Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2005), *19*, 89–111. Printed in the USA.
Copyright © 2005 Cambridge University Press 0890-0604/05 $16.00
DOI: 10.1017/S0890060405050092

A model of function-based representations

MICHAEL VAN WIE, CARI R. BRYANT, MATT R. BOHM, DANIEL A. MCADAMS,
AND ROBERT B. STONE

Department of Interdisciplinary Design Engineering, University of Missouri–Rolla, 1870 Miner Circle,
Rolla, Missouri 65409, USA

(Received May 28, 2004; Accepted December 15, 2004)

Abstract

The need to model and to reason about design alternatives throughout the design process demands robust representation schemes of function, behavior, and structure. Function describes the physical effect imposed on an energy or material flow by a design entity without regard for the working principles or physical solutions used to accomplish this effect. Behaviors are the physical events associated with a physical artifact (or hypothesized concept) over time (or simulated time) as perceived by an observer. Structure, the most tangible concept, partitions an artifact into meaningful constituents such as features, Wirk elements, and interfaces in addition to the widely used assemblies and components. The focus of this work is on defining a model for function-based representations that can be used across various design methodologies and for a variety of design tasks throughout all stages of the design process. In particular, the mapping between function and structure is explored and, to a lesser extent, its impact on behavior is noted. Clearly, the issues of a function-based representation's composition and mappings directly impact certain computational synthesis methods that rely on (digitally) archived product design knowledge. Moreover, functions have already been related to not only form, but also information of user actions, performance parameters in the form of equations, and failure mode data. It is essential to understand the composition and mappings of functions and their relation to design activities because this information is part of the foundation for function-based methods, and consequently dictates the performance of those methods. Toward this end, the important findings of this work include a formalism for two aspects of function-based representations (composition and mappings), the supported design activities of the model for function-based representations, and examples of how computational design methods benefit from this formalism.

Keywords: Computational Design Methods; Design Knowledge Representation; Function-Based Synthesis

1. INTRODUCTION

Functions are used to model designs in a manner that captures information distinct from, but related to, the physical structure and behavior of a design solution. A function describes the physical effect imposed on an energy or material flow by a design entity without regard for the working principles or physical solutions used to accomplish this effect. The functional language used in this work is the Functional Basis developed by Stone and Wood (2000), reconciled by Hirtz et al. (2002), and tested for utility (Ahmed & Wallace, 2003; Kurfman et al., 2003). Although a detailed review of this language is not part of this work, the functional basis describes a function as a transformation between a set of input flows to a set of outputs. Here, flows represent energy, material, or signals that flow into and out of the physical artifact being represented functionally. In this language, the function for a motor can be represented as "convert" where the input flow is electrical energy and the output flow is mechanical rotation. Other languages, concepts, and definitions of functions are available that take different perspectives (Umeda & Tomiyama, 1997; Chittaro & Kumar, 1998; Deng, 2002). Hubka and Eder (2001) note that there are several versions of function definitions, usage, and their utility. The functional basis is a suitable functional language but an open problem, and the focus of this work is the specific nature of how functions are used in design, and particularly, how the mapping of functionality to physical solutions can and should occur. A key aspect in this regard is the representation of not only functions but also the other aspects of design that functions represent or to what these

Reprint requests to: Robert B. Stone, 102A Basic Engineering Building, Department of Interdisciplinary Design Engineering, University of Missouri–Rolla, 1870 Miner Circle, Rolla, MO 65409, USA. E-mail: rstone@umr.edu

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

are related. Wood and Greer (2001) identify the development of hybrid representations that integrate both functions and other design specifications as an opportunity for advancing function-based synthesis methods. The purpose of this work is to present a holistic model for function-based representations (FBRs) that will serve as a foundation for an underlying knowledge framework to support the development of artificial intelligence (AI) design techniques. Toward this end, the objectives of this work are the following:

1. define a formalism for two aspects of FBRs: composition and mappings;
2. examine how the model for FBRs supports design activities; and
3. present examples of how computational design methods can benefit from this formalism.

A common perception is the need for representations of function, behavior, and structure (Chandrasekaran & Josephson, 2000; Chang et al., 2000; Brown, 2003). Behavior, in the sense of physical events associated with a physical artifact (or hypothesized concept) over time (or simulated time) as perceived by an observer, is the least formalized of the three concepts, and addressing this topic is relegated to future work. Structure is the most tangible concept with various approaches to partitioning structure into meaningful constituents such as features (Brown, 2003), Wirk elements (Jensen, 2000), and interfaces (Ullman, 1997) in addition to the widely used assemblies and components. Depending on the structural definitions used, different schemes can be prescribed for allocating function to structure. One of the more formal approaches to dealing with these allocations or mappings is the development of a functional ontology (Kitamura & Mizoguchi, 1998). Regardless of the mapping scheme used, the primary significance of functionality is its independence with structure.

The independence of functionality from particular choices and details of structure is precisely how functions represent design concepts as an abstraction from the form or structure of these physical attributes. With this abstraction, perhaps the most significant, or at least most prevalent purpose of functional models, is to facilitate freedom in thinking and reasoning about design solutions without the greater restriction associated with the structure domain (Pahl & Wallace, 2002). Both psychological inertia and higher information content related to the structure domain lead to this restriction. Functional models in this sense are a lean medium for cognitive processing, but as Chandrasekaran and Josephson (2000) indicate, especially for computer applications, this information requires an explicit description of how functional knowledge is represented.

Two different types of abstractions exist for functional representations. First is the hierarchy defined in the functional language itself, such as the three levels of the functional basis (Stone & Wood, 2000). These levels offer a general-specific classification of function terms. For example, the function "channel" at the primary level subsumes a more refined set of functions such as "import" and "export" at the secondary level. The second type of abstraction is that of system hierarchy that is sometimes referenced as aggregation–decomposition or a part-of relation. This refers to the specification of a functional model where certain high-level functions are decomposed into lower level functions. The Functional Analysis System Technique (Miles, 1961; Otto & Wood, 2001) for developing functional models offers a hierarchical structure for describing the functionality at multiple levels of system detail. Similarly, the use of a black box model shows a high level structure while a functional model provides greater detail. Kitamura and Mizoguchi (2004) also distinguish between the above two types of abstraction.

Although benefits of abstraction are readily apparent, the independence of functionality from structure is complicated by how functions are allocated to physical solutions particularly in a reverse engineering or similar mode of analysis. Functional representations, physical representations, and physical devices themselves exhibit hierarchical organization and some basic questions are appropriate (Kueneke, 1991). What are the elements of this information set? Is this set of constructs comprehensive and irreducible? Given an established composition, how are functions mapped to physical artifact information? How does this mapping scheme relate to the design activity being conducted? More broadly, the concern is how the quality of functional models can be best realized in terms of the composition and mapping characteristics for various design activities as prescribed in design methods.

Prior work has addressed the manner in which functionality is allocated to the various physical system levels in various degrees (Kitamura & Mizoguchi, 1998; Jensen, 2000; Brown, 2003; Shi & Schmidt, 2003), but the authors are not aware of a comprehensive treatment of function allocation in a general context that accounts for the various types of design activities and conditions in which functional modeling is employed. To understand the relationship between functionality and the different physical scales on which functionality is manifested, one task of this research is to examine the issue of resolution as it applies to functional models.

From a modeling perspective, the quality of information contained in a model is important because the model must be predictive in ways that enable design (McAdams & Dym, 2004). In the specific case of functional modeling, the issue of prediction capability is perhaps less significant than the issue of facilitating divergent thinking, at least in a synthesis mode. Similarly, precision and accuracy in the analysis mode is important because, as prior work shows, functional models can be used in a reverse engineering approach to archive design information (Bohm, Stone, & Szykman, 2003) and to use such archived knowledge for design synthesis (Gietka et al., 2002; Strawbridge et al., 2002). Clearly, the issues of composition and mappings propagate beyond

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

reverse engineering to design synthesis when considering that certain computational synthesis methods rely on data archived from these reverse engineering analyses. Moreover, functions have already been related to not only form, but also information of user actions (Janhager, 2003), performance parameters in the form of equations (Bryant et al., 2001; Yekula et al., 2003), and failure mode data (Tumer & Stone, 2003). It is essential to understand the composition and mappings of functions and their relation to design activities because this information is part of the foundation for function-based methods, and consequently dictates the performance of those methods.

2. DEVELOPING COMPOSITIONS AND MAPPINGS OF FBRs

The goal of developing the FBR model is to establish the informational items relevant to FBR and the relationships among them. As for model composition, we seek a comprehensive and irreducible set that make up the hierarchy of terms in both the function and structure domains. The mappings among these elements is then developed to show how these items can be associated, stored computationally, and used for automated design purposes. Much of this task is deciphering how considerable efforts from prior research can be joined to form a model of FBRs.

2.1. Composition of FBRs

2.1.1. Function

A primary issue for the composition of elements in the function domain is the concept of resolution. Low resolution implies coarse or abstract partitioning of functionality where this resolution can be interpreted in terms of both the functional language and the functional model. For the resolution within the model, a comparison between a black box model and a functional model in Figure 1 illustrates this point with a device designed to tag trees. Here, the black box is a single function that encapsulates the functionality of the entire functional model into what is taken to be the overall device function. The process of decomposing functions into lower level functions can continue until further decomposition no longer benefits the design activity. This open-ended interpretation places no restriction on the upper or lower bounds of resolution where the terms of the Functional Basis can be used at any resolution level. Selection of best resolution depends of this design activity, and is examined in Section 2.2.

In addition to functions, the term module is a distinct entity that simply describes a set of functions. Although a single function such as "convert" is indicated as a module based on the modular heuristics by Stone et al. (2000), a module is more commonly a group of two or more functions.

The concepts of modules and functions deal with customer-driven functions, that is, functions that are directly related to customer needs. The function "store paper" for a printer is an example. Supporting functions recently investigated by Bohm and Stone (2004) and referred to as metafunctions by Kitamura and Mizoguchi (1999), are functions of the physical artifact acting on itself rather than external flows. Specifically, supporting functionality is defined as "functions that describe manufacturing, assembly, and support features present in the embodied form of a product" (Bohm & Stone, 2004). For example, the function "secure lid" for a CD holder describes the action of the compliant mechanism detent of the base as it acts on the lid by securing it. This distinction between customer-driven and supporting functionality is significant, and suggests that supporting functions are another basic entity in the FBR model. The hierarchy of a functional model is the only factor imposing variations in resolution.

Beyond the resolution of the functional models is the issue of the functional vocabulary. Three levels of abstraction are specified in the functional basis for both the functions and flows (Stone & Wood, 2000). No constraints are placed on how the different levels are used or mixed within a functional model other than certain restrictions due to the compatibility between functions and flows. Additionally, the customer needs and type of design problem influence the choice of resolution level. Despite this flexibility, the functional basis offers a comprehensive set of customer-driven functions that offer consistency of function terminology.

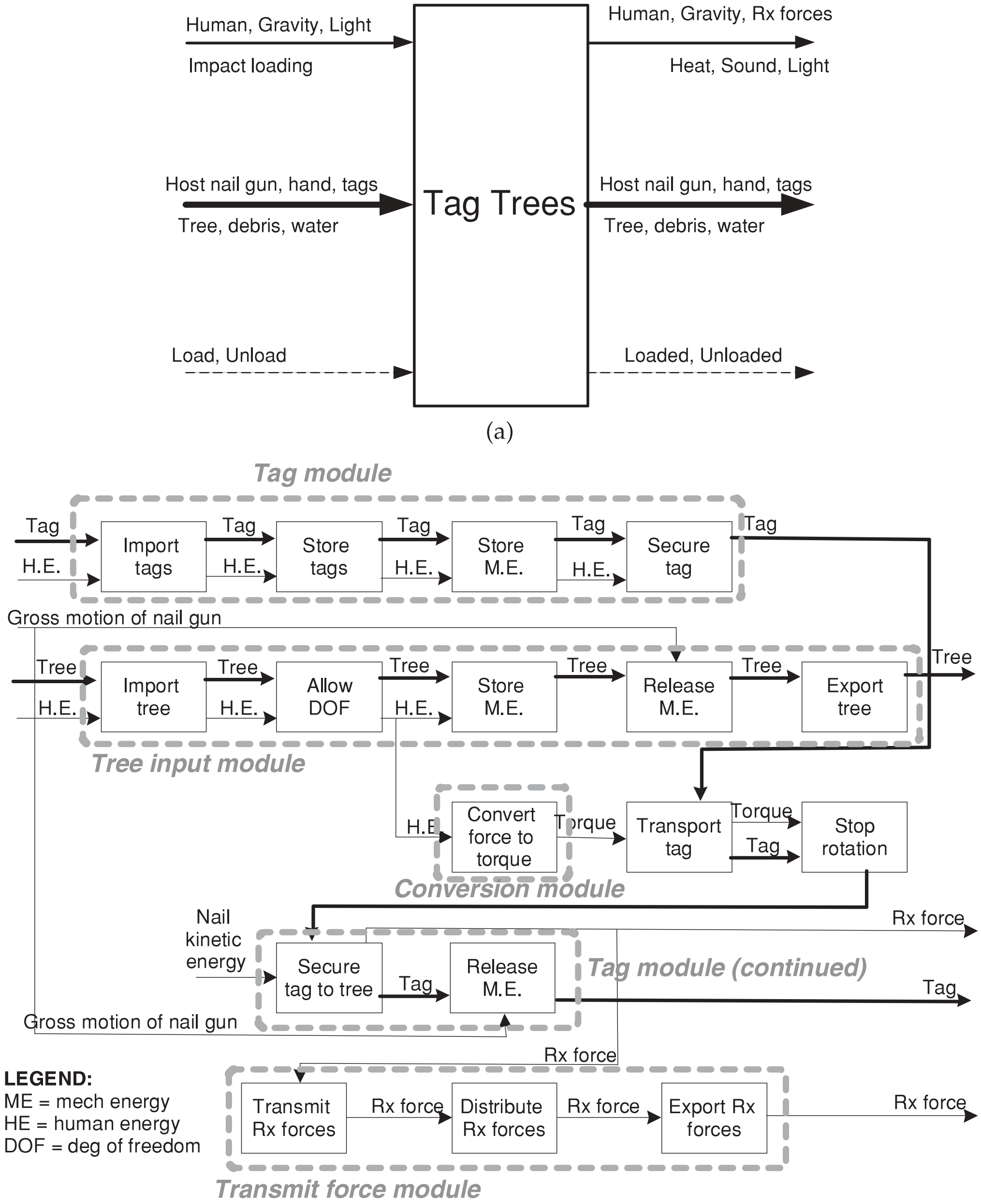
2.1.2. Form

Resolution with respect to physical partitions is more complicated. Here resolution is in terms of the degree of partitioning of a physical artifact in which the partitioned elements can be associated with one or more functions. The central issue of physical resolution is the manner in which a product can be divided along meaningful lines at various degrees of granularity. This is a description of only how a physical item is divided and is separate from the issue of how functions are allocated or associated with the partitioned elements of the product. It may be the case that while relatively high physical resolution is used, the allocation of function is kept at a low level of resolution. The issue of allocation is addressed in Section 2.2.

It is important to note that because functionality of a product is sometimes specified before the physical artifact exists or is even conceptualized as in the case of original design, the issue of resolution and therefore partitioning becomes difficult. In the case of original design it is irrational to define resolution in terms of the degree of physical partitioning because nothing yet exists to partition. For purposes of defining a set of physical elements for the FBR, this problem is avoided by treating resolution strictly in the context of a descriptive application such as reverse engineering when the physical device already exists and the functional model is constructed to represent this artifact.

As an upper bound of resolution, the complete product or system as a whole seems the obvious choice. Exploiting

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press



(a)

(b)

Fig. 1. Different functional model resolutions.

this idea, the next issue becomes the definition of the complete system boundary. Generally, an individual product can be bounded as a system by its physical boundary, although certain exceptions apply. Product family design is one exception where a set of products may make up a system boundary to illustrate the overall functional behavior of the portfolio rather than individual products. Additionally, the functional interactions among the members of the product family may be described if the complete functional model contains the product family set. A second exception is in the context of sustainable design or design for the environment. Given the importance of life-cycle effects in sustainable design, it is desirable to introduce systems and processes into the system model that would otherwise normally be outside the system boundary. In this case, the upper bound may include a wide spectrum of entities ranging from design teams, manufacturing units, the customer, product take-back facilities, recycling and refurbishing units, and so forth. From these two exceptions we define a physical element named metaproduct systems.

Skipping to the opposite end of the resolution spectrum at the subcomponent level, features (Brown, 2003) or Wirk elements (Jensen, 2000) are suitable concepts as they account for functionality at the smallest meaningful levels with the

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

possible exception of mechanical devices at the micro- or nanoscale. For example, the tip of a screwdriver clearly exhibits a function distinct from other portions of the tool, just as the edge of a knife differs in functionality from the remainder of the blade. Defining elements that can separate these physical aspects is the task. Wirk elements are selected here as the low-end physical element because they are clearly defined as geometrical properties of the designed artifact. However, to capture the properties of features, as explained by Brown (2003), that are not readily allocated to a region of an artifact, these properties are addressed outside the scope of the physical elements and are discussed in the next section.

Between the upper bound of metaproduct and the lower bound of Wirk elements, the only intermediate levels that seem meaningful and distinct are assemblies and components. Each involves a fundamental discontinuity in the makeup of physical products. These levels are described in Table 1.

Selection of the levels in Table 1 is based on the requirement that each level offer a clearly distinct degree of resolution. The design data model discussed recently by Aurisicchio et al. (2003) is based on matrices developed by Blessing (1994), which include the product, assembly, and component levels. The proposed levels in Table 1 differ by treating the products as an assembly. In addition, the Wirk elements are included to address the subcomponent level. At the low-resolution end, the metaproduct level is very high level, and offers an overall perspective. The assembly level has perhaps the widest breadth of the four levels by including all assemblies whether small modules or major assemblies containing multiple subassemblies. This broadly applicable level serves as a basis for a hierarchical approach to function-based product descriptions. The Vehicle Platform Partitioning System used by General Motors is one industry example of such a hierarchical strategy in practice today (Bohm, Stock, et al., 2003). Compared with the assembly level that is applicable down to a component, the component level is narrowly defined to capture single piece parts. For many original equipment manufacturer products such as motors, bearings, hydraulic cylinders, and so forth, these items overall would be captured as an assembly rather than a component. This approach differs from what is perceived as conventional practice, which is to treat such items as "components." At the finest level of resolution, Wirk elements discussed by Jensen (2000) are the foundation for the smallest partitioning that seems reasonable. These elements allow description of functionality within components. The use of these Wirk elements enables the description of functionality within components in terms of geometrical properties. This particular level is rich in supporting functionality.

2.1.3. *FBR model elements*

In addition to functional elements and physical elements, a broader perspective is taken to include three fundamental categories of product design information in the FBR model. These are the design, performance, and noise spaces that are respectively described using design variables (D), noise variables (N), and performance metrics (P), where each of these is comprehensive and measurable as described in Otto and Wood (2001). Briefly, the performance metrics are indicators of design objectives, design variables represent those items of a design for which a designer has a choice, and noise variables represent items that are beyond the control of the designer. Each of these reflects, to various degrees, the behavior of a design. Customer needs, for example, are accounted for by performance metrics that reflect the customer need objective. The general format is $P = f(D, N)$, where D includes the specification of function and structure as these and other examples are shown in Table 2. The scope of relations that are available in this framework is comparable to those achieved by the "object in the system" approach taken by Stetter et al. (2001).

2.2. Mappings of FBRs

Figure 2 illustrates the combined concepts of composition and mappings. Here the issue of composition lies in the type of primitives used across both the function and structure domains in their respective hierarchies and individual levels of abstraction. Mappings refer to the associations

Table 1. *Resolution levels and partitioned elements*

Resolution Level	Partitioned Element
Metaproduct	Product family Life-cycle system
Assembly	Single product (multiple components) Physical module
Component	Single piece part
Wirk element	Point Line, curve Surface Volume

Table 2. *Design and noise variables and performance metrics*

Design Variables	Noise Variables	Performance Metrics
Functions	Environmental conditions	Failures
Physical elements	User activities	Number of functions, parts, etc.
Dimensions	Evolutionary triggers	Flexibility
Material specification	Manufacturing defects	Cost
Manufacturing process		Component sourcing
Potential changes		Precision

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

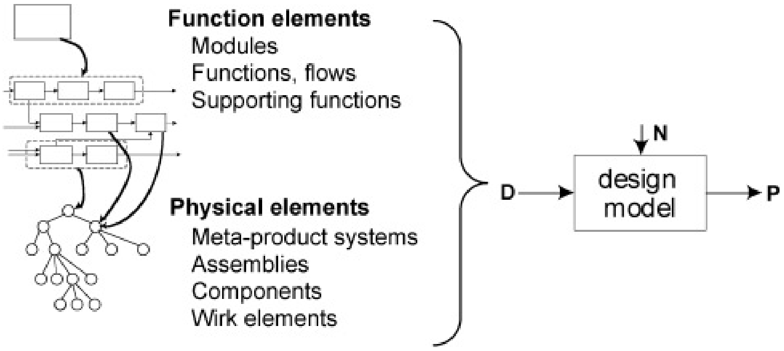


Fig. 2. Elements of function and form as design variables.

among these elements and to relevant factors that constrain the design. These mappings are considered in the next section.

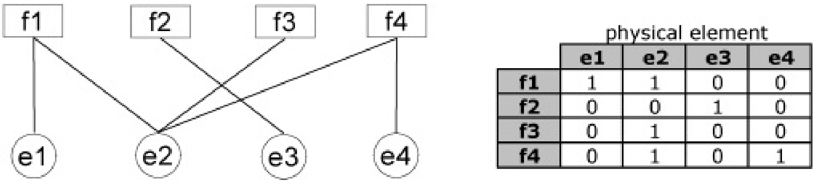
Function allocation refers to the mapping or association of function elements with physical elements. Function allocation is independent of resolution, but the scheme used to associate functionality with the physical elements influences the effectiveness of functional modeling. In this context, the effectiveness of the function allocation scheme is dependent on the type of design task or activity. The main purpose in examining function allocation is that previous work does little to address how allocation should be performed to maximize the effectiveness of functional modeling. It seems that the allocation strategy is left to the discretion of the designer. Given the variation in the level of detail that is possible with different approaches toward allocation, it is reasonable to assume that this performance can be improved by certain use of function allocation. A matrix approach for storing function allocation information is presented, followed by a discussion of allocation with respect to two prominent design activities: analysis and synthesis.

2.2.1. Adjacency matrix representation for FBR mappings

Regardless of the scheme used for function allocation, the basic structure used to store these relations is a graph where an adjacency matrix can then be formed, as shown in Figure 3. Such matrices offer a format that is suitable for computation. Examples of functions to physical elements at different levels are given in Figure 4. In each of these cases, these matrices can be expanded to include large quantities of archived data from analyses of existing devices and used to populate a morphological matrix of candidate physical solutions based on an initial set of functions (Strawbridge et al., 2002). This prior work denotes these matrices containing archived information as χ matrices, and this notation is adopted here. In general, χ matrices can be formed between any two informational items. The correlation of function and failure modes (Tumer & Stone, 2003) is one case of an association between design variables and an indicator of performance metrics. This approach provides a very broad capability for making associations. Some examples of potential χ matrices include mappings from functions to assemblies, supporting functions to Wirk elements, components to their nominal design flexibility, components to components [design structure matrix (DSM)], modules to functions, and so forth. Allocating functions with physical solutions at these different resolution levels (assemblies, components, etc.), allow separate physical solution searches for each level. For example, physical solutions at the Wirk element level could be searched to identify candidate interfaces and, in the process, screen out solutions at other resolution levels. In addition, this archived data at different resolution levels can be used to support studies that examine the relation between functions and their common implementations in terms of physical resolution.

2.2.2. Function allocation

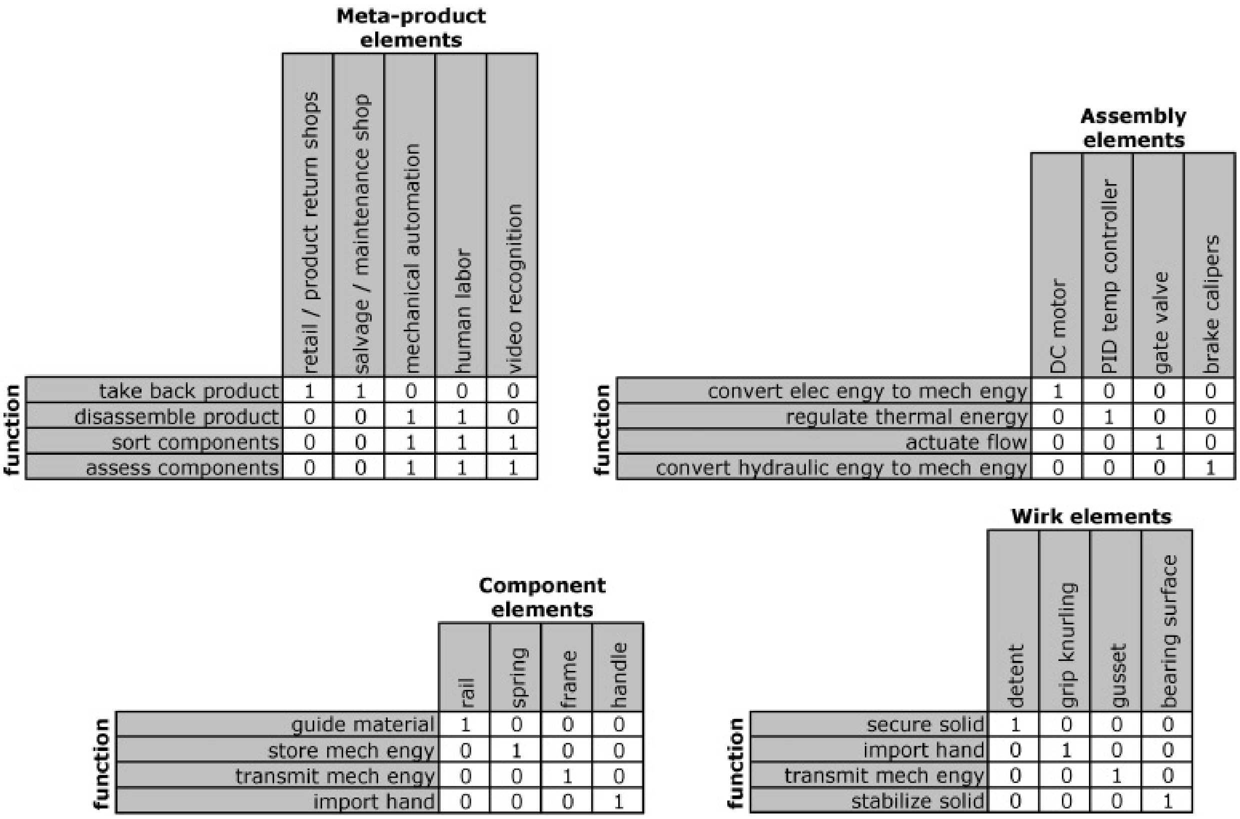
Function allocation deals with the issue of resolution used to associate a functional representation with a physical solu-



	physical element			
	e1	e2	e3	e4
f1	1	1	0	0
f2	0	0	1	0
f3	0	1	0	0
f4	0	1	0	1

Fig. 3. Function-based associations.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press



function	Meta-product elements: retail / product return shops	salvage / maintenance shop	mechanical automation	human labor	video recognition
take back product	1	1	0	0	0
disassemble product	0	0	1	1	0
sort components	0	0	1	1	1
assess components	0	0	1	1	1

function	Assembly elements: DC motor	PID temp controller	gate valve	brake calipers
convert elec engy to mech engy	1	0	0	0
regulate thermal energy	0	1	0	0
actuate flow	0	0	1	0
convert hydraulic engy to mech engy	0	0	0	1

function	Component elements: rail	spring	frame	handle
guide material	1	0	0	0
store mech engy	0	1	0	0
transmit mech engy	0	0	1	0
import hand	0	0	0	1

function	Wirk elements: detent	grip knurling	gusset	bearing surface
secure solid	1	0	0	0
import hand	0	1	0	0
transmit mech engy	0	0	1	0
stabilize solid	0	0	0	1

Fig. 4. Examples of function associations in χ matrices at four different resolutions.

tion. Resolution in this sense refers to two items: the level of detail in the functional model itself, and the physical resolution of the artifact being represented. A high-resolution functional model would have a large number of functions to describe the system of interest. Similarly, a high-resolution description of a physical system would take a magnification approach to account for the functionality of physical features at small scales.

One purpose of functional modeling during synthesis is to enhance the solution search process by providing the designer a means for thinking about solutions independent of physical form. Outside the scope of computational approaches, manual use of functions works well when the following two, somewhat conflicting, constraints are met. First, the functional model must be sufficiently abstract that normal mental cues that funnel and restrict the designer to a previously used experience set are no longer active. This directly facilitates the kind of "outside the box" thinking that is desirable. The second constraint requires sufficient descriptive power from the functional model so the solution presents enough detail to be worthwhile for advancing the design.

These two constraints are representative of two extreme case approaches that can be employed during function-based synthesis. A small common DC electric motor is a good example that illustrates both of these approaches. In the case of a motor, the overall functionality may be described as simply "convert electrical energy to mechanical energy." This allocates a single overall function to the complete motor assembly that consists of lower level assemblies (housing, armature, etc.) and components (magnets, bushings, etc.). At the other end of the spectrum, each function for all levels of detail down to the Wirk element level could be described. In addition, this entire set could be attributed collectively to the overall motor assembly. This subsumes all functionality contained within the motor. Particularly for a motor, the approach using an overall single function of "convert" is probably a more effective use of function allocation because these motors are often selected and purchased rather than designed in-house. However, a functional model with greater detail may be appropriate for situations when the motor is custom designed. These two cases are referenced simply with the level of detail in their approach.

It may seem counterproductive to ever need great levels of functional detail (e.g., down to and including functionality at Wirk element level) during design synthesis. However, several factors, which are examined below in Table 3, could warrant such detail. A designer may choose to utilize a high level or very detailed approach or an intermedi-

Table 3. *Design issues and effects on function allocation*

Design Issue	Effect on Function Allocation
Scale of system	Large systems require a large number of functions if allocated at either fine resolution or in a manner that subsumes lower partition levels.
Risk of functionality	If a needed function involves high risk, then greater functional detail may be necessary to provide redundancy or counteractions to mitigate these risks.
Confidence in function selection	Solutions established with high confidence do not require great detail except for purposes of interfacing with other functions.
Redesign	Portions of a design solution that change or are likely to change require greater detail than those areas that remain the same.
Energy and material flows	Some functionalities may require greater detail based on the energy and material flows for a given function. Here, energy and material flows refer to the inputs and outputs of a given function (e.g., electrical energy flows into a function describing a motor and mechanical rotational energy flows out).
Interfaces	Large numbers of interfaces require greater detail in function allocation.
Stage of design	Early design stages may require less functional detail than later stages.

ate level of detail in how functions are allocated. Clearly, there are distinctions among these approaches, yet little guidance is presented in the literature for the most effective approach for a particular design task. Although the best approach is dependent on the particular circumstances of the designer and the design problem, we address this issue by examining how different factors affect the allocation of functions during functional modeling. This provides a hypothesized effect for the level of functional allocation with respect to certain design issues that are commonly encountered.

As an overall observation, the best level of functional detail is ultimately dependent on the potential benefits that could be gained from the given detail being considered. Functions at any level chosen should aid the designer to think, reason, reduce uncertainty, and explore concepts. It appears that the choice of function allocation in terms of detail is highly variable and dependent on the particular product, the type of design being performed, and project constraints such as time available to the designer.

The particular issue involving the stage of design raises some points that are not typically addressed when discussing functional modeling. Normally, function-based synthesis is utilized in the early stages of design but it is plausible that the benefits of reasoning with functions can be realized by using a function-based approach throughout the design process. One consequence of using functional modeling during the later stages of design is that such an approach accounts for increasing levels of detail in a design in a manner that still allows a degree of divergent thinking afforded by the abstraction of functions. For example, the design of a computer mouse could take advantage of functional modeling by identifying functions of hand input surfaces. These functions, in turn, may spawn new ideas for shaping the exterior body and interfaces. Although relatively high level abstractions and little detail in function allocation may be necessary during the up-front functional modeling performed during the early stages of a computer mouse, the embodiment and detailed design stages could potentially benefit from employing functional modeling at a fairly detailed level during these later stages. Of course, the functional modeling used in later stages may be limited in scope and restricted to particular portions of the design because functional modeling need not demand a complete and exhaustive model at these later design stages regardless of the function allocation detail.

2.3. FBR model development summary

The FBR model is a collection of ideas explained in the general context of design, noise, and performance metrics. These ideas are individually established largely in prior work but interpreted here in a manner that define a full scale of resolution in both the function and physical domains, are explained in a general context of design, noise, and performance metrics, and offer a matrix approach for storing function-based and other relevant design information. The FBR model includes a composition of both function and physical elements that are comprehensive across the resolution scale of interest and irreducible in the sense that the distinguishing characteristics of the terms specified exhibit negligible redundancy. Because these terms are presented in the context of the broader set of design, noise, and performance parameters, many relations become readily identifiable. These associations in matrix form are a consistent and computable format for capturing not only function-based information as it relates to structure and other parameters, but also for capturing information between any two meaningful variables in the FBR model. Discussions are given for the issue of function allocation in terms of design synthesis; yet, it is clear that other design tasks are relevant. To understand the role of FBRs throughout design, the following section examines the FBR model with respect to a recently developed ontology of generic engineering design activities (Sim & Duffy, 2003).

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

3. FBR MODEL IN SUPPORT OF DESIGN ACTIVITIES

Perhaps the best measure of utility for the FBR model is evidenced by how the FBR model supports design activities. Despite the above discussion, which treats only the broad activities of design synthesis, a much larger set of activities are relevant. To assess the FBR model with respect to a comprehensive set of activities, the proposed ontology by Sim and Duffy (2003) is used. Tables 4, 5, and 6 present the assessment of the FBR model where the first two columns are used from Sim and Duffy (2003) to clearly describe the design activity. In the second column, the knowledge change refers to the impact that the design activity imposes on the input knowledge compared to the output knowledge resulting from the activity. The third column is an estimate of how the FBR model supports the knowledge change for a given activity. For some activities, multiple knowledge changes are listed, and when appropriate, the third column treats these distinctly. In some of these cases, such as the "decomposing" activity, the FBR support description applies to the activity overall.

Table 4. *Design definition activities, knowledge changes, and how the FBR model supports them*

Design Definition Activities	Knowledge Change as Result of Activity	How FBR Model Supports Knowledge Change of Given Activity
Abstracting	Knowledge abstractions that depict useful relationships of the evolving design concepts	Abstractions are provided within the functional basis, in hierarchies of functions, and four levels of structural resolution.
Associating	Knowledge of novel ways in which ideas/concepts are linked	FBR supports a large set of potential associations between design variables, noise variables, and performance metrics, the combination of which account for what the designer can control, what cannot be controlled but which affects noise and performance.
Composing	Knowledge of function to means structure	χ matrices provide explicit information about the relation between function and form at multiple abstraction levels (e.g., Fig. 4).
Decomposing	Knowledge of part/subpart or system/subsystem hierarchy of a product Knowledge of overall function in terms of subfunction mappings, function component mapping	FBR supports the decomposition of existing systems at both function and form levels where one example is the archival of historical data for existing instances of system hierarchy through reverse engineering of products.
Defining	Specifications that satisfy design requirements	The FBR model supports specification definition through energy, material, and signal flows in combination with performance parameters. Specifically, a parameter can be first identified from the functional model flows and subsequently given a max, min, or target value in order to form a specification.
Generating	A mapping of function/subfunction hierarchy to solution principle/component for a given design problem	FBR supports function to solution principle mappings through the χ matrices that provide explicit information about the relation between function and form at multiple abstraction levels (e.g., Fig. 4).
	A mapping from function to working principles to structure	Individual mappings for each are supported: working principles are supported by matrices that define both function and form topology (working principles themselves are supported insofar as form elements reflect and represent working principles).
	A mapping from function to design parameters to structural forms	Design parameters are explicitly supported, as are structural forms (e.g., components).
	A mapping from function to behavior to structural forms to explain the causal relationships among them	Behavior, although not explicitly addressed, is captured to some degree implicitly through the performance parameters.
Standardizing	Knowledge of nature of the standardization and the basis for it (i.e., standardization based on similarity)	Classes of components are supported in order to distinguish, e.g., between custom and OEM components
Structuring/configuring	Knowledge grouping system to system interactions based on system similarity or functional dependency	The topology of both function and physical elements is captured as well as the association between physical elements and functions.
Synthesizing	A final design that satisfies all the design requirements expressed in some design documents detailing all aspects of the design for manufacture	FBR support progressions from function to form and progressions at different abstraction levels within each of these domains.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

Table 5. *Design evaluation activities, knowledge changes, and how the FBR model supports them*

Design Evaluation Activities	Knowledge Change as Result of Activity	How FBR Model Supports Knowledge Change of Given Activity
Modeling	A more detailed description of the design, depending on the type of analysis	Detailed descriptions are supported insofar as elements as low as Wirk elements and supporting functions are represented.
Analyzing	Knowledge that a particular design in terms of form displays behavior(s) that meets design criteria	Associations with performance metrics directly account for design criteria.
Evaluating	Knowledge of the assessment and rationale of the quality of design solution(s)	The rational for a design is not captured, although a method using the FBR can create this rationale with a log of decisions.
Decision making	From a potential set of design solutions, the knowledge of the rationale for the choice of best design(s)	The FBR model itself does not contain decision operations, although knowledge for making various decisions is embedded in the FBR model.
Simulating	Conditional predictions of the expected behavior of the artifact	FBR supports simulation based on archived performance relations of prior instances of actual performance.
Testing/experimenting	Knowledge of the design's compliance with specification	FBR supports virtual testing based on archived performance relations of prior instances of actual performance.

Based on this assessment, it is apparent that both the design definition and evaluation categories in Tables 4 and 5 are well supported, while the design management activities are much less relevant to the FBR model. The support for the substantial number of design activities implies that the basic informational items within the FBR model are comprehensive for design over a broad range of activities. This does not suggest that the FBR model serves as a design method because no logic, operations, or other processing actions are part of the model. However, given the relevance of the model to several design activities, it is reasonable to claim that the FBR model is a useful foundation for the development of design methods that could potentially span a large number of design activities. Specifically, the FBR model in terms of its composition indicates the specific information that may be useful for a function-based design method, while the mappings are suggestive of the type of relations that may be useful. The FBR model also indicates that a matrix-based approach is a useful format for capturing these relations.

Table 6. *Design management activities, knowledge changes, and how the FBR model supports them*

Design Management Activities	Knowledge Change as Result of Activity	How FBR Model Supports Knowledge Change of Given Activity
Constraining	Knowledge of the feasible design solution space	Archived solutions can be constrained based on several potential preferences.
Exploring	Knowledge of potential designs as candidates for solutions	FBR supports exploration through extensive representation of function, form, noise, and performance parameters.
	Knowledge of design strategies to choose from	Not applicable for the FBR
Identifying	Knowledge of important issues that influence the design, leading to a better design strategy or design	FBR directly supports the use of noise variables that provide clues regarding the predesign context.
Information gathering	Additional knowledge of the design or the process	Not applicable for the FBR
Planning	A specific sequence of tasks for a given design project that minimizes design iteration	Not applicable for the FBR
Prioritizing	Knowledge of specific ordering of goals and subgoals for a design task/project	Not applicable for the FBR
Resolving	Specific knowledge of conflict resolution relating to the design or process	Not applicable for the FBR
Scheduling	A specific sequence of tasks for a given design project that minimizes time to market	Not applicable for the FBR
Selecting	Knowledge of the optimal choice for a specific design or process	Perhaps not *optimal*, but the FBR supports archived information that is historically *workable*.
Searching	Knowledge that contributes to the definition of the design	FBR supports a large amount of archived data for searching.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

4. FBR INSPIRED METHODS AND APPLICATIONS

To further explore the FBR model potential, this section examines specific methods that can benefit from the FBR model. The point is to show examples of how the concepts from the FBR model (e.g., resolution of function and physical artifacts; design, noise, and performance parameters; matrix-based mappings of function and form information) can inspire and support function-based design methods and applications. First, two reverse engineering applications are examined where both use the FBR model for knowledge archival to support analysis type activities, while the second section examines specific uses of the FBR model to support synthesis operations in AI applications.

4.1. Reverse engineering knowledge archival for AI: Computational product analysis and description

4.1.1. Product repositories: Improvements and new directions

The main purpose of product repositories is to capture and store design information in a format that facilitates analyses or empirical studies of historical product data, promotes consistency of design information, and allows past design data to be reused for design synthesis activities. Given this purpose, the FBR model indicates that design information is captured for multiple levels of product abstraction. Functional models are separated in terms of function, modules, and supporting functionality in addition to the three abstraction levels specified in the functional basis language. Similarly, physical data can be recorded at the metaproduct, assembly, component, and Wirk element levels to clearly distinguish the physical attribute under consideration. For example, the combined information of supporting functionality and Wirk element data allows for capture of functionality for physical interfaces.

Regardless of the resolution level chosen, the consistency of data collected is improved by maintaining resolution consistency. This offers an improvement over current methods in which both the resolution and allocation of function information may fluctuate during reverse engineering activities due to the perceived issues of importance from the person performing the task. In addition to consistency is the issue of completeness. By capturing product data systematically at each individual level, a more comprehensive view of the product is achieved. Currently, several items regarding function and form are captured in the University of Missouri–Rolla online design repository as shown in Figure 5 (Bohm, Stone, & Szykman, 2003; Bohm & Stone, 2004).

More generally, the FBR model indicates product data relating function and form data to performance information is archived. In addition to the current interest in capturing failure data (Tumer & Stone, 2003), potentially many additional performance-related factors could be stored. Similarly, information regarding noise variables could be correlated with both function and form. As an example, designs are driven by customer needs, although these customer needs are not entirely bounded or necessarily reflected directly by customer. Precipitating events that make up the predesign context are to a great extent out of the control of the designer and are thus considered noise variables. For the case of a redesign, certain events would be considered evolution triggers that drive the design in particular directions. Examples of these include competitor actions, statutory changes, cyclic (e.g., annual) changes, and so forth (Van Wie et al., 2004). A reasonable hypothesis is that elements of this predesign context are correlated with elements of function and form or changes in function and form from previous designs. Subsequently, this knowledge could be used during synthesis activities to generate candidate solutions in parallel with the conventional approach of using customer needs, as shown in Figure 6. It is important to note that this is a supplement to customer needs rather than a replacement.

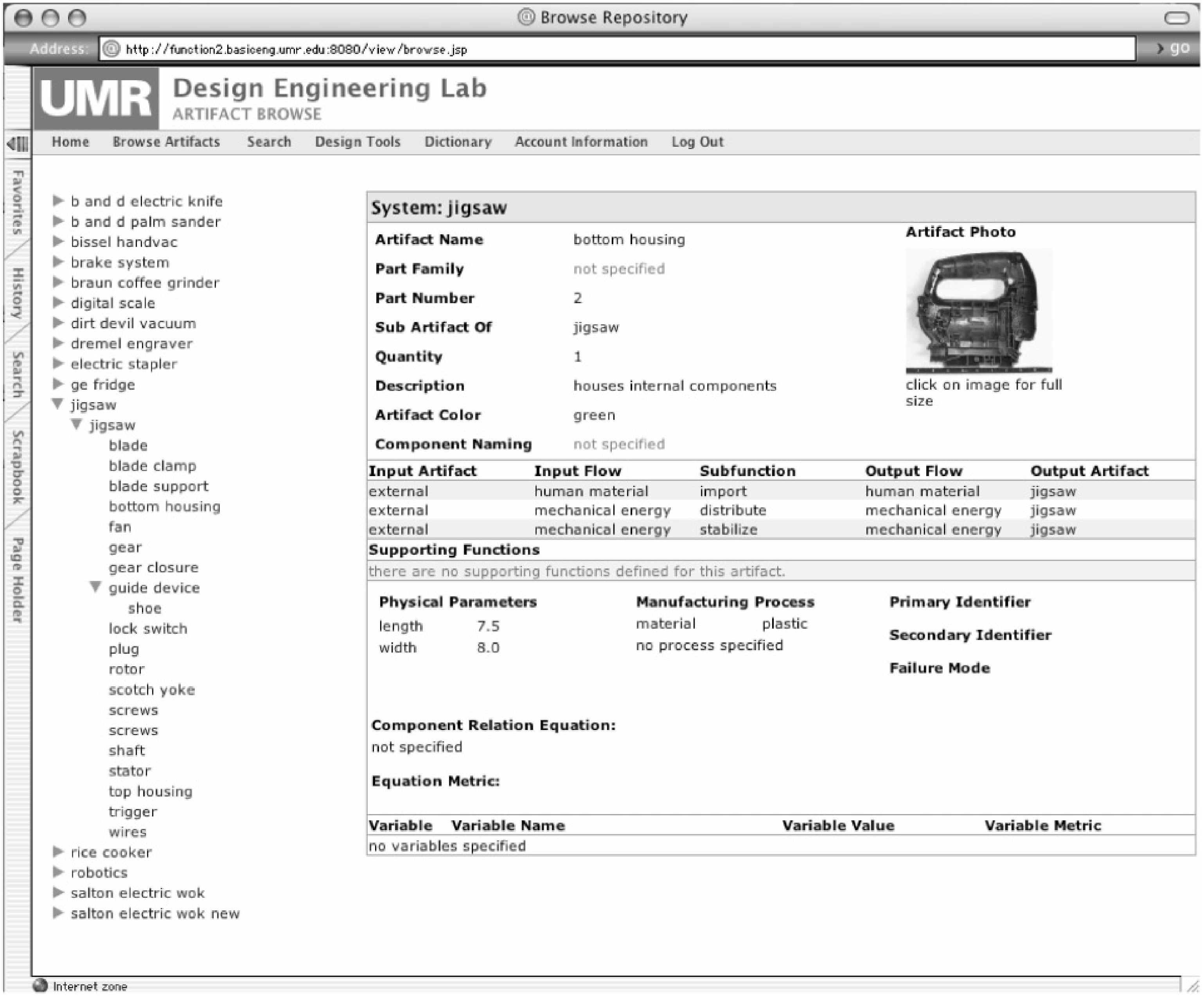
4.1.2. Patent description, patent claims data structure, and automated patent claim discovery

Patents are a particular type of product representation where descriptions of both function and form at different levels of abstraction are used to describe an invention and document the boundaries of legal protection regarding an invention. Although patent content is controlled as a legal matter, only the basic elements of the patent description and patent claims are addressed here. As a guideline, clarity and comprehensiveness are essential to a good claim. Normally a patent will include several claims that are supported by the description of the device. In this manner, the claims are an interpretation of the product description written with specific legal wording and phrasing.

Because of the need for clear and comprehensive descriptions of both function and form, patent claims are a potential application for automation of portions of the patent claim process. Although thoroughly addressing the feasibility of such a goal is well beyond the scope of this work, the following presents an overview of how the concepts of the FBR model could be used toward the creation of such an application.

The core of a patent is the portion called simply the “description” of the invention that is given in both textual and graphical format. The purpose of this description is to provide roughly the same information regarding the function and form of the device as equivalent to that which would be available if the device were physically in possession of the patent reader. This allows one to understand very well both the physical form of the invention in terms of geometry and material, and the manner in which the physical constituents of the invention act on energy and material and interact with each other.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press



http://function2.basiceng.umr.edu:8080/view/browse.jsp

- ▶ b and d electric knife
- ▶ b and d palm sander
- ▶ bissel handvac
- ▶ brake system
- ▶ braun coffee grinder
- ▶ digital scale
- ▶ dirt devil vacuum
- ▶ dremel engraver
- ▶ electric stapler
- ▶ ge fridge
- ▼ jigsaw
  - ▼ jigsaw
    - blade
    - blade clamp
    - blade support
    - bottom housing
    - fan
    - gear
    - gear closure
    - ▼ guide device
      - shoe
    - lock switch
    - plug
    - rotor
    - scotch yoke
    - screws
    - screws
    - shaft
    - stator
    - top housing
    - trigger
    - wires
- ▶ rice cooker
- ▶ robotics
- ▶ salton electric wok
- ▶ salton electric wok new

System: jigsaw

Artifact Name	bottom housing
Part Family	not specified
Part Number	2
Sub Artifact Of	jigsaw
Quantity	1
Description	houses internal components
Artifact Color	green
Component Naming	not specified

Artifact Photo

click on image for full size

Input Artifact	Input Flow	Subfunction	Output Flow	Output Artifact
external	human material	import	human material	jigsaw
external	mechanical energy	distribute	mechanical energy	jigsaw
external	mechanical energy	stabilize	mechanical energy	jigsaw

Supporting Functions

there are no supporting functions defined for this artifact.

Physical Parameters

length	7.5
width	8.0

Manufacturing Process

material	plastic
no process specified	

Primary Identifier

Secondary Identifier

Failure Mode

Component Relation Equation:

not specified

Equation Metric:

Variable	Variable Name	Variable Value	Variable Metric
no variables specified			

Fig. 5. The jigsaw product information in the design repository.

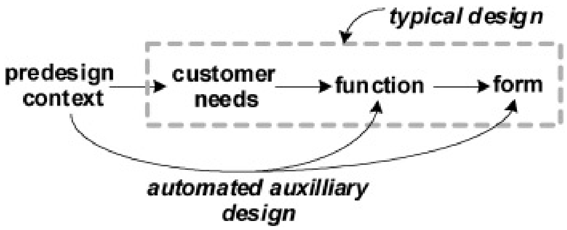


Fig. 6. Bringing design upstream: the use of archived predesign contexts for synthesis.

The results from this research can be applied to this description. The actions of the invention noted above describe the invention's functionality and supporting functionality, respectively. The physical form is described by the use of assembly, component, and Wirk element references to physical artifacts at these levels. Ultimately, this is an expanded and more comprehensive version of the product repository that currently exists (Bohm, Stone, & Szykman, 2003). For example, a patent description could be comprised of multiple associations between functions and physical elements in the form of χ matrices. Based on this description, claims can also be addressed in a similar manner.

The two main requisites for a claim are clarity and comprehensiveness. In current practice, claims are written using the format consisting of a *preamble* and a *body*. The preamble simply names the item that is to be claimed where the body describes the elements that manifest this claimed item. Consider the patent claim for a mechanical pencil (US Patent 6,705,789). One specific claim for this patent is stated by the following:

> A mechanical pencil comprising:
>
> a tubular member having a front end and a rear end;
>
> a slide member disposed at the front end of the tubular member for axial sliding movement therein, the slide member having a lead passageway for receiving a pencil lead; and
>
> a lead advancement mechanism mounted for axial movement within the tubular member and having a chuck body for undergoing advancing movement to advance the pencil lead through the lead passageway of the slide member and toward the front end of the tubular member and for undergoing retracting movement toward the rear end of the tubular member, the chuck body having a plurality of projections for engagement with the slide member so that the slide member undergoes retracting movement with the chuck body.

Dissection of this text indicates how functions, supporting functions, assemblies, components, and Wirk elements are related among each other. The phrase "a tubular member having a . . ." is captured by an association between a component and two Wirk elements. The phrase "a slide member . . ." is more complex and requires the use of both functionality and supporting functionality. Here, a slide member is associated with supporting functionality of the tubular member. The slide member is associated with both the functionality of importing "pencil lead" and the Wirk element of the inner bore or "lead passageway." These relations are illustrated in Figure 7, where this collective set of relations is represented by several matrices. Individually, these matrices are variants of the χ presented earlier where relations can be defined among the physical elements and the two types of functionality.

The graph in Figure 7, representing a claim, is a subgraph of the graph for the invention description that describes all the relations among physical elements and functionality. This suggests that the enumeration and discovery of patent claims for an invention is the result of searching for subgraphs within the product description that are unique relative to prior art where such prior art could be stored in a digital patent repository.

Numerous issues complicate the search for claims with this approach such as the large amount of data to be searched and the standard needed to qualify equivalence, similarity, and uniqueness of these candidate claims during a search operation. Nevertheless, the information content within the proposed scheme is, as an approximation, quite comprehensive of both the physical and functional characteristics of mechanical inventions with respect to the information used in a patent. This suggests that the underlying information provides sufficient resolution for the requisite standards to be applied. A benefit of using this approach is that it estab-

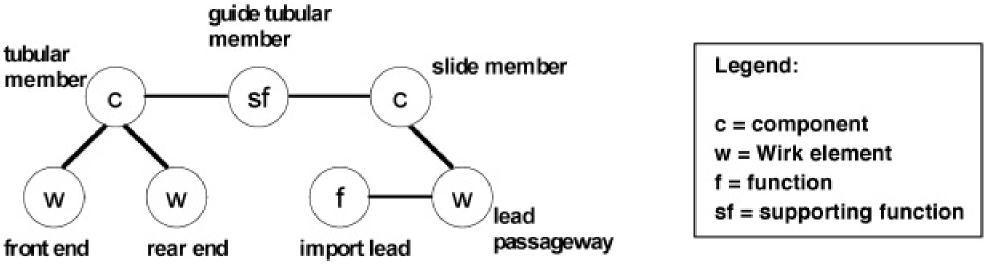


Fig. 7. Example relations for a mechanical pencil patent claim.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

lishes clear criteria for patent claims. These criteria could be helpful for determining the outcome of patent infringement issues. One reason that patents utilize such lengthy legal descriptions is that patent claims need to clearly and comprehensively describe the system and the bounds of legal protection. Employing a computational approach based on a well-defined scheme of functionality, physical elements, and their relationships is promising in this regard. As for the process of claim enumeration, the graph-based representation scheme may serve as a data structure for searches, but the subgraphs themselves would benefit from a transformation to a natural language format so that interpretation of claims can occur in a similar manner as the conventional approach.

4.2. Function-based synthesis operations for AI: Search, evaluate, and sort

Three specific operations involved in AI applications include searching, evaluating, and sorting. The following three sections demonstrate how the FBR model enhances the capabilities of these operations in computational product design methods.

4.2.1. Searching at multiple levels of abstraction

A method currently exists for using product data archived in matrix format to generate candidate physical solutions given some functional specification (Strawbridge et al., 2002). This same matrix manipulation process for this technique is demonstrated here to show how additional information suggested by the FBR model is included in automated concept generation methods. The first two examples are illustrated with a redesign case study in which a gas powered nail gun is adapted to serve as a tree tagging device. In this consulting project, the design objective is to create a proof of concept prototype to demonstrate how such a device can give land surveyors the ability to automatically dispense and attach colored tags to trees without the manual labor of holding the tag and attaching the tag with a hammer and pouch of nails. An existing nail gun is selected as a host system following an assessment of customer needs, the expected activities involved in user operation, and consideration of other alternative approaches such as staple guns and human-powered devices similar to hammers. Given this overall system choice, a functional model is created, as shown in Figure 8. In conjunction with this step, an initial architecture is proposed and given in Figure 9 for the layout of major modules based on spatial constraints.

Given this architecture and functional model, a major task is to determine how the tags stored on the side of the nail gun can be moved into place along the nail axis using the energy supplied by the operator pressing the nail gun against the tree. An illustration of this desired effect based on two different approaches is shown in Figure 10.

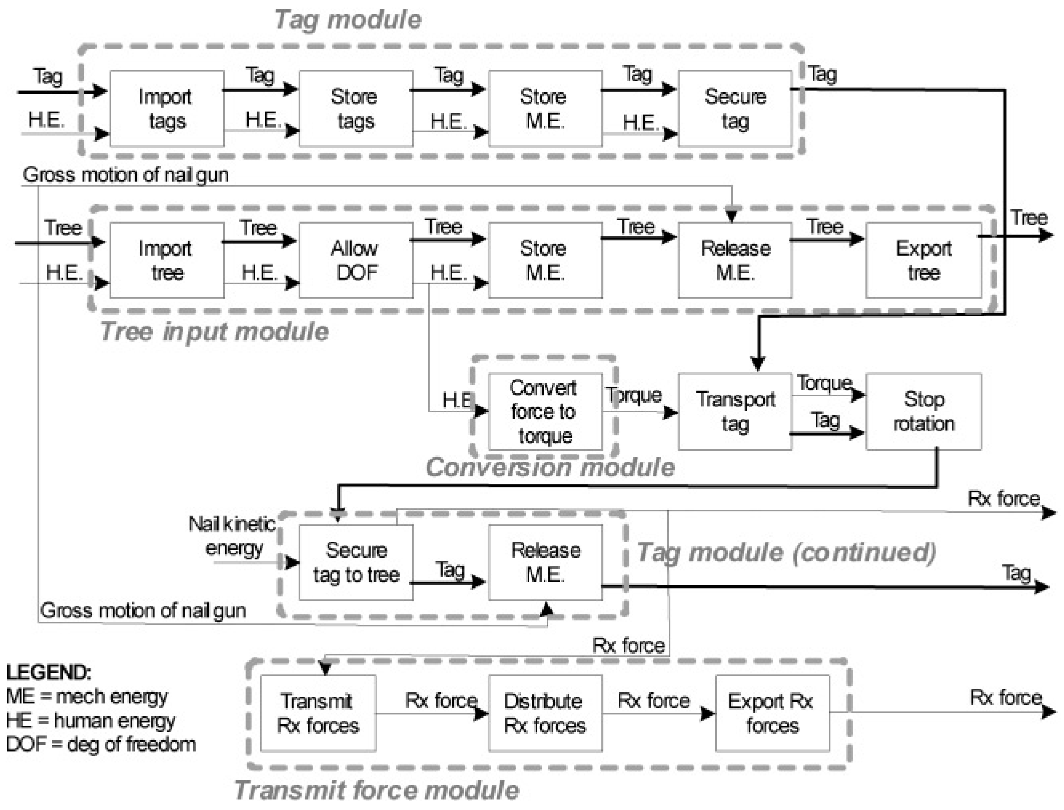


Fig. 8. The functional model for a tree tagging device.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

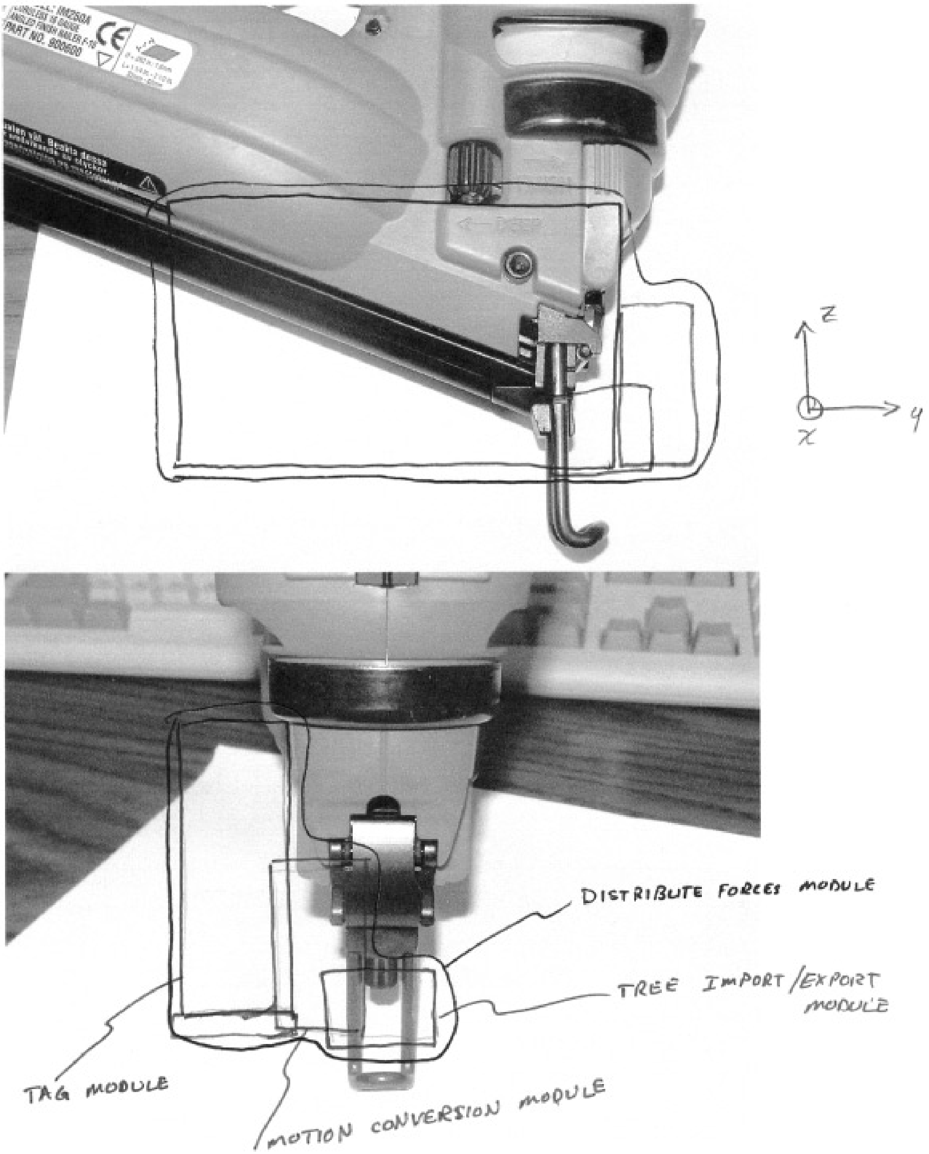


Fig. 9. The initial architecture to integrate with the existing product.

Upon selection of the rotational mode of tag motion, a search can be made for solutions that achieve this effect based on the desired functionality of "convert force to torque." This functionality is explicitly shown in the functional model. However, the functional model does not show a more abstract level of functionality that subsumes this "convert" function along with the "allow degree of freedom" and others. This more abstract level could be captured in a singular "convert *x*-motion to *y*-motion." Generation of these additional functions, like the original functional model, is simply a brainstorming exercise and not a computational task. However, the basic problem is that of organizing and aggregating lower level functions associated with higher level functions (Umeda et al., 1996; Gero & Kannnengiesser, 2004). Here this burden is placed directly on the designer for mentally developing a functional model and any ancillary functions derived from the functional model. The two convert functions, at different levels of functional abstraction, can then be searched for solutions based on data archived in χ matrices. Associations between these functions and past solutions consist of those shown in Figure 11. These particular archived solutions are at an assembly level of abstraction that are interpreted as a working principle level.

Given this historical design data, Figure 12 illustrates the matrix multiplication technique given in Strawbridge et al. (2002) for this search at the assembly level of structural abstraction. The example χ matrix is premultiplied by a filter matrix. For clarity, functions excluding the conversion functions are shown as place holders, although all func-

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

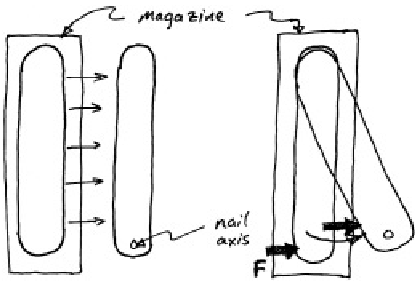


Fig. 10. Tag rotation from the storage module to a position in line with the nail firing axis.

tions in the functional model would normally be represented. This process identifies those assemblies archived in the χ matrix that match the functions designated with a 1 along the diagonal of the filter matrix. Beyond searches for assemblies, similar searches can be performed at any other level of abstraction depending on the specific needs of the designer.

4.2.2. *Evaluation and iteration of candidate solutions*

Upon completion of a search, evaluation becomes a key activity to select, modify, or backtrack to a previous step in the design. The FBR model suggests that in addition to populating χ matrices with mappings from function to form, other associations between elements within the general classes of design variables, noise variables, and performance parameters can also be formed. Given that failure mode data is classified as a performance-related issue, the technique for mapping failure mode data to components is a special case of this more general perspective. In the nail gun redesign example, it is reasonable to believe that archived data relating to retrofit type cases would include

Function	Candidate Assembly solutions
Convert force to torque	Motor & switch Rack & pinion Helical cam Bevel gear set

Function	Candidate Assembly solutions
Convert x-motion to y-motion	Linkage 1 Linkage 2

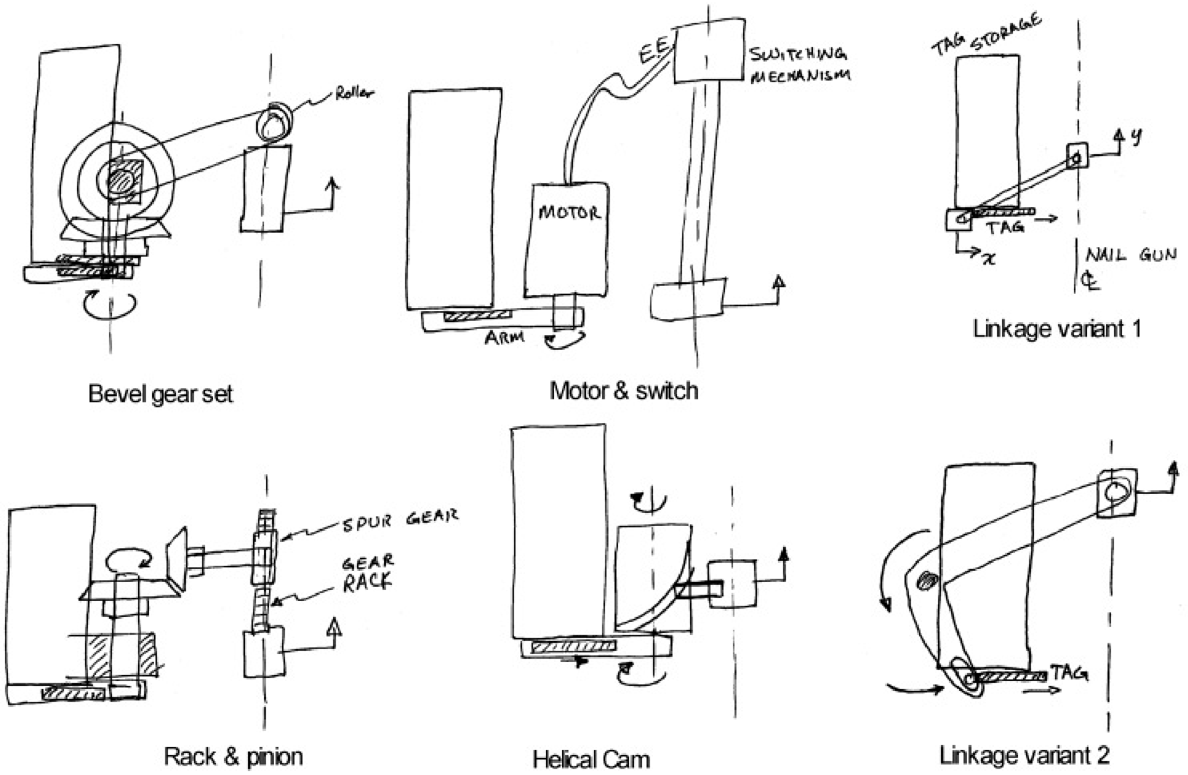


Fig. 11. Archived solutions at the assembly level of abstraction.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

	function 1	function 2	function 3	Convert force to torque	Convert y-motion to x-motion	function 6
function 1						
function 2						
function 3						
Convert force to torque				1		
Convert y-motion to x-motion					1	
function 6						

X

	p.e.	p.e.	Bevel gear set	Rack and pinion	Motor and switch	Helical cam	Linkage 1	Linkage 2
function 1	1	1	1					
function 2						1	1	
function 3				1				1
Convert force to torque			1	1	1	1		
Convert y-motion to x-motion		1					1	1
function 6	1			1		1	1	

=

	p.e.	p.e.	Bevel gear set	Rack and pinion	Motor and switch	Helical cam	Linkage 1	Linkage 2
function 1								
function 2								
function 3								
Convert force to torque			1	1	1	1		
Convert y-motion to x-motion							1	1
function 6								

Fig. 12. The generation of candidate assembly level solutions.

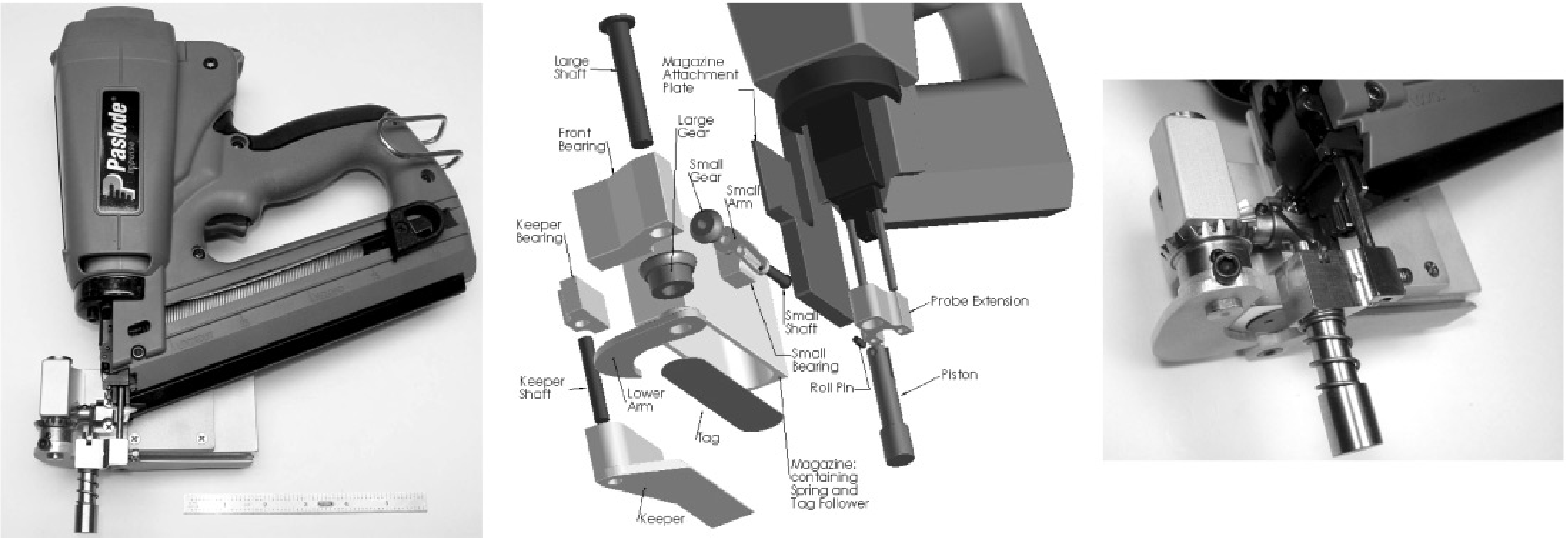


Fig. 13. The final proof of the concept.

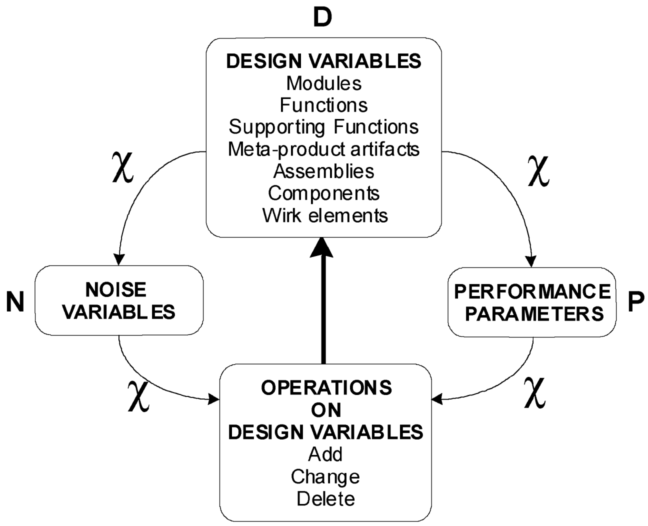
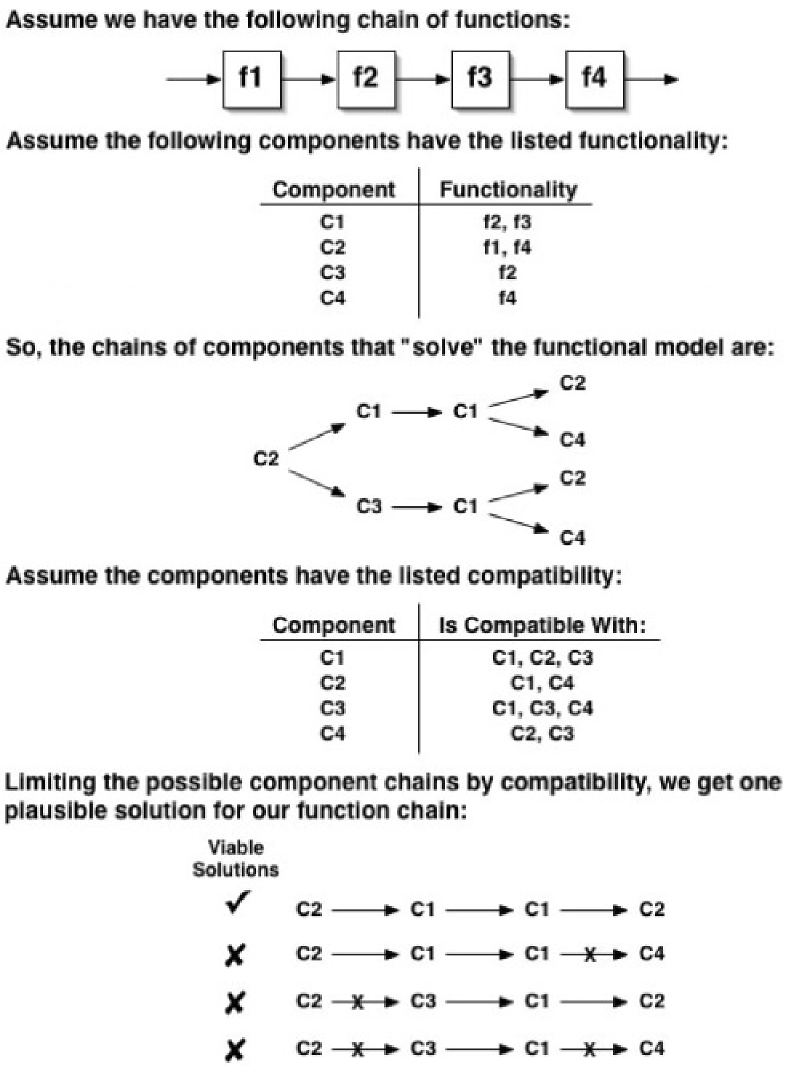


Fig. 14. The relations between various elements in the form of χ matrices that facilitate computational evaluation and iteration.

structural design solutions for adaptor plates because interfacing old to new is a common issue among retrofit design scenarios. Figure 13 shows that an adaptor plate is the main element interfacing the prototype unit with the existing product.

Specifically, a χ matrix relating the predesign context or evolutionary trigger of "retrofit" would be related to an adaptor plate. As a further extension, data could be archived to map such predesign contexts to the type of design changes that have occurred in the past. For example, the occurrence of lawsuits (a noise variable) is correlated and archived with design changes such as "add safety guard." Such design changes in general involve adding, changing, or deleting design elements (function or form). Similarly, these design changes may also be correlated with certain performance conditions. As a trivial example, a failing component will likely be correlated with deleting and subsequent replacement with an alternative. Figure 14 is a schematic for the different general categories of χ matrices that can be formed


Assume we have the following chain of functions:

Assume the following components have the listed functionality:

Component	Functionality
C1	f2, f3
C2	f1, f4
C3	f2
C4	f4

So, the chains of components that "solve" the functional model are:

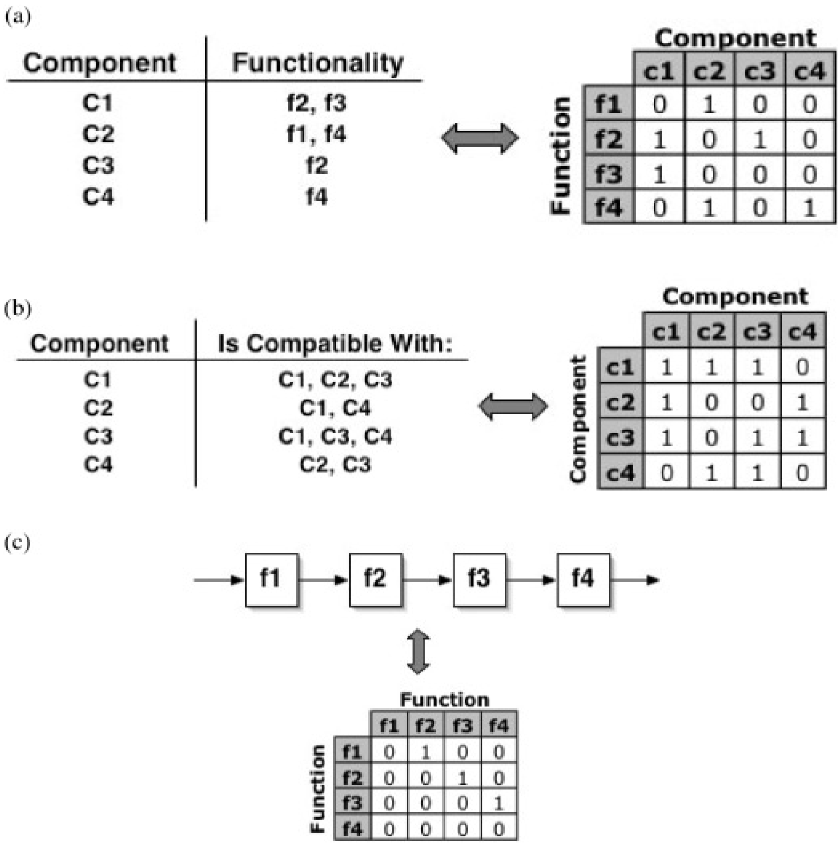
Assume the components have the listed compatibility:

Component	Is Compatible With:
C1	C1, C2, C3
C2	C1, C4
C3	C1, C3, C4
C4	C2, C3

Limiting the possible component chains by compatibility, we get one plausible solution for our function chain:

Fig. 15. A schematic of the proposed method of concept generation.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press



Component	Functionality
C1	f2, f3
C2	f1, f4
C3	f2
C4	f4

	c1	c2	c3	c4
f1	0	1	0	0
f2	1	0	1	0
f3	1	0	0	0
f4	0	1	0	1

Component	Is Compatible With:
C1	C1, C2, C3
C2	C1, C4
C3	C1, C3, C4
C4	C2, C3

	c1	c2	c3	c4
c1	1	1	1	0
c2	1	0	0	1
c3	1	0	1	1
c4	0	1	1	0

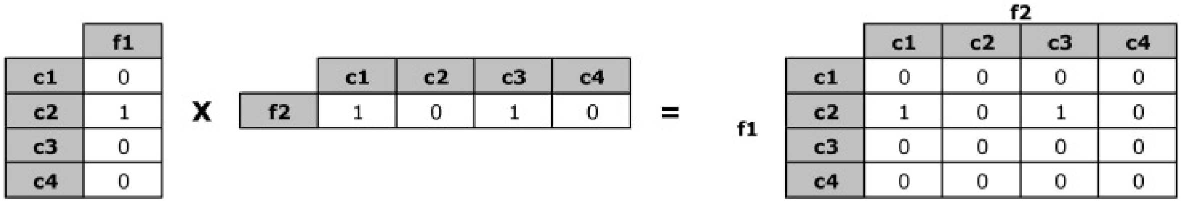
	f1	f2	f3	f4
f1	0	1	0	0
f2	0	0	1	0
f3	0	0	0	1
f4	0	0	0	0

Fig. 16. (a) Function–component, (b) design structure, and (c) functional model connectivity matrices.

from the FBR model. In this figure, the χs indicate the two items for which associations are made. By including the mappings to design changes, a loop for design iteration emerges. If initiated with an existing design, knowing the performance parameters and predesign context for the product, this loop can be interpreted as an automated redesign process. Details for controlling this iteration process for this endeavor is left for future work. Although not shown in Figure 14, associations between design variables (e.g., modules to functions) can also be stored in χ matrices to capture relations such as hierarchical decomposition of functions similar to that achieved in the function–means structure that is adopted in the function-based synthesis methodology of Schemebuilder (Bracewell, 2002).

4.2.3. Sorting candidate solutions

In the context of automated function-based synthesis, the above two sections illustrate how candidate solutions are searched given historical information of past solutions and evaluated given historical information of the performance from these past solutions. One of the challenges is to group, sort, or otherwise identify workable solutions from incompatible ones. Previous work has reported success with clustering strategies based on similarity of solution (Chakrabarti et al., 2002). The following presents an outline of a proposed algorithm for the task of identifying compatible solutions where this compatibility is based on archived product knowledge of past instances of component–component relations.



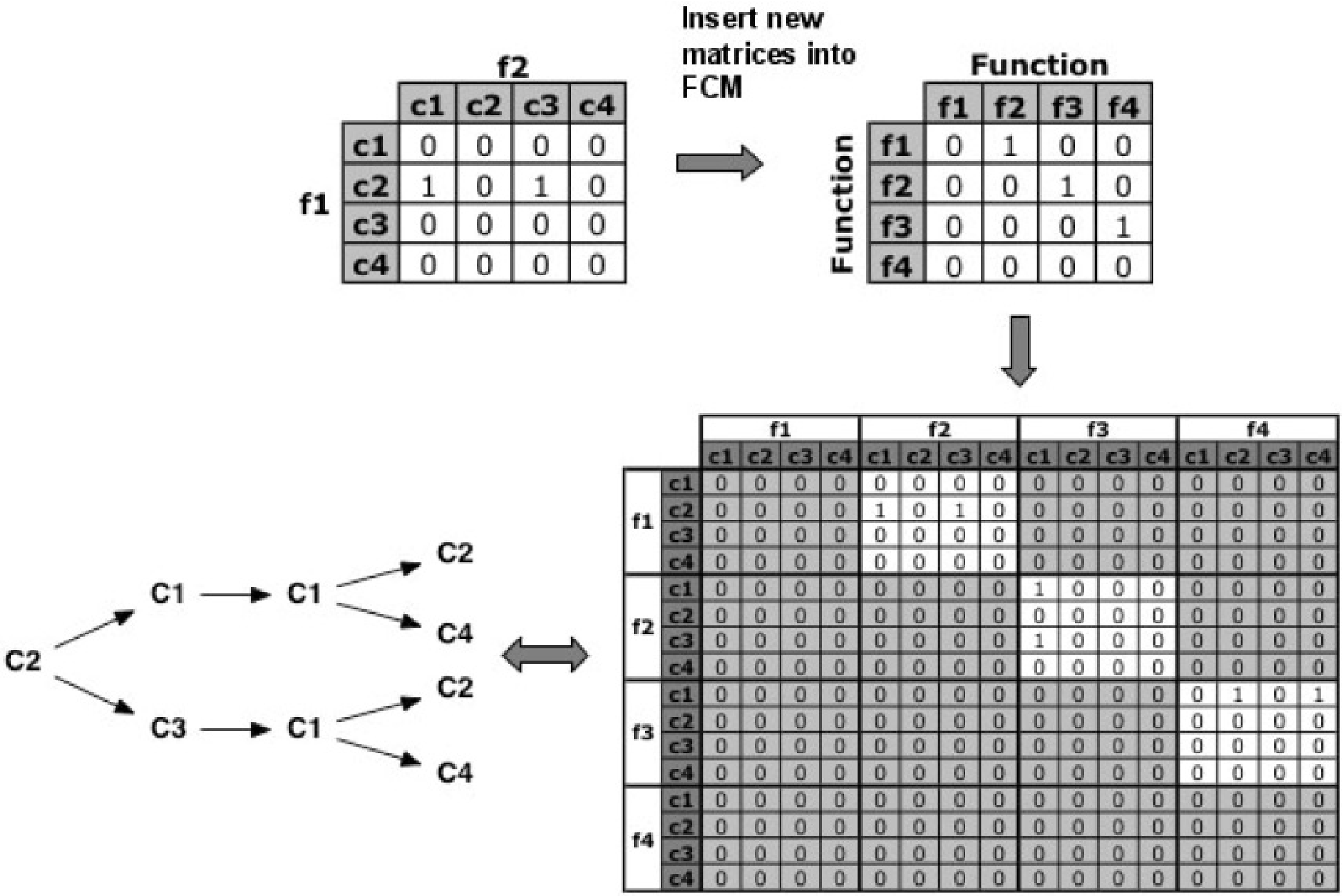
	f1
c1	0
c2	1
c3	0
c4	0

	c1	c2	c3	c4
f2	1	0	1	0

	c1	c2	c3	c4
c1	0	0	0	0
c2	1	0	1	0
c3	0	0	0	0
c4	0	0	0	0

Fig. 17. Matrix multiplication of vectors from the function–component matrix (FCM) leads to a series of matrices describing possible component design solutions.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press



		f2			
		c1	c2	c3	c4
f1	c1	0	0	0	0
	c2	1	0	1	0
	c3	0	0	0	0
	c4	0	0	0	0

	f1	f2	f3	f4
f1	0	1	0	0
f2	0	0	1	0
f3	0	0	0	1
f4	0	0	0	0

		f1				f2				f3				f4			
		c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4
f1	c1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
	c3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f2	c1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	c2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	c4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f3	c1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
	c2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f4	c1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 18. Matrices created from Figure 3 are inserted into the function connectivity matrix generated from the functional model.

Given a χ matrix for functions to components, functions in the functional model are mapped to lists of components that are capable of solving each function. The tree of possible component chains is then pruned by eliminating component connections that are not feasible according to component–component compatibility. A schematic of this process is shown in Figure 15.

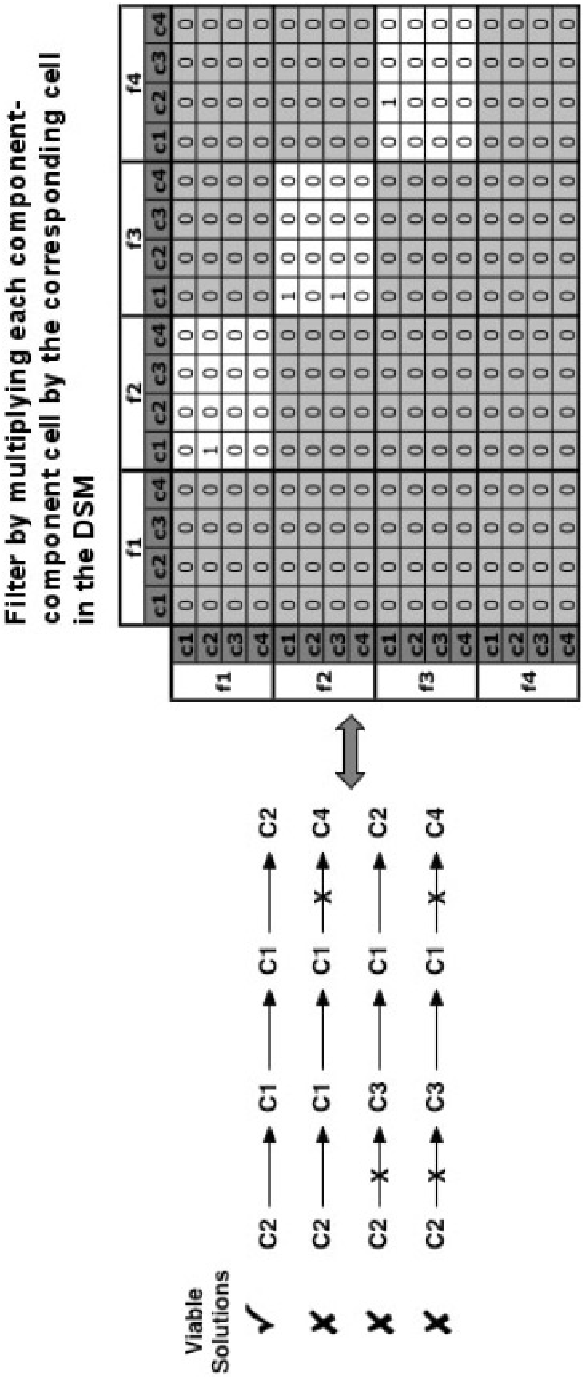
Specific χ matrices, including the function–component matrix (FCM) that describes the function–component relationships (Fig. 16a) and the DSM that describes the component–component compatibility (Fig. 16b), coupled with a matrix defining the connectivity from the functional model (Fig. 16c) can be manipulated to produce the filtered matrix of design solutions.

Rows corresponding to the functions present in the designer's functional model are extracted from the FCM and multiplied together to produce matrices of possible component chains linking each pair of functions together as given in Figure 17.

These matrices can then be entered into the function connectivity matrix as shown in Figure 18 and then filtered with the DSM to remove component–component combinations that are incompatible as illustrated in Figure 19. The matrices generated represent a component tree of design solutions that has been pruned of infeasible designs. If the component connections of existing products are used to fill the DSM, then design solutions can be ranked according to historical occurrence of component connections. In other words, the most common design solutions to the functional model would bubble to the top of the list of generated concepts. Other factors, such as failure modes derived from a separate evaluation step, could also be used as filters during the sorting stage to help limit and rank design solutions.

5. DISCUSSION AND FUTURE WORK

This work presents a model of FBRs that describes both the functional and structural elements of this model as they relate to resolution of knowledge representation. Although these elements have individually been introduced in prior research efforts, this work identifies a sufficient and irreducible set of these items in a composite manner that accounts for a full range of function and structure resolutions applicable in mechanical design. In addition to the basic items of function and structure, the FBR model incorporates the broader concepts of design variables, noise variables, and performance parameters where the specifications of function and form are subsumed into the set of design variables. As a whole, this provides a new coherent framework that allows a great number of associations or mappings to be made between the items comprising the FBR model. A matrix-based approach is demonstrated as a computable means for handling these associations. Given the FBR composition and their mappings, an initial assessment of the model is made with respect to a comprehensive set of design activities. Potential of the FBR model is developed further



		f1				f2				f3				f4			
		c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4
f1	c1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	c3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f2	c1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	c2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	c4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f3	c1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	c2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f4	c1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	c4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 19. Cells of the possible design solution matrices are multiplied with corresponding cells of the design structure matrix (DSM) to produce a filtered list of design solutions.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

through descriptions of how the concepts of the FBR are used for both existing and proposed AI methods ranging from knowledge archival in product repositories to computational synthesis operations that support concept generation.

A direct benefit of the FBR model is that it clarifies and demonstrates how function-based information can be related to not only structure, but also to variables that reflect design performance in a computable format. Given that product behavior is intimately tied with performance, the use of performance information as indicated by the FBR model seems promising for future investigations of product behavior. Although design synthesis based on archived knowledge is specifically addressed in this work, another benefit that such archived product knowledge provides is a convenient source of data for empirical studies. Development of prescriptive design methods, whether heuristics or advanced processes, rests on the information gleaned from a descriptive evaluation of how products are and should be designed. The FBR model presented in this work is suggestive of many function-based relations that could be archived in a computational format and examined during an empirical study to develop this descriptive product information. In addition to empirical studies, future work should explore and develop AI design methods using the computational relations that the FBR model provides.

REFERENCES

Ahmed, S., & Wallace, K. (2003). Evaluating a functional basis. *ASME Design Engineering Technical Conf. Proc*. Paper No. DET2003/DTM-48685.

Aurisicchio, M., Bracewell, R., & Wallace, K. (2003). A design data model to support rationale capture and functional synthesis. *ASME Design Engineering Technical Conf. Proc*. Paper No. DET2003/DTM-48672.

Blessing, L. (1994). *A process-based approach to computer supported engineering design*. PhD Thesis. Enschede, The Netherlands: University of Twente.

Bohm, M., & Stone, R. (2004). Representing functionality to support reuse: conceptual and supporting functions. *ASME Design Engineering Technical Conf. Proc*. Paper No. DET2004/DTM-57693.

Bohm, M., Stone, R.B., & Szykman, S. (2003). Enhancing virtual product representations for advanced design repository systems. *ASME Design Engineering Technical Conf. Proc*. Paper No. DETC2003/CIE-48239.

Bohm, M., Stock, M., Hutcheson, R., McAdams, D., Stone, R., & Donndelinger, J. (2003). *Opportunities for application of formal functional modeling methods in advanced vehicle development*. General Motors contract report. Detroit, MI: General Motors.

Bracewell, R. (2002). Synthesis based on function–means trees: Schemebuilder. In *Engineering Design Synthesis* (Chakrabarti, A., Ed.), pp. 199–212.

Brown, D.C. (2003). Functional, behavioral and structural features. *ASME Design Engineering Technical Conf. Proc*. Paper No. DET2003/DTM-48684.

Bryant, C., Kurfman, M., Stone, R., & McAdams, D. (2001). Creating equation handbooks to model design performance parameters. *ICED01—Int. Conf. Engineering Design*, Glasgow.

Chakrabarti, A., Langton, P., Liu, Y., & Bligh, T. (2002). An approach to compositional synthesis of mechanical design concepts using computers. In *Engineering Design Synthesis* (Chakrabarti, A., Ed.), pp. 179–198. New York: Springer–Verlag.

Chandrasekaran, B., & Josephson, J. (2000). Function in device representation. *Engineering with Computers 16*, 162–177.

Chang, E., Li, X., & Schmidt, L. (2000, May 21). The need for a form, function, and behavior-based representation system. Available on-line at www.enme.umd.edu/DATLab

Chittaro, L.,& Kumar, A. (1998). Reasoning about function and its applications to engineering. *Artificial Intelligence in Engineering 12*, 331–336.

Deng, Y. (2002). Function and behavior representation in conceptual mechanical design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 16(4)*, 343–362.

Gero, J., & Kannengiesser, U. (2004). The situated function–behaviour–structure framework. *Design Studies 25(4)*, 373–391.

Gietka, P., Verma, M., & Wood., W. (2002). Functional modeling, reverse engineering, and design reuse. *ASME Design Engineering Technical Conf. Proc*. Paper No. DETC2002/DTM-34019.

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.

Hubka, V., & Eder, W. (2001). Functions revisited. *ICED01—Int. Conf. Engineering Design*, Glasgow.

Janhager, J. (2003). Hierarchical decomposition of technical functions and user actions. *ASME Design Engineering Technical Conf. Proc*. Paper No. DETC2003/DTM-48642.

Jensen, T. (2000). Function integration explained by allocation and activation of Wirk elements. *ASME Design Engineering Technical Conf. Proc*. Paper No. DETC2000/DTM-14551.

Keuneke, A. (1991). Device representation—the significance of functional knowledge. *IEEE Expert 6(1)*, 22–25.

Kitamura, Y., & Mizoguchi, R. (1998). Functional ontology for functional understanding. *AAAI Proc. 12th Int. Workshop on Qualitative Reasoning (QR-98)*, pp. 77–87.

Kitamura, Y., & Mizoguchi, R. (1999). Meta-functions of artifacts. *AAAI Proc. 13th Int. Workshop on Qualitative Reasoning (QR-99)*, pp. 136–145.

Kitamura, Y., & Mizoguchi, R. (2004). Ontology-based systematization of functional knowledge. *Journal of Engineering Design 15(4)*, 327–351.

Kurfman, M., Stock, M., Stone, R., Rajan, J., & Wood, K. (2003). Experimental studies assessing the repeatability of a functional modeling derivation method. *Journal of Mechanical Design 125(4)*, 682–693.

McAdams, D., & Dym, C. (2004). Modeling and information in the design process. *ASME Design Engineering Technical Conf. Proc*. Paper No. DETC2004/DTM-57101.

Miles, L.D. (1961). *Techniques of Value Analysis and Engineering*. New York: McGraw–Hill.

Otto, K., & Wood, K. (2001). *Product Design: Techniques in Reverse Engineering and New Product Design*. Englewood Cliffs, NJ: Prentice–Hall.

Pahl, G., & Wallace, K. (2002). Using the concept of functions to help synthesise solutions. In *Engineering Design Synthesis* (Chakrabarti, A., Ed.), pp. 109–119. New York: Springer–Verlag.

Shi, H., & Schmidt, L. (2003). Partitioning knowledge for generative configuration design. *ASME Design Engineering Technical Conf. Proc*. Paper No. DETC2003/DTM-48686.

Sim, S., & Duffy, A. (2003). Towards and ontology of generic engineering activities. *Research in Engineering Design 14*, 200–223.

Stetter, R., Abmann, G., & Viertlbock, M. (2001). Functional product modeling—new methods for the generation of product functions. *ICED01—Int. Conf. Engineering Design*, Glasgow.

Stone, R., & Wood, K. (2000). Development of a functional basis for design. *Journal of Mechanical Design 122(4)*, 359–370.

Stone, R., Wood, K., & Crawford, R. (2000). A heuristic method for identifying modules for product architectures. *Design Studies 21(1)*, 5–31.

Strawbridge, Z., McAdams, D., & Stone, R. (2002). A computational approach to conceptual design. *ASME Design Engineering Technical Conf. Proc*. Paper No. DETC2002/DTM-34001.

Tumer, I., & Stone, R. (2003). Mapping function to failure mode during component development. *Research in Engineering Design 14(1)*, 25–33.

Ullman, D. (1997). *The Mechanical Design Process*, 2nd ed. New York: McGraw–Hill.

Umeda, Y., & Tomiyama, T. (1997). Functional reasoning in design. *IEEE Expert 12(2)*, 42–48.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., & Tomiyama, T. (1996). Supporting conceptual design based on the function–behavior–state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10(3)*, 275–288.

Van Wie, M., Stone, R., & McAdams, D. (2004). Sustainable design through flexible product evolution. *ASME Int. Mechanical Engineering Congr*. Paper No. IMECE2004-60667.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press

Wood, K., & Greer, J. (2001). Function-based synthesis methods in engineering design—State of the art, methods analysis, and visions for the future. In *Formal Engineering Design Synthesis* (Antonsson, E., & Cagan, J., Eds.), pp. 170–227. New York: Cambridge University Press.

Yekula, R., McAdams, D., & Stone, R. (2003). Functional and mathematical equivalence of mechanisms: a novel approach to integrating synthesis and design analysis. *ASME Design Engineering Technical Conf. Proc*. Paper No. DETC2003/DTM-48663.

Michael Van Wie is a Postdoctoral Research Fellow in the Department of Interdisciplinary Design Engineering at the University of Missouri–Rolla. He received his BS from Texas A&M University and his MS and PhD from the University of Texas at Austin. His research is aimed at the development of formalisms, methods, and tools that improve the ability of designers to perform their engineering tasks. Dr. Van Wie's research includes theoretical studies to establish the basic underpinnings of how design is and should be performed and more applied work that examines these new theories through case studies involving actual design problems from industry.

Cari R. Bryant received her BS in mechanical engineering from the University of Missouri–Rolla in 2000. In 2003 she received an MS in mechanical engineering and an MS in biomedical engineering from the University of Michigan while doing research in the University of Michigan Orthopaedic Research Laboratories. She is now working on her PhD in mechanical engineering in the Design Engineering Laboratory at the University of Missouri–Rolla.

Matt R. Bohm received his BS in mechanical engineering from the University of Missouri–Rolla in 2002. In 2004 he received his MS in mechanical engineering from the University of Missouri–Rolla while doing research in the Design Engineering Laboratory. He is now working on a PhD in computer science at the University of Missouri–Rolla through the Design Engineering Laboratory.

Daniel A. McAdams is an Assistant Professor of mechanical engineering in the Department of Mechanical and Aerospace Engineering at the University of Missouri–Rolla. He joined the department in October of 1999 after receiving his PhD from the University of Texas at Austin. Dr. McAdam's research interests are in the area of design theory and methodology, product architecture, design for manufacturing, tolerance design, and modeling for design, with specific emphasis and applications in product design.

Robert B. Stone is currently an Associate Professor in the Interdisciplinary Engineering Department and the Director of The Student Design Center in the School of Engineering at the University of Missouri–Rolla. He joined Basic Engineering, a service teaching department, in January 1998, after completing his PhD in mechanical engineering from the University of Texas at Austin. Dr. Stone has utilized his design background to develop undergraduate interdisciplinary design methodology courses and graduate courses covering and implementing cutting-edge product design methods. Recently, he has led a planning grant effort to transform the Basic Engineering Department into the degree granting Interdisciplinary Engineering Department. Dr. Stone's research interests are design theory and methodology, specifically product architectures, functional representations, and design languages. He has received multiple awards related to his research in product design, and has authored a chapter on product architecture in a product design text. Prior to initiating his graduate work, Dr. Stone worked in the Missions Operation Directorate of NASA–Johnson Space Center as a Space Shuttle Flight Controller.

https://doi.org/10.1017/S0890060405050092 Published online by Cambridge University Press