**Proceedings of IDETC/CIE 2005:
ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California**

DETC2005/DAC-85313

A GROUP TECHNOLOGY BASED REPRESENTATION FOR PRODUCT PORTFOLIOS

Robert L. Jordan Jr., Michael Van Wie, Robert B. Stone
University of Missouri – Rolla
r.l.jordan@umr.edu, vanwie@umr.edu, rstone@umr.edu

Jiachuan Wang
United Technologies Research Center
wangj2@utrc.utc.com

Janis Terpenny
Virginia Polytechnic Institute and State University
terpenny@vt.edu

ABSTRACT

Repository based applications for portfolio design offer the potential for leveraging archived design data with computational searches. Toward the development of such search tools, we present a representation for product portfolios that is an extension of an existing Group Technology (GT) coding scheme. Relevance to portfolio design is treated with a case study example of a hand held grinder design. Results of this work provide a numerical coding representation that captures function, form, material and manufacturing data for systems. This extends the current GT line work by combining these four types of design data and clarifying the use of the functional basis in a GT code. The results serve as a useful starting point for the development of portfolio design algorithms, such as genetic algorithms, that account for this combination of design information.

1. INTRODUCTION

Portfolio design is the problem of designing a set of related systems that individually meet customer requirements and exhibit some degree of commonality. Difficulty arises because of the conflict between these two goals. Given the example of a power tool portfolio such as the Delta Toolset as shown in Figure 1, it is apparent that certain features are common among these products such as the reuse of components. It is important to look beyond components at other resources to identify core capabilities (Meyer and Utterback, 1993). Beyond components are additional common features in terms of perhaps several different categories such as manufacturing processes, materials, and even abstractions such as product function. The aggregate of these common aspects is the platform for the portfolio or product family.

Figure 1. Delta Toolset – drill, sander, circular saw, jigsaw, flashlight

Consider the problem of adding an additional product, such as a light duty hand held grinder to this existing portfolio. One strategy is to view the problem from an optimization perspective where the goal is to maximize commonality and satisfy performance criteria for the new grinder product. Such an approach ideally should account for the large number of candidate solutions that are possible in terms of various combinations of components, manufacturing, material choices, etc. Even in the case of this toolset where each product contains only about 50 components, the combinatorial complexity suggests that computational approaches other than complete enumeration or other exhaustive searches are indicated. In this work, we develop a portfolio representation in order to address portfolio design, such as the grinder design, through evolutionary computation approaches, which appear suitable for this type of optimization problem.

Recently, genetic algorithms have been successfully used for product family design (D'Souza and Simpson, 2003; Li and Azarm, 2002). It is clear that the representation scheme is a critical element to a successful genetic algorithm (Goldberg, 2001; Fogel and Angeline, 1997; Russell and Norvig, 2003). Many encoding options are possible such as simple binary strings or hierarchical tree structures (Yoshimura and Izui, 2002; Wang eg al., 2005). The purpose of this research is to develop a representation for product portfolios that accounts for selected data currently in the product repository under development at the University of Missouri – Rolla (UMR) (Bohm and Stone, 2003). Specifically, we account for function, artifact, manufacturing, and material choice information. Note that here we refer to an artifact as a physical embodiment such as an assembly or component. This paper reports on our efforts to adopt an approach taken from Group Technology as a means to encode this product data for future use in the implementation of genetic algorithm based search tools.

2. PROBLEM CLARIFICATION AND RELATED WORK

The following presents an overview of background material to better describe the context and constraints for the current work. Each section is presented with respect to the design of a new grinder variant from the Delta Toolset in order to more fully explain the types of design issues related to design problems at the scale of this case study.

2.1 Repository Based Portfolio Design

The product repository at UMR incorporates several types of product data including both function and form information (Bohm and Stone, 2003). In terms of portfolio design for the Delta Toolset, the repository can offer the user support in different modes. The user can search and browse through existing artifacts of the toolset in order to manually formulate a grinder design variant. Alternatively, the user can obtain output from the repository in the form of a morphological matrix, which is a recent addition to the repository feature set. The following describes a manual design (as opposed to an automated algorithm) process involving the repository. The intent is to sequentially identify the needs, functions, and components for the platform of the Delta Toolset given the introduction of the grinder to the portfolio.

2.1.1 Gather Customer Needs

For the case when the user performs a manual design for the grinder, the process can proceed according to the following scenario. First, a set of customer needs is gathered for both hand held tools in general and for the proposed grinder design specifically. These customer needs are processed according to a recently developed platform design method (Kurtadikar, 2004), which partitions the set of needs into platform needs and unique needs. It is important for the reader to understand that such separation between platform and unique needs may involve many driving factors including, for example, edicts from higher management where such influences to date have not been well understood from a design theory stance. Nevertheless, given some initial set of customer needs, a group of platform needs is identified as shown in Table 1.

Four of the 28 needs in Table 1 are shown with an asterisk. These four cases involve needs that are found not relevant to include in the remainder of the design for the following reasons. "Leave unmarked / non-marred surface" was not relevant to the grinder since the entire purpose of a grinder is to affect the surface with sandpaper. "Versatile" and "reliable / durable" are not issues that affect platform design for the grinder case study. Versatility in this case is effectively subsumed into other 'ease of use' related needs that are well accounted for already. Reliability is perhaps a greater embodiment problem than a platform design issue. Finally, the need for a good brand name is simply out of the scope of the design problem.

Table 1. Platform Customer Needs

use only 1 battery - not multiple
fast to use
intuitive controls
long life of accessories - sand pad, saw blade, etc.
crisp manipulation of controls
*leave unmarked / non-marred surface
well secured battery
accept standard accessories / attachments
light force to trigger
fool proof - safe
*good brand name
inexpensive and available replacement batteries
minimize noise
support different levels of abrasion (rough to polish)
easy battery removal / replacement
indicate position to saw in a straight line
fit hand
minimize number of fingers required to manipulate controls
*versatile
effective on mulitple material types (grinder)
solid feel (not too light or flimsy)
control chatter / vibration / grabbing
quick recharge time
easy to keep steady / stable
high powered
easy to change bits
run a long time on a charge
*reliable / durable

2.1.2 Functional Design

Given this set of platform needs, functions are generated for each need. This is accomplished by first searching each existing toolset product in the repository and enumerating the set of functions in the existing toolset. Next, functions are selected from this list in order to address each of the customer needs identified above. Functions that are repeated are pruned and the final list of identified functions for the platform of the grinder variant is established as shown in Table 2. As a point of reference, the grinder functions differ from the Delta Toolset sander only by the two functions shaded in gray. Even these two functions are present in at least one other product in the Delta Toolset.

Table 2. Grinder Functions

convert electrical energy
distribute mechanical energy
guide electrical energy
guide human energy
import human energy
import solid material
position human energy
secure human energy
secure solid material
separate solid material
store electrical energy
supply electrical energy
transfer electrical energy
transfer mechanical energy

2.1.3 Embodiment Design

Upon establishing needed functionality, candidate physical solutions are generated by identifying all artifacts from the existing portfolio that are used to embody the list of functions in Table 2. A morphological matrix is produced from this set of alternatives and a partial view of this result is shown in Figure 2. At this point, the designer may generate alternative concept variants using alternative combinations of suitable artifacts. Performance models may be used for individual alternatives to evaluate suitability of a given alternative as it relates to the customer needs for the grinder variant.

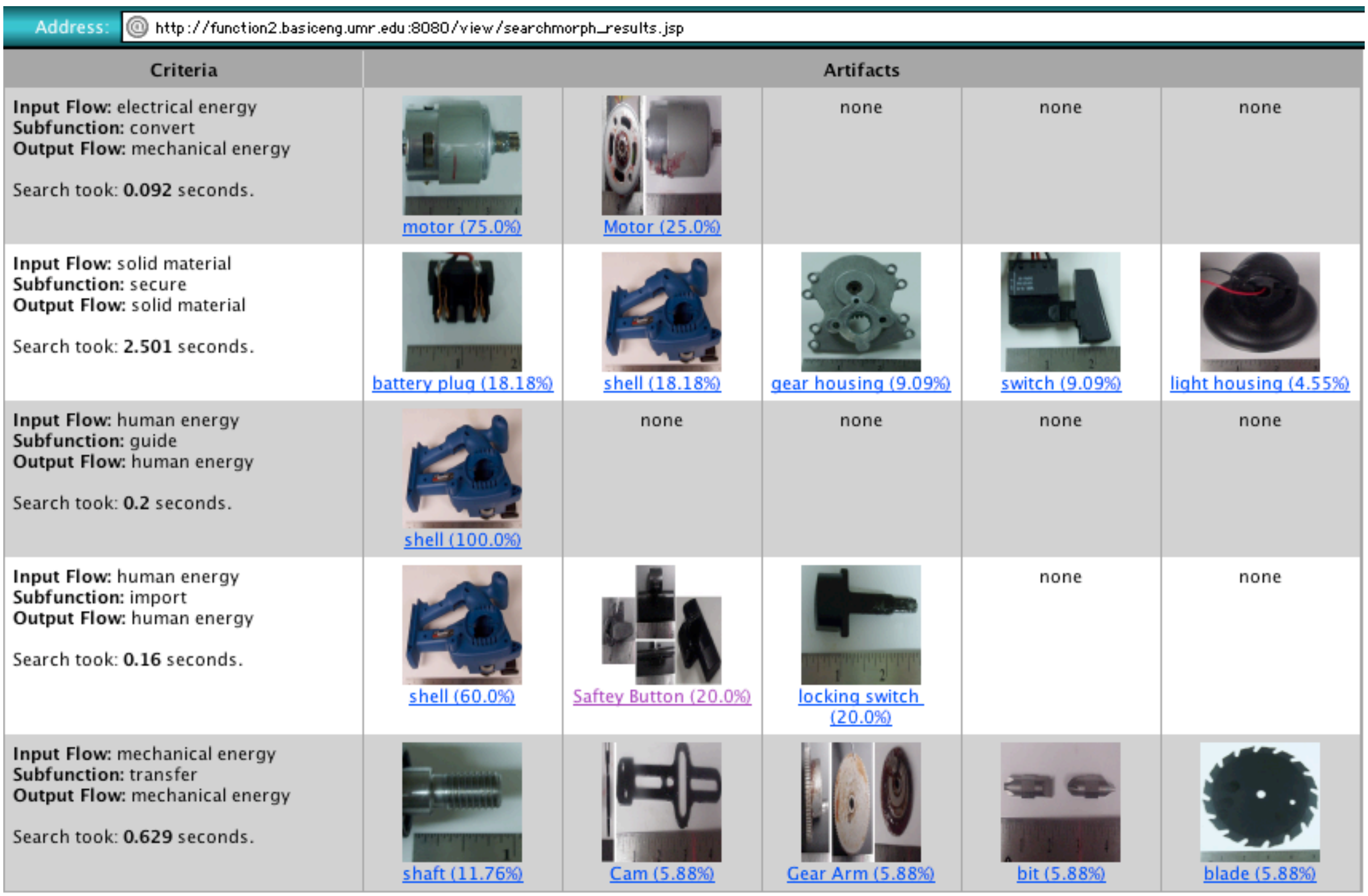

Figure 2. Partial Morphological Matrix for Grinder Variant (solutions searched from existing Delta Toolset)

The repository currently facilitates portfolio design insofar as one can browse the existing product family and automatically generate morphological matrix searches given some set of specified functionality. Our vision is to move toward a suitable optimization approach such as genetic algorithms in conjunction with the repository in order to enhance the engineer's ability to generate solutions with a greater number of factors including, for example, manufacturing issues. The next two sections highlight repository capabilities.

2.2 Repository Design Tools

The existing repository at UMR consists of tools such as a morphological matrix generator that can return existing candidate physical solutions based on a specified function list. One next step in ongoing development of this repository is to develop a portfolio design tool that expands on this feature set in order to perform portfolio design more effectively and efficiently than manual search.

Genetic algorithms are an appropriate technique for dealing with the large combinatorial problem of searching for not only components as illustrated in the previous section, but also functional descriptions and various embodiment characteristics like manufacturing choices of those components. Commonality in a broad sense is sought for all resources and processes used in product development. This is an interpretation similar to the notion of core capabilities by Meyer and Utterback (1993). Encapsulating repository data in a representation suitable for search techniques such as genetic algorithms is needed.

2.3 Representing Product Portfolios

A product portfolio consists of multiple products each of which are composed of multiple functions, artifacts, and embodiment properties. A variety of representations are appropriate for portfolios depending on the particular design application. Nanda et al. (2004) investigate the use of Semantic Web technologies to develop ontologies of families. Related work has demonstrated the use of web-based product family visualization methods coupled with an optimization algorithm for designing product families (Mulberger and Simpson, 2004). D'Souza and Simpson (2003) represent a set of design variables for product families in a string format that is subsequently used with a genetic algorithm optimization approach. This particular approach employed a relatively narrow perspective of the product family by restricting the representation to a small (~5) number of design variables. Insofar as a given optimization approach, such as genetic algorithms, is feasible, representations like those similar to Group Technology are needed to incorporate a larger set of product family data. Recent work by Al-Ahmari (2004) employs a Group Technology coding representation with a fuzzy clustering approach in order to select and form part families. The following section presents an overview of Group Technology.

2.4 Group Technology

Group Technology (GT) is a coded representation of information about a specified part or artifact usually given in an alphanumeric string. Such a coding scheme can be chain-type (list), hierarchical (tree), or some hybrid type for example. The basic idea behind GT, first created by Mitrofanov, is not new (Opitz, 1971). GT is a manufacturing philosophy that supports methods to exploit commonality in design, assembly, fabrication, and material characteristics of an artifact. Comparison of two different GT codes can allow for estimates of product similarity.

Currently, there are several GT coding schemes for individual mechanical parts. The scheme by Opitz has been most generally used as the basic framework for understanding coding systems. The Opitz code can be applied to machine parts, non-machined parts, purchased parts, and considers both design and manufacturing information (Opitz, 1969). Henderson et al (1988) developed a classification to automatically generate the DCLASS GT code of rotational products from a 3-D CAD database. Chen (1989) developed a computerized GT coding system that operates on a product design specified in the IGES format. In this system, an IGES file is converted into a customized product description file which is then transformed into a format from which the GT codes can be extracted. Bhadra and Fischer (1988) developed a GT classification to catalog and code rotational symmetric parts. Shah and Bhatnagar (1989) developed an automated GT coding system based on the Opitz coding scheme for machined parts. This system assigns pre-defined classification codes for each attribute of its feature-based CAD system. The extensive information captured by the taxonomy codes is used to determine individual feature characteristics and the relationships between features and entire parts. MICLASS was developed from the Organization for Industrial Research, Inc. and is the most commonly used code system in metal manufacturing. DCLASS is from Brigham Young University and KK-3 was developed in Japan for the Promotion of Machine Industry. Currently, there is no broad consensus for a particular code used for the classification of parts. Most coding schemes have been specifically engineered for a company or industry (Girdhar and Mital, 2001a).

As noted previously, Opitz's work has been the basis of several GT codes. Opitz's goal was to create a code that would be a numerical representation of workpieces and their attributes. Girdhar and Mital revisited GT and Opitz's code focusing on expanding the GT part coding for functionality. Girdhar and Mital theorized that the addition of a code incorporating function would help in the selection of artifacts to fulfill specific product needs. In Part 1 of their work, the main focus was on the development of a basis for functionality coding. The code that was presented can be seen in Table 3.

Table 3. Function Code taken from (Girdhar and Mital, 2001a)

Primary Code	Basic Function (charachteristic)	Secondary Code	Sub-class	Function Code
1	CHANGE (type)	1	Convert	11
		2	Create	12
		3	Destroy	13
2	VARY (magnitude)	1	Regulate	21
		2	Change Form	22
		3	Actuate	23
		4	End	24
3	CONNECT (number)	1	Couple	31
		2	Mix	32
		3	Separate	33
4	CHANNEL (place)	1	Transfer	41
		2	Transmit	42
		3	Guide	43
5	STORE (time)	1	Intake	51
		2	Extract	52
		3	Hold	53

In Part II of Girdhar and Mital's work, the presented function code was added to Opitz's Code to allow for coding of workpieces. Since many workpieces can serve multiple functions, there is a place holder for 3 functions. Girdhar and Mital stated these to be Primary, Secondary, and Tertiary functions. This designation is confusing since the code uses the functional basis terminology of a hierarchy (primary, secondary, and tertiary) instead to simply refer to different instances of function that have no hierarchical property. Moreover, the various levels of the functional basis were used inconsistently across the different "levels" of primary, secondary, and tertiary. We believe this is an incorrect use of the functional basis and our work corrects this issue.

Workpieces having less than 3 functions would have zero's in the place of the missing functions. A list of 231 workpieces was compiled and coded with just the function code. The expanded Opitz code can be seen in Table 4. The shaded region represents the Function Code as done by Girdhar and Mital.

Table 4. Expanded Opitz Code from (Girdhar and Mital, 2001b)

Form Code					Supplementary Code				Function Code						Secondary Code
									Primary Function		Secondary Function		Tertiary Function		
Digit 1	Digit 2	Digit 3	Digit 4	Digit 5	Digit 6	Digit 7	Digit 8	Digit 9	Digit 10	Digit 11	Digit 12	Digit 13	Digit 14	Digit 15	Digit 16 and onwards
General Shape (Part Class)	External shape and relevent form elements	Internal shape and main bores	Surface plane machining	Auxiliary holes and gear types	Dimensions	Material	Original shape of raw materials	Accuracy	Basic Function	Function Sub-Class	Basic Function	Function Sub-Class	Basic Function	Function Sub-Class	Company specific (production operation type & sequence)

3. OBJECTIVES

The purpose of this work is to develop a Group Technology based coding scheme to represent product portfolios. As a practical matter, this representation is at the product level given that modeling of the portfolio is simply an aggregate of individual products. However, some discussion is given toward the issue of how exactly this aggregation can occur. The primary deliverable is a set of codes that capture function, component, manufacturing, and material issues. The secondary deliverable is a preliminary examination of the application of these codes for use in repository based design tools for product portfolio design. Specifically, genetic algorithm search methods are considered.

4. METHODOLOGY

The following covers the methods and procedure for the development of the new coding scheme, its elements, and its connection with a repository based system. Figure 8 illustrates where the GT code presented next fits into an overall process of portfolio design. The process begins customer needs analysis and proceeds through functional design and morphological matrix generation. The coding scheme is used to capture function, component, material, and manufacturing data in a computable representation for use in a portfolio design algorithm of choice.

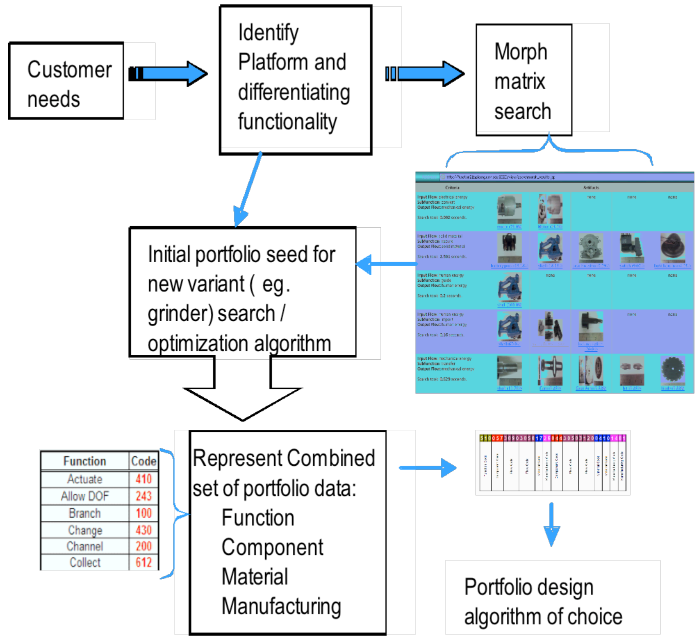

Figure 3. Overall Portfolio Design Process

4.1 Coding Scheme Development

The new code is a representation of a subset of the information currently established in the UMR design repository (http://function.basiceng.umr.edu). The code is broken down into five elements: Component, Material, Manufacturing, Function, and Flow.

The first five digits in Opitz's GT code which represents Component Class, External Shape Elements, Internal Shape Elements, Surface Machining, and Auxiliary Holes and Gear Teeth, respectively, is replaced with a simpler code taken from a component basis (Greer, et al., 2003). The component basis is a standard naming convention for mechanical parts. The component basis is a list of human-made mechanical transmission artifacts as functional forms, geometric shapes, simple machines, and natural forms. Each artifact is congruent with a set of synonymous artifacts and each artifact is represented with a three digit code. The code is a simple numerical assignment of a physical artifact based upon its position within the component basis.

Currently, the component basis has 92 distinctly different artifacts with a portion of the Component Basis shown in Table 5. The complete code is given in Appendix A.

Table 5. Component Basis Code Excerpt

Code	Name	Synonyms
00	Acoustic Insulator	silencer
01	Agitator	stirrer, mover
02	Airfoil	wing
03	Axle	stub axle, beam axle, axle shaft
04	Battery	
05	Bearing	journal bearing, thrust bearing

The next element in the code is the material. There is a Hierarchical Organization of Material Alternatives, which is broken down into 4 levels (Poli, 2001).

Level I: Selection between Metal and Plastics.
Level II: If metal, then cast or wrought. If plastic, then thermoplastic or thermoset.
Level III: Selection of type of material: Aluminum, Steel, Copper, ABS, nylon, etc.
Level IV: Selection of specific alloy or resin ie. Al A380.0, CRS ASTM A606, Polycarbonate

As seen above, the higher levels are most specific. The code being presented, allows for a Level III classification. This level was chosen because of the lengthy list of Level IV materials and that the Level IV materials change frequently. MatWeb shows a list of over 46,000 engineering materials (MatWeb, 2005). Level III proved to be the most specific list without much variation. A simplified list of engineering metals was created with the aid of Askeland's text *The Science and Engineering of Materials*. Askeland decomposed engineering materials into 6 categories: ferrous, nonferrous, ceramic, polymers, composites, and construction (Askeland, 1994). Each category is then broken down into sub-categories. The materials code was based on this sub-category list. An excerpt can be seen in Table 6. The complete scheme is shown in Appendix B.

Table 6. Material Coding Excerpt

Code	Material
00	Multiple Materials
01	Carbon/Low Alloy Steels
02	Tool Steels
03	Cast Irons
04	Stainless Steel
05	Beryllium
06	Copper Alloys

The code consists of 2 digits and a total of 27 materials. The next portion of code is the manufacturing element where a list of *all* the manufacturing processes seemed to be impossible due to the fact that there is a surplus of processes and multiple variations of each. A simple list was comprised from data taken from *The Material Selector*. The list covers metallic, plastic, machining, and joining manufacturing processes. This list is very simplistic and only covers a few manufacturing processes (Material Selector, 1980). This list and its code can be seen in Table 7 and the complete scheme in Appendix C.

Table 7. Manufacturing Code Excerpt

Code	Manufacturing
00	Machining
01	Bending
02	Sand Casting
03	Shell Mold Casting
04	Full Mold Casting
05	Permanent Mold Casting
06	Die Castings

The next code element is an adaptation of the Functional Basis. There have been several attempts at a making a concise list of all the possible function taxonomies and the taxonomy chosen for the Function Code is the Functional Basis (Hirtz et al., 2002). The code for the Functional Basis is given with a three digit numeric representation based upon the position of the function in the Functional Basis table, which can be viewed in Appendix D. A portion of the table and code is shown in Table 8.

Table 8. Functional Basis Code Excerpt

Primary		Secondary		Tertiary	
#	Definition	#	Definition	#	Definition
1	Branch	1	Separate	1	Divide
				2	Extract
				3	Remove
		2	Distribute		
2	Channel	1	Import		
		2	Export		
		3	Transfer	1	Transport
				2	Transmit
		4	Guide	1	Translate
				2	Rotate
				3	Allow DOF

100 - Branch
110 - Separate
111 - Divide
112 - Extract
113 - Remove
120 - Distribute
200 - Channel
210 - Import
220 - Export
230 - Transfer
231 - Transport
232 - Transmit
240 - Guide
241 - Translate
242 - Rotate
243 - Allow DOF

The final element of the code is the Flow Code. An essential part of functional modeling is the representation of the *flow* quantities that are inputs and outputs of functions (Stone and Wood, 2000). The flows are broken down into 3 primary classes: material, signal, and energy. Each one of these can be broken down into secondary and some tertiary classes. A portion of the code is given in Table 9. The Flow Code is a simple 4 digit number based upon the location of the flow within the Functional Basis Flow Table. The code can be seen in its entirety in Appendix E.

Table 9. Functional Basis Flow Code Excerpt

Primary		Secondary		Tertiary	
#	Definition	##	Definition	#	Definition
1	Material	1	Human		
		2	Gas		
		3	Liquid		
		4	Solid	1	Object
				2	Participate
				3	Composite
		5	Plasma		
		6	Mixture	1	Gas - Gas
				2	Liquid - Liquid
				3	Solid - Solid
				4	Solid - Liquid
				5	Liquid - Gas
				6	Solid - Gas
				7	Solid - Liquid - Gas
				8	Colloidal

1000 - Material	1060 - Mixture
1010 - Human	1061 - Gas - Gas
1020 - Gas	1062 - Liquid - Liquid
1030 - Liquid	1063 - Solid - Solid
1040 - Solid	1064 - Solid - Liquid
1041 - Object	1065 - Liquid - Gas
1042 - Participate	1066 - Solid - Gas
1043 - Composite	1067 - Solid - Liquid - Gas
1050 - Plasma	1068 - Colloidal

4.2 Coding Layout

The coding scheme is designed with the expectation that it will serve as a departure point for developing a genetic algorithm for performing product family design. Individual artifacts have multiple manufacturing processes, functions, flows, and materials. The following discussion provides a survey of potential implementations of the coding scheme. This is not an exhaustive enumeration of layouts options, but rather a brief look at the pros, cons, and challenges associated with implementing the coding scheme in a few different variations.

There are several different ways to aggregate and represent the combined elements of the coding scheme we present in this work. Here we examine lists and graphs. Lists are simple structures where one implementation is to simply allocate each of the coding scheme code types (eg. function code) to an element in a list, which overall describes an artifact of function. Using a list in this manner is a near alternative to the use of string of coding elements associated with a variable. Such a string implementation, which is somewhat conventional in genetic algorithm applications, is a poor option for the coding scheme presented in this work given the required string length for describing even small products. Scalability is more easily achievable with lists rather than single variable strings. Additionally, lists can be embedded with other data for needed information given a design algorithm.

Graphs offer a degree of connectedness that single level lists do not. In particular, a tree is an alternative for structuring the coding scheme. One key feature is the hierarchical aspect of nodes that are arranged in such a way to determine parent-child relations. This hierarchical property may be desirable for capturing multiple levels of an artifact or function structure. This is a reasonable approach for handling multiple levels of similarly structured coded information given that trees can be defined recursively. Another benefit of graphs in general is the convenience of using matrices (adjacency matrices). As an extension of using graphs to implement combined codes from the coding scheme, one may use such matrix-based approaches. This makes available a number of linear algebra techniques for handling product data.

Due to the various possibilities for arranging a numeric list, only a select few are shown here. The type of representation can be chosen by the customer based upon the customer needs such as the number of material and manufacturing processes that should be represented. The arrangements chosen to be represented here use a Component-centric representation and a Function-centric representation.

A Component-centric representation is a list of numbers that represent *a specific component* along with its possible functions, flows, materials, and manufacturing process. This list would allow for comparison of all available information for a specific component. An illustration can be seen in Figure 4. Note that the representation length is dynamic in that as many functions or flows, etc. can be represented as necessary.

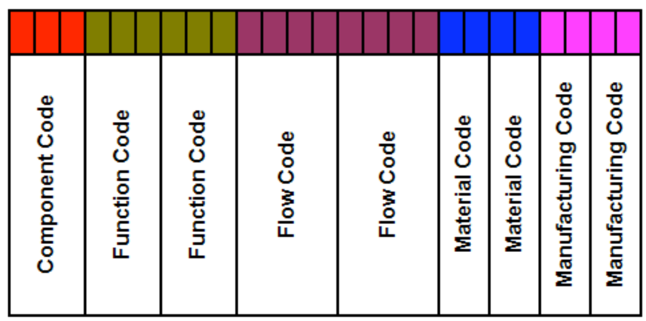

Figure 4. Component-centric Representation

A Function-centric representation is a numeric list that provides a specific function and all of its associated components, materials, and manufacturing for the specified function. In comparison with the Component-centric representation, a function perspective could be significantly longer due to the large number of components that embody a given function. Either code must be engineered toward a particular algorithm choice, which will have direct impact on the length of the representation. This layout can be seen in Figure 5.

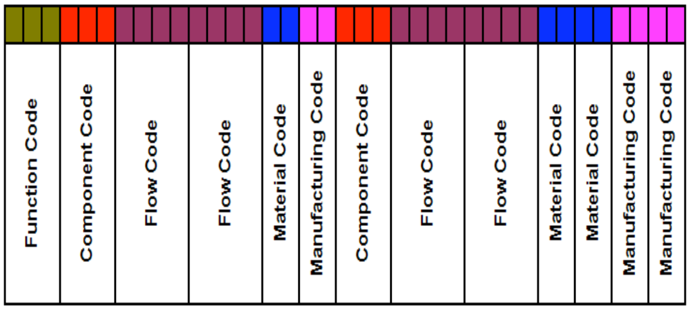

Figure 5. Function-centric Representation

A hierarchical Component-centric structure can be seen in Figure 6 and relates to the Delta Tool Set. The product portfolio breaks down into 5 products: sander, drill, flashlight, saw, and jig-saw. Each product can then be decomposed into artifacts. At this level the artifacts can be coded with respect to their synonymous components. At this level the code would become a hybrid, containing both a hierarchical and list structure.

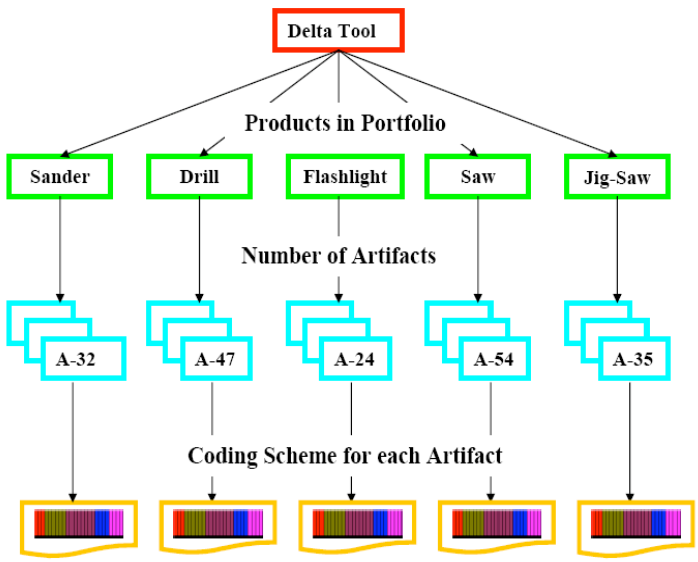

Figure 6. Product Portfolio Hierarchical Model Example

Similarly, another hierarchical representation can be based upon function. An example in Figure 7 is based on the *convert* function. One of the many possible components that has convert as a function is a lever. This figure illustrates a portion of all the components, flows, materials, and manufacturing processes for the function *convert* as instantiated by a lever.

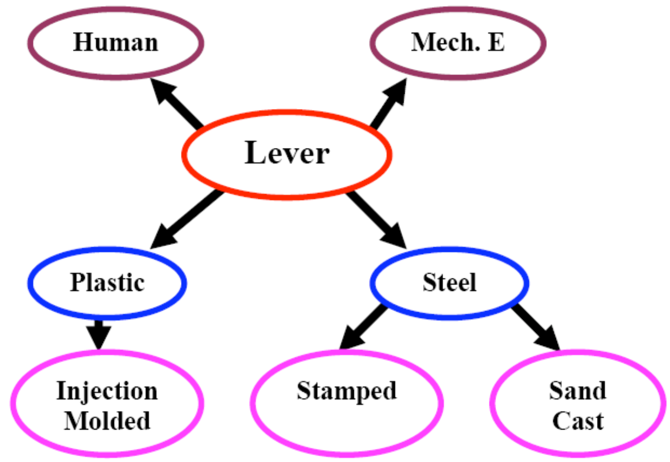

Figure 7. Covert Function Hierarchal Model Example

Details of any particular structure of a GT code for a genetic algorithm or other search method are ultimately best designed in conjunction with the algorithm. One recommendation at this point is that from the amount of information within the UMR Repository and the expectancy of its growth, it is reasonable to assume that if the code is represented as a single level string, the code would be too long to be practical. Beyond this, both lists and hierarchical implementations are potential viable options. These layouts are a precursor to the creation of a genetic algorithm for performing searchers of product family solutions in a top-down fashion given initial required functionality or perhaps a requirement based on other data in the coding scheme such as required components.

4.3 GA Search Method

As illustrated in Figure 6 of the previous section, an existing product portfolio in the design repository is a set of products, and each individual product can be GT coded with a number of artifacts using hierarchical graphs for example. Consider the scenario when a new product, such as the grinder, needs to be designed into an existing product family. The new design to be added into the product portfolio needs to be optimized such that it satisfies specific customer needs for desired performance requirements while maximizing commonality with the existing platform for reduced cost benefit from design and manufacturing reuse. Sometimes tradeoffs need to be made between performance and commonality (cost). Figure 8 illustrates a typical outline of genetic algorithm-based search method to realize this optimization process.

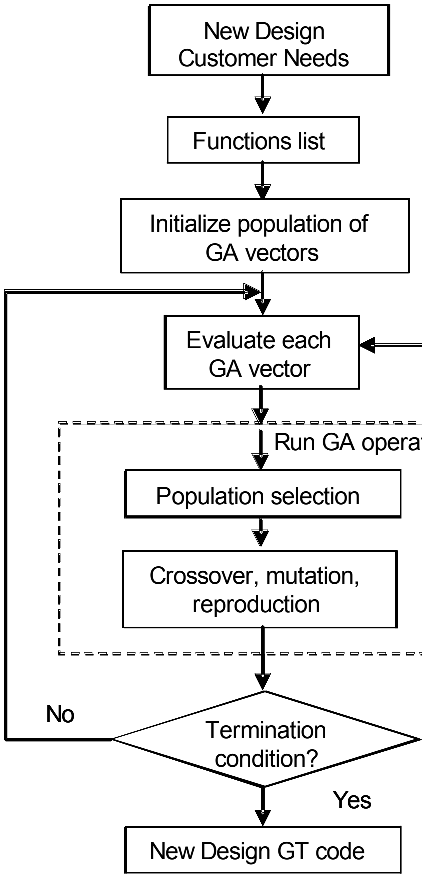

Figure 8. GA Search Procedure

First, specific customer needs for a new design are gathered and translated to corresponding functions, as described in Section 2.1. Here we encode new product designs using GA vectors, a vector of GA lists with each list representing one function GT code. Since functions and flows are known, the GA vector has a list of GT codes filled in with function codes and flow codes, while leaving other codes initialized randomly in the first generation. The evaluation of a GA vector is based on two evaluation criteria. The first criterion is whether the solution generated satisfies specified performance requirements. The second criterion is the commonality metric between the new design GT code and each individual product GT code in the product portfolio. Different measures can be assigned when there is function, flow, component, material, and manufacturing commonality. Suppose there are 5 products (A, B, C, D, E) in the portfolio, and their commonality measures with the new design are Fa, Fb, Fc, Fd, and Fe, with weights Wa, Wb, Wc, Wd, We, respectively, then the total commonality metric = (WaFa+WbFb+WcFc+WdFd+WeFe). The overall fitness evaluation of a given solution is the combination of both performance satisfaction and a commonality metric.

The GA vector population goes through generations of evolution until termination criteria are satisfied. The termination criteria can be the number of generations, the target maximum fitness value of the population, etc. This description of a GA search method above is generic given that this work does not specify a particular GA algorithm, a specific fitness function, or mutation / crossover operations. Rather, our focus is on the GT approach for product data representation with the intent to use this approach for more detailed GA development in future work.

5. EXAMPLES

5.1 Coding Example

In order to illustrate the Component-centric coding scheme, a shaft as shown in Figure 9 from the drive train assembly of the Delta Sander is selected. All the information for this code is taken from the repository and a representation of this data can be viewed in Appendix F. Figure 10 shows the code of the shaft and the information that each number sequence represents. This example simply illustrates a code for a single shaft component in list form Note that the code for other artifacts may be of different length.

Figure 9: Drive Train Assembly of Delta Sander

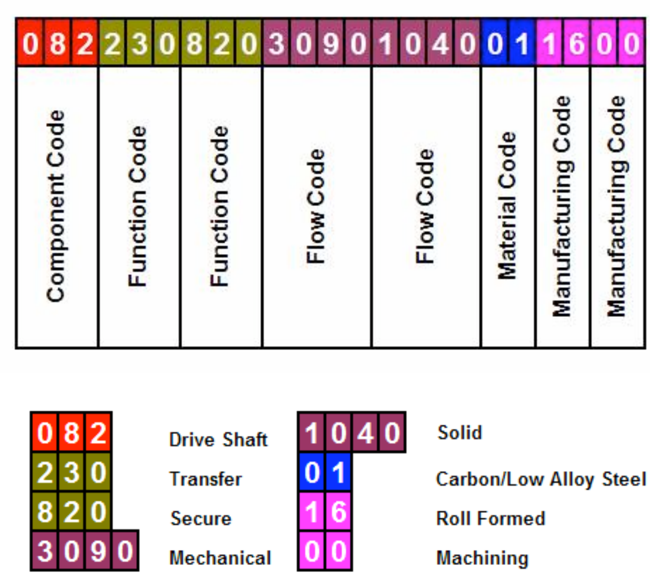

Figure 10. Component-centric Coding Example of Shaft

The second example is of a Function-centric code of the function *convert*. Again all the information needed for the code is established from data in the repository. This illustration only shows two components that achieve the convert function: a lever and a heating coil. As seen from Figure 11, components, flows, materials and manufacturing processes are included.

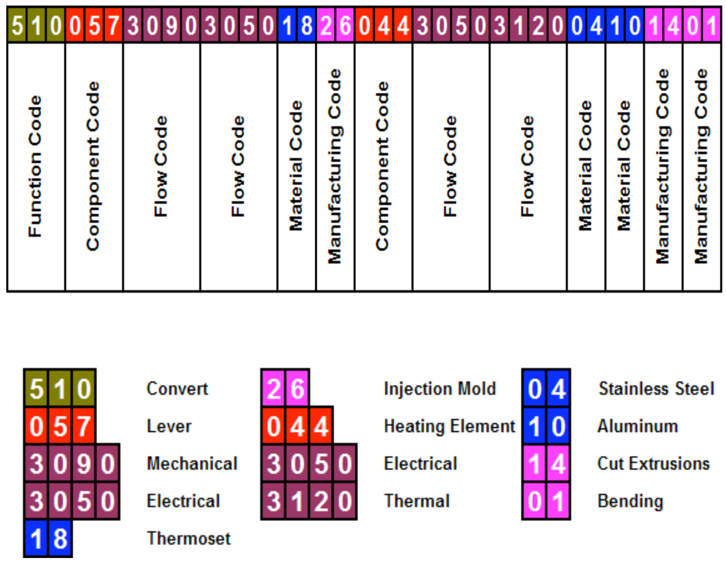

Figure 11. Function-centric Coding Example

5.2 Grinder Design Example

Consider the functions needed for the grinder as seen in Table 2. The grinder is very similar to the Delta Sander with the exception of two functions: position human energy, separate solid material. It is logical to find an artifact that is already being produced and satisfies the functions and the sander is a very near alternative. Using the UMR Repository and searching for commonality of function within the current Delta Tool Set, one finds that the Jigsaw and Circular Saw have the function *position human energy* and the Circular Saw has an artifact that satisfies the *separate solid material* function.

The *position human energy function* is manifested by the housing of the saw and jigsaw. This shows that the housing is a good option for this artifact solution. Since the saw, jig-saw, and grinder all have different structures, it is not practical to use identical housings. Looking deeper into the code will show that the housing for the saw and jig-saw are both made from thermoset plastics and are both manufactured by injection molding. This sub-group shows that it is possible to manufacture this housing using current materials and processes.

The blade of the circular saw is what fulfills the *separate solid material*. Clearly, a blade would not be satisfactory for a grinding tool. The basic operation of the grinder is to reduce material into smaller fragments and remove by the use of very small cutting surfaces. Obviously a blade would not accomplish this task, but can be used as a basis for a redesign of a similar artifact.

6. CONCLUSIONS AND FUTURE WORK

This work presents a Group Technology based representation for capturing four elements of product data: function, component, material, and manufacturing data. This representation is an incremental contribution toward new computational portfolio design methods. The method discussed has been shown as a candidate method to illustrate components and their attributes. We do not claim that this is an optimized solution, but rather one scheme to capture information associated with four elements of product data along with brief examination of lists and graphs for structuring this data. Explanation of the representation is aided in part through discussions of a grinder design example. These results serve as preliminary steps toward developing computational approaches such as genetic algorithms that account for multiple types of product data rather than more narrowly scoped genetic algorithms found in prior work related to portfolio design. A peripheral result of this research is a set of comments correcting the somewhat confusing use of the functional basis in a paper by Girdhar and Mital (2001b). The next step in this work is the development of a genetic algorithm or other search technique to implement the representation in a system that integrates with the existing UMR design repository. Further work should examine particular GA approaches using a GT based representation.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. IIS-032541. Any opinions, findings, and conclusions or recommendations presented in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Aho, A. and Ullman, J., 1995, *Foundations of Computer Science, C edition*. W. H. Freeman and Company.

Al-Ahmari, A., “Part Feature Selection and Family Design in Group Technology Using a Fuzzy Approach,” *International Journal of Computer Applications in Technology*, 21(3):129-136.

Askeland, D., 1994, *The Science and Engineering of Materials, 3rd ed*. Boston: PWS Publishing Company.

Bhadra, A. and Fischer, G.W., 1988, "A New GT Classification Approach: A Database with Graphical Dimensions," *Manufacturing Review*, 1(1): 44-49.

Bohm, M., Stone, R., Szykman, S., 2003, “Enhancing Virtual Product Representations for Advanced Design Repository Systems,” *ASME Proceedings of Design Engineering Technical Conference, DETC2003/CIE-48239*.

Chen, C., 1989, "A Form Feature Oriented Coding Scheme,". *Proceedings of the 11th Annual Conference on Computers and Industrial Engineering*, pp. 227-233.

D'Souza, B. and Simpson, T., 2003, "A Genetic Algorithm Based Method for Product Family Design Optimization," *Engineering Optimization*, 35(1):1-18.

Fogel, D. and Angeline, P., 1997, "Guidelines for A Suitable Encoding," *Handbook of Evolutionary Computation*, Back, T, Fogel, D., Michalewicz, Z (eds.).

Greer, J., Stock, M., Stone, R., Wood, K., 2003 "Enumerating the Component Space: First Steps Toward a Design Naming Convention for Mechanical Parts" *ASME Proceedings of Design Engineering Technical Conference*, DETC2003/DTM-48666.

Girdhar, A. and Mital, A., 2001a, "Expanding Group Technology Part Coding for Functionality: Part I – Developing a Functional Basis for Classification," *International Journal of Industrial Engineering – Applications and Practice*, 8(3):186-197.

Girdhar, A. and Mital, A., 2001b, "Expanding Group Technology Part Coding for Functionality: Part II – Functional Classification of Workparts and Application to Design," *International Journal of Industrial Engineering – Applications and Practice*, 8(3):198 – 209.

Goldberg, D., 2002, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer.

Henderson, M., and Musti, S., 1988, "Automated Group Technology Part Coding from a Three-Dimensional CAD Database,". *Journal of Engineering in Industry*, 110(3):278-287.

Hirtz, J., Stone, R., McAdams, D., Szykman, S., Wood, K., 2002, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts," Research in Engineering Design 13(2):65-82.

Kurtakikar, R., Stone, R., Van Wie, M., McAdams, D., 2004, "A Customer Needs Motivated Conceptual Design Methodology for Product Portfolios," *ASME Proceedings of Design Engineering Technical Conference, DETC2004/DTM-57289*.

Li, H. and Azarm, S., 2002, "An approach for product line design selection under uncertainty and competition," *ASME Journal of Mechanical Design*, 124(3), 385–392.

Materials Selector, 1980, December 1, Materials Engineering, pp.C166 - C177.

MatWEB: Material List, 2004, http://www.matweb.com/search/GetAllMatls.asp

Meyer, M. and Utterback, J., 1993, "The Product Family and the Dynamics of Core Capability," *Sloan Management Review*, Spring-93, pp.29-47.

Mulberger, J. and Simpson, T., 2004, "Advancements in a Web-Based Framework in Product Family Optimization and Visalization," *ASME Proceedings of Design Engineering Technical Conference, DETC2004/CIE-57688*.

Nanda, J., Thevenot, H., Simpson, T., Kumara, S., 2004, "Exploring Semantic Web Technologies for Product Family Modeling," *ASME Proceedings of Design Engineering Technical Conference, DETC2004/CIE-57683*.

Opitz, H., 1970, *A Classification System to Describe Workpieces,* Pergamon Press Ltd.

Opitz, H., and Wiendahl, H.P., 1971, "Group Technology and Manufacturing Systems for Small and Medium Quantity Production," *International Journal of Production Research*, 9(1):181-203.

Opitz, H., Eversheim, W., Wiendahl, H.P., 1969, "Workpiece Classification and it Industrial Application," *International Journal of Machine Tool Design Research*, 9, 39-50.

Poli, C., 2001, *Design for Manufacturing: A Structured Approach,* Butterworth-Heinemann.

Russell, S. and Norvig, P., 2003, *Artificial Intelligence: A Modern Approach*, 2nd ed., Prentice Hall.

Shah, J. and Bhatnagar, A., 1989, "Group Technology Classification from Feature Based Geometric Models," *Manufacturing Review*, 2(3):204-213.

Stone, R. and Wood, K., 2000, "Development of a Functional Basis for Design," Journal of Mechanical Design 122(4):359-370.

Wang, J., Fan, Z., Terpenny, J. P., Goodman, E. D., 2005, "Knwoledge Interaction with Genetic Programming in Mechatronic Systems Design Using Bond Graphs," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(2):172-182.

Yoshimura, M. and Izui, K., 2002, "Smart Optimization of Machine Systems Using Hierarchical Genotype Representations," *ASME Journal of Mechanical Design*, 124(3):375-384.

Appendix A: Component Basis Code

Code	Name	Synonyms
00	Acoustic Insulator	silencer
01	Agitator	stirrer, mover
02	Airfoil	wing
03	Axle	stub axle, beam axle, axle shaft
04	Battery	
05	Bearing	journal bearing, thrust bearing
06	Belt	strap, girdle,band,restraint,strip
07	Bladder	baloon, inner tube, membrane
08	Blade	cutting edge,knife,razor,scraper
09	Bracket	Cantilever, console, corbel, strut
10	Burner	
11	Cable	wire,lead,chain,rope
12	Cam	eccentric,cam plate,camshaft
13	Cap	stopper,plug,bung
14	Carousel	
15	Chip	integrated circuit, transistor
16	Choke	throttle
17	Circuit Board	circuit card, board, card
18	Coating	Inside layer, lining, facing, liner
19	Comb	rake
20	Condenser	
21	Container	Receptacle, receiver, holder
22	Cover	top, lid, hood, shield, shroud, guard
23	Cushion	filling, wadding, pad
24	Diode	
25	Displacement Gauge	
26	Divider	diaphragm, partition, panel, wall, barrier
27	Door	Gate, flap, access panel, entrance
28	Driveshaft	output shaft, input shaft, jack shaft, half shaft
29	Electric Cord	
30	Electric Insulator	insulation
31	Electric Motor	actuator, dc motor
32	Electric Resistor	
33	Electric Wire	coil, transformer
34	Evaporator	
35	Extension	
36	Fan	windmill, impeller, propeller
37	Flywheel	inertia wheel, momentum wheel
38	Friction Enhancer	
39	Gear	cog wheel,rack,pinion,ring,sun planet
40	Gripper	grip manipulator,graber
41	Guide	v-guide, channel, pilot, track, path, way, locating hole, pathway, trace, jig pin
42	Handle	handle,hand hold
43	Heat Exchanger	intercooler,platen,radiator
44	Heating Element	loop,spiral,helix
45	Hinge	pivot,axis,pin,hold down, jam,post,peg,dowel
46	Housing	main body, container, box, shell, holder, casing, crate, crust, chest, skin, armor, housing, skin, sheath, envelope, wrapping, cage, enclosure
47	Hydraulic Coupler	
48	Inclined plane	
49	Indicator Light	
50	Inductor	coil, transformer
51	Insert	grommet, eyelet, bushing
52	Junction	split, junction, yoke, fork
53	Key	Half-moon key, cotter key, shear key
54	Latch Release	catch, pawl, lock
55	Lens	
56	Level Gauge	

Code	Name	Synonyms
57	Lever	bar, peddle, rocker arm, lever arm
58	Link	Connection, pawl, rod, strut, brace, cross piece, girder
59	Magnet	lodestone, electromagnet
60	Material Filter	clarifier, separator
61	Needle	Spine, stylus
62	Nozzle	jet, injector, fuel injector
63	Nut-Bolt	Wing nut, lug nut, female screw
64	Piston	ram, plunger
65	Potentiometer	pot
66	Pressure Gauge	
67	Pulley	step pulley
68	Pump	
69	Punch	die, stamp
70	Reservoir	cup, container, receiver, vessel, holder
71	Rivet	Pop-Rivet
72	Rotational Coupler	union, compression coupling, clamping coupling
73	Rotor	Disk, impeller, hub, spindle, nave, indexer, index head
74	Scoop	ladle, dipper, skimmer, shovel, bucket, scoop, spoon, cup
75	Screw	Jackscrew, power screw, drive screw, lead screw, set screw, machine screw
76	Seal	gasket, o-ring
77	Signal Filter	
78	Sled	shoe, runner, skid
79	Socket	port, outlet, tray, dish, repository
80	Speed Gauge	
81	Spring	cantilever spring, coil spring, leaf spring, plate spring, torsion spring
82	Sprocket	
83	Stator	Stator Plate
84	Stop	snubber, travel limiter,bumper
85	Support	stand, foot, foundation, buttress, crutch, leg, seat, slab, scaffold, brace, bed, stanchion, reinforcement, base, pillar, column, joist, sole plate, anchor, pedestal, jig, fixture,table, underpinning, piling, bench, crutch, platen, saddle, prop, spine, backbone, undercarriage, caliper
86	Switch	knob, button, flip-flop, toggle
87	Thermal Insulator	
88	Thermostat	
89	Tube	pipe, cylinder, conduit, channel, duct, nipple, sleeve
90	Valve	louver, regulator, tap, flap valve, rotary valve
91	Wheel	rim, disk, tire

Appendix B: Material Code

Code	Material
00	Multiple Materials
01	Carbon/Low Alloy Steels
02	Tool Steels
03	Cast Irons
04	Stainless Steel
05	Beryllium
06	Copper Alloys
07	Nickel
08	Cobalt
09	Titanium Alloys
10	Aluminum
11	Refractory Metals
12	Lead
13	Zinc
14	Ceramics
15	Glass
16	Refractories
17	Thermoplastics
18	Thermosets
19	Adhesives
20	Elastomers
21	Particulate Composits
22	Fiber-Reinforce Composites
23	Laminar Composites
24	Wood
25	Plywood
26	Concrete
27	Asphalt

Appendix C: Manufacturing Code

Code	Manufacturing
00	Machining
01	Bending
02	Sand Casting
03	Shell Mold Casting
04	Full Mold Casting
05	Permanent Mold Casting
06	Die Castings
07	Plaster Mold Castings
08	Ceramic Mold Castings
09	Investment Castings
10	Centrifugal Mold Casting
11	Screw Machine Parts
12	Powder Metal Parts
13	Electroformed Parts
14	Cut Extrusions
15	Sections Tubing
16	Roll Formed Parts
17	Cold/Hot Rolled
18	Photo fabricated Parts
19	Open Die Forgings
20	Closed Die Forgings
21	Upset Forgings
22	Cold Headed Parts
23	Impact (cold) Extruded Parts
24	Drawn Parts
25	Stampings
26	Injection Molding
27	Coatings
28	Arc Welding
29	Oxyacetylene Welding
30	Electron Beam Welding
31	Resistance Welding
32	Friction Stir Welding
33	Brazing
34	Soldering
35	Adhesive Bonding
36	Threaded Fastening
37	Riveting/Metal Stitching

Appendix D: Function Code

Primary		Secondary		Tertiary	
#	Definition	#	Definition	#	Definition
1	Branch	1	Separate	1	Divide
				2	Extract
				3	Remove
		2	Distribute		
2	Channel	1	Import		
		2	Export		
		3	Transfer	1	Transport
				2	Transmit
		4	Guide	1	Translate
				2	Rotate
				3	Allow DOF
3	Connect	1	Couple	1	Join
				2	Link
		2	Mix		
4	Control Magnitude	1	Actuate		
		2	Regulate	1	Increase
				2	Decrease
		3	Change	1	Increment
				2	Decrement
				3	Shape
				4	Condition
		4	Stop	1	Prevent
				2	Inhibit
5	Convert	1	Convert		
6	Provision	1	Store	1	Contain
				2	Collect
		2	Supply		
7	Signal	1	Sense	1	Detect
				2	Measure
		2	Indicate	1	Track
				2	Display
		3	Process		
8	Support	1	Stabilize		
		2	Secure		
		3	Position		

Function	Code
Actuate	410
Allow DOF	243
Branch	100
Change	430
Channel	200
Collect	612
Condition	444
Connect	300
Contain	611
Control Magnitude	400
Convert	510
Couple	310
Decrease	422
Decrement	432
Detect	711
Display	722
Distribute	120
Divide	111
Export	220
Extract	112
Guide	240
Import	210
Increase	421
Increment	431
Indicate	720
Inhibit	412
Join	311
Link	312
Measure	712
Mix	320
Position	830
Prevent	441
Process	730
Provision	600
Regulate	420
Remove	113
Rotate	242
Secure	820
Sense	710
Separate	110
Shape	433
Signal	700
Stabilize	810
Stop	440
Store	610
Supply	620
Support	800
Track	721
Transfer	230
Translate	241
Transmit	232
Transport	231

Appendix E: Flow Code

Primary		Secondary		Tertiary	
#	Definition	##	Definition	#	Definition
1	Material	01	Human		
		02	Gas		
		03	Liquid		
		04	Solid	1	Object
				2	Participate
				3	Composite
		05	Plasma		
		06	Mixture	1	Gas - Gas
				2	Liquid - Liquid
				3	Solid - Solid
				4	Solid - Liquid
				5	Liquid - Gas
				6	Solid - Gas
				7	Solid - Liquid - Gas
				8	Colloidal
2	Signal	01	Status	1	Auditory
				2	Olfactory
				3	Tactile
				4	Taste
				5	Visual
		02	Control	1	Analog
				2	Discrete
3	Energy	01	Human E		
		02	Acoustic		
		03	Biological		
		04	Chemical		
		05	Electrical		
		06	Electromagnetic	1	Optical
				2	Solar
		07	Hydraulic		
		08	Magnetic		
		09	Mechanical	1	Rotational
				2	Translational
		10	Pneumatic		
		11	Radioactive/Nuclear		
		12	Thermal		

Flow	Code
Acoustic	3020
Analog	2021
Auditory	2011
Biological	3030
Chemical	3040
Colloidal	1068
Composite	1043
Control	2020
Discrete	2022
Electrical	3050
Electromagnetic	3060
Energy	3000
Gas	1020
Gas - Gas	1061
Human	1010
Human E	3010
Hydraulic	3070
Liquid	1030
Liquid - Gas	1065
Liquid - Liquid	1062
Magnetic	3080
Material	1000
Mechanical	3090
Mixture	1060
Object	1041
Olfactory	2012
Optical	3061
Participate	1042
Plasma	1050
Pneumatic	3100
Radioactive/Nuclear	3110
Rotational	3091
Signal	2000
Solar	3062
Solid	1040
Solid - Gas	1066
Solid - Liquid	1064
Solid - Liquid - Gas	1067
Solid - Solid	1063
Status	2010
Tactile	2013
Taste	2014
Thermal	3120
Translational	3092
Visual	2015

Appendix F: Repository Artifact Layout

System: delta sander

Artifact Name	shaft
Part Family	not specified
Part Number	27
Sub Artifact Of	drive train
Quantity	1
Description	not specified
Artifact Color	silver/gold/ copper
Component Naming	driveshaft

Artifact Photo

click on image for full size

Input Artifact	Input Flow	Subfunction	Output Flow	Output Artifact
gear	mechanical energy	transfer	mechanical energy	offset cover
spacer 3	solid material	secure	solid material	internal
spacer 4	solid material	secure	solid material	internal
Supporting Functions				
gear	solid material	secure	solid material	internal
bearing 1	solid material	guide	solid material	internal
bearing 2	solid material	guide	solid material	internal
bearing 3	solid material	guide	solid material	internal
spacer 1	solid material	secure	solid material	internal
spacer 2	solid material	secure	solid material	internal
offset cover	solid material	secure	solid material	internal