matrix.

Table 6. Architecture Notation – An Architecture Workframe

Layout Diagram	Description	Minimum Inputs	Outputs
Spatial Constraints Diagram	A diagram mapping spatial constraints to spatial regions	Customer needs, Requirements, Functional Model, One Physical Solution choice	Spatial constraints External Physical interfaces
Function Layout Diagram	A diagram mapping functions to spatial regions	Spatial Constraints Diagram Physical product (if redesign)	Function to form mapping Candidate physical modules and partition choices
Physical Solution Diagram	A conceptual physical solution for a subset of the spatial diagram	Function Layout Diagram Alternative physical solution choices	Spatial layout of physical solution Relative motion
Partition Diagram	A tree structure	Physical Solution Diagram	Physical partitioning into modules and components Assembly choices
Manufacturing Diagram	A diagram mapping manufacturing choices to spatial regions	Physical Solution Diagram	Manufacturing choices
Product Family Diagram	A diagram mapping product family elements to spatial regions	Function Layout Diagram	Product family elements

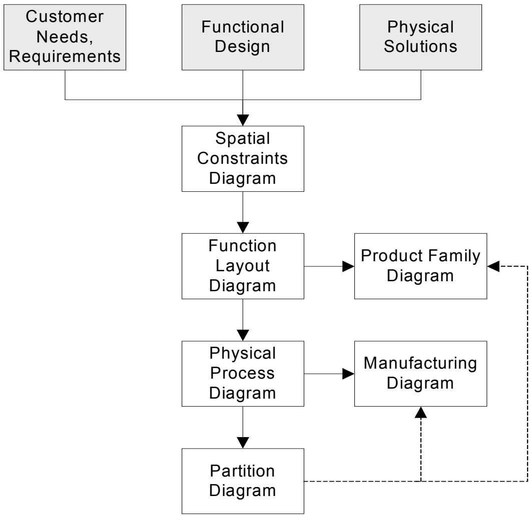


Figure 3. Representation Development Sequence

Each of the six diagrams is defined in the next section and includes a set of nomenclature and a procedure for specifying each. A Pilot G-2 pen is used as an example to illustrate the six diagrams. A few definitions are relevant to diagram terminology:

Spatial Diagram: A notation that represents the product space overall or regions of material in the product space. The diagram may be realized with a sketch, solid model, or appropriate data structure.

Product space: The spatial volume bounded by a product.

Region: A closed volume including the interior and edges. A region can be either the product space or a subset of the product space.

Figure 4. Pilot G-2 pen

Figures 4 through 10 illustrate the representation for a Pilot G-2 pen. While this example represents an existing product, it shows the type of information that a designer would consider during a design exercise for the case of either original or redesign. Beginning with the spatial constraints diagram in Figure 5, an exterior boundary of the product is established including all material and energy flows based on functional models or a black box. This silhouette is one solution of the form boundary given the set of inputs to the problem. By selecting an alternative set of physical solutions from these inputs, the designer has the option of generating alternative spatial constraints diagrams. This provides a means for developing alternative layouts in a reasonably systematic manner. Following the generation of one or more spatial constraints diagrams, the subsequent five representation diagrams may be developed similarly.

4.1.1 Spatial Constraints Diagram (SCD)

The purpose of the spatial constraints representation is to show geometric constraints of the product. Table 7 provides the general nomenclature. Note that the SCD is a silhouette of the product much like a control volume. External flows show the main interactions with the environment, and the dimensions give some idea about the scale and known constraints. This diagram mostly involves decisions about how the customer needs are manifested on the external shape of the product. Both the main product lines as manifested partially by the product outline and external interactions, as they relate to visualizing customer-product interaction, have been shown to be important factors during sketching activities [35,36].

Definition: The spatial constraints diagram consists of a roughly sized product boundary, all external energy, material, and signal flows oriented in space, and known geometric constraints specified.

Diagram Generation Procedure:

1. Reproduce the functional black box.
2. In order of importance, spatially orient the energy, material, and signal flows based on requirements information, functional information, and the currently selected set of physical solutions.
3. Establish a product boundary by reshaping the black box boundary according to the magnitude and direction of the energy flows and the size, shape, type, and amount of the material flows and their relative location.
4. Identify and label dimensional constraints.

Table 7. Spatial Constraints Nomenclature

Notation	Definition	Rule Set
→	Energy, Material, or Signal flows external to the product boundary	Show flows relative to the product boundary in terms of position and orientation Flows are equivalent to the flows on the black box diagram
□	Boundary of some product region.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region Multiple boundaries of regions collectively form the product boundary
Text	Dimensions or descriptions of geometric constraints	Spatial information shown where relevant

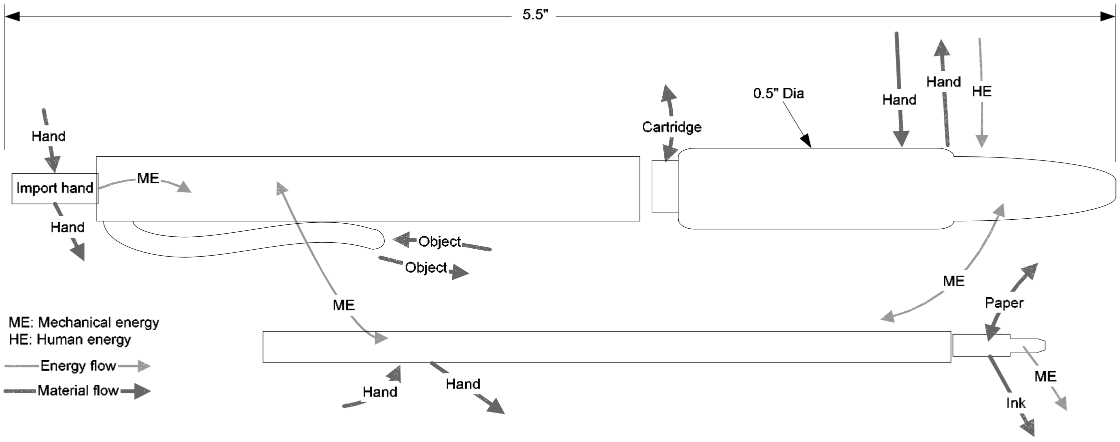


Figure 5. Spatial Constraints Diagram

4.1.2 Function Layout Diagram (FLD)

Proceeding inward from the product boundary, the mapping of function to form is established with the function layout diagram. This mapping is accomplished by simply associating a function with a spatial region. Here the regions of the product which differ based on function alone are identified and annotated. Advantages and compromises with respect to function location can be seen more clearly as all functions are described spatially in the product boundary. For example, the length of energy and material flow may appear as no concern in the function structure while the function layout diagram may indicate an excessive distance in the flowpath. The function layout diagram adds spatial perspective to the traditional function structure and therefore enriches the designer's viewpoint.

The function layout is based somewhat on the wirk element concept [31] in that spatial regions are associated with some functionality. The main purpose of this notation is to show the basic internal functions and flows in a spatial format rather than the traditional block diagram style of a function structure. Table 8 gives the nomenclature, and an example of a function layout is given in Figure 6. One outcome of this particular notation is the ability to predict candidate physical modules in a manner not previously reported [37]. The FLD is where several architectural decisions are made and many options are considered. In fact, one can create many candidate FLD's for the same SCD.

Definition: The function layout diagram consists of a product boundary, all flows external and internal to the device, layout elements, and layout intersections.

Diagram Generation Procedure:

1. Reproduce the product boundary based on the spatial constraints diagram and include the flows but do not include dimensions.
2. In order of flow importance, establish regions for each functional module first and then each function while maintaining the same functional topology from the function structure.
    a. Establish a layout intersection (dashed line) for each functional module identified in the function structure. Module identification in the function structure is described by Stone [4].
    b. Establish a layout element (solid line) for each function in the function structure.
    c. Establish internal flows that connect layout elements and layout intersections.
    d. Size the layout elements and intersections based on the physical solutions size, the magnitude and direction of the energy flows, and the size, shape, type, and amount of the material flows and their relative location.

Table 8. Function Layout Nomenclature

Notation	Definition	Rule Set
→	Energy, Material, or Signal flows external and internal to the product boundary	Show flows relative to the product boundary in terms of position and orientation Show flows from input to output along the same function path as the function structure
□	Layout element – a region that corresponds to some function	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
⬚	Layout intersection - A region that is a unique intersection of two or more layout elements.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
Text	Flow and function descriptions	Function information is shown for each layout element and layout intersection All functions from the function structure should be identified

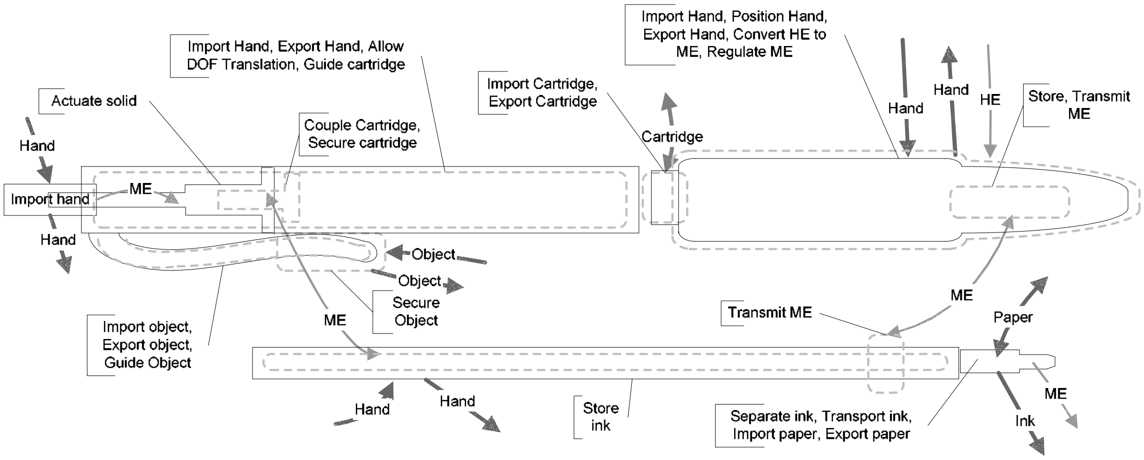


Figure 6. Function Layout Diagram

4.1.3 Physical Solution Diagram (PSD)

The physical solution diagram is intended to address the choices from the morphological matrix [38] for a set of functions and the relative spatial arrangement of those morphological choices. Table 9 provides the nomenclature, and an example is given in Figure 7. Note that gross motions are given while some, such as the motion of the planetary gear set, are not. One reasonable threshold for selecting those motions to include is to consider how significant the motion is to the physical solution. This allows the designer to be flexible when including physical motions. This diagram is important since it provides the designer with a check to see if the corresponding FLD is feasible.

Once the spatial layout of functions is developed, the physical process diagram is generated by selecting those physical solutions that are consistent with the spatial constraints diagram and the function layout diagram. This amounts to documenting the particular set of physical solutions that formed the basis of the above two diagrams and finding and documenting any other remaining physical solutions that would also be suitable in conjunction with the other workframes that are already developed. This offers a somewhat efficient opportunity to identify multiple physical solutions within the same diagram. In the case of the pen for example, the clip is associated with two possible alternative physical solutions. First there is the compliant mechanism which is what this particular pen uses. The other solution is a separate metal clip which is also a viable alternative. By annotating these alternative physical solutions in this manner, a designer can efficiently document multiple form concepts that match the constraints imposed from the other representation diagrams, namely the spatial constraints and the function layout diagrams.

Definition: The physical solution diagram consists of a product boundary, physical solution descriptions (text, schematic, or sketch) and arrows indicating motion of parts.

Diagram Generation Procedure:

1. Identify a layout element or layout intersection.
2. Indicate (with text, schematic, or sketch) the physical solution that performs the functionality of that layout element or layout intersection.
3. Repeat steps 1 and 2 for all layout elements and layout intersections.
4. Identify regions in the device that exhibit relative motion.
5. Label those relative motions using an arrow to indicate direction.

Table 9. Physical Solution Nomenclature

Notation	Definition	Rule Set
▭	Layout element – a region that corresponds to some function	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
⬚	Layout intersection - A region that is a unique intersection of two or more layout elements.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
→	Relative motion of a region	Indicates a significant relative motion corresponding to a physical solution at some region

⤵		May generally be translation or rotation
Text	Physical solution descriptions	Indicates one or more physical solutions that are consistent with a spatial region Alternative physical solutions can be indicated on the same layout provided that both solutions share approximately the same geometric specifications

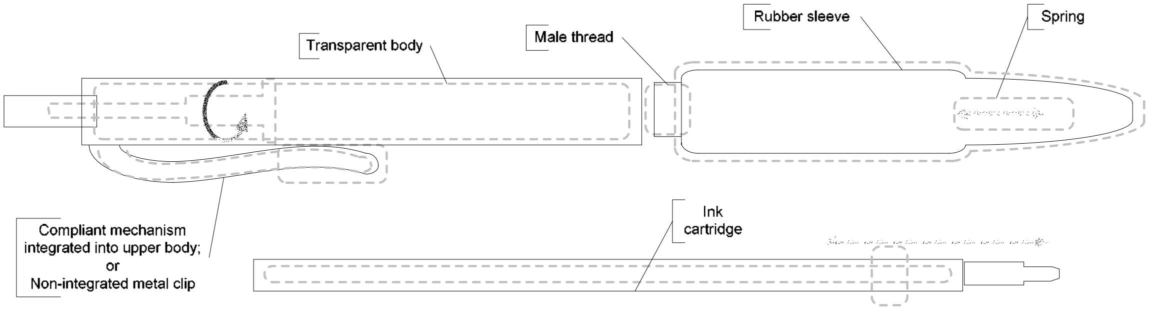


Figure 7. Physical Process Diagram

4.1.4 Partition Diagram (PD)

While the physical process diagram captures the nature of the solution in terms of form by means of a textual based annotation, the details of assembly, modularity, and interfaces are somewhat left to the imagination. The following partition diagram helps solve this problem by addressing these items through a Design Structure Matrix (DSM) [12]. This component-component matrix shows the physical modules as indicated with the thick boxes. Here a physical module takes the definition by Greer [32] – A set of components that hold together in a stable configuration with no external effort required to maintain that stability. Additionally, interfaces are indicated as 1's for each element in the matrix where an interface between two components is defined as:

Interface: A spatial region where energy and/or material flow between components or between a component and the external environment [37].

The purpose of the partition scheme is to establish physical modules, components, and the manner in which they are connected. An example of the pen DSM is shown in Figure 8. Note that only half of the matrix is shown due to symmetry. The PD notation is based on the DSM, and an extension of the DSM called the branch diagram, which is a tree structure. Nomenclature for both of these items is given in Table 10 and background on the development of the branch diagram is given in [37].

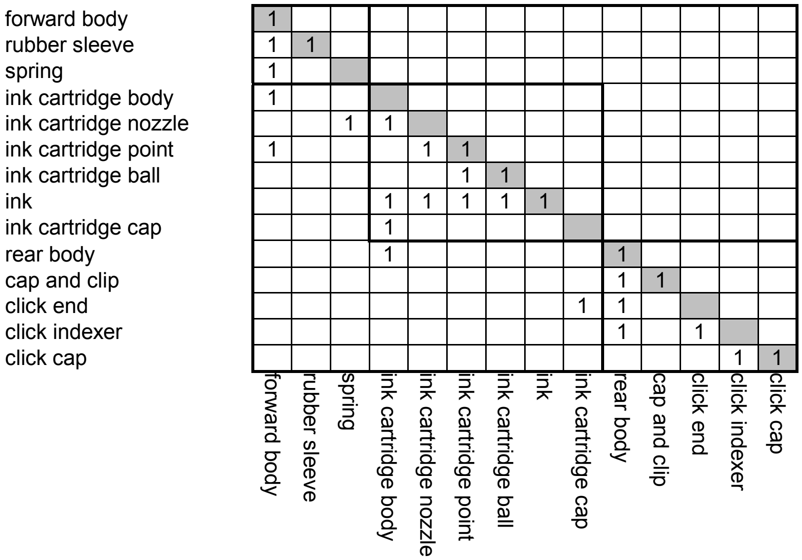
Definition: The partition diagram consists of the hierarchical relations among modules, components, and their physical interfaces.

Diagram Generation Procedure:

1. Partition regions of the device physical solution diagram into physical modules and components.
2. Generate a DSM including physical interfaces among components based on the partitioning scheme.
3. Generate a branch diagram based on the DSM.

Table 10. Partition Nomenclature

Notation	Definition	Rule Set
Design Structure Matrix (DSM)	A *component – component* matrix showing the relation between components and interfaces	Each element can indicate the existence of an interface (1) or not (0) Each off-diagonal element represents an interface between two components Each diagonal element represents an interface between a component and the environment external to the device Boxes within the matrix indicate a physical module among those components
Branch Diagram	A tree structure showing the hierarchy among physical modules and components	The main parent node is the product and subsequent lower levels are physical modules and components subsumed according to the DSM Squares represent physical modules and these correspond 1 to 1 with the physical modules in the DSM Circles represent components



	forward body	rubber sleeve	spring	ink cartridge body	ink cartridge nozzle	ink cartridge point	ink cartridge ball	ink	ink cartridge cap	rear body	cap and clip	click end	click indexer	click cap
forward body	1													
rubber sleeve	1	1												
spring	1													
ink cartridge body	1													
ink cartridge nozzle			1	1										
ink cartridge point	1				1	1								
ink cartridge ball						1	1							
ink				1	1	1	1	1						
ink cartridge cap				1										
rear body				1						1				
cap and clip										1	1			
click end									1	1				
click indexer										1		1		
click cap													1	1

Figure 8. Partition Diagram

4.1.5 Manufacturing Diagram (MD)

The purpose of this diagram is to establish the manufacturing choices for components and modules. The primary format for this notation is a spatial diagram although other representations such as a branched diagram or bill of materials could potentially be substituted depending on the particular emphasis of the designer. The nomenclature as shown in Table 11 is reused to some degree from the previous diagrams. Figure 9 illustrates the MD for the pen.

Definition: The manufacturing diagram consists of manufacturing choices such as material, processing, in-house sourcing, or OEM sourcing associated with a spatial region.

Diagram Generation Procedure:

1. Identify the material and manufacturing choice for each region or component or module.
2. Identify in-house or OEM sourcing.

Table 11. Manufacturing Nomenclature

Notation	Definition	Rule Set
	Layout element – a region that corresponds to some function	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
	Layout intersection - A region that is a unique intersection of two or more layout elements.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
Text	Descriptions of manufacturing choices	Indicates the choice manufacturing with respect to a product region

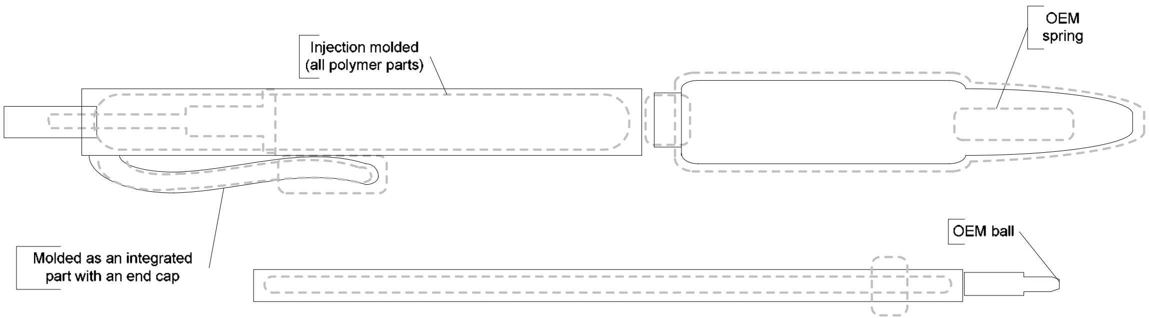


Figure 9. Manufacturing Diagram

4.1.6 Product Family Diagram (PFD)

The purpose of the product family diagram is to highlight those regions or parts of the product that are common, similar, or different than others in the product family. Nomenclature for this diagram is shown in Table 12. Note the distinction between common and cousin regions which is based on work from [39]. Figure 10 provides a hypothesized estimate of those items likely to be common among other (currently nonexistent) variants of the G-2 pen family.

Definition: The product family diagram consists of common, similar, and unique items among a product family with respect to the current device being designed. These items may include functions, modules, or components.

Diagram Generation Procedure:

1. Identify common parts and modules or identify common layout elements and layout intersections.
2. Identify similar parts and modules or identify similar layout elements and layout intersections.

Table 12. Product Family Nomenclature

Notation	Definition	Rule Set
□	Layout element – a region that corresponds to some function	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
⬚	Layout intersection - A region that is a unique intersection of two or more layout elements.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
■	A region common among other variants in the family portfolio	Used when a region is common to a region from one or more other variants in the family portfolio
▨	Region that is cousin (similar, but not unique nor drastically different) with a region from other variants in the family portfolio	Used when a region is similar to a region from one or more other variants in the family portfolio
Text	Common or cousin region descriptions	Indicates the nature of the region that is common or cousin – such a description may be in terms of either functionality ***or*** the physical solution

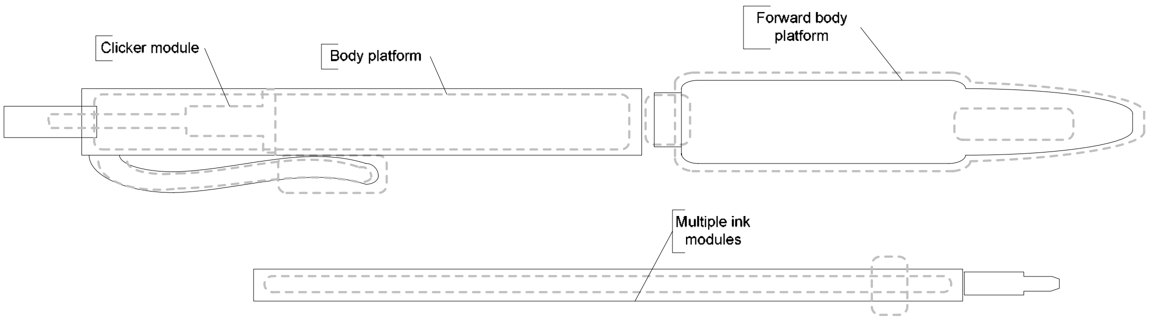


Figure 10. Product Family Diagram

5.0 REPRESENTATION EFFECTIVENESS

In order to evaluate the representation, an experiment was performed that involved a comparison of three groups of engineering students using three different representation techniques. The three techniques tested included 1) the experimental method presented in this paper, 2) a comparable technique from the Ulrich and Eppinger text [19], and 3) no prescribed technique at all.

Freshman engineering students from the University of Missouri-Rolla participated in the experiment where they were divided into three groups. Two person teams were formed within each group. In addition to the experimental technique and Ulrich and Eppinger's method, a third group was only given a morphological matrix and no architecture technique for developing concept variants. Group make-up is shown in Table 13. A description of a design problem to create a better mouse trap was given to all three groups and was presented in terms of the following information: customer needs, a black box, functional model, and a morphological matrix.

Following the experiment, each concept was evaluated according to three measures: feasibility, originality, and detail. Results were formed by averaging the ratings as judged by several graduate engineering students at the University of Texas at Austin (a different institution from where the experiment was run) who were blind to the knowledge of which concepts were associated with a given test group. Based on the results, it is clear that the student performance of the experimental group with respect to generating feasible and detailed solutions was better than both control groups. A t-test on the data shows no statistical difference between any groups in terms of originality of solution. However, the same test verifies a significant difference between experimental and both control groups in terms of detail and feasibility of solutions. Figure 11 illustrates the results and Table 14 shows the probability that the means between the test groups are distinct [37].

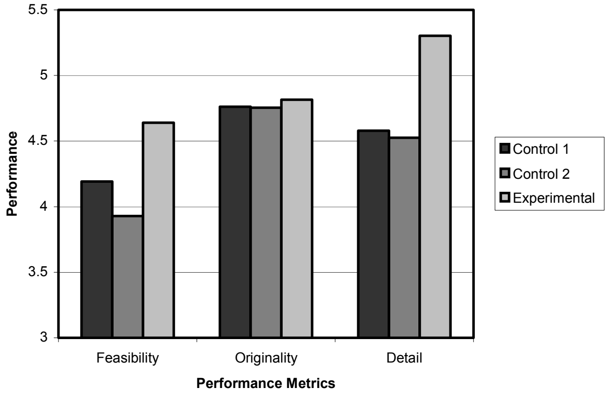


Figure 11. Evaluation of Architecture Designs

Table 13. Experimental Configuration

Group	Method Used	No. of Teams
Control 1	No method	17
Control 2	Ulrich & Eppinger's technique [19]	18
Experimental	Presented in this paper	18

Table 14. Probability of Distinct Results

	Feasibility	Originality	Detail
Comparative Groups	t-test (%)	t-test (%)	t-test (%)
Experimental / Control 1	89	20	97
Experimental / Control 2	98	19	97
Control 1 and Control 2	62	2	15

6.0 CONCLUSIONS AND FUTURE WORK

This paper presents the architecture workframe to better link functional design to embodiment design. As a notation, the workframe establishes a new domain within the design process thus facilitating a stepping stone to effectively shorten the leap from function to form. Even within the workframe itself, the architecture design process is further partitioned into incremental steps that guide the designer from known information such as previously defined constraints to increasingly greater levels of detail and specification. Although there is still a fundamental discontinuity between function and form, the representation is a powerful aid because it directs the designer through a series of small manageable steps. A particularly attractive feature of the workframe is that it has low overhead requirements in terms of the resources needed to employ the technique. Currently, the representation can be implemented manually with pencil and paper. This allows for the representation to impact very quickly a potentially large audience from novice designers in academics to practicing designers in industry.

Probably the most characteristic property of the representation is the incremental level of detail among the six diagrams. This allows the designer to take small steps toward a solution by beginning with known parameters, such as the external flows, and leading toward more detail without overwhelming the designer in any given step. This demonstration of incremental solution development is another useful contribution and it is based on the proven strategy of divide and conquer.

Given the nature of embodiment design, which has a similar process, this work suggests that improvements in embodiment design might also be achieved by the development of such a representation analogous to the architecture representation in this work. In addition, this work indicates that a construct such as an 'architecture domain,' defined by the representation, is a useful approach. Therefore this work provides a foundation for establishing such a domain in the overall scheme of the design process. In systems engineering work in particular, where architecture design and embodiment design are currently intermingled together as a single task, inclusion of this domain as a distinct design phase can offer the benefit of an additional verification step prior to embodiment design. This again is more consistent with an incremental approach which is generally effective. One important issue is how granular the design process should become. In the case of architecture design, it appears than the finer resolution afforded by the proposed representation is beneficial. However, at some point these benefits are expected to diminish and it is not clear just when this will occur.

REFERENCES

[1] Ullman, D., 2001, "Toward the ideal mechanical engineering support system," *Research in Engineering Design*. **13**: 55-64.

[2] Yu, J., Gonzales-Zugasti, J., Otto, K., 1998, "Product Architecture Definition Based Upon Customer Demands," *ASME Design Engineering Technical Conference Proceedings, DETC98/DTM-5679.*

[3] Blackenfelt, M., 2001, *Managing Complexity by Product Modularisation: Balancing the Aspects of Technology and Business During the Design Process*, PhD Thesis, Royal Institute of Technology, Stockholm.

[4] Stone, R., 1997. *Towards a Theory of Modular Design*. PhD Thesis, The University of Texas at Austin.

[5] Stone, R. B., Wood, K. L., Crawford, R. H., 2000, "Using Quantitative Functional Models to Develop Product Architectures," *Design Studies*, **21**(3): 239-260.

[6] Allen, K. R. and Carlson-Skalak, 1998, "Defining Product Architecture During Conceptual Design," *ASME Design Engineering Technical Conference Proceedings, DETC98/DTM-5650.*

[7] Zamirowski, E. and Otto, K., 1999, "Identifying Product Portfolio Architecture Modularity Using Function and Variety Heuristics," *ASME Design Engineering Technical Conference Proceedings, DETC99/DTM-8760.*

[8] Erens, F. and Verhulst, K., 1997, "Architectures for Product Families," *Computers in Industry*, **33**: 165-178.

[9] Dahmus, J. B., Gonzales-Zugasti, J. P., Otto, K. N., 2001, "Modular Product Architecture," *Design Studies*, **22**(5):409-424.

[10] Martin, M. and Ishii, K., 2002, "Design for variety: developing standardized and modularized product platform architectures," *Research in Engineering Design*, **13**:213-235.

[11] Rosen, D. W., 1996, "Design of Modular Product Architectures in Discrete Design Spaces Subject to Life Cycle Issues," *ASME Design Engineering Technical Conference Proceedings, DETC96/DAC-1485.*

[12] Sosa, M., Eppinger, S., Rowles, C., 2000, "Designing Modular and Integrative Systems," *ASME Design Engineering Technical Conference Proceedings, DETC00/DTM-14571.*

[13] Antonsson, E. and Cagan, J. (eds), 2001, *Formal Engineering Design Synthesis*, Cambridge University Press, United Kingdom.

[14] Chakrabarti, A. (ed), 2002, *Engineering Design Synthesis*, Springer, London.

[15] Pahl, G. and Beitz, W., 1996, *Engineering Design: A Systematic Approach*, Springer-Verlag, New York.

[16] Hansen, C. T. and Andreasen, M. M., 2002, "Two approaches to synthesis based on domain theory," *Engineering Design Synthesis*, Chakrabarti (ed), Springer, London.

[17] IEEE Computer Society, 1998, *IEEE Standard for Application and Management of the Systems Engineering Process - IEEE 1220,* New York.

[18] Barkmeyer, E. J. (ed), 1997, "SIMA Reference Architecture – Part 1." NIST 5939 Report, Gaithersburg, MD.

[19] Ulrich, K. and Eppinger, S., 2000, *Product Design and Development*, Irwin McGraw-Hill, Boston.

[20] Cutherell, D., 1996, "Product Architecture," *The PDMA Handbook of New Product Development,* New York, John Wiley & Sons, 217-235.

[21] Otto, K. and Wood, K., 2001, *Product Design – Techniques in Reverse Engineering and New Product Development*, Prentice Hall, New Jersey.

[22] Holtta, K., Tang, V., Seering, W. P., 2003, "Modularizing Product Architectures Using Dendrograms," (*Working Paper)* Center for Innovation in Product Development, MIT.

[23] Wood K. L. and Greer, J. L., 2001, "Function-Based Synthesis Methods in Engineering Design," *Formal Engineering Design Synthesis,* Antonsson and Cagan (eds), Cambridge University Press, United Kingdom.

[24] Chakratarti, A. and Bligh, T., 1994., "An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part I: Introduction and Knowledge Representation," *Research in Engineering Design*, **6**:127-141.

[25] Campbell, M. I. , Cagan, J., Kotovsky, K., 2000, "Agent-based synthesis of electromechanical design configurations," *ASME Journal of Mechanical Design*, **122**:1-9.

[26] Connolly, T. J., 2000, *Synthesis of multiple-energy active elements for mechanical systems*, PhD Thesis, The University of Texas at Austin.

[27] Schmidt, L. and Cagan, J., 1995, "Recursive Annealing: A Computational Model for Machine Design." *Research in Engineering Design*, **7**:102-125.

[28] Gorti, S. R. and Sriram, R. D., 1996, "From symbol to form: a framework for conceptual design," *Computer-Aided Design*, **28**(11):853-870.

[29] Markman, A. B., 1999, *Knowledge Representation*, Mahwah, Lawrence Erlbaum Associates, New Jersey.

[30] Finke, R., Ward, T., Smith, S., 1992, *Creative Cognition – Theory, Research, and Applications*, MIT Press.

[31] Jensen, T., 2000., "Function Integration Explained by Allocation and Activation of Wirk Elements," *ASME Design Engineering Technical Conference Proceedings, DETC00/DTM-14551.*

[32] Greer, J., 2002, *Effort Flow Analysis: A Methodology for Directed Product Evolution Using Rigid Body and Compliant Mechanisms*, PhD Thesis, The University of Texas at Austin.

[33] Stone, R. B., Wood, K. L., 1999, "Development of a Functional Basis for Design," *ASME Design Engineering Technical Conference Proceedings, DETC99/DTM-8765.*

[34] Liddament, T., 2000, "The myths of imagery," *Design Studies*, **21**(6):589-606.

[35] Dahl, D. W., Chattopadhyay, A., Gorn, G. J., 2001, "The importance of visualization in concept design," *Design Studies*, **22**(1):5-26.

[36] Tovey, M., Porter, S., Newman, R., 2003, "Sketching concept development and automotive design," *Design Studies*, **24**(2):135-153.

[37] Van Wie, M., 2002, *Designing Product Architecture: A Systematic Method*, PhD Thesis, The University of Texas at Austin.

[38] Zwicky, F., 1948, "The Morphological Method of Analysis and Construction," *Courant*, Anniversary Volume.

[39] Kleespies, H., 2002, Seminar on the Joint Strike Fighter Program, The University of Texas at Austin.