5-2009

A STUDY IN THE INFORMATION CONTENT, CONSISTENCY, AND EXPRESSIVE POWER OF FUNCTION STRUCTURES IN MECHANICAL DESIGN

Chiradeep Sen
Clemson University, csen@clemson.edu

A STUDY IN THE INFORMATION CONTENT, CONSISTENCY, AND EXPRESSIVE POWER OF FUNCTION STRUCTURES IN MECHANICAL DESIGN

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

---

by
Chiradeep Sen
May 2009

Accepted by:
Joshua D. Summers, Committee Chair
Gregory M. Mocko
Georges M. Fadel

ABSTRACT

In engineering design research, function structures are used to represent the intended functionality of technical artifacts. Function structures are graph-based representations where the nodes are functions, or actions, and the edges are flows, or objects of those actions. For the consistent description of artifact functionality, multiple controlled vocabularies have been developed in previous research. The Functional Basis is one such vocabulary that provides for a set of verbs and a set of nouns, organized in the three-level hierarchy. This vocabulary is extensively studied in design research. Two major application of this vocabulary are the Design Repository, which is a web-base archive of design information of consumer electro-mechanical products obtained through reverse engineering, and the functional decomposition grammar rules that synthesizes sub-functions or elementary actions of a product from the overall function or goal of the product. However, despite the Functional Basis' popularity, the usefulness of its hierarchical structure has not been specifically tested. Additionally, although this vocabulary provides the verbs and nouns, no explicit guideline for using those terms in function structures has been proposed. Consequently, multiple representational inconsistencies can be found in the function structures within the Design Repository. The two research goals in this thesis are: (1) to investigate if the hierarchy in the Functional Basis is useful for constructing function structures and (2) to explore means to increase the consistency and expressive power of the Functional Basis vocabulary.

To address the first goal, an information metric for function structures and function vocabularies is developed based on the principles of Information Theory. This metric is applied to three function structures from the Design Repository to demonstrate that the secondary level of the Functional Basis is the most informative of the three. This finding is validated by an external empirical study, which shows that the secondary level is used most frequently in the Design Repository, finally indicating that the hierarchy is not useful for constructing function structures.

To address the second research goal, a new representation of functions, including rules the topological connections in a function structure, is presented. It is demonstrated through experiments that the new representation is more expressive than the text-based descriptions of functions used in the Functional Basis, as it formally describes which flows can be connected to which functions. It is also shown that the new representation reduces the uncertainty involved in the individual function structures.

DEDICATION

To the memory of Atai, who taught me the language of music

ACKNOWLEDGMENTS

I thank my research co-advisors, Dr. Joshua D. Summers and Dr. Gregory M. Mocko, for teaching me how to do research, for pushing me when I slacked, for helping me with my research with their criticism and advice, and for being dependable friends. I owe much of my academic, professional, and moral growth over the past two years directly to them. I also thank Dr. Georges M. Fadel, my advisory committee member, for helping me through the review of this thesis.

I thank the fellow graduate students of the Automation in Design group (AiD) and the Engineering Information Management Laboratory (EIML) at Clemson University for their friendship and support and for continuously enriching my thoughts with alternate viewpoints. Specifically, I thank Ben Caldwell for conducting the preliminary empirical study that inspired the research presented here. I also thank Carl Lamar, Essam Namouz, Jay Richardson, Yvette Price, James McLellan, Jonathan Thomas, David Stowe, Gary Palmer II, and Beshoy Morkos.

Finally, I thank my wife Sunrita, my parents Arup and Reba, my sister Paramita, and Dr. Ashok Roychoudhury and his family for their constant love and support, without which this work would not be possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE: THESIS OVERVIEW, SUMMARY OF RESEARCH GAPS, RESEARCH QUESTIONS, AND HYPOTHESES

In engineering design, models are used to represent different aspects of the design artifact. For example, requirements lists capture the customer's needs, computational geometric models represent the spatial form of the solution, and structural or thermal analysis models simulate the product's response to the operating conditions. Artifact functionality is one aspect that helps the designer to understand how a product works and to search for new designs solutions [Ullman, 1992; Otto & Wood, 2001; Pahl et al., 2007; Ulrich & Eppinger, 2008]. In this case, a formal representation that supports consistent modeling of functionality could enable automated (computerized) reasoning that can in turn assist designers to accomplish those tasks faster and more effectively. To this end, the development of a formal representation of artifact functionality is the overall research objective of this thesis. The following section briefly identifies some critical research gaps in this area, and summarizes the contributions of this thesis in addressing those gaps.

1.1 Summary of Research Gaps in Function Representations

Multiple representations are studied in engineering design research for describing artifact functionality, as elaborated in Chapter Two and Chapter Three. Notable examples include the graph-based function structures of Pahl and Beitz [Pahl et al., 2007], the Function-Behavior-Structure model (FBS) proposed by Gero and colleagues [Gero, 1990; Gero & Kannengiesser, 2000; Gero & Kannengiesser, 2002], the Function-Behavior-State (FBSt) model of Tomiyama and colleagues [Umeda & Tomiyama, 1995; Erden et al., 2008], the Structure-Behavior-Function (SBF) model offered by Goel and colleagues [Bhatta et al., 1994; Goel & Bhatta, 2004], the representation of function as the artifact's role and effects, as suggested by Chandrasekaran and colleagues [Chandrasekaran & Josephson, 1997; Chandrasekaran & Josephson, 2000; Chandrasekaran, 2005], and the affordance-based view of functionality, proposed by Maier and Fadel [Maier & Fadel, 2001; Maier & Fadel, 2002; Maier, 2008]. Many of these representations are not logically or mathematically rigorous enough to support automated reasoning, as discussed in Chapter Three.

By contrast, the function structure representation is recognized here to be reasonably consistent and rigorous (Section 3.3), and therefore, is undertaken for evolving into a more formal version. Function structures are graph-based representations where nodes are *functions* or actions executed by the artifact and edges are *flows* or objects those actions. Figure 1 shows an example of a function structure for a conceptual carpet-tile packing machine.

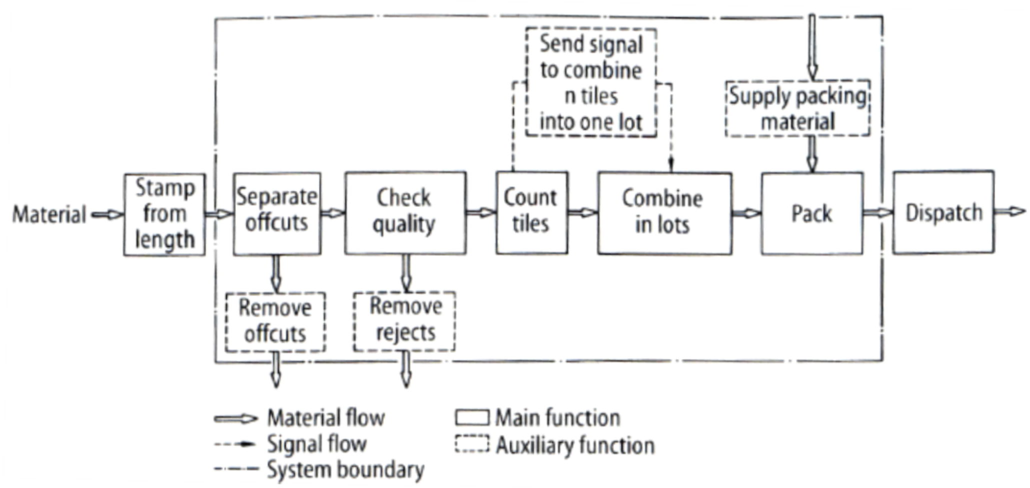


Figure 1: Function structure of a carpet-tile packing machine [Pahl et al., 2007]

Here the blocks describe the actions taken by the machine, while the arrows indicate the flows of carpet tile, offcuts, rejects, and packing material through the machine. Function structures are widely studied in engineering design research due to their simple underpinnings and multiple applications in original design and reverse engineering [Kurfman et al., 2000; Stone & Wood, 2000; Otto & Wood, 2001; Hirtz et al., 2002; Kurfman et al., 2003; Bohm & Stone, 2004; Sridharan & Campbell, 2005; Bohm et al., 2006; Caldwell & Mocko, 2007; Kurtoglu, 2007; Pahl et al., 2007; Caldwell et al., 2008; Ulrich & Eppinger, 2008; Sen et al., 2010]. However, a major limitation of this representation is that the connections between the functions and flows in a function structure are not formally controlled, leaving room for representational inconsistency in the models. A motivation behind this thesis is to develop an evolved representation of functions to address this research gap.

1.2 Summary of Research Gaps in Function Vocabularies and their Applications

The development of a controlled vocabulary is recognized as a viable first step toward formalizing a domain [Summers et al., 2001; Luger, 2002]. Consequently, several vocabularies are proposed in previous research for describing functionality, as discussed in Section 3.4. One of those vocabularies, the Functional Basis [Stone & Wood, 2000; Hirtz et al., 2002], is widely studied in academic research and has been used in multiple applications [Tumer & Stone, 2001; Arunajadai et al., 2002; McAdams & Wood, 2002; Sridharan & Campbell, 2004; Sridharan & Campbell, 2005; Stone et al., 2005; Bryant et al., 2006; Vucovich et al., 2006]. This vocabulary consists of 53 verbs and 45 nouns organized in a three-level hierarchy, as shown in Table 1 and Table 2.

Table 1: Functional Basis verbs hierarchy [Hirtz et al., 2002]

Primary	Secondary	Tertiary
Branch	Separate	Divide
		Extract
		Remove
	Distribute	
Channel	Import	
	Export	
	Transfer	Transport
		Transmit
	Guide	Translate
		Rotate
		Allow DoF
Connect	Couple	Join
		Link
	Mix	
Control Magnitude	Actuate	
	Regulate	Increase
		Decrease
	Change	Increment
		Decrement
		Shape
		Condition
	Stop	Prevent
		Inhibit
Convert	Convert	
Provide	Store	Contain
		Collect
	Supply	Supply
Signal	Sense	Detect
		Measure
	Indicate	Track
		Display
	Process	
Support	Stabilize	
	Secure	

Table 2: Functional Basis nouns hierarchy [Hirtz et al., 2002]

Primary	Secondary	Tertiary
Material	Human	
	Gas	
	Liquid	
	Solid	Object
		Particulate
		Composite
	Plasma	
	Mixture	Gas-Gas
		Liquid-Liquid
		Solid-Solid
		Solid-Liquid-Gas
		Colloidal
Signal	Status	Auditory
		Olfactory
		Tactile
		Taste
		Visual
	Control	Analog
		Discrete
Energy	Human	
	Acoustic	
	Biological	
	Chemical	
	Electrical	
	Electromagnetic	Optical
		Solar
	Hydraulic	
	Magnetic	
	Mechanical	Rotational
		Translational
	Pneumatic	
	Radioactive/Nuclear	
	Thermal	

The functionality of an artifact can be described by forming predicates using the terms listed in these two tables. For example, the functionality of a storage cell can be described as “store energy”, and the functionality of an electric motor can be described as “convert electric energy to rotational mechanical energy”. By using a controlled vocabulary the consistency of term selection for function description can be increased.

Two of the Functional Basis’ more noteworthy applications are the Design Repository developed at the Missouri University of Science & Technology [Bohm et al., 2003; Bohm & Stone, 2004; Bohm et al., 2005; Bohm et al., 2006], and the grammar rules for the automated functional decomposition developed at University of Texas, Austin [Sridharan & Campbell, 2004; Sridharan & Campbell, 2005]. The Design Repository[1] is a web-based archive of design information of consumer electro-mechanical products obtained through reverse engineering, as discussed in Section 3.4.2. The grammar rules make use of historical data stored within the Design Repository to identify trends of artifact function. Once a few trends are identified, they are applied to new design problems to produce design concepts. These rules are discussed in Section 3.4.3.

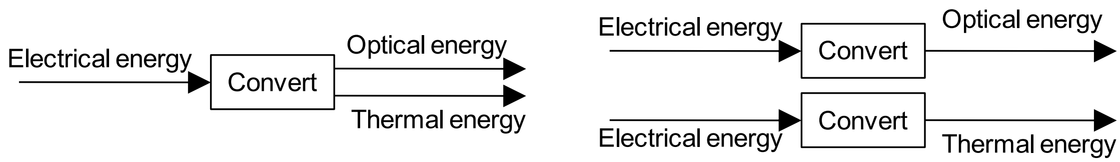
Despite its popularity in academic research, the major limitations of the Functional Basis and its applications are (1) the lack of formalism for building connections between functions and flows, (2) the lack of expressive power of the vocabulary terms, (3) the bottom-up research approach that cannot guarantee adequacy of the vocabulary, and (4) the lack of uniform usage of the hierarchy in function structures. These points are discussed below.

1.2.1 The Lack of Connection Formalism

Although the controlled vocabulary can increase the consistency of terms usage in function structures, it provides no guideline about how these terms should be connected in a model. For example, it does not specify whether the conversion of electrical energy to light and heat in an incandescent lamp should be modeled as one conversion action

[1] http://repository.designengineeringlab.org/, accessed on January 27, 2009

with one input (electrical energy) and two outputs (optical and thermal energies) as shown in Figure 2(a), or with two separate functions, one each for the two conversions as shown in Figure 2(b).



(a) Using one function **(b) Using two functions**

Figure 2: Two possible ways of modeling the simultaneous conversion of one energy type into two other types

The connections between the functions and flows in specific models are presently left to the designer’s preference. Since the function structures stored in the Design Repository were created by various designers, including faculty members, graduate students, and visiting students of the Missouri University of Science & Technology, these models demonstrate a variety of personal preferences, leading to representational inconsistency (Section 3.4.4).

1.2.2 The Lack of Expressive Power of Vocabulary Terms

The above example also illustrates lack of expressive power of the vocabulary terms and the function structure representation as a whole. As discussed in Chapter Six, the definitions of the Functional Basis verbs (0) indicate that not all verbs and nouns are compatible for topological connections. By formally capturing this knowledge more powerful reasoning can be potentially performed on the function structures, making the vocabulary terms and the representation as a whole more expressive. However, due to the lack of topological guidelines in the Functional Basis, this knowledge remains to be captured.

1.2.3 The Bottom-Up Research Approach

The third limitation of the Functional Basis lies in its empirical research approach, which relies heavily upon historical data. For example, the Functional Basis was evolved in a so called *bottom-up* manner, by examining existing artifacts and including the terms required to describe functions that were not noticed in previously studied artifacts. In this manner, the vocabulary grew to a natural maturity level where no additional terms were required to describe subsequently investigated artifacts. Therefore, the adequacy of this vocabulary is based upon empirical validation, but there is no theoretical underpinning explaining why those terms should be necessary or sufficient for describing artifact functionality. In fact, an external empirical investigation has revealed that the function structures in the Design Repository contain up to 25% non-Functional Basis terms, indicating that the vocabulary may not be adequate [Caldwell et al., 2008].

Similarly, the trends that are used to derive the functional decomposition grammar rules [Sridharan & Campbell, 2004; Sridharan & Campbell, 2005] are based upon historical data of decomposition found in the Design Repository. Notably, these rules do produce multiple ways of decomposing a function structure, and therefore are examples of automated reasoning about functionality. However, due to the bottom-up approach, the ideas generated are limited to combinatory variations of previously known solutions, rather than being synthesized from fundamental logical arguments about which functions can be decomposed into which others, called the *top-down* approach. As a result, the rules are not necessarily sufficient for generating solutions to new problems. Moreover, due to the inadequacy of the vocabulary and inconsistency of modeling formalism, the rules are also subject to similar limitations.

1.2.4 The Non-Uniform Utilization of the Hierarchy

An empirical study shows that the three hierarchical levels of the Functional Basis vocabulary are not used uniformly in the function structures within the Design Repository, though they are created by the same research group that developed the Functional Basis [Caldwell et al., 2008]. Specifically, the study reveals that over 90% of terms used in those models belong to the secondary level of the vocabulary, indicating that the hierarchy may not be useful for constructing function structures.

These findings bring under scrutiny the validity of the reverse engineering approach (bottom-up) of creating function vocabularies. Therefore, it is important to objectively examine the usefulness of the Functional Basis, specifically, its three-level hierarchy. These findings also illustrate the lack of representational rigor and expressive power of the function structure representation, and provide motivation for enhancing it to a more rigorous version. The motivation behind this thesis is therefore multifold:

1. to develop a means to quantify the usefulness of a function vocabulary,
2. to assess the usefulness of hierarchical organization of terms in the Functional Basis, and
3. to develop a means to enhance the expressive power of the function structure representation.

The research questions presented in the next section address these motivations.

1.3 **Research Questions**

The first research question is aimed at assessing the usefulness of the Functional Basis hierarchy. The sub-questions need to be answered in order to address the overall question, or to validate the answer to the overall question.

RQ-1. Are the hierarchical levels of the Functional Basis equally useful for constructing function structures?

a. What metric should be used to quantify the usefulness of a function structure?
b. What metric should be used to quantify the usefulness of a vocabulary?
c. What is the practical interpretation of the metric of usefulness?
d. Is the assessment of the hierarchy supported by experimental results?

The second research question is aimed at improving the quality of the function structure representation.

RQ-2. How can the function structure representation be made more expressive?

a. What metric should be used to measure the representation's expressiveness?
b. Which elements can be formally represented to increase the expressiveness?

1.4 **Research Hypotheses**

Based on the empirical study [Caldwell et al., 2008], it is hypothesized that the secondary level of the Functional Basis is the most useful of the three. The metric of usefulness proposed in this thesis is adapted from Information Theory [Hartley, 1928; Shannon, 1948], originally developed in the field of communications. This metric, called *information content,* is a measure of the amount of information designers can extract from a specific model built upon a specific representation. Practically, the metric is interpreted as the number of questions that the model can answer about the described object. Chapter Four presents the details of this metric. As discussed in Chapter Six, this metric can be adapted to assess the usefulness of function structures and vocabularies, as well as the expressiveness of the representation. The expressiveness of the function structure representation is enhanced in this thesis by formally capturing the meaning of each verb within the Functional Basis vocabulary. The following section summarizes the above discussion in form of research hypotheses.

Hyp-1. The secondary level of the Functional Basis is the most useful one for constructing function structures.

Hyp-2. The concept of information entropy in Information Theory research can be modified to create a metric to assess the usefulness of specific function structures, function vocabularies, and the expressiveness of function representations.

Hyp-3. Function structures can be made more expressive by formally describing the meanings of the function verbs within the Functional Basis.

CHAPTER TWO: ENGINEERING DESIGN REPRESENTATIONS AND THEIR QUALITY MEASURES

2.1 Models and Representation in Engineering Design

Models are used in engineering design to describe facts about the design product, the design process, or the design problem in a form that can support reasoning about the domain. For example, computational geometric models (Computer-Aided-Design models or CAD models) represent the spatial form of the solution, structural or thermal analysis models simulate the product's response to the operating conditions, and engineering databases can be used to capture product specifications. A definition of the term *model* is found in artificial intelligence research as follows: "To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A" [Minsky, 1965]. Thus, a model is an abstraction of reality that can be used to answer questions about the reality. For example, the CAD-models can answer geometry-related questions, such as "What is the nominal clearance between the bearing and the shaft journal in a particular design?" The analysis models can answer questions related to the operating conditions, such as "what is the expected minimum oil film thickness between the shaft and the bearing under a specific load and speed?" Similarly, the databases can answer specification-related questions, such as "how many standard bearing oils are available that can support a certain load at a certain speed?"

Each model, in turn, is built upon a representation, which formulates the underlying framework for capturing knowledge. For example, CAD models are built upon the Boundary Representation (B-Rep), which is a graph-based representation for organizing geometric entities and their relations [Baumgart, 1974; Zeid, 2007]. Similarly, the analysis models are built upon the finite element representation, which describes spatial forms as discrete elements and provides for the mathematical framework that combines the element-wise response into the estimated response of the whole model [Brenner & Scott, 2008]. In the case of engineering databases, the entity-relation schema [Chen, 1976; Chen, 1980; Chen, 1981] formulates the framework required for describing the entities, attributes, and their relations within a domain of interest. By ensuring that the underlying representation is logically and mathematically rigorous, the consistency of the specific models can be ensured.

Multiple representations are studied in engineering design research to describe the functionality of technical artifacts. Each representation describes a different aspect of functionality and helps to answer different questions. For example, the function structure representation [Pahl et al., 2007] captures the aspect of *transformation* of material, energy, and information through an artifact in terms of input-output relations of flows passing through the artifact. This model supports reasoning about how one function can be decomposed into multiple sub-functions [Sridharan & Campbell, 2005; Pahl et al., 2007], the search for working principles for the sub-functions [Pahl et al., 2007], or the search for components that embody those working principles [Kurtoglu et al., 2005a]. Another model, called the affordance model [Maier & Fadel, 2001; Maier & Fadel, 2002; Maier, 2008], captures the aspect of *abilities* of the artifact to perform certain tasks. This model can answer questions about what a user can achieve by using the artifact in a given environment [Maier, 2008]. A third model, named the contact and channel model [Albers et al., 2008], captures the aspect of *parity* of working surfaces that achieve certain functionality together. This model supports the reasoning that two surfaces must interact in order to accomplish a function. The models mentioned above coexist in engineering design research since none is considered more correct than the others, and each offer the potential to answer some questions about artifact functionality. Due to this multitude of options, it becomes important to objectively evaluate the quality of function models and representations. Toward this end, one objective of this thesis is to develop a mathematical metric of measure the usefulness of models created with the function structure representation [Pahl et al., 2007]. The findings of this study are used to answer RQ-1.

2.2 Assessing Quality of Models and Representations in Engineering Design

The usefulness of a model lies in its ability to facilitate the design process by helping the designer to make decisions [Radhakrishnan & McAdams, 2005]. This quality, in turn, depends on the type and amount of information designers can represent and manipulate using the representation [Summers et al., 2001; Summers, 2005]. The quality of a representation is measured from multiple aspects. Consistency, completeness, and uniqueness are regarded as basic qualities of a representation [Rich & Knight, 1991; Winston, 1992; Luger, 2002]. In measuring how efficiently knowledge is represented, the term *expressiveness,* or expressive power, is used in artificial intelligence literature [Rich & Knight, 1991; Winston, 1992; Baader, 1996; Luger, 2002]. In turn, expressiveness is attributed to multiple aspects of the representation. Specifically, *expressive adequacy* [Baader, 1996] or *coverage* [Summers, 2005] refers to which knowledge elements are represented and which are not. *Distinction ability* defines the level of detail or resolution at which the representation can distinguish between closely resembling entities [Woods, 1983]. *Extensibility* means the provision for creating new elements for representing new situations [Summers, 2005]. The *types of elements,* such as object, relations, or attributes, that the representation is composed of are considered as another measure of expressiveness [Summers, 2005]. *Succinctness* means the compactness of the description of the concepts represented [Coste-Marquis et al., 2004]. Finally, *mappability* between two representations refers to the ability to translate one representation to the other and vice-versa [Baader, 1996]. For example, a definition of expressiveness of a formal language relies upon measuring if all strings in a first-order-logic-based language can be expressed in the language under examination and vice-versa [Baader, 1996]. This notion of expressiveness also appears in the relation between the formal languages within the Chomsky hierarchy [Linz, 2006]. In computation theory, the Chomsky hierarchy describes four major classes of formal grammars and corresponding formal languages in a containment relationship, where each lower-level grammar is a subset of (is contained within) a higher level grammar, in terms of the languages it can *express* [Mateescu & Salomaa, 1997; Hopcroft et al., 2001; Linz, 2006]. From higher to lower levels, these languages are named as regular languages, context-free languages, context-sensitive languages, and recursively enumerable languages [Mateescu & Salomaa, 1997; Hopcroft et al., 2001; Linz, 2006]. In terms of expressiveness, for example, since all regular languages can be generated with context-free grammars, but not all context-free languages can be expressed as regular expressions, the context free languages are said to be more expressive than the regular languages [Linz, 2006]. The same idea has been adapted to measure the expressive power of planning formalisms and systems [Nebel, 2000]. In engineering design, the same approach has been used to evaluate the expressiveness of a geometric and parametric representation schema [Summers, 2005]. In this case, the representation under examination is claimed to be at least as expressive as first order predicate calculus by showing that it can be translated into first order predicate calculus statements without any loss of information [Summers, 2005].

In this thesis, the quality of function structures is measured in terms of the amount of information the model reveals to the designer about the product it describes [Sen et al., 2010]. In this case, information content is a suitable surrogate to expressiveness, in the sense that a higher amount of information makes a model more expressive, which in turn potentially empowers the designer to make better decisions. Therefore, a metric of information content of function structures could help designers select the most expressive function structure out of many. Such a metric could help answer questions such as "How much information is *generated* by creating this function structure?", "How much information is *contained* in this function structure?", or "How much information in *transacted* when this function structure is exchanged between designers"?

Information content of a function structure is interpreted in terms of the number of questions that can be answered about the modeled artifact using the model. Alternately, it is interpreted as the number of questions that must be answered about the artifact so that the function structure can be reproduced by a person who is not observing its original version. These two interpretations are synonymous, as a higher number of answers required to reproduce the model indicates that more information about the product is encoded in the model to start with, making it harder (more answers) to decipher all the information stored in it. These two interpretations are used in evaluating the usefulness of the Functional Basis hierarchy and to measure the expressiveness of the graph-based representation. For example, between two models of the same product built on the same representation, the model that answers more questions about the product is considered more expressive, and the quantitative score of information content of that model is expected to be higher than the other. On the other hand, if two representations are used to model the same product, the model that requires fewer answers in order to be reconstructed is considered to be based on a more expressive representation, since a lower score of information content in the model itself indicates that less information was required to express the same product using that representation.

Using the above view of evaluating expressiveness of models and representations, this thesis evaluates the of function structures, the graph-based representation of Pahl & Beitz [Pahl et al., 2007], and the Functional Basis vocabulary [Stone & Wood, 2000; Hirtz et al., 2002], leading to answering RQ-1. This thesis also presents a means to enhance the expressiveness of the Functional Basis vocabulary by formally representing the meanings of the functions within the Functional Basis. The findings of this study are used to answer RQ-2. In the next chapter, different viewpoints and models of artifact functionality, particularly the function structures representations, are reviewed, and the Functional Basis vocabulary is discussed.

CHAPTER THREE: ARTIFACT FUNCTION AND ITS REPRESENTATIONS IN ENGINEERING DESIGN

This chapter presents a discussion on the major representations of artifact functionality proposed in previous research and their limitations. The function structure is discussed along with its potential and limitations toward being evolved into a formal representation. The Functional Basis vocabulary, the Design Repository, and their applications are discussed and their limitations are illustrated.

3.1 Definitions of Function in Previous Research

The functionality of technical systems has been defined by various authors from different viewpoints. Pahl & Beitz define a function as "the intended input/output relation of a system whose purpose is to perform a task" [Pahl et al., 2007]. Thus, this definition focuses on the inner workings of the system rather than the system's interaction with the surroundings. Ullman defines a function as "the desired output from a system" [Ullman, 1992]. This definition focuses on the *purpose* of a system, leading to the simplification that two systems have the same function as long as they produce the same output, irrespective of their inputs. Thus, in this viewpoint, an electric motor and an internal combustion engine have the same function, that of producing rotational kinetic energy.

Another set of function definitions can be found in design literature that attempt to include the system, its intended purpose, its actual behavior, and its surroundings in the definition. For example, Bobrow defines function as "the relation between the goal of a human user and the behavior of a system" [Bobrow, 1984]. Umeda and Tomiyama define function as “a description of behavior recognized by a human through abstraction in order to utilize it”. Chandrasekaran and Josephson give an ontological definition of function as a set of constraints that a new object (artifact) introduced to an ontological world must satisfy so that the effect of that object is manifested as a desired role [Chandrasekaran & Josephson, 1997]. Thus, these definitions are concerned about not only the system, but also the system’s effect when it is submerged into its surroundings.

Another viewpoint of functions is discussed in reverse engineering. Otto & Wood provide a systematic method of describing functions of an existing artifact through product tear down and cataloging [Otto & Wood, 2001]. While this approach uses the input-output transformation concept of Pahl and Beitz, the key difference is that this approach focuses on the *actual* functionality found through reverse engineering rather than the intended purpose. Despite their apparent differences, all of these definitions define function as a description of what the system does, either in terms of inner workings or as its effect on the environment, or both.

3.2 Representations of Function in Previous Research and their Limitations

Multiple representations of functions are studied in engineering design research. Notable examples include the Function-Behavior-Structure model (FBS) of Gero and colleagues [Gero, 1990; Gero & Kannengiesser, 2000; Gero & Kannengiesser, 2002], the Function-Behavior-State (FBSt) model of Tomiyama and colleagues [Umeda & Tomiyama, 1995; Erden et al., 2008], the Structure-Behavior-Function (SBF) model of Goel and colleagues [Bhatta et al., 1994; Goel & Bhatta, 2004], the representation of functions as the artifact’s roles and its effects on the environment proposed by

Chandrasekaran and colleagues [Chandrasekaran & Josephson, 1997; Chandrasekaran & Josephson, 2000; Chandrasekaran, 2005], and the Affordance-based view of functionality, proposed by Maier and Fadel [Maier & Fadel, 2001; Maier & Fadel, 2002; Maier, 2008]. Each of these models addresses the problem of describing artifact functionality from a different viewpoint, and potentially serves unique reasoning types. These models are reviewed below to illustrate the research gap. In Section 3.3 the graph-based function structure representation [Pahl et al., 2007] is separately reviewed, as it is used in this thesis to develop a new representation to address those gaps.

3.2.1 The Function-Behavior-Structure Model (FBS)

Gero and colleagues modeled functionality as a tripartite interaction between the *function*, *behavior* and *structure* of the system in the so called Function-Behavior-Structure model (FBS) [Gero, 1990]. A graphical description of this model is shown in Figure 3. Here functions (F) are the intended actions of the artifact, which are translated by the designer into a set of expected behaviors (B_e). Structure (S) refers to the specific form of a design solution, and behavior refers to the actual performance of the structure (B_s). The structure leads to the design description (D) at the end of the design process. In this figure, the arrows indicate the direction of transitions from one concept to another, as described by Gero [Gero, 1990].

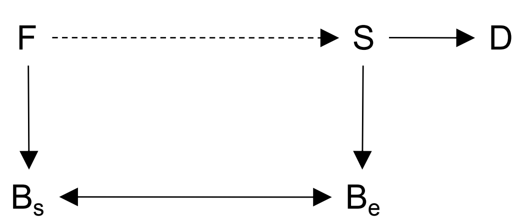


Figure 3: The Function-Behavior-Structure (FBS) model [Gero, 1990]

According to this model the steps in design are: *formulation, synthesis, analysis, evaluation, reformulation, and production of design description* [Gero, 1990]. This model recognizes that design is an iterative activity and provides for the *reformulation* of the design, where the designer analyzes the difference between B_s and B_e in the present cycle and assigns new B_e for the next cycle. The designer can assess the design only thorough the available representations, which describe the *situatedness* of the artifact in its environment [Gero & Kannengiesser, 2000; Gero & Kannengiesser, 2002]. Including the artifact, environment, and designer in a single representation increases its coverage, but at the cost of reasoning power. Specifically, the decision-making process of the designer falls under the purview of cognitive psychology, and are difficult to model due to the lack of a single theory of human cognition. As a result, though the FBS model covers some important entities and their interactions, it is not suitable for building a formal, computer-reasonable representation.

3.2.2 The Function-Behavior-State Model (FBSt)

A closely resembling model to the FBS model, called the Function-Behavior-State model (FBSt), is presented by Tomiyama and colleagues [Umeda & Tomiyama, 1995]. The notions of function and behavior are similar to those presented in the FBS model. The state of an artifact is the set of situations within the artifact and its surroundings that constitute a mode of operation of the artifact. For example, in the design of a photocopier machine, the two states of the drum subsystem are “to charge the drum” and “to discharge the drum”. This representation has been conceptually modeled as an entity-relation-attribute (ERA) framework [Umeda & Tomiyama, 1995]. However, no software implementation of this schema was found during the literature review for this thesis.

3.2.3 The Device-Centric and Environment-Centric Views of Function

Chandrasekaran and colleagues described function from two viewpoints, namely the *device-centric view* and the *environment-centric view* [Chandrasekaran & Josephson, 1997; Chandrasekaran & Josephson, 2000; Chandrasekaran, 2005]. The first view describes the artifact in terms of its *role* in serving the user. The second view captures the artifact's effect on the environment. The authors propose a preliminary ontology to describe functions. However, in addition to multiple definitions of the word function, the ontology includes the concepts and sub-concepts for the environment, the designer, the user, the intended and unintended behaviors, the modes of deployment by the user, and the structure of the artifact. Since each of these terms can be interpreted from multiple viewpoints, consistent definitions are difficult to achieve in many of these cases. For example, the authors identify six different meanings of the word *behavior* and point out the ambiguity they cause to the ontological description of the term [Chandrasekaran & Josephson, 2000]. Similar to the FBS model, this ontology has a broad scope that includes ambiguous terms, and has not been embodied into software applications.

3.2.4 The Affordance-based Model of Functionality

The affordance-based model tries to capture functionality in terms of what the designer can accomplish—both in positive and harmful ways—using an artifact situated in a given environment [Maier & Fadel, 2001; Maier & Fadel, 2002; Maier, 2008]. The affordance of a product is defined as "what the product provides, offers, of furnishes to a user or another product" [Maier, 2008]. The authors argue that since most products are intended to be used by a human user, directly or indirectly, a more comprehensive way to model the product is not by expressing what the product does, rather, by describing what the user can accomplish with the product. The authors identify multiple affordances that a product must offer in the course of its lifecycle, such as *afford manufacture, afford maintenance, afford human use, afford desired purposes but do not afford undesired purposes, afford sustainability,* and *afford retirement* [Maier, 2008]. From this viewpoint, different products may have various affordances for the same purpose. For example, both a chair and a briefcase afford the *sitting on* and *storing documents.* However, the chair affords *sitting on* better than *being used as a document storage*, while these affordances for the briefcase are in the reverse order. However, affordance is not yet mathematically formalized, and is not suitable for automated reasoning in its present state.

By contrast to the above examples the function structure representation is reasonably consistent and has a more focused scope that includes only the inner workings of the artifact. This representation is widely studied in engineering design research [Stone & Wood, 2000; Otto & Wood, 2001; Bohm et al., 2006; Pahl et al., 2007; Ulrich & Eppinger, 2008] and has been utilized in automated reasoning with artifact functionality [McAdams & Wood, 2002; Sridharan & Campbell, 2004; Kurtoglu et al., 2005a; Sridharan & Campbell, 2005; Bryant et al., 2006; Kurtoglu, 2007]. This representation is reviewed in detail next.

3.3 <u>Function Structures, their Applications, and their Limitations</u>

Function structures are graph-based representations of artifact functionality where nodes are *functions* or actions executed by the artifact and edges are *flows* or objects those actions. Three elementary concepts for constructing function structures, namely *functions, flows,* and *system boundary* are discussed in previous research along with preliminary rules for using those entities [Pahl et al., 2007]. A function is a transformative action that receives a set of flows and transforms that into another set. The flows are of three main types, namely, *material*, *energy*, and *signal* [Rodenacker, 1971; Pahl et al., 2007], while the functions are of two broad types: *main* and *auxiliary*. The main functions are critical to the overall functionality of the artifact, while the auxiliary functions are not. The system boundary defines the scope of the model by encompassing the graph entities (nodes and edges) that fall within the design scope. A preliminary vocabulary of graphical entities for constructing function structures is shown in Figure 4 [Pahl et al., 2007], where these entities are distinguished with their unique line fonts.

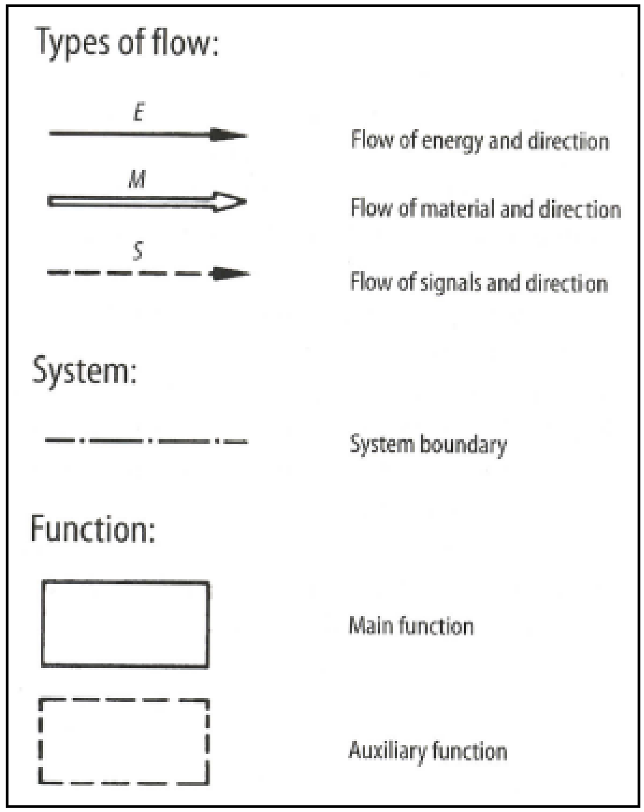


Figure 4: Preliminary vocabulary of graphical elements [Pahl et al., 2007]

Based on this vocabulary of symbols the function structure of a conceptual carpet tile packing machine is shown in Figure 5.

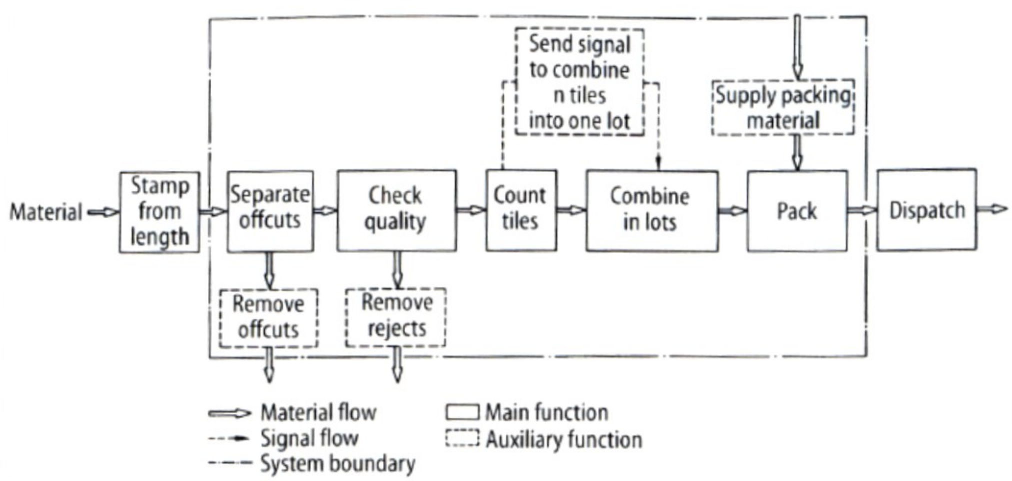


Figure 5: Function structure of a carpet-tile packing machine based on the graphical vocabulary of Figure 4

In this figure, the seven blocks drawn in solid lines are the main functions. The four blocks in dotted lines are the auxiliary functions, while the chain-dotted line represents the system boundary. Two functions, *Stamp from Length* and *Dispatch*, are not included in the system boundary, as overall function of the system is *packing the tiles* rather than cutting or dispatching them. Each function describes a transformative action between its inputs and outputs, which may be a change in flow types, such as the conversion of water (liquid) into ice (solid), or a change in flow parameters, such as changing the temperature of water through a cooler. In Figure 5, the two flows of *Material* (carpet tile) that enter and leave the function *Check Quality* are different, as they are in different states in terms of whether quality has been checked on them or not.

Ideally, if each instance of *Material* was identified with different symbols the function structure would be more expressive, as it would support additional reasoning such as counting the number of different states in which the flow exists in the machine. As seen in Figure 5, the flows are not uniquely identified, which is a limitation of this representation. Additionally, this function structure does not utilize any energy flow, implying that no energy is required for the machine's operation, which is inconsistent with the law of conservation of energy. This example illustrates that the construction of function structures is not formalized enough to ensure model consistency. In this thesis, the function structure representation is evolved to address these research gaps.

Function Structures within the Design Repository

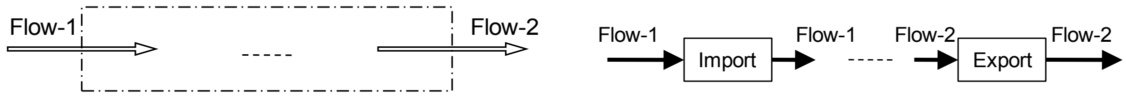
In this thesis, function structures stored in the Design Repository are investigated for studying their information content and representation formalism. These function structures are generally follow the graphical representation above, with two exceptions:

1. The material flows are shown using bold solid lines instead of double line fonts (Figure 6). In this thesis, bold solid line fonts are used for maintaining consistency with the source function structures.

(a) Double line font used by [Pahl et al., 2007]　　**(b) Thick solid single line font used in the Design Repository**

Figure 6: The modified line font for material flows used in Design Repository

2. The function structures in the Design Repository represent the system boundary with two functions—*Import* and *Export*—instead of using a line font, as shown in Figure 7. These models are built using the Functional Basis vocabulary (Section 3.4), which provides for these two verbs for this purpose. In Figure 7 (a) and (b), the inner details of the models are represented by the dotted line in the middle of each figure. In both cases, an arbitrary flow named Flow-1 is entering the system and another flow named Flow-2 is leaving the system. Despite the graphical differences, these two representations of entry and exit of flows are logically the same.



(a) Using the system boundary [Pahl et al., 2007]

(b) Using the *Import* and *Export* functions

Figure 7: The two representation of entry and exit of flows

In addition to the graphical representation, some critical issues with function modeling are discussed by Pahl and Beitz [Pahl et al., 2007], as follows.

1. *Causal dependencies between functions*, where the occurrence of one function is dependent on the occurrence of another. By establishing such dependencies, more expressive models can be constructed. For example, in order to represent that the *Check Quality* function in Figure 5 is not executed unless the *Separate Cutoffs* function has succeeded, a formal logical control needs to be included in the model. Such controls are not explicitly studied in previous research.
2. *Logical relations between functions*, which provide for modeling simultaneity or conjunction (AND), exclusivity or disjunction (OR), and negation (NOT) between functions. An example of conjunction is the simultaneous production of light and heat in an incandescent lamp discussed in Section 1.2. Formal modeling of such relations has not thoroughly investigated previously.
3. *Functional states of artifacts*, where an artifact executes a set of functions in one state and another set of functions in another state. An example of states was discussed using the charging and discharging states of photocopier drums in the FBSt model (Section 3.2.2).

4. *Relation between the artifact and the environment, including the user.* This area has been studied in the representations discussed in Section 3.2. Pahl and Beitz also provide a conceptual model, shown in Figure 8which distinguishes between the intended functionality, the positive and negative side effects, the interaction between the artifact and the user through a feedback loop, and the disturbing effects (noise) from the environment that are functionally undesired in the system. As discussed before, including these concepts potentially makes the model more comprehensive, but makes automated reasoning difficult.

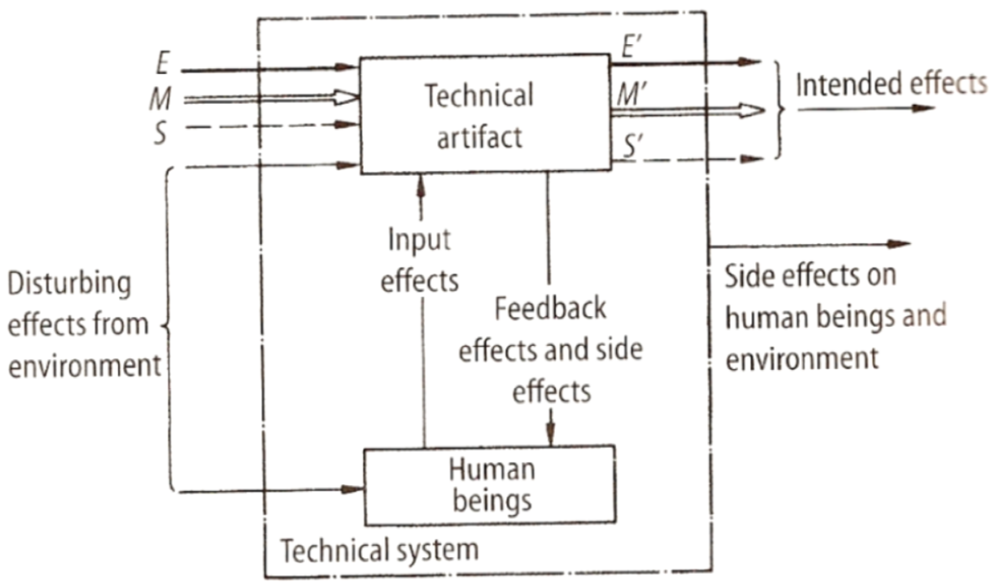


Figure 8: Interrelationships between the artifact and the user [Pahl et al., 2007]

5. *Intelligence, creativity, and other cognitive actions of designers,* and finally,
6. *The composition and decomposition of functions*, which is the *aggregation* relation between the overall function of the artifact and the sub-functions carried out by its components and subsystems. This aspect of function modeling is discussed in further detail in Section 3.3.2.

Function structures are used in engineering design research to represent the intended functionality of new products [Pahl et al., 2007] as well as to document the functionality of existing products through reverse engineering [Otto & Wood, 2001], as discussed next.

3.3.1 Function Structures in Reverse Engineering

Reverse engineering is a design approach where an existing artifact is studied to understand the design decisions and functionality through systematic product tear-down, the lessons from which are then applied to improve future designs [Otto & Wood, 2001]. In reverse engineering, function structures are used to understand and document the functionality of existing artifacts [Otto & Wood, 2001]. As mentioned in Chapter One, the function structures stored in the Design Repository are produced through this method. For example, the function structure shown in Figure 9 is obtained from the Design Repository, and it was constructed through reverse engineering of a commercial product, the Supermax Conair hair dryer.

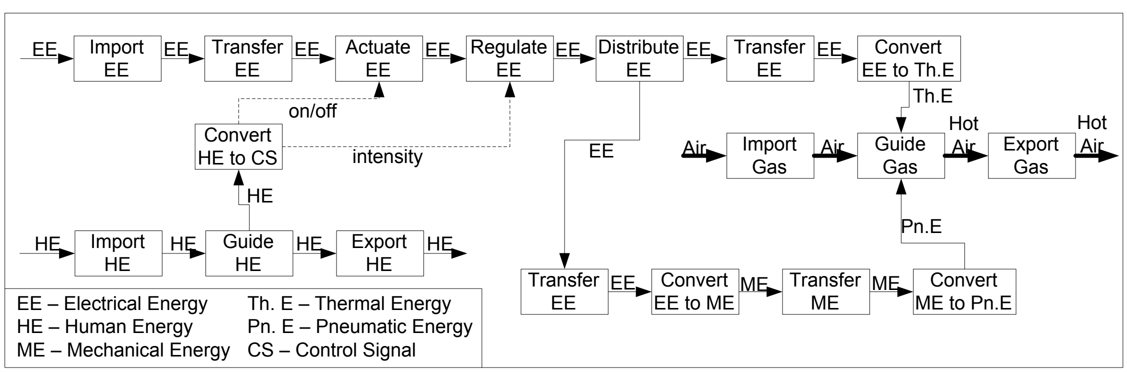


Figure 9: Function structure of the Supermax Conair hair dryer created through reverse engineering and stored within the Design Repository

In Figure 9, the abbreviated names of the flows are expanded in the bottom left corner of the figure. Since a function is assigned to each investigated component of the product, this model describes a component-wise account of functionality. For example, the function *Convert EE to ThE* represents the electric heater of the hair dryer, while the functions *Convert EE to ME* and *Convert ME to PnE* represent the electric motor and the fan impeller. In this manner, function structures allow for documenting the findings of a product tear down activity, and analyzing the functionality of the artifact later.

A similar approach to reverse engineering is adopted in engineering forensics, which is the investigative study of field-failures of engineering artifacts, with the aim of determining the root cause of the failures, so that they can be prevented in future designs [Noon, 2001]. An example of this approach is the investigation of failures of U.S. Army helicopters carried out by Collins and colleagues, which resulted in an early controlled vocabulary of functions [Collins et al., 1976]. Later, automated failure prediction of new designs was studied utilizing the failure data of products stored in the Design Repository [Tumer & Stone, 2001].

A benefit of reverse engineering is that once a suitably large archive of product design information is established it can be used to assist new designs. An example of such reuse is found in the Concept Generator tool [Bryant et al., 2006], which is a software application that suggests design concepts and components from the overall functional description of a new product, using the design information stored in the Design Repository. Similarly, the grammar rules for automated functional decomposition discussed in Chapter One utilize historical data of typical functional transformations in similar products stored in the Design Repository [Sridharan & Campbell, 2004; Sridharan & Campbell, 2005]. The Design Repository archive is of particular interest to this thesis, as the function structures stored there are used to evaluate the Functional Basis vocabulary. This archive and few of its applications are discussed in Section 3.4.2.

3.3.2 Function Structures in New Product Design

In new product design, function structures are used to expand the design space, and to look for solution principles and components. Pahl and Beitz recommend that function structures be used to express the functionality of the artifact in a solution-neutral form [Pahl et al., 2007]. Solution-neutral function structures allow for broader solutions search, as they represent the abstract functional description rather than suggesting form-specific solutions. For example, the intended overall functionality of a prime mover is represented in solution-neutral form as “Convert chemical energy to mechanical energy”, as shown in Figure 10.

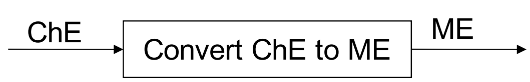
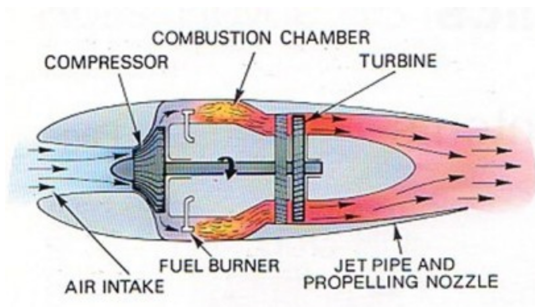


Figure 10: Solution-neutral description of the function of a prime mover

Due to its solution-neutral nature, this function structure can lead to multiple solutions that are different in working principles but satisfy the overall functionality, such as an internal combustion engine, a jet engine, and a battery-motor assembly (Figure 11).

(a)
Internal combustion engine



(b)
Jet engine

(c)
Battery-motor assembly

Figure 11: Three concepts that satisfy the function shown in Figure 10

Both the internal combustion engine and the jet engine convert the chemical energy of fuel to rotational mechanical energy of the output shaft, but employ different working principles: the internal combustion engine converts the thermal energy into the reciprocation of the piston that is in turn converted into rotation of the crankshaft, whereas the jet engine converts the thermal energy of the gas directly into the rotation of turbine blades. By contrast, the battery-motor assembly converts the chemical energy of the electrolyte in the battery into electrical energy, which is then converted to mechanical energy using the motor. However, the overall functionality of all three solutions is identical, which is described by the function structure in Figure 10. By contrast, if the function description included form-specific or solution-specific details, such as shaft, turbine, piston, or electrolyte, the solutions would be fewer in number, as all of the solutions in Figure 11 do not use all of those details. Interestingly, this function structure is also satisfied by a dynamite stick and a gun, both of which convert the chemical energy of explosives (dynamite, gun powder) into kinetic energy of projectiles (rocks, bullet).

Function structures can be composed or decomposed to represent different levels of resolution (granularity) of details in the model. In composition, the overall functionality achieved by a set of related functions and flows is expressed by a single function and a set of flows. For example, in Figure 9, the three functions that describe the conversion of EE to ME, the transfer of ME, and the conversion of ME to PnE can be composed into one function that accounts for the terminal conversions (EE to PnE), hiding the intermediate functions. The resulting composed function structure is shown in Figure 12, where the long block in bottom right represents the composed function.

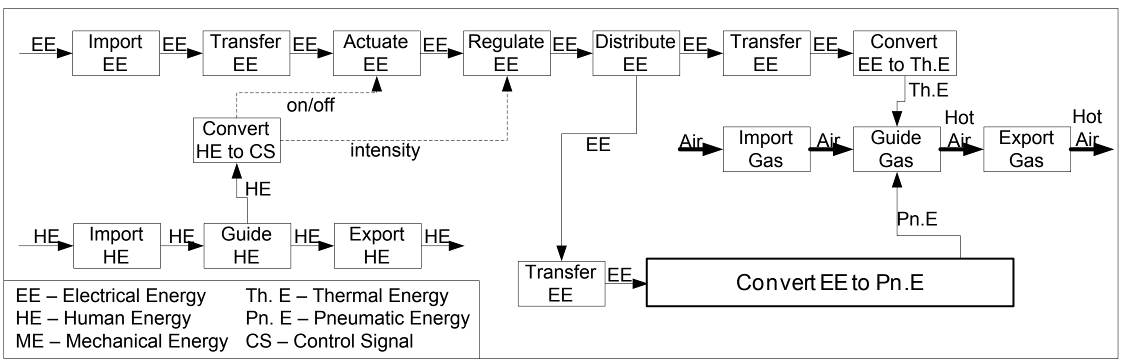


Figure 12: Partially composed version of the function structure shown in Figure 9

In this manner, a function structure can be successively composed into fewer functions that represent overall functionality and hide intermediate details. Ultimately, this process leads to the composition of the entire function structure into a single function and a set of flows that describes the overall functionality of the whole product. This model is called the *black-box* function structure [Pahl et al., 2007], which is shown in Figure 13 for the hair dryer product of Figure 9. A comparison between these two figures indicate that the decomposed version reveals more information about the product using eighteen functions and 24 flows, while the black-box describes the overall functionality using one function and seven flows.

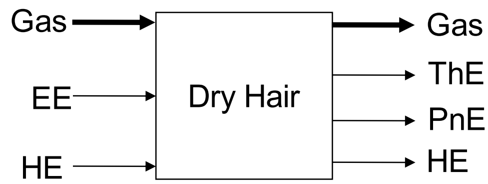


Figure 13: Black-box function structure for the hair dryer product

Converse to composition, a function structure can be decomposed to capture increasing levels of detail. Starting with a black-box or a relatively composed version of the model, individual functions can be broken down into multiple sub-functions and associated flows in such a way that the broken down version represents the original functionality as a whole [Pahl et al., 2007]. An example of decomposition would be to start with Figure 13 and gradually break that down to obtain Figure 9. In addition to revealing more details about the product, functional decomposition helps in the search for design solutions. These solutions can then be organized using a morphological matrix [Pahl et al., 2007] and combined into multiple working structures that represent different design concepts. Functional decomposition has also been used in directly identifying components that meet certain functions [Kurtoglu et al., 2005a; Kurtoglu et al., 2005b].

3.3.3 Limitations of Function Structures

As demonstrated above, the graph-based function structures representation is consistent in terms of the vocabulary of symbols. However, a few limitations need to be addressed in order to evolve it into a more formal representation.

1. Beyond the three flow types and two function types the representation does not provide for any classification of entities.
2. The decision of whether a function is main or auxiliary is subjective and dependent on the designer.
3. The terms used in creating a function structures (Figure 5) are drawn from the natural English dictionary, which makes this model humanly interpretable yet not suitable for automated reasoning, unless the semantics of those words are formally captured.
4. Beyond the general rule that a function transforms the input flows to output flows, there are no specific rules available for controlling how the flows and functions must be joined together in a function structure. This limitation was discussed in Section 1.2 using Figure 2, where two ways of representing the conversion of electrical energy to light and heat in an incandescent lamp were compared. Another example was cited in this chapter using Figure 5, which violates the law of conservation of energy due to the lack of formalism of constructing the models. Additionally, there is no formal guideline to choose the level of detail (decomposition) in a model or the level of specificity of the functional terms. In the present form of the function structure representation, these decisions are informal and human-dependent. As a result, function structures are subject to representational inconsistency.

The issues of classification of functions and flows (point 1 above) and that of a standard vocabulary of terms (point 3 above) have been studied in engineering design research over the past three decades [Collins et al., 1976; Kirschman & Fadel, 1998; Szykman et al., 1999; Stone & Wood, 2000; Hirtz et al., 2002]. The Functional Basis mentioned in Chapter One is one outcome of these research efforts, which are discussed in the next section. However, the issue of controlled connections between functions and flows (point 4 above) has not been specifically addressed in previous research. This research gap is addressed in this thesis by developing a formal description of functions.

Despite these limitations, function structures are widely used in engineering design research, as they are built upon a simple principle—flow transformation—which can be mapped to the principles of conservation of mass and energy, making the representation fundamentally robust. They are free from the relatively ill-defined aspects of user-interaction or the environment, making them simple to interpret. They can be represented in solution-neutral ways, making them sufficiently abstract for product modeling. Due to their graph-based representation, function structures can potentially be formalized to support graph-theoretic reasoning, and ultimately evolved into a formal representation to support automated reasoning about product functionality. Due to these reasons, function structures are selected for further formalization.

3.4 Function Vocabularies, the Functional Basis, its Applications and Limitations

As discussed in Section 2.2, consistency is a critical requirement for representations in order to support unambiguous interpretation and reasoning on the models. As a first step toward establishing consistency in function structures, controlled vocabularies are explored in engineering design research [Collins et al., 1976; Kirschman & Fadel, 1998; Szykman et al., 1999; Stone & Wood, 2000; Hirtz et al., 2002]. Collins and colleagues identified 46 elemental functions within mechanical components from failure studies of U.S. army helicopters [Collins et al., 1976]. Based on this vocabulary, Kirschman and Fadel described functions within consumer products using four groups: *motion*, *control*, *power/matter*, and *enclose* [Kirschman & Fadel, 1998]. Keuneke identified four function keywords that describe the functionality of mechanical artifacts, namely, *ToMake, To-Maintain, ToPrevent, and ToControl* [Keuneke, 1991]. A separate research at the National Institute of Standards and Technologies (NIST) resulted in a function vocabulary for consumer products, referred here as the NIST vocabulary [Szykman et al., 1999]. The Functional Basis was derived from the NIST vocabulary in a joint effort between industry and academia at the Missouri University of Science & Technology [Stone & Wood, 2000], and reconciled in 2002 [Hirtz et al., 2002]. This vocabulary is discussed below.

3.4.1 The Functional Basis

The Functional Basis contains 53 verbs and 45 nouns organized in a three-level hierarchy, as shown in Table 1 and Table 2. The verbs are meant for use in the functions, while the nouns are meant for flows in the function structures. For ease of reference, these two tables are repeated below.

Table 1(repeat): Functional Basis verbs hierarchy [Hirtz et al., 2002]

Primary	Secondary	Tertiary
Branch	Separate	Divide
		Extract
		Remove
	Distribute	
Channel	Import	
	Export	
	Transfer	Transport
		Transmit
	Guide	Translate
		Rotate
		Allow DoF
Connect	Couple	Join
		Link
	Mix	
Control Magnitude	Actuate	
	Regulate	Increase
		Decrease
	Change	Increment
		Decrement
		Shape
		Condition
	Stop	Prevent
		Inhibit
Convert	Convert	
Provide	Store	Contain
		Collect
	Supply	Supply
Signal	Sense	Detect
		Measure
	Indicate	Track
		Display
	Process	
Support	Stabilize	
	Secure	

Table 2 (repeat): Functional Basis nouns hierarchy [Hirtz et al., 2002]

Primary	Secondary	Tertiary
Material	Human	
	Gas	
	Liquid	
	Solid	Object
		Particulate
		Composite
	Plasma	
	Mixture	Gas-Gas
		Liquid-Liquid
		Solid-Solid
		Solid-Liquid-Gas
		Colloidal
Signal	Status	Auditory
		Olfactory
		Tactile
		Taste
		Visual
	Control	Analog
		Discrete
Energy	Human	
	Acoustic	
	Biological	
	Chemical	
	Electrical	
	Electromagnetic	Optical
		Solar
	Hydraulic	
	Magnetic	
	Mechanical	Rotational
		Translational
	Pneumatic	
	Radioactive/Nuclear	
	Thermal	

The left column in each table above is called the primary level, with the middle column being the secondary level and right column being the tertiary level. The primary level is considered a *higher* level than the secondary, and the tertiary level is considered a *lower* level than the secondary. As noted in Chapter One, the functionality of an artifact can be described by forming predicates using the verbs and nouns listed in these tables. For example, the function of an electric motor can be described as “convert electrical energy to mechanical energy”. The hierarchy of terms is used to control the specificity of function description. For example, to describe the output energy from the motor more specifically, the secondary term *mechanical energy* can be replaced with a suitable tertiary term that is a taxonomical child of the term *mechanical energy* – in this case, *rotational mechanical energy*. Conversely, if a lower resolution of description is required, the primary term *energy* can be used, hiding the details that the energy is *rotational* (tertiary), or even *mechanical* (secondary).

The Functional Basis was incrementally developed by examining the functionality of existing products through systematic reverse engineering, and including the functional terms (verbs and nouns) that were necessary to describe the newly found functions [Bohm et al., 2003; Bohm & Stone, 2004; Bohm et al., 2005; Bohm et al., 2006]. The findings of these reverse engineering studies are recorded in the Design Repository, which later emerged as a web-based archive for storing design information of products. This repository is discussed next.

3.4.2 The Design Repository

The Design Repository is a web-based archive of design information of consumer electro-mechanical products obtained through reverse engineering. The functional information of these products are captured by first tearing down the product using the protocol of Otto and Wood [Otto & Wood, 2001], followed by cataloging the function of each component or sub-system using the Functional Basis vocabulary, and finally, connecting the functions by tracking the flows within the product [Kurfman et al., 2000], creating a function structure.

There are 130 total products available in the repository[2]. Graph-based function structures are available for approximately half of these products, while function-component matrices and assembly structures are available for all. Some of the information stored in the repository is not directly related to functionality, such as geometric dimensions, material, failure modes, and manufacturing process. Figure 14 shows a screenshot of the Design Repository webpage, illustrating the data stored for a specific component (heating coil frame) of a specific product (Supermax hair dryer).

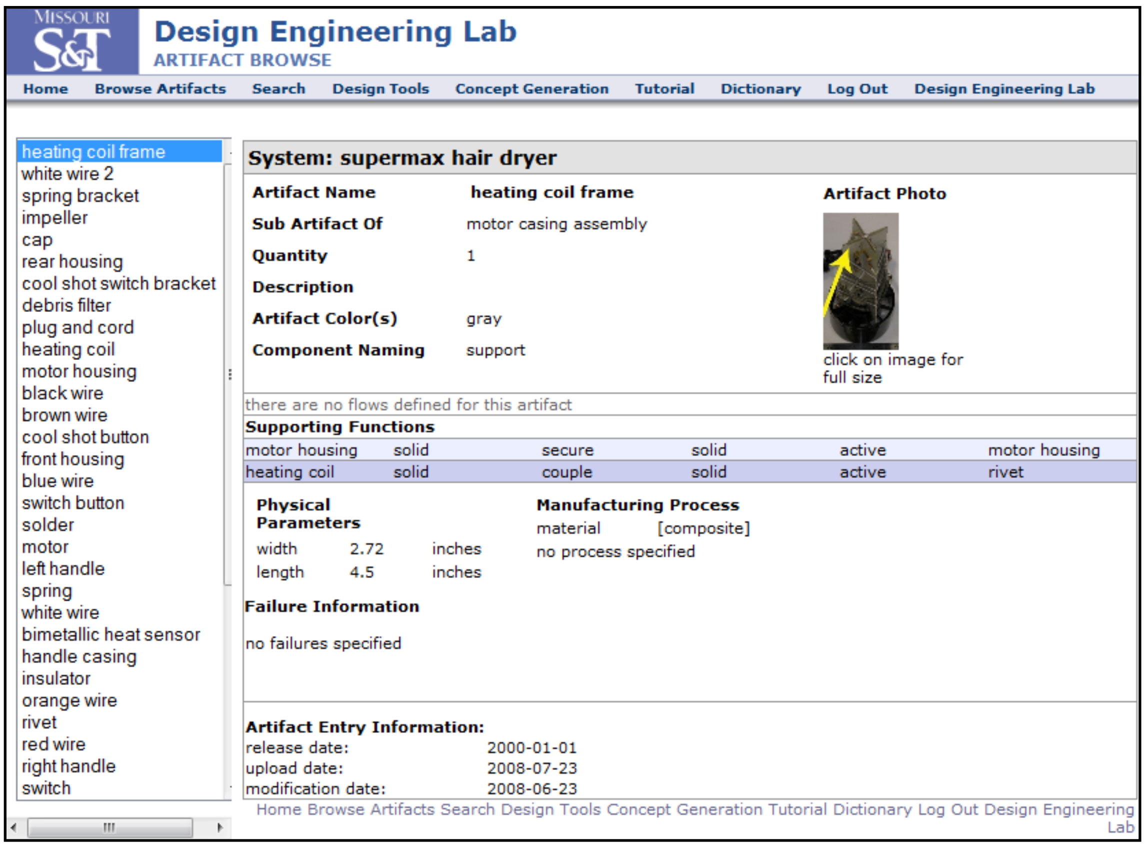


Figure 14: The artifact browser in the Design Repository showing the heating coil frame of the Supermax hair dryer

[2] http://repository.designengineeringlab.org/, accessed on January 27, 2009

The functions of the components and subsystems are captured in the function list using the Functional Basis vocabulary. For example, in the case of the heating coil the supporting function is listed as to couple solid to solid, where *couple* and *solid* are Functional Basis terms. Additionally, graphical function structures, such as Figure 9, and component-function matrices that store the function of each component in the product, are included in the database. Unfortunately, these graphical models are static and do not directly support computational reasoning. Previous studies indicate that in some cases inconsistencies exist between the function structure and the component-function matrices [Caldwell et al., 2008].

The verbs and nouns in the Functional Basis vocabulary are explicitly defined in textual form within the Design Repository's dictionary page. Figure 15 shows the definition of the function *Import,* which is highlighted with the rectangle.

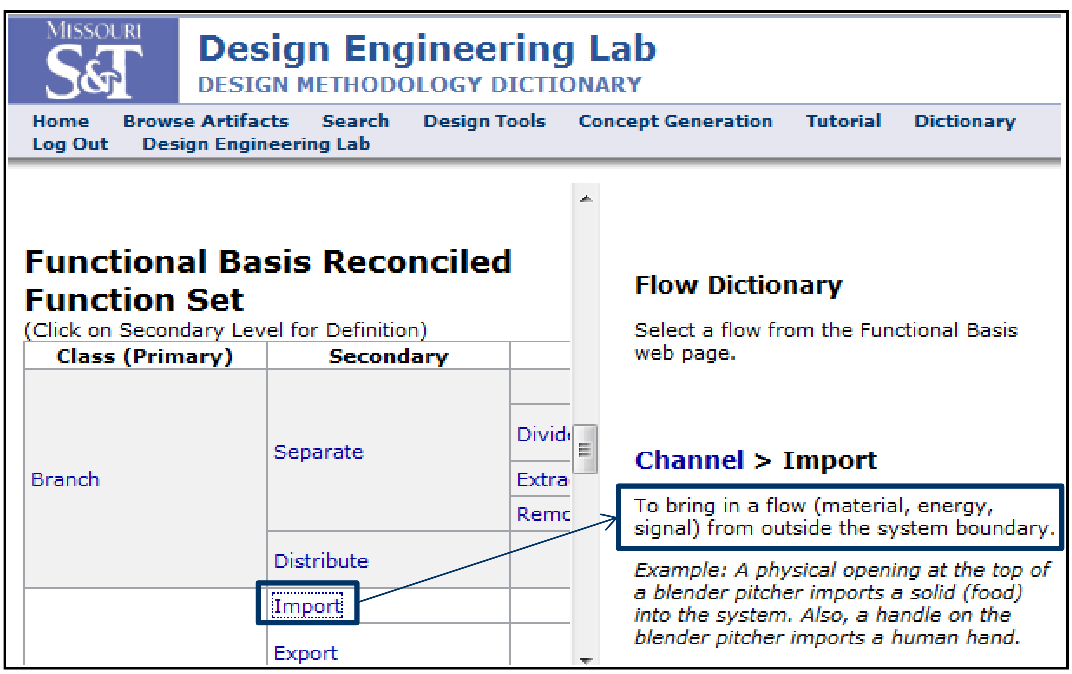


Figure 15: Definition of the Functional Basis terms in the Design Repository

The definitions of all the verbs in the vocabulary are provided in 0. Since the vocabulary represents these terms only by their names rather than by formally capturing their definitions, the definitions are not used to control the construction of the function structures. In this thesis, an evolved representation of functions is presented that captures this missing semantic information.

The Functional Basis and the Design Repository are widely studied in design literature, and have been utilized in several academic applications. For example, the Concept Generator tool suggests component layouts for new design concepts using the component-function matrices of similar products stored in the Design Repository, similar to an automated morphological analysis [Bryant et al., 2006; Vucovich et al., 2006]. Similarly, a failure analysis tool, named the Function-Failure Design Method (FFDM), has been designed to predict potential failure modes in the conceptual design phase of new designs based on the archived failure history of components performing similar functions [Tumer & Stone, 2001; Arunajadai et al., 2002; Stone et al., 2005]. This vocabulary has also been used for analyzing functional similarity between products, which relies upon identifying similar occurrences of function-flow pairs between two function structures [McAdams & Wood, 2002]. The Functional Basis has been extended to formulate a vocabulary of standard mechanical components [Kurtoglu et al., 2005a; Kurtoglu et al., 2005b]. Finally, the Functional Basis and Design Repository have been used in automated decomposition of function structures [Sridharan & Campbell, 2004; Sridharan & Campbell, 2005], which is discussed in the next section.

3.4.3 The Functional Decomposition Grammar Rules

The grammar rules for automated functional decomposition utilize the historical product design information within the Design Repository to identify trends in functional transform in typical electro-mechanical products [Sridharan & Campbell, 2004; Sridharan & Campbell, 2005]. Once the trends are established, the rules apply those trends on the black-box function of a new design, decomposing that to represent new concepts. A total of 69 rules have been reported in previous literature, which were obtained from investigating trends in 32 different products within the Design Repository [Sridharan & Campbell, 2005].

The software implementation of these rules operates by first identifying locations within a function structure where a rule can be applied. These locations are called *active centers* [Sridharan & Campbell, 2004; Sridharan & Campbell, 2005]. For example, the incoming flows to the black-box function structure are considered as active centers such that the function *Import* is applied to them. This rule is a consequence of the modeling approach in the Design Repository, where all incoming flows are introduced to the system through the function *Import,* as discussed in Section 3.3. After applying this rule, each imported flow constitutes a new active center, creating provision for directing it to a sub-function. This transition is shown in Figure 16.

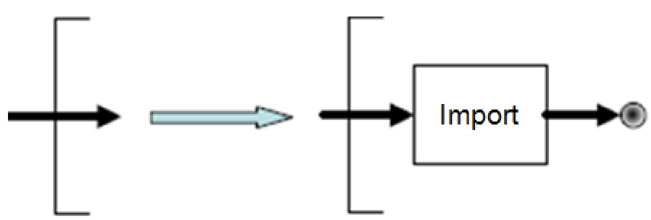


Figure 16: Active center produced by the rule that applies the function *Import* to an incoming flow [Sridharan & Campbell, 2005]

In this figure the left side indicates an incoming flow to the black-box function. The right side shows the change in the function structure due to applying the rule that created an instance of the function *Import* and the also created a new active center (marked by the encircled dot). The rule that is applied to the new active center depends on the previously established trends. For example, the typical function structures within the Design Repository indicate that once electrical energy is imported to a system, it is transmitted to a switch, followed by being actuated by the switch. Therefore, considering that the incoming flow in Figure 16 was electrical energy, the second rule replaces the instance of *Import* with the function sequence *Import-Transmit-Actuate,* as shown in Figure 17.

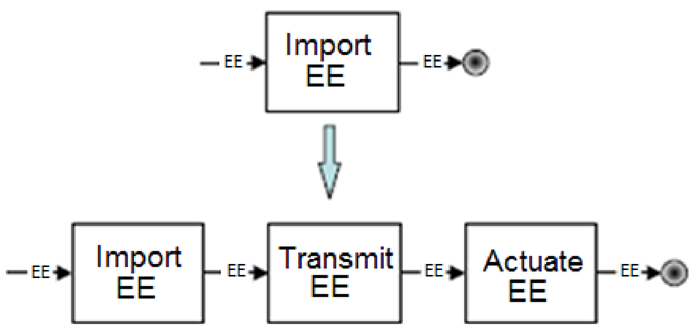


Figure 17: The effect of a rule that ensures that electrical energy is transmitted to a switch and actuated by it, after being imported to the system [Sridharan & Campbell, 2005]

In this figure, the top and bottom parts show the function structure before and after applying the second rule. Notably, this transformation also produced a new active center to which a third rule can be applied, resulting into further decomposition of the model. At the end of this process, a concluding rule assigns the *Export* function to the open-ended flows, thus terminating the decomposition process.

This automated decomposition approach has been experimentally tested several times, and a software application has also been created that automates the process of choosing the appropriate rule for an active center and applying it. However, this approach has some limitations, which are discussed in the next section.

3.4.4 Limitations of the Functional Basis and its Related Research

Four critical limitations of the Functional Basis and its applications, including the Design Repository, were discussed in Chapter One. These are: (1) the lack of formalism for building connections between functions and flows, (2) the lack of expressive power of the vocabulary terms, (3) the bottom-up research approach that cannot guarantee adequacy of the vocabulary, and (4) the lack of uniform usage of the hierarchy in function structures. These limitations have been discussed in the previous sections during discussing the function structure representation and the Functional Basis vocabulary. Here the first limitation, the lack of connection formalism, is discussed with an example to illustrate the extent of inconsistency in the function structures in the Design Repository. Figure 18 shows the hair dryer function structure of Figure 9, with the inconsistencies pointed out in italicized texts, which are discussed below.

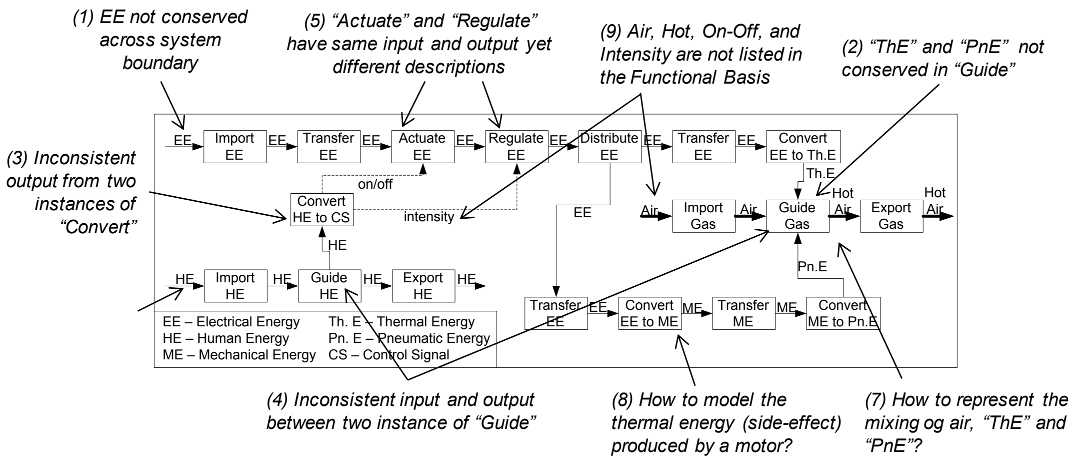


Figure 18: Consumer hair dryer function structure in the Design Repository

1. The flow of *EE* is not conserved across the whole model. It can be traced within the model till it is converted into *ThE* and *PnE*, which are then added to an instance of *Guide*. Beyond this point, no energy flow can be traced. Though it can be humanly reasoned that the thermal and pneumatic energy flows are added to the gas flow producing a hot stream of air, it is not consistent, as in the present state, the model violates the law of conservation of energy.
2. For the same reason, the function *Guide* at the right side of the model also violates the law of conservation of energy.
3. The function *Convert* that converts *HE* to *control signals*, has a different number of output flows than the other instances of *Convert* in the model. The lack of formalism lies in the fact that there is no restriction on how many flows can be produced under a conversion action. Further, the conversion of energy into signals is as a violation of the law of conservation, unless it is explicitly stated that signals are equivalent to energies.

4. A similar inconsistency of number of input and output flows can be seen between the two instances of *Guide.*
5. Conversely, the functions *Actuate* and *Regulate* have the same number and type of flows as input and output, yet they have different names and purposes. As the functions are not uniquely identifiable through their associated flows, the names are the only means of identifying the functions. Yet, the definitions of those names are not included in the formal representation.
6. It is not clear if human interaction should be modeled as *human material (HM)* or *human energy (HE)*, as both would suffice in this case.
7. There is no consistent protocol about how to model mixing of material and energy flows. The model shows that the flows of *ThE* and *PnE* are mixed to the gas using the function *Guide.* The definition of *Guide* in 0 does not allow such actions, rather the function *Mix* does.
8. The function structure shown here does not provide for modeling the side effects, such as the heat (*ThE*) produced by the motor as it converts *EE* to *ME*.
9. Finally, the model contains several non-Functional Basis terms, such as *on-off, intensity, air* and *hot.*

The above mentioned inconsistencies arise from the lack of a rigorous formalism to control the topological connections between the functions and the flows. Additionally, an empirical study shows that the three hierarchical levels of the vocabulary are not used uniformly within the Design Repository [Caldwell et al., 2008]. Eleven randomly chosen function structures were selected for this study from the Design Repository, which contained 115 products at that time. The terms used in those function structures were categorized according to the hierarchy of the vocabulary. The counting revealed that the models use up to 25% non-Functional Basis terms for functional description, indicating that the vocabulary is inadequate for its purpose. Additionally, it was found that above 90% of the Functional Basis terms used in those models are drawn from the secondary level, indicating that the hierarchical organization of term in the vocabulary is not useful.

Due to these limitations, it becomes important to objectively examine the usefulness of the Functional Basis vocabulary, specifically, its hierarchical organization. In order to address this task, first a mathematical metric of usefulness is required. This metric is developed in the next chapter, and applies to the Functional Basis in Chapter Five.

CHAPTER FOUR: THE INFORMATION THEORY OF FUNCTION STRUCTURES

This chapter develops the metrics of usefulness of function models and function vocabularies. First a general form of the metric is derived from the principles of Information Theory. Next, two different metrics are developed for assessing two aspects of usefulness of the function structures. A practical interpretation of the metric is presented as the number of questions that can be answered about the described artifact using the function structure.

4.1 The Basics of Information Theory

Information Theory, originally developed in the context of communication, provides a mathematical measure for information content of a message produced by a discrete source [Hartley, 1928; Shannon, 1948]. In this context, a message constitutes of a stream of events that carries some information. Conversely, an event is a unit block of information in a message. The source is discrete if the events occur as distinct units of the message with no provision for partial occurrence. The source is linear if the events are produced sequentially. The events in the message are selected from a predefined, finite list of allowed events or controlled vocabulary, where each event has a known probability of occurrence in the message. Under these premises, the information content of a single event in a message is given by [Shannon, 1948]:

$$I_i = -K \cdot \sum_{j=1}^{n} p_j \cdot \log_b p_j \qquad \text{Eq.1}$$

where:

I_i is the information content of a single event in the message

K is a constant for scaling the information content between different information sources

n is the size of the finite predefined vocabulary

j is the counter of the elements in the vocabulary

p_j is the probability of the j-th element of the vocabulary occurring in the message

b is a positive integer, the base of the logarithm

The constant K scales the quantity inside the summation sign and assumes different values for different sources. Hence K can be used to compare information content across different design representations. This comparison is reserved for future work. In this thesis, only a single representation, the function structure, is studied. Therefore K is arbitrarily defined as unity. The premises of Information Theory are next mapped to the features of function structures to justify the use of the information metric.

4.2 <u>Correspondents of Information Theory in Function Structures</u>

Function structures, such as those stored in the Design Repository, can be viewed as the union of two non-intersecting sets: the set of functions and the set of flows. Each of these sets consists of discrete elements, as the individual function and flow instances are discrete entities in the model. Thus, the model, as a whole, is a discrete domain. Though the model contains all the elements in a graphical representation, for the sake of computing information content the elements are considered sequentially, making the model linear. Further, the elements of the model shown in Figure 9 are ideally drawn from a specific level of the Functional Basis (Table 1and Table 2), which are finite vocabularies of predefined sizes. The probability of occurrence of terms in the Functional

Basis and their dependencies have been studied in previous research [Kurtoglu et al., 2005b; Sridharan & Campbell, 2005]; however, so far no conclusion has been generally accepted. Therefore, a uniform distribution of independent probabilities of functions and flows over the respective vocabularies is assumed here. Under these assumptions, a function structure behaves like a linear discrete source of information. These assumptions are formally stated below.

Assumptions:

1. *A function structure is a linear source, i.e., the functions and flows are encountered by the observer in a sequential fashion.*
2. *The probability of occurrence of verbs and nouns of the Functional Basis in a function structure is uniformly distributed over the respective vocabularies*
3. *The probability of occurrence of the verbs and nouns of the Functional Basis in a function structure is independent of the other verbs and nouns used in the model*

Based on the above assumptions, the concepts of message, event, source, vocabulary and probability distribution are mapped between Information Theory and the corresponding concepts in function structures in Table 3.

Table 3: Correspondents of Information Theory in function structures

Concepts in Information Theory	Correspondents in function structures
Message	The set of functions and the set of flows in a function structure
Events	Individual functions and flows in a function structure
Discrete, linear source	The function structure
Finite predefined vocabulary	The list of verbs and nouns in a specific level of the Functional Basis

Concepts in Information Theory	Correspondents in function structures
Probability distribution of events over the vocabulary	Assumed uniform over the Functional Basis (Assumption 2 above)

4.3 <u>Information Metric for Functional Elements – General Form</u>

Under Assumptions 2 and 3, and setting K = 1, Eq.1 undergoes the following change.

$$I_i = -K \cdot \sum_{j=1}^{n} p_j \cdot \log_b p_j = -(1) \cdot \sum_{j=1}^{n} p \cdot \log_b p = -n \cdot p \cdot \log_b p = -\log_b p = \log_b (n)$$

Eq.2

where:

I_i is the information content of a single event

j is the counter of the elements in the vocabulary

p_j is the probability of the j-th element of the vocabulary occurring in the message

p is the uniform probability of all elements in the vocabulary, under Assumption 2

$n = 1/p$ is the size of the vocabulary

b is a positive integer, the base of the logarithm

Since Eq.2 is obtained by applying the assumptions that map the premises of Information Theory on to the function structures, Eq.2 represents the information content per element in a function structure. The base of logarithm b is essentially a scaling factor for I_i. As shown in Eq.3, changing the base from b to c scales I by a constant, $\log_b(c)$.

$$\log_b(x) = \log_c(x) \times \log_b(c) \qquad \text{Eq.3}$$

Therefore, the base can be arbitrarily chosen, as long as the choice is consistently maintained for all computations. Here, the value 2 is selected as it provides an intuitive practical interpretation of the metric, as will be discussed in Section 4.5. The choice of the base determines the unit of information, which, for b=2, is bits [Shannon, 1948].The unit information content per element of the function structure is thus simplified to:

$$I_i = \log_2(x) \text{ bits/element} \qquad \text{Eq.4}$$

where:

I_i is the unit information, that is, information per element of the model

x is the number of terms in the vocabulary from which the element is drawn

For y distinct elements in the function structure, the total information content is given by:

$$I = \sum_{i=1}^{y} I_i = \sum_{i=1}^{y} \log_2(x) = y \cdot \log_2(x) \text{ bits} \qquad \text{Eq.5}$$

where:

I is the information content of the all elements in the message

I_i is the information content of the i-th element of the message

x is the number of terms in the vocabulary from which the elements are drawn

y is the number of elements in the functional model

Eq.4 and Eq.5 are defined as the general metrics of information content of function structures in this research. This measure of information content has previously been used to measure size complexity of engineering models [Summers & Shah, 2003; Summers & Ameri, 2008].

4.4 <u>Information Content of Functions and Flows in Function Structures</u>

Each element (function or flow) of a function structure contributes to the information revealed by the model to the designer, since by the removal of an element, or a set of elements, the model captures less information about the described product than the initial version. Thus, the information content of the whole function structure is expected to be a function of the set of functions and flows, of the form $I_{FM} = f(V,N)$, where V and N are the respective sets of functions (verbs) and flows (nouns) in the model. The function *'f'* describes how the information from the function and flow instances contributes to the total information content. In this thesis, two possible definitions of *'f'* are identified, namely, *element-wise* and *combined* information content. Both definitions are discussed in this section.

It is noteworthy that the topology of a function structure, meaning the connectedness of the functions with the flows, also contributes to its informativeness, as the model becomes more informative to the designer when the functions and flows are arranged in the topological arrangement rather than in a flat list. However, since the Functional Basis contains vocabularies of only verbs (used in the functions) and nouns (used in the flows), and does not provide for any formalism for the topological construction of function structures, this element of information is not counted in the metrics in this chapter. The computation of topological information content is presented in Chapter Six, where the total information content of the function structures is computed using two approaches.

4.4.1 Element-wise Information Content of Function Structures

The element-wise information content of a function structure is the algebraic sum of information contributed by the individual elements. As explained earlier, there is no intersection between the sets of functions and flows in typical function structures, as a function cannot serve also as a flow, and the vice-versa. Therefore, the contributions by the sets of functions and flows are to be algebraically summed in order to compute the total information content of the model. Based on this argument, if the number of verbs and nouns in the respective vocabularies are given by x_V and x_N, and the number of functions (verb instances) and flows (noun instances) in a specific function structure is given by y_V and y_N, the following metrics are obtained from Eq.5.

Definitions:

1. *Information content of functions in a function structure:*

$$I_V = y_V \cdot \lceil \log_2 (x_V) \rceil \text{ bits} \qquad \text{Eq.6}$$

2. *Information content of flows in a function structure:*

$$I_N = y_N \cdot \lceil \log_2 (x_N) \rceil \text{ bits} \qquad \text{Eq.7}$$

3. *Information content of the whole function structure (element-wise):*

$$I_{FM} = I_V + I_N = y_V \cdot \lceil \log_2 (x_V) \rceil + y_N \cdot \lceil \log_2 (x_N) \rceil \text{ bits} \qquad \text{Eq.8}$$

In the above three equations, and the other equations of information metric in this thesis, the symbol $\lceil \ \rceil$ is used to round the number inside the symbol to its nearest higher integer. This symbol reads as "the ceiling of", in accordance with the *ceiling* function in the C programming language [Kernighan & Ritchie, 2004]. The necessity of this rounding operation is explained in Section 4.5.

4.4.2 Combined Information Content of Function Structures

The combined information content of a function structure is based on a combined vocabulary obtained by concatenating the individual vocabularies of verbs and nouns. If the number of verbs and nouns in the respective vocabularies are given by x_V and x_N, and the number of functions (verb instances) and flows (noun instances) in the function structure is given by y_V and y_N respectively, the following metrics are obtained from Eq.4 and Eq.5.

Definitions:

1. *Combined information content per element:*

$$I_{i(N+V)} = \lceil \log_2(x_V + x_N) \rceil \text{ bits/element} \qquad \text{Eq.9}$$

2. *Combined information content of the whole model:*

$$I_{N+V} = (y_V + y_N) \cdot \lceil \log_2(x_V + x_N) \rceil \text{ bits} \qquad \text{Eq.10}$$

The combined metric is built upon a hypothetical merged vocabulary of size $(x_V + x_N)$. This merger eliminates the effect of size difference, if any, between the vocabularies of verbs and nouns, as the combined metric computes information content based on the enlarged, unified vocabulary. In the case of the Functional Basis, the sizes of the verb and noun vocabularies at the secondary level are comparable: 21 verbs and 20 nouns. However, at the primary level, the sizes are significantly different: eight verbs and three nouns. In such cases, by using the combined metric, both functions and flows are assigned the same weight carried by the size of the vocabulary. Further, the factor $(y_V + y_N)$ accounts for the total number of elements, functions and flows, in the function structure. Thus, if a function structure has significantly more functions than flows, or the vice-versa, the element-wise metric would compute a significantly different values for these two elements, while the combined metric would compute a unified value considering equal weights for the functions and the flows carried by the numbers of their instances in the model. Thus, the combined metric is expected to be neutral to the choice of function structures and the relative densities of functions and flows in them for the experiments discussed in Chapter Five. These two metrics, the element-wise and the combined, are used in this thesis to measure information content of function structures.

4.5 Practical Interpretations of the Information Metrics

A practical interpretation of information content of a function structure is the number of questions that must be answered about the model in order to be able to reconstruct the model without directly viewing it. Conversely, information content can be viewed as the number of questions that the function structure answers about the product to the designer. These interpretations agree with the practical notion of information, where more answers about a domain of interest generally imply more facts being obtained about the domain. This interpretation is explained with an illustration in Figure 19.

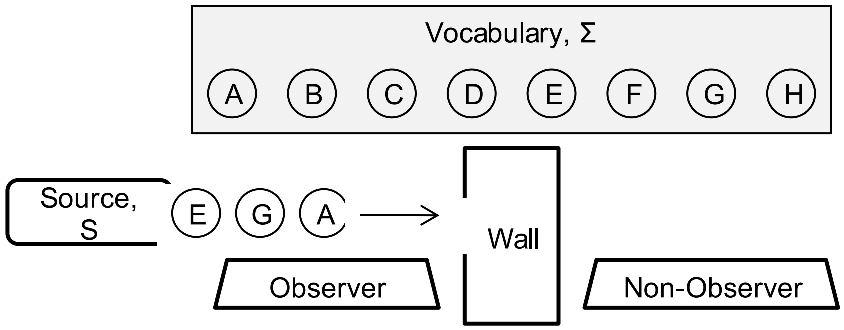


Figure 19: Illustrative scheme of communication between designers [Sen et al., 2010]

Here, a finite vocabulary Σ, containing elements A through H, is used to describe a model M, that uses one instance of each of elements E, G, and A. These elements are analogous to the functions and flows in a function structure, while the vocabulary Σ is analogous to the Functional Basis function set and flow set. A designer, who is observing the function structure, is transmitting information about the individual elements (functions and flows) to another designer, who cannot view the function structure. With each element transmitted, the non-observer comes to know more about the function structure M, without directly viewing it. Thus, an important question arises, "What is the value of the information transmitted by the observer per element?"

To answer this question, let the communication setup change so that the non-observer is required to determine the events by asking binary questions to the observer. Binary questions are answered either yes or no. Under this condition, the non-observer can identify an element within the model by asking binary questions to the observer in such a way that the search space of possible elements, namely, the vocabulary Σ, is successively narrowed down using a binary search tree, like the game twenty-questions. Starting with a vocabulary of size x, the size of the search space reduces with each question following the geometric series $\{x, x/2, x/4 \ldots 4, 2, 1\}$, until the correct element is found. For example, in order to identify the element G in the model, the non-observer asks the following questions in succession and gets the following answers:

1. Is the element in the list {A,B,C,D}? Answer = No
2. Is the element in the list {E,F}? Answer = No
3. Is the element in the list {G}? Answer = Yes, at which point the element is uniquely known to be G.

An assumption in this scenario is that the non-observer and the observer both know the vocabulary Σ, on which the model is built. Also, as pointed out in Assumption-2, all elements in the vocabulary are equally probable to occur in the model. Therefore, the best bet for the non-observer is always to split the remaining search space in the middle, as the equal probabilities prevent him from taking any guess at a more likely solution. Under these conditions, the minimum number of binary questions that the non-observer needs to ask for each element in the model is the logarithm of the size of the vocabulary, analogous to the depth of the binary search tree [Kruse & Ryba, 1999]. This number is given by:

$$N_{\min} = \log_2(x) \qquad \text{Eq.11}$$

where:

N_{min} is the minimum number of binary questions required to determine the element

x is the number of terns in the vocabulary, Σ

Thus, it can be argued that in the initial communication setup, the non-observer was receiving a value of $\log_2(x)$ with the description of each element because the information received from the observer was equivalent to receiving answers to $\log_2(x)$ questions. The form of the expression in Eq.11 is identical with the general form of the information metric in Eq.4. Therefore, the information content of each element practically represents the minimum number of binary questions that must be asked in order to identify an element within the function structure.

In this manner, the non-observer can duplicate the entire function structure, element by element, by asking $\log_2(x)$ questions for each element. At this point, due to the equality between the original and the duplicate models, it can be argued that all the usefulness associated with the original model is also available to the rebuilt model. Thus, the usefulness of the original model can be thought of having been transmitted, though indirectly, from the observer to the non-observer in the form of answers to a finite number of questions. The metric, therefore, represents the practical usefulness of the function structure. Further, since number of questions cannot be a fraction, a whole question needs to be counted for the fractional part of the logarithm, resulting in the need for the ceiling function in the equations.

From a different viewpoint, the information content of a function structure is a measure of the uncertainty involved in the model. In this viewpoint, the non-observer is totally uncertain about the individual functions and flows in the model to start with. As he asks more questions and determines more functions and flows in the model, his uncertainty about the model decreases. Once all the functions and flows are known, the entire model is known to the non-observer, and his uncertainty about the model reduces to zero. The number of questions can therefore represent the initial uncertainty of the model. For a function structure with a high initial uncertainty, the number of questions required to resolve the uncertainty is expected to be proportionally high. This interpretation is obtained from the classical communication theory [Shannon, 1948]. This view has also been adopted in engineering design research, where information-based uncertainty has been described as a source of complexity [Suh, 1990; Summers & Shah, 2003; Ameri et al., 2008]. This uncertainty-oriented view of information is used in Chapter Six, as it provides a natural interpretation of the results presented there.

4.6 <u>Internal Validation of the Information Metrics</u>

Four requirements for information metrics are discussed in Information Theory literature [Shannon, 1948; Carter, 2006]. The metrics presented in Section 4.4 are validated against these requirements to ensure that by adopting Assumptions 1, 2, and 3, the fundamental premises of Information Theory are not lost in these metrics.

4.6.1 Requirement 1

Information is always a non-negative quantity [Carter, 2006]. In a function structure there is always at least one function and at least one flow ($y_V \geq 1$, $y_N \geq 1$). Without a function, the function structure cannot represent any transformative action, and therefore, is invalid and useless. Similarly, a function receives at least one flow as input and produce at least one as output. The number of flows associated with a function, as input or output, is called the cardinality of the function. Theoretically, in extreme cases of cardinality, a function may be associated with only one flow, either as input or as output. For example, the verb *store* in the Functional Basis is used to represent storing actions in functions, and typically receives and incoming flow with no output flow coming out from the function. Thus, the minimum cardinality of a function (verb instance) is one. Without any input or output flow, the function becomes redundant in the function structure as due to isolation, it cannot contribute to the total outcome of the model. Similarly, any usable function vocabulary, including the Functional Basis, must contain at least one verb and one noun each ($x_V \geq 1$, $x_N \geq 1$), as otherwise the vocabulary cannot be used to construct both functions and flows in a function structure. Due to these lower limits of unity, the minimum value of the expressions in Eq.5 is $I_{\min} \geq (1) \times \log_2(1)$, i.e., $I_{\min} \geq 0$. Therefore, the metric satisfies this requirement.

4.6.2 Requirement 2

If an event has probability of 1, no information is obtained from its occurrence [Carter, 2006]. In function structures, the events are analogous to the functions and flows, and this condition implies that there is only one verb or one noun repeatedly used in the functions and flows of the function structure. In that case, the term becomes fully predictable and no additional information is gained by knowing about its occurrence. Mathematically, by setting $x_V = 1$ and $x_N = 1$ in Eq.6 and Eq.7, both I_V and I_N vanish. Thus, the metric satisfies this requirement.

4.6.3 Requirement 3

If two independent events occur, whose joint probability is the product of their independent probabilities, the total information obtained is the sum of their individual information [Carter, 2006]. If *'i'* and *'j'* are two elements of a vocabulary, with independent probabilities p_i and p_j, the probability of their joint occurrence is given by $p_i \times p_j$. Under Assumptions 2 and 3, $p_i = p_j = 1/x$, where x is the size of the vocabulary. Hence the probability of the joint occurrence of *'i'* and *'j'* is $(1/x) \times (1/x) = 1/x^2$, which is equivalent to the independent uniform probability of a single element in a vocabulary of size x^2. Thus, if the individual information content of events *'i'* and *'j'* are I_i and I_j, the information produced by their joint occurrence is obtained from Eq.4 as:

$$I_{i+j} = \log_2(x^2) = 2 \cdot \log_2(x) = \log_2(x) + \log_2(x) = I_i + I_j \qquad \text{Eq.12}$$

where:

i, j are two distinct elements of the vocabulary of size x

I_i is the individual information content of element i

I_j is the individual information content of element j

I_{i+j} the individual information content of an element in a vocabulary of size x^2

The metric, therefore, satisfies this requirement.

4.6.4 Requirement 4

Information is a monotonic continuous function of the probabilities, that is, a slight increase in the probabilities should always result into a slight increase in information [Shannon, 1948; Carter, 2006]. Figure 20 shows the plot of element-wise information against the size of the vocabulary, which satisfies the criterion due to the monotonically increasing nature of logarithms. As discussed in Requirement 1, the practically usable portion of the curve is in the range $x \geq 1$, because a null vocabulary $(x=0)$ is unusable for creating messages.

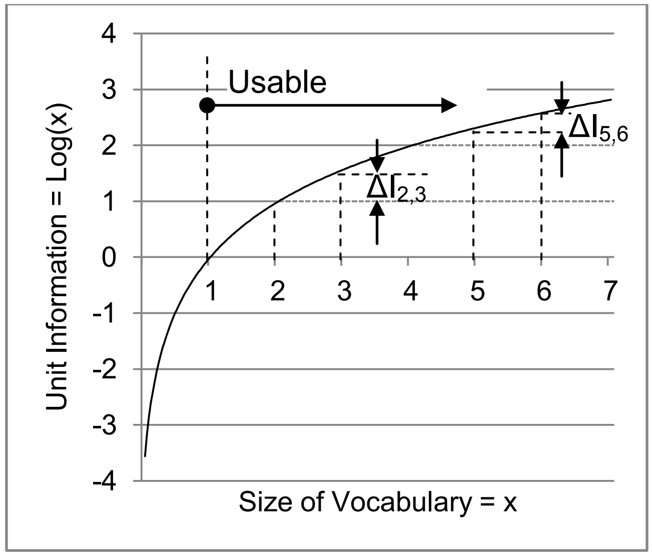


Figure 20: Element-wise information versus the size of vocabulary: Logarithmic plot with base 2

4.7 <u>Discussion</u>

In this section, important properties of the metric are reviewed and their implications to function structures are discussed.

4.7.1 Response to Variables

As seen in Eq.5, the information content I of a function structure increases linearly with the size of the model, y, and logarithmically with the size of the vocabulary, x. Thus, the metric is more sensitive to the change of model size than to the change of the vocabulary size. This implies that a means to arrive at larger models, such as decomposition, can help increase the informativeness of a model more than using a larger vocabulary to construct the model. Intuitively, in a large vocabulary, the distinction between the terms becomes gradually obscured. Hence the model's informativeness to the designer does not increase significantly.

4.7.2 Information Density of a Vocabulary

Since information is a monotonically increasing function of the vocabulary size, the information obtained from a larger vocabulary is always larger, but the increase in information gradually diminishes with increasing size of the vocabulary. As observed in Figure 20, the increase of information due to unit increase of the vocabulary size from 2 to 3, indicated by $\Delta I_{2,3}$, is larger than the increase in information due to the same increase in the vocabulary size from 5 to 6, indicated as $\Delta I_{5,6}$. This observation enables the formulation of a new quantity to assess the usefulness of the vocabulary itself. This quantity, termed information density, is defined below:

Definition

Information density of a vocabulary is the amount of information produced by a single event, measured per unit size of the vocabulary.

The information density of a vocabulary of size *x* is implies the usefulness of the vocabulary in terms of the *benefit* (information produced) over *cost* (size of the vocabulary), and is obtained by dividing both sides of Eq.4 by the size of the vocabulary, *x*.

$$I' = \frac{I_i}{x} = \frac{\lceil \log_2(x) \rceil}{x} \qquad \text{Eq.13}$$

where:

I' is the information density of the vocabulary

I_i is the information per element of the vocabulary

x is the size of the vocabulary

4.7.3 Quantity versus Quality

The metric provides a measure for only the quantity, not the quality, of information stored in a function structure. The numeric value of information can be increased merely by increasing the number of terms in the function structure or the vocabulary, even if the model does not describe the system correctly or consistently. The issues of correctness and consistency are addressed in Section 6.2, where a novel schema of function representation is developed and compared against the existing Functional Basis vocabulary. In the next section these information metrics are applied to evaluate the hierarchy of the Functional Basis vocabulary, using three function structures selected from the Design Repository.

CHAPTER FIVE: APPLICATION OF THE INFORMATION METRIC: EXPERIMENTAL EVALUATION OF THE HIERARCHY WITHIN THE FUNCTIONAL BASIS (EXPERIMENT-I)

The metrics developed in the previous chapter are applied to measure the information content of function structures through a series of experiments using three products within the Design Repository. The products are the Supermax hair dryer, the Delta jigsaw and the Brother sewing machine. These products are chosen as they use many of the Functional Basis' commonly used functions and one of them, the hair dryer, has been studied in previous function modeling research [Mocko et al., 2007; Caldwell et al., 2008]. Additionally, these products demonstrate a variety of sizes and function-to-flow ratios. While the function structure for Supermax hair dryer has 18 functions and 24 flows (a ratio of 0.75), the Brother sewing machine has 44 functions and 64 flows (a ratio of 0.69), and the Delta jigsaw has 17 functions and 42 flows (a ratio of 0.40). The results of these experiments are used to evaluate the usefulness of the Functional Basis vocabulary.

5.1 <u>Experimental Protocol</u>

Four experimental steps are defined in this section and illustrated through the Supermax hair dryer example. These steps are: 1) Model clean-up, 2) Translating the models across Functional Basis levels, 3) Defining the vocabularies, and 4) Computing the information content. The results of the experiments for the Delta jigsaw and the Brother sewing machine are shown in Appendix B and Appendix C.

5.1.1 Model Clean Up

The function structures are first corrected for representational inconsistencies. This process is explained with the help of the hair dryer function structure obtained from the Design Repository, shown in Figure 9. However, for the ease of reference, this figure is repeated below.

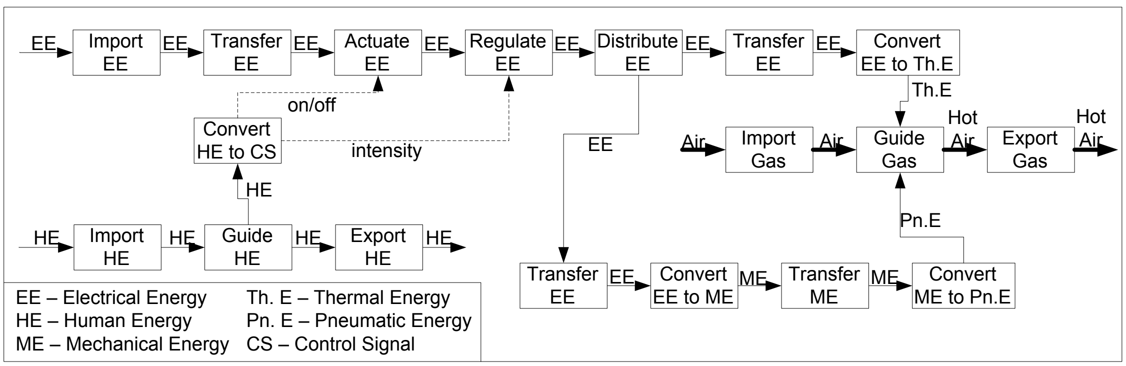


Figure 9 (repeat): Function structure of the Supermax Conair hair dryer created through reverse engineering and stored within the Design Repository

Flow Clean Up for Non-Functional Basis Terms

Figure 9 contains some non-Functional Basis terms, such as *hot, air, on-off,* and *intensity*. These terms are replaced with suitable terms from the Functional Basis, using the same hierarchical level as the remainder of the model, such as *gas* for *air*, and *control signal* for *on-off* and *intensity*. The adjective *hot* is dropped, since the Functional Basis does not provide any vocabulary of adjectives for gasses.

Function Clean-Up for Redundancies

The text within each block represents the transformative action carried out by the function. This text can be generally broken down in to two parts: a verb that indicates the transformative action, and one or two nouns that represent the objects or outcomes of that action. For example, in the block *Import EE* in the top left corner of this figure, the first word *Import* is the verb that represents the transformative action, while the second word, *EE* (electrical energy) is the object of that action. However, it can be noted from Figure 9 that the names of the flows provide enough information about the objects and outcomes of the functions. For example, in the function *Convert ME to PnE*, the incoming and outgoing flows are labeled as *ME* and *PnE*, making the nouns inside the block redundant. Therefore, all texts other than the function verb are omitted from the blocks.

Figure 21 shows the cleaned up function structure of the Supermax hair dryer, with the six corrections highlighted with circles. The function structures of Delta jigsaw and Brother sewing machine, as obtained from the Design Repository, are shown in Figure 31 (Appendix B) and Figure 35 (Appendix C).

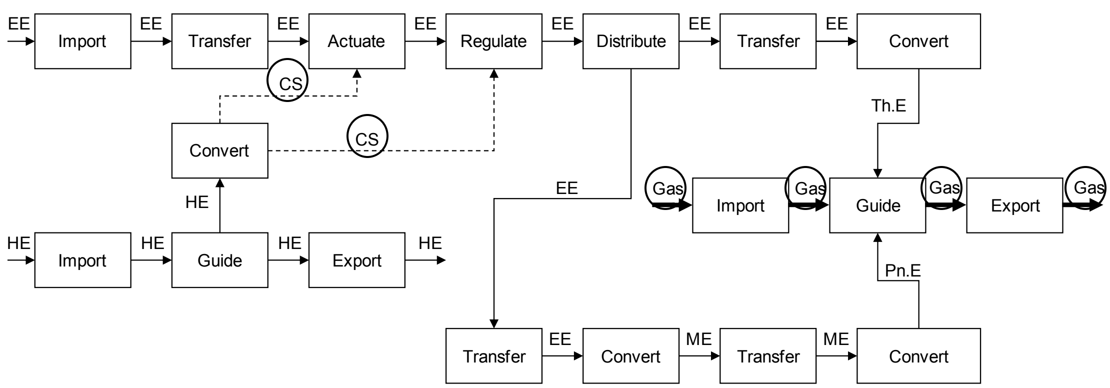


Figure 21: Hair dryer function structure defined with secondary verbs and secondary nouns, after clean up

5.1.2 Translating Function Structures across Functional Basis Levels

After a function structure is cleaned, it is *translated*, that is, redefined with verbs and nouns from other levels of the Functional Basis, without any change to its topology.

Since there are three hierarchical levels for both verbs and nouns in the Functional Basis, a model can be translated to 16 different *designations,* as shown in Table 4.

Table 4: Designation protocol of function structures

Noun Levels →				
3	M(0,3)	M(1,3)	M(2,3)	M(3,3)
2	M(0,2)	M(1,2)	M(2,2)	M(3,2)
1	M(0,1)	M(1,1)	M(2,1)	M(3,1)
0	M(0,0)	M(1,0)	M(2,0)	M(3,0)
	0	1	2	3
	Verb Levels →			

A model described with the mth level of the verbs hierarchy and the nth level of the nouns hierarchy of the Functional Basis is *designated* as M(m,n). For example, M(2,3) designates a model with secondary level verbs and tertiary level nouns. The bottom row M(m,0), and the left column M(0,n) designate models with only one type of terms. For example, M(3,0) designates a model described with tertiary level verbs in the functions but no nouns on the flows. These models are used for measuring element-wise information content, as their information is carried by only one type of element. The function structures on the diagonal are described with the same levels of verbs and nouns. These models are used here for measuring the combined information content. M(0,0) represents the empty function structure graph and contains zero information, which can be verified by setting zeros for y_V and y_N in Eq.10. The grey cells designate models described with mixed levels of verbs and nouns; these models are not used in these experiments, since conventionally, the function structures within the Design Repository are defined with the same hierarchical level of verbs and nouns.

When a function structure is translated from a lower to a higher level (upward translation), the taxonomical parent of each lower-level element is chosen as the new element. When a model is translated from a higher to a lower level (downward translation), each new element is chosen from the taxonomical children of the higher level element using engineering judgment. For example, the secondary function 'guide' in Figure 21 is translated to 'channel' in upward translation, while in downward translation 'allow DoF' is selected as the definition of 'allow DoF' (Appendix A) best matches with the actual function in the product. Thus, upward translations are more objective than downward translations. However, due to the assumed uniform probability distribution of terms over the vocabulary, the specific selection does not impact the numeric score of information content. In order to ensure that each higher level term is represented in the lower levels of the Functional Basis, secondary terms that are not categorized in the tertiary level are propagated, as is, to the tertiary level. For example, in Figure 21, the secondary verbs 'distribute', 'import' and 'export' are all propagated to the tertiary level at the time of translation. Figure 22 and Figure 23 show the hair dryer function structures of designations M(1,1) and M(2,2) respectively. These models are obtained by translation from Figure 21, which is of designation M(2,2).

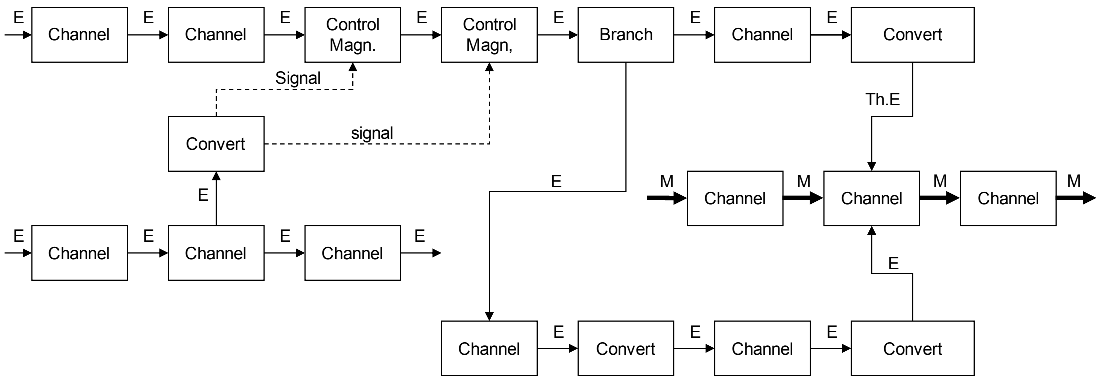


Figure 22: Hair dryer function structure defined with primary verbs and primary nouns, M(1,1)

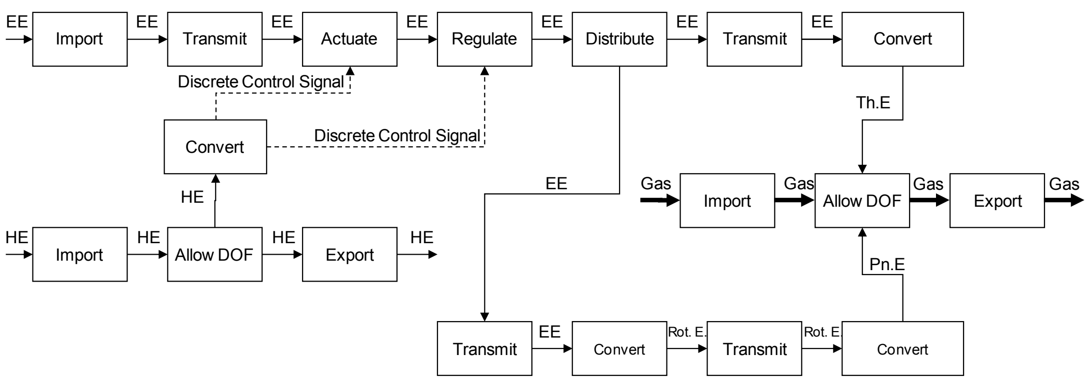


Figure 23: Hair dryer function structure defined with tertiary verbs and tertiary nouns, M(3,3)

5.1.3 Defining Three Types of Vocabularies for Computing Information Content

Due to the hierarchical arrangement of terms in the Functional Basis, a downward translation enables at least three interpretations of the lower level vocabulary, as defined below.

Definitions

1. *The fixed vocabulary of a given level is the collection of all terms in that level.*
2. *The used vocabulary of a given level and a given function structure described on that level is the set of terms that appear in the model.*

3. *The reduced vocabulary for a given function structure that is obtained by translation from a higher to a lower level is the set of all lower level terms that can be obtained as taxonomical children of the higher level terms used by the higher level function structure.*

In the hair dryer function structure, the fixed vocabulary of verbs for all models of designations M(1,n), M(2,n), and M(3,n) are given by the entire collection of verbs in the primary, secondary and tertiary levels of the Functional Basis: 8, 21, and 35, respectively. The used vocabulary of verbs for the models of designation M(1,n), M(2,n) and M(3,n) are the number of verbs appearing in the functions of Figure 22, Figure 21, and Figure 23, respectively, which are four, eight and eight. The reduced vocabulary of verbs for all models of designation M(1,n) is accepted to be identical with the fixed verb vocabulary of the same models, since primary models cannot be obtained in downward translation. Since the used vocabulary of verbs for M(1,n) consists of 'branch', 'channel', 'control', and 'convert', the reduced vocabulary for M(2,n) is taxonomically obtained as the following list: 'separate' and 'distribute' (obtained from 'branch'), 'import', 'export', 'transfer', and 'guide' (obtained from 'channel'), 'actuate' and 'regulate' (obtained from 'control magnitude'), and 'convert' (obtained from 'convert') – a list of 11 verbs. Similarly, for all models of designation M(3,n), the reduced verbs vocabulary is of size 12.

In a similar way, the nouns vocabularies of the fixed, used, and reduced types are determined for each row in Table 4. The combined vocabularies are obtained by adding up the sizes of the corresponding verb and noun vocabularies. Table 5 shows a summary of the verb, noun, and combined vocabularies of the fixed (F), used (U), and reduced (R) types, for all ten designations. In each cell under columns U and R, the values separated by commas represent vocabulary sizes for the Supermax hair dryer, the Delta jigsaw, and the Brother sewing machine respectively. The fixed vocabulary size is a property of the vocabularies, not the models, hence remains equal for all products in each level.

Table 5: Summary of vocabulary sizes for the Supermax hair dryer, Delta jigsaw and Brother sewing machine function structures

Noun Levels →	F	U	R	F	U	R	F	U	R	F	U	R
3	36	7,10,9	9,12,16							71	15,23,24	21,36,34
2	20	7,8,7	20,20,20				41	15,21,17	31,38,33			
1	3	3,3,3	3,3,3	11	7,10,8	11,11,11						
0	0	0	0	8	4,7,5	8,8,8	21	8,13,10	11,18,13	35	8,13,15	12,24,18
	0			1			2			3		

Verb Levels →

5.1.4 Computing Information Content

In order to compute information content, first the sizes of the respective function structures *(y_V and y_N)* are determined. Since there are 18 functions and 24 flows in the hair dryer function structure (Figure 21), the size of all function structures corresponding to the bottom row of Table 5, y_V, is 18, and the size of all function structures in the left column of Table 5, y_N, is 24. The size of all function structures on the diagonal, $y_V + y_N$, is 18 + 24 = 42. The empty model, M(0,0) is an exception, with size zero. The information content of the whole model is then computed by applying Eq.8 for element-wise information and Eq.10 for combined information. In each case, the result of the logarithm is rounded up to the next higher integer, since a whole binary question is counted for the fractional part of the logarithm. Notably, the rounding up is done *before* multiplying by *y*, as opposed to rounding the total information content obtained after multiplying by *y*, since according to the practical interpretation of information, each element of the model needs a finite number of questions to be fully known by the non-observer (see Section 4.5). For example, element-wise information content for M(2,0) and M(0,2) using the fixed vocabulary are computed using Eq.8 as $18 \times \log_2(8) + 0 = 18 \times 3 = 54$ bits, and $0 + 24 \times \log_2(20) = 24 \times 4.3 = 24 \times 5 = 120$ bits, respectively. Similarly, the combined information for M(2,2) using the fixed vocabulary is computed using Eq.10 as $42 \times \log_2(41) = 42 \times 5.4 = 42 \times 6 = 252$ bits. The results of the computations for the Supermax hair dryer are shown in Table 6. The results for the jigsaw and the sewing machine are shown in Table 17 (Appendix B) and Table 18 (Appendix C).

Table 6: Results: Information content of the Supermax hair dryer

Noun Levels ↑	F	U	R	F	U	R	F	U	R	F	U	R
3	144	72	96							294	168	210
2	***120***	72	120				252	168	210			
1	48	48	48	168	126	168						
0				54	36	54	***90***	54	72	108	54	72
	0			1			2			3		

Verb Levels →

5.2 Experimental Results

The results tabulated in Table 6 for the Supermax hair dryer function structure are summarized using bar charts for comparison in Figure 24. This figure shows the nine data points from the bottom row of Table 6, which are the element-wise information contents of the functions. The three clusters of bars represent the primary, secondary, and tertiary levels of the verbs hierarchy, corresponding to models of designation M(1,0), M(2,0), and M(3,0). Within each cluster, the individual bars represent information content using the fixed, used, and reduced vocabularies of verbs.

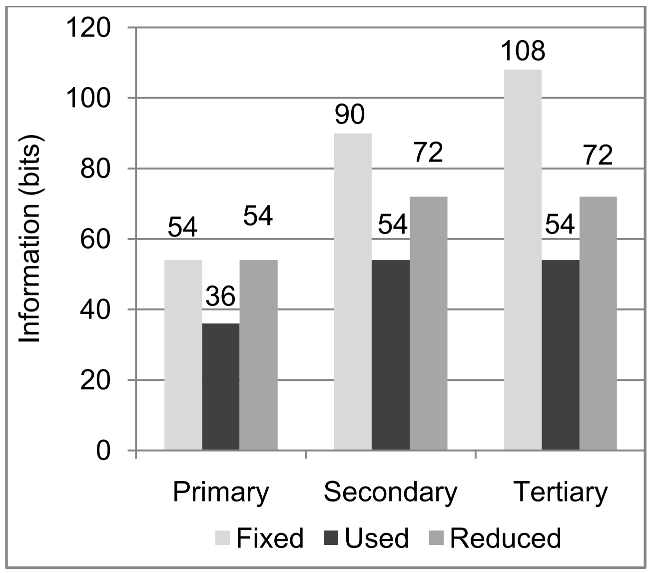


Figure 24: Supermax hair dryer information content: Functions only: M(1,0), M(2,0), M(3,0)

Similarly, Figure 25 shows the nine data points from the left column of Table 6, which are the element-wise information contents of flows in models of designation M(0,1), M(0,2), and M(0,3), and Figure 26 shows the nine data points from the diagonal of Table 6, which are the combined information contents of functions and flows in models of designations M(1,1), M(2,2), and M(3,3). The results for the Delta jigsaw function structures are shown in Figure 32, Figure 33, and Figure 34 in Appendix B. The results for the Brother sewing machine function structure are shown in Figure 36, Figure

37, and Figure 38 and Appendix C. These six figures are also organized in the same way as explained above.

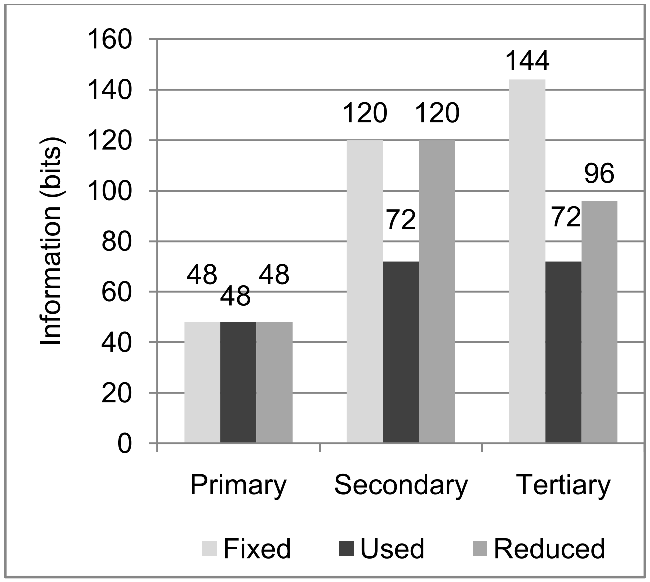


Figure 25: Supermax hair dryer information content: Flows only: M(0,1), M(0,2), M(0,3)

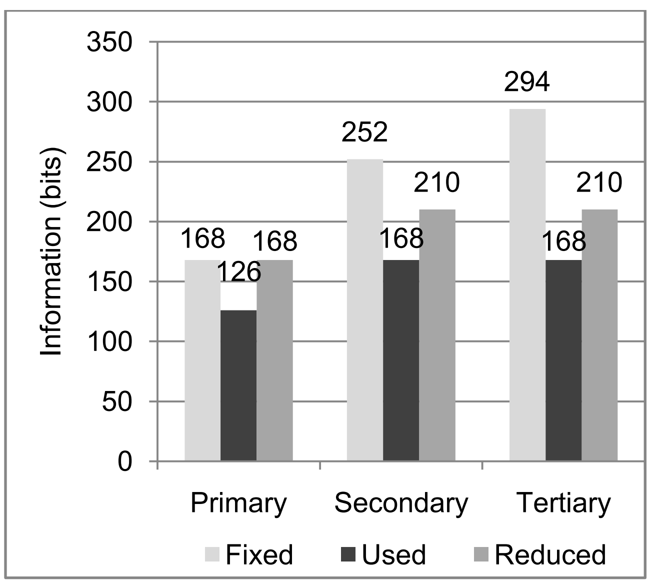


Figure 26: Supermax hair dryer information content: Combined: M(1,1), M(2,2), M(3,3)

The information density of the Functional Basis verbs and nouns are shown in Figure 27. These numbers are obtained from Eq.13. For example, the information density of the primary nouns, which has three elements, can be computed as $\lceil \log_2(3) \rceil / 3 = 0.67$ bits per noun, as shown in the figure.

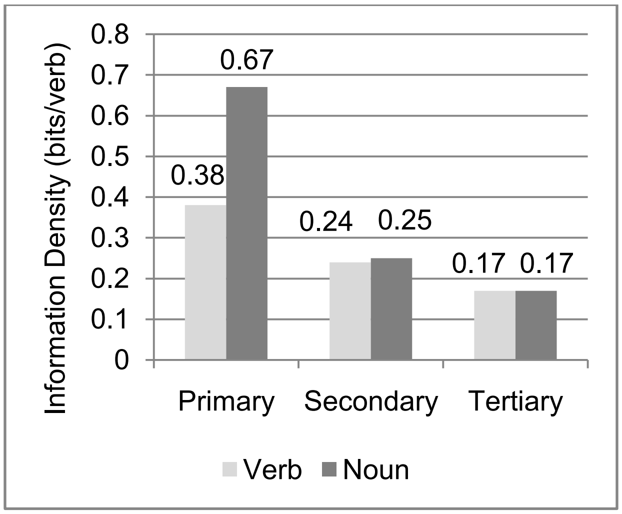


Figure 27: Information density of the Functional Basis verbs and nouns

5.3 <u>Observations and Analysis</u>

Table 7 summarizes the trends of information content based on the experimental results. There are 27 trends discussed, resulting from the combination of three products, three vocabulary types (fixed, used, reduced), and three metrics (functions, flows, and combined). $\Delta I_{I,II}$ represents the change in information content from the primary to the secondary level and $\Delta I_{II,III}$ indicates the change in information content from the secondary to the tertiary level. The symbols '+', '0', and '–' in a cell under $\Delta I_{I,II}$, for example, indicate that the information content based on the secondary level is greater than, equal to, or lower than the information content based on the primary level.

Table 7: Trend of information content across the Functional Basis levels

Voc. type	Metric type	Supermax hair dryer		Delta jigsaw		Brother sewing machine	
		$\Delta I_{I,II}$	$\Delta I_{II,III}$	$\Delta I_{I,II}$	$\Delta I_{II,III}$	$\Delta I_{I,II}$	$\Delta I_{II,III}$
Fixed	Function	+	+	+	+	+	+
	Flow	+	+	+	+	+	+
	Combined	+	+	+	+	+	+
Used	Function	+	0	+	0	+	0
	Flow	+	0	+	+	+	+
	Combined	+	0	+	0	+	0
Reduced	Function	+	0	+	0	+	+
	Flow	+	–	+	–	+	–
	Combined	+	0	+	0	+	0

Eight observations made on the results are presented here. They address the variation in information content across the hierarchical levels of the Functional Basis, the comparative increase of information across those levels, and the trends in information density.

1. Information content of function structures based on the fixed vocabulary monotonically increases from the primary to the secondary to the tertiary level of the Functional Basis (top three rows of data in Table 7). This trend is consistent for the function, flow, and combined metrics, for all three products examined. This trend is expected, as the vocabularies increase in size with the levels (see Table 5).
2. Information content of function structures based on the used vocabulary increases from the primary to the secondary level, but usually remains the same between the secondary and tertiary levels (middle three rows of data in Table 7). This trend is consistent in all but two out of nine cases. The two

exceptions occurred in the noun metrics in the Delta jigsaw and Brother sewing machine models, where the information content increased from the secondary to the tertiary level. But this increase is marginal: 168 – 126 = 42 bits in Delta jigsaw, and 256 – 192 = 64 bits. As a result, the overall information, shown by the combined information content, remains the same between the secondary and the tertiary levels for both products. This observation indicates that even though the vocabulary size increases between the levels, the usage of terms in function structures does not increase proportionately, which means that the tertiary level contains redundant terms, both verbs and nouns.

3. Information content of function structures based on the reduced vocabulary increase from the primary to the secondary level, but usually remain the same from the secondary to the tertiary level, in case of the functions and the combined metrics (first and third row of the last three rows of data in Table 7). This observation is consistent through all but one out of six cases: the function metric of the sewing machine. This trend is identical with Observation 2, and it reinforces the analysis that the tertiary level contains many redundant terms, which add little information content.
4. Information content of function structures based on the reduced vocabulary using the noun metric increases from the primary to the secondary level, but decreases from the secondary to the tertiary level (middle row of the last three rows of data in Table 7). As discussed in Section 5.1.3, the reduced

vocabulary is obtained in two steps. First, the used vocabulary of the higher level is determined. Next, upon downward translation, this used vocabulary expands into its taxonomical children of the lower level. The vocabulary first reduces then expands in this process. While the reduction depends entirely on the function structure, the expansion is entirely dependent on the hierarchical structure of the vocabulary. This observation, then, is a consequence of the fact that the hierarchical expansion of nouns from the primary to the secondary level is much higher than the expansion from the secondary to the tertiary level, which means that the Functional Basis noun hierarchy is an unbalanced taxonomy.

5. *All 27 trends consistently show a significant increase of information content from the primary to the secondary level,* (three columns under heading $\Delta I_{I,II}$ in Table 7). This observation indicates that the secondary level is more informative to the designer than the primary level. However, due to the mixed trends recorded under heading $\Delta I_{II,III}$, particularly in case of the used and reduced vocabularies, the tertiary level is not necessarily more informative to the designer than the secondary level. Table 8 shows some more trends in information content in form of a truth table. Each instance of $I_{m,n}$ represents the information content of a function structure of designation M(m,n). $I'_{m,n}$ indicates the information gradient of the vocabulary measured on model M(m,n). Each row in the Statement column contains a statement that predicts a relation between two quantities related to information content or information

density. Each statement is being evaluated from the experimental results. The status of the evaluation is indicated using symbols '1' for true and '0' for false in the three columns on the right. The Fixed, Used, and Reduced columns indicate the types of vocabulary used for computing information content. The three symbols inside each cell, separated by commas, indicate the status of the evaluation for the Supermax hair dryer, Delta jigsaw, and Brother sewing machine function structures. The trends that did not match the prediction are shaded.

Table 8: Truth table of trends in information content

Trend #	Statement	Fixed	Used	Reduced
1	$\frac{I_{2,0}}{I_{1,0}} > \frac{I_{3,0}}{I_{2,0}}$	1, 1, 1	1, 1, 1	1, 1, 1
2	$\frac{I_{0,2}}{I_{0,1}} > \frac{I_{0,3}}{I_{0,2}}$	1, 1, 1	1, 1, 1	1, 1, 1
3	$\frac{I_{2,2}}{I_{1,1}} > \frac{I_{3,3}}{I_{2,2}}$	1, 1, 1	1, 1, 1	1, 1, 1
4	$I'_{1,0} \geq I'_{2,0} \geq I'_{3,0}$	1, 1, 1	1, 1, 1	1, 1, 1
5	$I'_{0,1} \geq I'_{0,2} \geq I'_{0,3}$	1, 1, 1	1, 0, 0	0, 0, 1
6	$I'_{1,1} \geq I'_{2,2} \geq I'_{3,3}$	1, 1, 1	1, 1, 1	0, 0, 1
7	$I_{1,1} > I_{0,1} + I_{1,0}$	1, 1, 1	1, 1, 1	1, 1, 1
8	$I_{2,2} > I_{0,2} + I_{2,0}$	1, 1, 1	1, 1, 1	1, 1, 1
9	$I_{3,3} > I_{0,3} + I_{3,0}$	1, 1, 1	1, 1, 1	1, 1, 1

6. The proportional increase in information content from the primary to the secondary level is greater than the proportional increase from the secondary to the tertiary level (trends 1-3 in Table 8). This observation is consistent for all three products, for all three vocabulary types, and for all three metrics. Thus,

even though information contents based on the fixed vocabularies increase from the primary to the secondary to the tertiary level in all three products (Observation 1), the proportional increase gradually diminishes for all types of vocabularies in all products, the largest jump being in the downward translation from the primary to the secondary level of both verbs and nouns. This observation supports from a different viewpoint the analysis of Observation 5 that the secondary level is the most useful level in the Functional Basis.

7. *The information density based on the fixed vocabularies reduces from the primary to the secondary to the tertiary level* (trends 4-6 in Table 8). For example, in the case of the Supermax hair dryer, the density of the fixed verbs vocabulary for the primary, secondary, and tertiary levels is 0.364, 0.146, and 0.099 bits per verb. This trend indicates that the *usefulness* of a given level, in terms of *benefit* (information produced) over *cost* (size of the level), reduces with lower levels of the hierarchy. The tertiary level has the lowest information density of the three levels.

8. *The combined information content of function structures is greater than the sum of the element-wise information contents* (trends 7-9 in Table 8). This means that a combined model, described with verbs and nouns of the same hierarchical level, is more informative than the collection of two partial models, described with only verbs and only nouns of the same level. This observation is intuitively explainable, since, given the two partial models,

some human interpretation or value-added activity is required to synthesize them into the combined model. The difference between the information content of the combined model and the sum of the information content of the partial models accounts for this added value in the model.

5.4 Conclusions from Experiment-I

The conclusions about the Functional Basis hierarchy and the information metric are summarized below based on the results of Experiment-I.

5.4.1 Conclusions about the Functional Basis Hierarchy

The secondary level of the Functional Basis vocabulary is clearly the most useful level of the three, in case of both verbs and nouns. The primary level has too low information content, which results from the low number of terms that is insufficient to provide the necessary specificity of function description, making the level less useful than the secondary. The tertiary level is problematic as it has too many redundant terms, which provide only a marginal benefit over the secondary level, but at the cost of a poor information density. In fact, in some cases, the information content actually reduces upon a downward translation from the secondary to the tertiary level, making the tertiary level more discouraging to the designer. Overall, the secondary level appears to be the most preferred of the three levels, providing a good balance between information content and information density. In previous research, an empirical study revealed that about 92% of the Functional Basis terms in function structures within the Design Repository belong to the secondary level [Caldwell et al., 2008]. This empirical observation reinforces the above conclusions, provided that the function structures used in that study was constructed correctly using the Functional Basis.

5.4.2 Conclusions about the Information Metric

The information metric acts as a measure of the usefulness of function structures and the vocabulary, and behaves in agreement with practical expectations. It produces larger values for larger vocabularies and larger function structures, has a reasonable practical interpretation (number of questions), satisfies the required criteria set by Information Theory research, and predicts trends in information content of function structures that is practically reasonable. These observations indicate that the metric is internally valid, that is, it is mathematically and logically consistent within its own definition. This internal validity ensures that the metric does not provide logically inconsistent results, such as predicting negative information content or lower information content for a larger model. However, internal validation does not assess if the metric indeed represents the usefulness of a function structure as perceived by the designer. Therefore, it is not conclusively proved that the practical value of a function structure depends only on its size and the size of the vocabulary.

As discussed in Section 3.3.3, the Functional Basis only provides for the verbs and nouns, but not for any formalism for constructing the connections between those terms, leading to representational inconsistencies. This inconsistency can be illustrated by comparing Figure 9 with Figure 21, where the former contains redundant texts in the blocks, and the latter is cleaned up from those redundancies. If the information metric is applied to these function structures without recognizing the redundancies, Figure 9 would produce higher information content than Figure 21, because it accounts for the same information element more than once (flow nouns within blocks and on the arrows). The information content would be a misleading metric in this case, as the redundant model is practically less useful due to information cluttering and redundancy. This example illustrates that the models must be based upon a consistent representation before the information metric can be applied to them. Once such formalism is established, the metric can be externally validated through user experiments to test if it predicts higher information content for function structures that are considered more useful or valuable by the human designer.

5.4.3 Gap Analysis: Topology as a Source of Information in Function Structures

The metrics discussed in this chapter are based on the assumption that function structures are linear sources of information, where the functions and flows are encountered by the designer one by one, as discrete packets of information (Assumption 1). This assumption was necessary to map the premises of function modeling to those of Information Theory, upon which these metrics are built [Sen et al., 2010]. However, as seen in Figure 21, Figure 22, and Figure 23, function structures are non-linear representations, where all elements—the functions, the flows, and the connections between them—are presented to the observer simultaneously. Specifically, one component of information that is not accounted for by the information metrics is the topological connections between the functions and flows. Despite these limitations, the metric is applicable to measure the usefulness of the Functional Basis, as this vocabulary only provides for verbs and nouns, but not topology. However, the need for capturing topological information can be illustrated by considering two models, one of which displays the functions and flows in a list and other being a complete function structure. The metric would compute the exact same information content between these models, as it ignores the topology. Yet, to the designer, the topological arrangement reveals more information about the product's functionality than the list. It is, therefore, important to extend the metric to measure topological information of function structures.

In the next chapter, this topological information is investigated from two approaches. The first approach is based upon the assumption that any flow can originate from any function in the model and terminate into any other function. The second approach is based upon additional knowledge about the compatibility of functions and flows that limit the topological connection options. The comparison between these approaches lead to a consistent and logically rigorous representation of functions that increases the expressive power of the function structure representation, and reduces the uncertainty involved with individual function structures.

CHAPTER SIX: MEASURING TOPOLOGICAL INFORMATION CONTENT OF FUNCTION STRUCTURES (EXPERIMENT-II)

In this chapter, the information metric is extended to measure the topological information content of function structures. The topological inconsistency discussed in Chapter One and Chapter Three is addressed by capturing the verb definitions stored in the Design Repository. The representation that evolves out of this exercise is then used to rebuild the function structure of the hair dryer product used as example in Chapter Five and to measure information content on it. Results indicate that the evolved representation increases the consistency of models, makes the vocabulary more expressive, and reduces the uncertainty associated with individual function structures.

The two approaches of computing topological information content of function structures investigated here differ in terms of the available knowledge about the compatibility between functions and flows within a model. The issue of compatibility arises as the number and types of incoming and outgoing flows of a given function leads to a limited number of possible combinations that are compatible with the function, according to the function's definition. For example, the function *Import* is defined in the Design Repository as "to bring in a flow (material, energy, signal) from outside the system boundary". Accordingly, a combination of flows where the input is different than the output is not compatible with *Import*, as such a difference suggests a *conversion* during the *importing* action. This example illustrates the presence of implicit topological or relational knowledge within the definitions of the functions. However, unless these definitions are formally represented, they cannot be used in constructing or analyzing function structures. As explained next, the lack of such formalism results into increased uncertainty in the function models.

In section 4.5, information content of function structures was interpreted as the initial uncertainty of the model. Here this viewpoint of information content is used to analyze the topological information content. The uncertainty associated with the topology of a function model arises from the multiple options available to each flow for its origin (tail of arrow) and destination (head of arrow). This uncertainty exists even if the flow itself is known and can be resolved by asking binary questions to determine which functions are the origin and destination of the flow. This situation of *not knowing* the topology of the flow is illustrated in Figure 28, where the flow is marked as known but its origin and destination are unknown. The answers to such questions obtained for all the flows in the model collectively represent a description of the model's topology, and therefore represent the topological information content of the model.

Figure 28: Uncertain origin and destination of a known flow

In the absence of a formal representation of topological knowledge, all functions in the model need to be considered as possible sources and destinations, as a given flow could originate from any function in the model and terminate on to any other function. This approach is the first approach investigated here and is used as a baseline of computing topological information (Section 6.1). This approach corresponds to the current state of the Functional Basis, as this vocabulary does not provide any guideline for topological constructs in function structures. In Section 6.2.1, a representation for topological knowledge is developed and applied to the verbs within the Functional Basis, leading to an enhanced vocabulary, each element of which is a set of rules describing the function and its topological compatibility in a unified form. In the second approach (Section 6.2.2), this enhanced vocabulary is used to compute the information content of the hair dryer function structure. In this case, the number of origin and destination options of each flow is less than the first approach due to the formal representation of the knowledge about limited compatibility. Thus fewer binary questions are required to determine the model's topology. In this manner, the availability of prior topological knowledge reduces the model's uncertainty and makes the representation (enhanced vocabulary) more expressive, as explained in Section 2.2.

6.1 Approach-1: Topological Uncertainty in Function Structures without Formal Representation of Topological Knowledge

In order to demonstrate the contribution of topological information content toward the total information content of function structures, only the element-wise information contents of functions and flows based on the fixed vocabularies are considered here. Since there is no vocabulary for topological connections, the concepts of *used* or *reduced* vocabularies are not applicable to topology. The set of functions in the function structure constitute the search space of origins and destinations for the flows, and therefore is analogous to the fixed vocabulary for computing topological information.

The information content (element-wise, fixed) from the functions and flows in a function structure can be computed using Eq.6 and Eq.7 respectively. These computations are captured for the hair dryer function structure in Table 6. In that table, the cell with the bold italicized text in the left column represents the information content of the flows based on the fixed secondary vocabulary, I_N = 120 bits. Similarly, the cell with bold italicized text in the bottom row quantifies the information content of the functions, I_V = 90 bits. However, for the sake of completeness of the example, these calculations are repeated here. The hair dryer function structure has eighteen functions and 24 flows, while the Functional Basis secondary level has 21 verbs and twenty nouns. These values are used in the equations below to compute the element-wise information content of the model contributed by the functions and the flows.

$$I_V = y_V \cdot \lceil \log_2(x_V) \rceil = 18 \cdot \lceil \log_2(21) \rceil = 90 \text{ bits} \qquad \text{Eq.14}$$

$$I_N = y_N \cdot \lceil \log_2(x_N) \rceil = 24 \cdot \lceil \log_2(20) \rceil = 120 \text{ bits} \qquad \text{Eq.15}$$

Once the information content of functions and flows are determined for the function structure, the only missing information about the model is that associated with its topology. In the absence of prior topological knowledge, if there are y_V functions in a function model, then each flow has (y_V+1) options for its origin, as each flow can originate from any of the functions within the model, as well as from the environment. In the topological sense, the environment behaves as a function, as it can be the origin or destination for any flow. Further, if it is assumed that a flow cannot terminate back to its origin, then each flow has one less option for its destination than its origin optinos. Thus, the number of possible destinations is y_V+1-1= y_V. Therefore, the total number of combinatory possibilities for the origin and destination of the flow is (y_V+1) × y_V. If there are y_N flows in the model, the total number of binary questions required to determine the model's topology is given by the term I_T in Eq.16:

$$I_T = y_N \cdot \left\lceil \log_2 \left[(y_V + 1) \times y_V \right] \right\rceil \quad \text{Eq.16}$$

Eq.16 quantifies the topological uncertainty in the function model without any topological knowledge. The topological uncertainty of the hair dryer function model is computed using this equation in Table 9.

Table 9: Topological uncertainty in the hair dryer function structure without topological knowledge representation

Number of functions in the model, y_V	18
Number of flows in the model, y_N	24
Topological uncertainty (bits) $I_T = y_N \cdot \left\lceil \log_2 \left[(y_V + 1) \times y_V \right] \right\rceil$	216

The total uncertainty in the hair dryer function model is calculated in Table 10, where of I_V, I_N, and I_T are obtained from Eq.14, Eq.15, and Table 9.

Table 10: Total uncertainty in the hair dryer function structure without topological knowledge (Approach-1)

Uncertainty Components	**Uncertainty (bits)**
Uncertainty from functions, I_V	90
Uncertainty from flows, I_N	120
Uncertainty from model topology, I_T	216
Total in the Function Model, $\boldsymbol{I_{FM} = I_V + I_N + I_T}$	**426**

As seen in Table 10, topological uncertainty contributes a significant portion of the total uncertainty of the model: 216 out of 426 bits, which is approximately 51%. This component is also significantly higher than the uncertainty due to the functions and flows in the model. Notably, this topological uncertainty is caused by the same number of flows (y_N) that contribute to the flow uncertainty (I_N), but the effect is magnified in case of topology due to the large number of combinatory possibilities for the origins and destinations, each of which is equally probable. In this research, this explosion of topological uncertainty is attributed to the lack of formal representation of the topological knowledge. The large number of topological combinations arises from the open assumption that a flow can originate from or terminate to any function, which is not necessarily true for all functions and flows. However, in order to use the more realistic number of combinations, which is potentially lower than the number of options based on the exhaustive combinations, the knowledge about topological compatibility between functions and flows need to be formally represented. In the following section, this knowledge representation is developed.

6.2 Approach-2: Topological Uncertainty in Function Structures with Formal Representation of Topological Knowledge

In order to compute the uncertainty in the presence of topological knowledge, first a formal representation of this additional knowledge is needed. This new representation is developed in Section 6.2.1. The uncertainty is then computed in Section 6.2.2.

6.2.1 Representation of Topological Knowledge

In order to formally represent the topological knowledge, a function is represented in this research as a triple {*Name*, *In_List*, *Out_List*}, instead of only its name, as done in the Functional Basis. The first attribute, *Name,* is a string indicating the name of the function, which is identical to the literal string (name) used to identify the function in Table 1 and Table 2.

Table 1. The second attribute, *In_List,* is the list of input flows accepted by the function. The third attribute, *Out_List,* is the list of output flows produced by the function. In case of the Functional Basis, each member of these two lists is a Functional Basis flow term. For example, the instance of the function *Import* in Figure 21 that represents the input of electrical energy to the system can be expressed as {"Import", {EE}, {EE}}, and the function *Distribute* that breaks the flow of EE into two flows of EE can be expressed as {"Distribute", {EE}, {EE, EE}}.

Along with this new triple-based description of individual functions, a set of rules is used for each verb in the vocabulary to control the valid input and output flows that can be associated with an instance of that verb. These rules are extracted from the definition of the verbs within the Design Repository. For example, the verb *Import* is defined as "to bring in a flow (material, energy, signal) from outside the system boundary". From this definition the following rules can be extracted:

1. *The function operates on one flow at a time.*

   This can be formally expressed as the rule:

$$\text{In\_List} = \{I_1\} \qquad \text{Eq.17}$$

   indicating that the size of *In_List* is unity, where I_1 is the only input flow.

2. *The incoming flow does not undergo any change within the scope of the function, other than being imported to the system, indicating that the incoming and outgoing flows are identical.*

Consequently, the list *Out_List* must contain only one flow, O_1, which is identical to I_1. The resulting rules are:

$$\text{Out\_List} = \{O_1\} \qquad \text{Eq.18}$$

$$I_1 = O_1 \qquad \text{Eq.19}$$

3. *The flows (incoming or outgoing) can be of any one type within the classes Material (M), Energy (E), and Signal (S).*

   This fact leads to the rule:

$$I_1 = O_1 \in \{M \cup E \cup S\} \qquad \text{Eq.20}$$

4. *The incoming flow always originates in the environment, and the outgoing flow goes to another function within the model, but does not go back to the environment.*

   The set of vertices in the function model graph, denoted by V, is defined here as the functions in the model, plus the environment. This definition implies that the environment is indistinguishable from a function in the topological sense, since it can be the origin or destination of a flow, just as any function in the model. Thus, the following rules can be written:

$$Origin(I_1) = Env \in V \qquad \text{Eq.21}$$

$$Destination(O_1) \in V - \{Env\} \qquad \text{Eq.22}$$

5. *Additionally, it is assumed that a flow cannot terminate back to the same function from which it originated.*

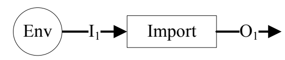
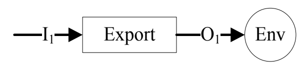
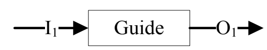
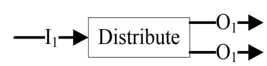
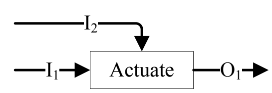
This rule is not explicit in the definition of *Import,* but is a reasonable assumption, as allowing a flow to terminate to its origin creates provision for infinite looping of a flow without any change being done to it between such loops, and thereby rendering the flow itself redundant for the overall functionality of the product. This reasoning leads to the rule:

$$Destination(O_1) \neq Origin(O_1) \qquad \text{Eq.23}$$

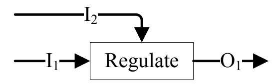
Here the methods *Origin()* and *Destination()* operate on a flow to determine its origin and destination functions. In this example, the first four rules, Eq.17 through Eq.20 control the number and types of flows that can be associated with the function. These rules are called the *compatibility rules*. The last three rules, Eq.21 and Eq.23, control the origin of the incoming flow and the destination of the outgoing flow. These rules are called the *connection rules*. The compatibility and connection rules together represent the topological knowledge for the function *Import.*

In this manner, each verb in the Functional Basis can be represented as a triple and its accompanying rules. Such an exercise would result into a new vocabulary, isomorphic to the Functional Basis function set, each element of which is a description of the function in the triple notation and its rules. For brevity, only the functions used in the hair dryer function model (Figure 21) are presented using this enhanced representation in Table 11. The Functional Basis definition of each function is provided in the second column to justify the rules. The third and fourth columns show the rules for each function.

Table 11: Function triples and topological rules for the hair dryer functions

Verb	Definition[3]	Compatibility Rules	Connection Rules	Triple Notation and Template
Import	To bring in a flow (material, energy, signal) from outside the system boundary.	In_List = $\{I_1\}$ Out_List = $\{O_1\}$ $I_1 = O_1 \in \{M \cup E \cup S\}$	$Origin(I_1) = Env \in V$ $Destination(O_1) \in V - \{Env\}$ $Destination(O_1) \neq Origin(O_1)$	{Import, {I1}, {O1}} 
Export	To send a flow (material, energy, signal) outside the system boundary.	In_List = $\{I_1\}$ Out_List = $\{O_1\}$ $I_1 = O_1 \in \{M \cup E \cup S\}$	$Origin(I_1) \in V - \{Env\}$ $Destination(O_1) = Env \in V$	{Export, {I1}, {O1}} 
Guide	To direct the course of a flow (material, energy, signal) along a specific path.	In_List = $\{I_1\}$ Out_List = $\{O_1\}$ $I_1 = O_1 \in \{M \cup E \cup S\}$	$Origin(I_1) \in V - \{Env\}$ $Destination(O_1) \in V - \{Env\}$ $Origin(I_1) \neq Destination(O_1)$ $Destination(O_1) \neq Origin(O_1)$	{Guide, {I1}, {O1}} 
Transfer	To shift, or convey, a flow (material, energy, signal) from one place to another.	In_List = $\{I_1\}$ Out_List = $\{O_1\}$ $I_1 = O_1 \in \{M \cup E \cup S\}$	$Origin(I_1) \in V - \{Env\}$ $Destination(O_1) \in V - \{Env\}$ $Origin(I_1) \neq Destination(O_1)$ $Destination(O_1) \neq Origin(O_1)$	{Transfer, {I1}, {O1}} 
Distribute	To cause a flow (material, energy, signal) to break up. The individual bits are similar to each other and the undistributed flow.	In_List = $\{I_1\}$ Out_List = $\{O_1, O_1, \ldots\}$ $I_1 = O_1 \in \{M \cup E \cup S\}$ $\lvert \text{Out\_List} \rvert = n > 1$ n is a positive integers	$Origin(I_1) \in V - \{Env\}$ $Destination(O_1) \in V - \{Env\}$ $Destination(O_1) \neq Origin(O_1)$	{Distribute, {I1}, { O1, O1, ... n terms}}  $n = 2$
Actuate	To commence the flow of energy, signal, or material in response to an imported control signal.	In_List = $\{I_1, I_2\}$ Out_List = $\{O_1\}$ $I_1 = O_1 \in \{M \cup E \cup S\}$ $I_2 = CS \in S$	$Origin(I_1) \in V - \{Env\}$ $Origin(I_2) \in V - \{Env\}$ $Destination(O_1) \in V - \{Env\}$ $Destination(O_1) \neq Origin(O_1)$	{Actuate, { I1, I2}, {O1}} 

[3](http://repository.designengineeringlab.org/, accessed on January 27, 2009)

Verb	Definition[3]	Compatibility Rules	Connection Rules	Triple Notation and Template
Regulate	To adjust the flow of energy, signal, or material in response to a control signal, such as a characteristic of a flow.	In_List = $\{I_1, I_2\}$ Out_List = $\{O_1\}$ $I_1 = O_1 \in \{M \cup E \cup S\}$ $I_2 = CS \in S$	$Origin(I_1) \in V - \{Env\}$ $Origin(I_2) \in V - \{Env\}$ $Destination(O_1) \in V - \{Env\}$ $Destination(O_1) \neq Origin(O_1)$	{Regulate, { I_1, I_2}, {O_1}} 
Convert	To change from one form of a flow (material, energy, signal) to another.	In_List = $\{I_1\}$ Out_List = $\{O_1\}$ $I_1, O_1 \in \{M \cup E \cup S\}$ $I_1 \neq O_1$	$Origin(I_1) \in V - \{Env\}$ $Destination(O_1) \in V - \{Env\}$ $Destination(O_1) \neq Origin(O_1)$	{Convert, {I_1}, {O_1}} 

Each row in the fifth column of Table 11 shows the triple notation of each function, and a graphically equivalent representation of the rules. In each case, the string within the block represents the function name, with the exception of the environment which is represented as a circle in order to distinguish it from the functions. The incoming arrows are members of *In_List*, while the outgoing arrows belong to *Out_List*. The strings written on the arrows are the names of individual flows, and match with the symbols used in the rules of the third column. These graphical representations are called *function templates* in this research.

A review of Table 11 reveals that the templates are not unique unless the names are included in them. The compatibility rules are mostly unique, with the exception of A*ctuate* and *Regulate,* which are both logically and topologically identical. Also, some templates are over-defined. For example, the inclusion of the environment in templates of *Import* or *Export* makes their name in the blocks redundant. Addressing these inconsistencies requires the use of additional rules and graphical elements, which are out of the scope of this thesis, yet is reserved for future work. However, despite the aforementioned inconsistencies, the templates are useful in their present form for demonstrating the effect of topological knowledge on the uncertainty of function models.

The adequacy and consistency of the enhanced vocabulary are also outside the scope of this thesis, and are reserved for future work. However, it can be argued that if the original functions and their definitions are adequate and consistent for describing design artifacts, the enhanced version should also be adequate and consistent for the same purpose since the only change incurred through this enhancement is the inclusion of additional knowledge without loss of any existing knowledge.

The impact of this new representation on the uncertainty of function structures is of concern to this thesis. In the next section the topological uncertainty of the hair dryer function structure is computed with this enhanced vocabulary.

6.2.2 Computing Uncertainty in Function Structures with Topological Knowledge

In this section, the uncertainty in the hair dryer function structure is computed using the enhanced vocabulary, in terms of the number of binary questions, as explained in Section 4.5. The computation decomposes the total uncertainty into three components of the model: the function templates, the flows attached to the templates, and the connections between the templates. In each case, uncertainty is computed in terms of the number of binary questions. Thus, the total uncertainty in the model is the sum of the number of questions required to determine these three. By asking enough questions to fully describe these three parts, the hypothetical non-observer of Section 4.5 can gather enough information about the model so that he can to reconstruct the model at his end. Hence, all the uncertainty in the model is accounted for in the three parts.

While computing uncertainty of the function structure in terms of number of binary questions, it is assumed that the number of functions and flows in the model are known to the non-observer. Thus, when all the functions and flows are determined through binary question, the non-observer knows to stop asking further questions. In the special case of the *distribute* function, the value of n (the number of outgoing flows) is also assumed to be known to the non-observer. By this assumption, the non-observer can reconstruct the function with the correct number of outgoing flows. The computation of the total uncertainty of the hair dryer function structure in three parts is illustrated below.

Part-1: Uncertainty from Function Template Instances

As illustrated in Eq.14, the number of binary questions required to determine the functions in the model, as with approach 1, is $I_V = y_V \times \log_2(x_V) = 18 \times \log_2(21) = 90$, for 18 function instances in the model, and 21 functions in the vocabulary. Thus, by asking 90 binary questions, the non-observer can determine how many instances of each function are used in the model. For example, in case of the hair dryer function model, the non-observer finds the followings: there are eight different functions in the model: *Import* (3 instances), *Transfer* (4 instances), *Guide* (2 instances), *Export* (2 instances), *Distribute* (1 instance), *Actuate* (1 instance), *Regulate* (1 instance), and *Convert* (4 instances). Thus, the number of questions required to determine the template names is equal to the number of questions required to determine the functions in approach-1: ninety in both cases.

Part-2: Uncertainty in Flows attached to the Templates using Compatibility Rules

Once the function template instances are known, the non-observer can determine the flows associated with each function template using the compatibility rules. For example, from the rule $I_1 \in \{M \cup E \cup S\}$ in the rule set of the function *Import*, it is known to the non-observer that the options for I_1 and O_1 includes all members in sets of Material (*M*), Energy (*E*), and Signal (*S*) in the Functional Basis nouns set: a total of 20 items. Thus, the number of binary questions required to determine I_1 is

$$Info_{I1} = \lceil \log_2(20) \rceil = \lceil 4.32 \rceil = 5 \text{ bits} \qquad \text{Eq.24}$$

Further, from the rule $I_1 = O_1$, it is known that no additional question is necessary to determine O_1, once I_1 is determined. The rules In_List={I_1} and Out_List={O_1} suggest that there are no other flows than I_1 and O_1 involved in the function. Therefore, the total number of questions to determine the flows associated with the function *Import* is 5. In terms of uncertainty, the topological uncertainty of each instance of *Import* is:

$$I_{\text{Import}} = \lceil \log_2(20) \rceil = \lceil 4.32 \rceil = 5 \text{ bits/instance} \qquad \text{Eq.25}$$

According to Table 11, the compatibility rules of *Import* are identical with those for the functions *Export, Guide,* and *Transfer*. For the function *Distribute*, only the second rule is different from the second rule of *Import*. However, since there is only one flow, O_1, repeated *n* times in Out_List, and since the rule $I_1 = O_1$ holds by the definition of *Distribute*, the non-observer can conclude that once I_1 is determined by asking binary questions, no additional question is necessary for determining any of the instances of O_1. Therefore, in terms of topological uncertainty, *Distribute* is identical with *Import*. These findings lead to the following conclusion.

$$I_{\text{Export}} = I_{\text{Guide}} = I_{\text{Transfer}} = I_{Distribute} = I_{\text{Import}} = 5 \text{ bits/instance} \qquad \text{Eq.26}$$

For the function *Actuate*, there are two input flows listed in *In_List*. However, by definition, the flow I_2 is hardcoded to be a *control signal*, a secondary signal class within the Functional Basis flow set. Therefore, there is no uncertainty associated with this flow, as no questions are necessary to determine it. Apart from I_2, the remaining compatibility rules are identical with *Import.* Hence, the uncertainty involved in each instance of *Actuate* is:

$$I_{\text{Actuate}} = \lceil \log_2(20) \rceil = \lceil 4.32 \rceil = 5 \text{ bits/instance} \qquad \text{Eq.27}$$

The compatibility rules for *Regulate* are identical with that of *Actuate*, hence, the uncertainty involved in each instance of *Regulate* is:

$$I_{\text{Regulate}} = I_{\text{Actuate}} = \lceil \log_2(20) \rceil = \lceil 4.32 \rceil = 5 \text{ bits/instance} \qquad \text{Eq.28}$$

For the function *Convert* (see Table 11), though the first three compatibility rules are identical with *Import*, the rule $I_1 \neq O_1$ makes this function different from *Import*. Due to this rule, the uncertainty needs to be computed for the incoming and outgoing flows separately. For the incoming flow, all 20 elements in the unified list of Material (M), Energy (E), and Signal (S) are available as options, hence the number of questions required to determine this input flow is:

$$Info_{I_1} = \lceil \log_2(20) \rceil = \lceil 4.32 \rceil = 5 \text{ bit} \qquad \text{Eq.29}$$

However, for the outgoing flow, the number of options is one less than 20, since this flow could not be the same as the incoming flow, by the definition of *Convert*. Thus, the number of questions required for determining this flow is:

$$Info_{O_1} = \lceil \log_2(19) \rceil = \lceil 4.25 \rceil = 5 \text{ bits} \qquad \text{Eq.30}$$

Though the numbers of questions required to identify the two flows are equal after rounding up, they are fundamentally different. The total uncertainty involved in the topology of each instance of *Convert* is therefore:

$$I_{\text{Convert}} = \lceil \log_2(20) \rceil + \lceil \log_2(19) \rceil = 10 \text{ bits/inst} \qquad \text{Eq.31}$$

Based on the findings of Eq.25 through Eq.31, the uncertainty due to the flows attached to the templates in the hair dryer function structure is as tabulated in Table 12. The second column of this table summarizes the values from Eq.25 through Eq.31, while the third column lists the number of instances of each function within the hair dryer function model. The fourth column computes the total uncertainty contributed by the flows attached to templates, as the product of the respective cells in the second and third column.

Table 12: Uncertainty from the flows (I_F) in the function templates of the hair dryer function structure

Function	**Uncertainty (bits/instance)**	**Number of instances**	**Total uncertainty (bits)**
Import	5	3	15
Export	5	2	10
Guide	5	2	10
Transfer	5	4	20
Distribute	5	1	5
Actuate	5	1	5
Regulate	5	1	5
Convert	10	4	40
TOTAL (I_F)		**18**	**110**

From the above discussion, it follows that after asking 90 questions for the functions and 110 questions for the flows attached to the templates, the non-observer knows all the function templates and flows in the model. However, the connections between them are yet to be determined. This intermediate state of knowledge about the function structure is shown in Figure 29.

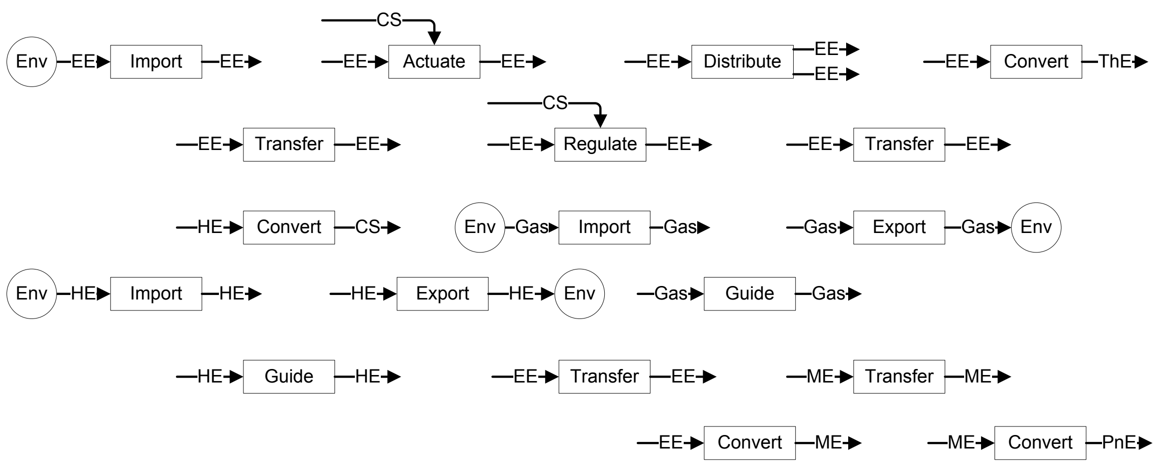


Figure 29: Intermediate state of the function structure: Disconnected function templates obtained by asking 90+110 = 200 binary questions

As discussed before, once these connections are determined by asking more questions, no more information will be required for the non-observer to successfully reconstruct the function model. At that point, it can be argued that the entire uncertainty of the function model is removed as the model is fully known. The computation of this last component of uncertainty is shown in the next section.

Part-3: Uncertainty in the Connections between Templates using Connection Rules

In order to determine the connections between the templates the non-observer may pick the outgoing flows from a template one at a time and considers the other templates as the possible destination. Alternately, the non-observer can pick an incoming flow to a template, and consider from which other templates that flow could have originated. For simplicity, the first approach is illustrated here for determining the connections.

For the example hair dryer function structure, the flow of EE coming out of the template of *Import* in the top left corner of Figure 29 can terminate into any template that accepts EE as an input. There are nine templates in the function model that accept EE as an input. However, some templates have identical description in terms in the triple notation, suggesting that they are indistinguishable from each other. For example, all instances of *Transfer* have the same triple: {"Transfer", {EE}, {EE}}, and do not count as multiple destination options for the EE flow under consideration. By contrast, the two instances of *Convert* that accept EE as an input are different, as they have different triples: {"Convert", {EE}, {ThE}} and {"Convert", {EE}, {ME}}. Thus, the reduced options for the destination of the said EE flow are *Actuate, Distribute, Convert* (with output of ThE)*, Convert* (with output of ME)*, Transfer, and Regulate* – a total six options. Thus, by asking $\lceil \log_2(6) \rceil = \lceil 2.58 \rceil = 3$ questions, the non-observer can determine that the EE flow terminates into a template of *Transfer.* The connection between *Import* and *Transfer* shown in Figure 30 can be built by the non-observer at this point.

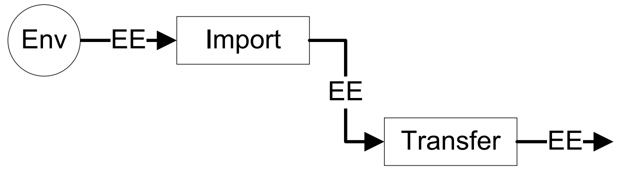


Figure 30: Connection between *Import* and *Transfer,* determined by using the connection rules for the outgoing flow of EE from the function *Import*

In this manner, all the connections in the function model can be determined by asking binary questions. The number of destination options and binary questions required to determine the destination of each flow is shown in Table 13. The eighteen rows in the first column correspond to the eighteen templates in Figure 29. The second column shows the outgoing flows from each template. Since distribute has two outgoing flows, there are total nineteen rows in the second column for eighteen functions. The third column lists the possible destinations for the outgoing flow, and the fourth column gives the size of this list. The last column calculates the uncertainty involved in those options, equivalent to the number of binary questions to find the actual destination, in bits.

There are two special decisions required for completing the computation in Table 13. The first one pertains to the instance of the function *Guide* that indicates the flow of the gas through the hair dryer. As seen in the hair dryer function structure in Figure 21, the outgoing flows of ThE and PnE from the two instances of Convert are terminated on this instance of *Guide*. However, there is no provision for multiple incoming flows in the definition of *Guide* in the Design Repository, which also reflects in the compatibility rules of the function in Table 11. These additional incoming flows are inconsistencies in the model, which was inherent to the function structure from the Design Repository. Correcting function models for such modeling inconsistencies is out of the scope of this thesis. However, in each of these two flows (ThE and PnE), the number of destination option is arbitrarily assigned as 1, as seen in row 8 and row 19 of Table 13.

Table 13: Connection uncertainty (I_C) in the hair dryer function structure

Sl. No.	Template in Triple Notation	Out Flow	Destination Options	# Options	Uncertainty
1	{"Import",{EE},{EE}}	EE	{Transfer, Actuate, Regulate, Distribute, Convert, Convert}	6	3
2	{"Transfer",{EE},{EE}}	EE	{Actuate, Regulate, Distribute, Transfer, Convert, Convert}	6	3
3	{"Actuate",{EE, CS},{EE}}	EE	{ Transfer, Regulate, Distribute, Convert, Convert }	5	3
4	{"Regulate",{EE,CS},{EE}}	EE	{ Transfer, Actuate, Distribute, Convert, Convert }	5	3
5	{"Distribute",{EE},{EE,EE}}	EE	{ Transfer, Actuate, Regulate, Convert, Convert }	5	3
6		EE	{ Transfer, Actuate, Regulate, Convert, Convert }	5	3
7	{"Transfer",{EE},{EE}}	EE	{Actuate, Regulate, Distribute, Transfer, Convert, Convert}	6	3
8	{"Convert",{EE},{ThE}}	ThE	{}	1	0
9	{"Convert",{HE},{CS}}	CS	{Actuate, Regulate}**	2	1
10	{"Import",{HE},{HE}}	HE	{Guide, Export, Convert}	3	2
11	{"Guide",{HE},{HE}}	HE	{Export, Convert}	2	1
12	{"Export",{HE},{HE}}	HE	{Env}	1	0
13	{"Import",{Gas},{ Gas }}	Gas	{Guide, Export}	2	1
14	{"Import",{ Gas },{ Gas }}	Gas	{Export}	1	0
15	{"Export",{ Gas },{ Gas }}	Gas	{Env}	1	0
16	{"Transfer",{EE},{EE}}	EE	{Actuate, Regulate, Distribute, Transfer, Convert, Convert}	6	3
17	{"Convert",{EE},{ME}}	ME	{Transfer}	1	0
18	{"Transfer",{ME},{ME}}	ME	{Convert}	1	0
19	{"Convert",{ME},{PnE}}	PnE	{}	1	0
	TOTAL (I_C)				**29**

Similarly, in Figure 21, the instance of {"Convert", {HE}, {CS}} has two outgoing flows of CS, which is in contradiction with the definition of *Convert* in the Design Repository. The definition of *Convert,* as well as the compatibility rules, indicates that there is only one incoming and one outgoing flow associated with this verb. This instance of *Convert* is another example of modeling inconsistency inherent to the Design Repository, which is out of the scope of this thesis. Specifically for this function structure, one instance of the CS flow coming out of the function {"Convert",{HE},{CS}} is ignored, as seen in row 9 of Table 13.

Finally, the total uncertainty in the function model can be computed by adding the three components – functions, flows attached to the templates, and the connections. This calculation is presented in Table 14.

Table 14: Total uncertainty in the hair dryer function structure with topological knowledge (Approach-2)

Uncertainty Components	**Uncertainty (bits)**
Uncertainty from functions (I_V)	90
Uncertainty from template Flows (I_F)	110
Uncertainty from connections (I_C)	29
Total in the Function Model ($I_{FM} = I_V + I_F + I_C$)	**229**

6.3 Comparison between the Two Approaches of Topological Uncertainty

The total information contents of the hair dryer function structure based on the two approaches discussed here are compared in Table 15.

Table 15: Comparison between the two approaches

Uncertainty Components	**Notation and magnitude of uncertainty (bits)**	
	Approach-1	**Approach-2**
Functions	$I_V = 90$	$I_V = 90$
Flows	$I_N = 120$	$I_F = 110$
Connections	$I_T = 216$	$I_C = 29$
Total	**426**	**229**

As seen in Table 15, the total uncertainty of the model is reduced from 426 bits in Approach-1 (Table 10) to 229 bits in Approach-2 (Table 14): a reduction of nearly 46%. Both approaches rely on determining the functions first, thus incurring the same amount of uncertainty (number of questions) in doing so: 90 bits. The uncertainties contributed by the flows, I_N in Approach-1 and I_F in Approach-2, are comparable in size: 120 bits for I_N and 110 bits for I_F. However, significant difference is observed between the third components: 216 bits for I_T (Approach-1), and 29 bits for I_C (Approach-2). Both of these components represent the uncertainty involved in the connectedness within the model. However, as the additional knowledge of compatibility and connection is made available within the enhanced vocabulary, the number of possible destinations for the flows is smaller in Approach-2 than in Approach-1, resulting into less uncertainty. For example, seven out of the nineteen flows in Table 13 have only one destination option, owing to this prior knowledge. In each of these seven cases, the contribution to connection uncertainty is zero in Approach-2, compared to five bits in Approach-1, as can be derived from Table 10 (90 / 18 = 5). Approach-1 depends on an exhaustive search based on the assumption that any flow could go from any function to any other function. As a result, the number questions necessary to determine the connections is much higher.

6.4 Conclusions from Experiment-II

The high level of topological uncertainty in Approach-1 indicates that the topological arrangement in the function model bears a large share of the model's information. In the case of the hair dryer function model the share of topological information is 216 bits out of 426 bits total: approximately 51%. This observation agrees with the discussion in Section 5.4.3 that the topological arrangement makes the function model much more informative to the designer than a mere listing of the functions and flows.

Second, the reduction of the topological uncertainty in Approach-2 from Approach-1 indicates that by formally representing the topological knowledge, the uncertainty of the models can be significantly reduced. While in Approach-1 the non-observer has to exhaust all options of origins and destinations to determine the topology of the model, the representation of the rules in Approach-2 makes more knowledge available to the designer for analyzing or interpreting the models, thereby requiring less uncertainty to be resolved. Essentially, fewer questions need to be asked.

In terms of expressiveness, the enhanced vocabulary is more expressive than the original Functional Basis verb set, as it gives the designer more information about the rules that control the model before creating a function model. As discussed, a measure for expressive power of a representation is to test if it supports models that can be created by other representations (mappability). The enhanced vocabulary is isomorphic to the Functional Basis function set, meaning that for every function in the Functional Basis, there is a template in the enhanced vocabulary. Additionally, the enhanced vocabulary contains compatibility and connection information about the functions that embody knowledge about the *relations* between the functions. Thus, the expressiveness of the new vocabulary is higher than the Functional Basis in terms of *types of elements*.

Additionally, the new vocabulary can be used to create function models without relying on human judgment of function-to-flow compatibility. This change potentially supports more consistent and objective function modeling than the Functional Basis. The inconsistencies in the models stored in the Design Repository (Figure 18) result from a lack of a topological formalism that can be addressed with these rules.

Finally, the experimental results begin to illustrate that developing formalisms for controlling the topology of function structures is beneficial, both in terms of expressiveness of the vocabulary and consistency of the models. The compatibility rules and connection rules used here to enhance the Functional Basis vocabulary are nothing but formal representations of the function definitions that already exist within the Design Repository. However these definitions do not contribute to constructing models as they exist only in text-based format, which is reliable only for human interpretation. This experiment and the associated development of the new function representation demonstrate a means to capture this semantic information into a formal representation so that function modeling becomes more formal, potentially computer implementable, and more consistent.

Some limitations of the function templates are the over-definition and non-uniqueness. Resolving these issues potentially enhances the expressiveness of the vocabulary farther but requires additional graphical and logical elements. However, despite these limitations, the main idea of increasing expressiveness of the vocabulary has been demonstrated here. In the following chapter, the overall conclusions of this thesis are presented and some opportunities for future extensions to this research are identified.

CHAPTER SEVEN: ANSWERS TO THE RESEARCH QUESTIONS AND OVERALL CONCLUSIONS

Based on the two experiments presented in Chapter Five and Chapter Six, the research questions presented in the beginning of the thesis (Section 1.3) are answered. In the following sections, the sub-questions are answered first, and the main questions are answered later by combining the answers to the sub-questions.

7.1 Answers to RQ-1 and its Sub-Questions

RQ-1.a.	What metric should be used to quantify the usefulness of a function structure?
Answer 1.a.	The usefulness of different function structures constructed with the same vocabulary can be compared by comparing their information content, which can be quantified using Eq.8 or Eq.10 in terms of the size of the vocabulary and the sizes of the function structures.

As discussed in Section 4.4, the size of the model and the vocabulary can be interpreted in two ways: *element-wise*, and *combined*. Element-wise information content accounts for the separate contributions from the functions and flows toward the total information content of the model. The combined metric considers the entire model as a single source of information, ignoring the separate identities of the functions and flows. Therefore, this metric is insensitive to the ratio of functions to flows in specific models or in specific vocabularies.

RQ-1.b.	What metric should be used to quantify the usefulness of a vocabulary?
Answer 1.b.	The information metric presented in Eq.8 and Eq.10, combined with information density (Section 4.7.2) can be used to measure the usefulness of the vocabulary.

Information content indicates the number of questions that can be answered about the product using a function structure, while information density measures information produced by each term in a vocabulary, indicating the compactness of information in the vocabulary. To measure the usefulness of the vocabulary, both of these metrics need to be used. Information density alone cannot serve this purpose, as a high density can be achieved simply by limiting the size of the vocabulary. For example, the primary level of the Functional Basis nouns has the highest information density of all levels (Figure 27) because it has only three terms. In this case, information density indicates a high usefulness of that level, which is false, as the low specificity of terms prevent the expression of useful details of the product. This *false-positive* identification of the primary level can be prevented by comparing information content between the three levels, which reveals that the primary level produces the least information of all levels (Figure 25).

RQ-1.c.	What is the practical interpretation of the metric of usefulness?
Answer 1.c.	The information content of a function structure represents the number of questions that can be answered about the product using the model. Considering that the lack of information represents uncertainty, the metric

measures the total uncertainty apparent to a designer about the product's functions.

These viewpoints are discussed in Section 4.5 and Figure 19 where a designer tries to replicate a function structure by asking questions to another designer who is observing the model. Notably, the function structures are composed of only three sources of information: functions, flows, and connections. Under this scheme, the non-observer can completely determine each component by asking I_V questions for the functions, I_N questions for the flows, and I_T questions for the connections. At this point, the non-observer has enough information to reconstruct the model. As this reconstructed model is identical in terms of the three components of information, it can be argued that all design activities and reasoning that could be supported by the original function structure can also be supported by the reconstructed model. In this sense, all the value or information stored in the original model is transferred to the non-observer, and the number of questions can be used to measure this information. In the second viewpoint, the non-observer is initially uncertain about the model. With the answer to each question this uncertainty is gradually reduced, diminishing to zero when all the facts (functions, flows, connections) about the model are known.

RQ-1.d. Is the assessment of the hierarchy supported by experimental results?

Answer 1.d. Yes, the findings of Experiment-I in Chapter Five are in agreement with the previously conducted empirical study [Caldwell et al., 2008] that proved that the secondary level is the most used level in the function structures within the Design Repository.

The empirical study [Caldwell et al., 2008] examined approximately 10% of the function structures stored in the Design Repository, and found that more than 90% of the Functional Basis terms in those models are drawn from the secondary level. This result indicates that the secondary level is potentially the useful of the three. The analysis of experiment-I (Sections 5.3 and 5.4.1) indicate that the secondary level has the most favorable combination of information content and information density, making it the most useful for constructing function structures. This analysis explains the results of the empirical study. By combining the answers to the above four sub-questions, RQ-1 can now be answered.

RQ-1.	Are the hierarchical levels of the Functional Basis equally useful for constructing function structures?
Answer 1.	No, the secondary level of the Functional Basis is more useful than the other two levels for constructing function structures, as it results into the best combination of information content and information density.

7.2 <u>Answers to RQ-2 and its Sub-Questions</u>

RQ-2.a.	What metric should be used to measure the representation's expressiveness?
Answer 2.a.	A comparative score of information content of the same function structure created with different representations can be indirectly used to compare the expressiveness of the representations, as a more expressive representation needs less information to describe the same product.

When the same representation and vocabulary are used to create different models, the model with the higher information content is more useful, as information content is proportional to the number of elements in the model, and therefore, is commensurate to the amount of design information presented there. By contrast, when two representations are used to construct function structures of the same artifact, the representation that results into the lower information content is considered to be more expressive, as less information was necessary to describe the artifact using that representation.

RQ-2.b.	Which elements can be formally represented to increase the expressiveness?
Answer 2.b.	The definitions of the Functional Basis verbs can be formally captured to produce the triple-based notation of functions, which combine the topological rules within the function templates, making them more expressive than the text-based representation of verbs in the Functional Basis.

In the text-based representation of verbs, no rules for topological connections are explicitly captured. Hence, each flow can potentially originate from any function in the model and terminate to any other function, leading to high level of uncertainty about the topological connections. By formally capturing the verb definitions, the rules for topological compatibility can be derived, which control the number of functions that can be origin or destination for a given flow. As a result, the topological uncertainty of the model decreases, requiring less information for describing the product, which implies that the new representation is more expressive than the text-based description of verbs in the

Functional Basis. Based on the answers to the above three sub-questions, the overall question RQ-2 can be answered now, as shown below.

RQ-2.	How can the function structure representation be made more expressive?
Answer 2.	The graph-based function structure representation can be made more expressive by formally capturing the definitions of the function verbs in the Functional Basis vocabulary, and formulating rules for the topological connections between the functions and the flows.

7.3 <u>Thesis Contributions and Concluding Remarks</u>

The main contributions of this thesis are twofold. First, this thesis presents a means to mathematically compute the information content of function structures and vocabularies. It presents two metrics of information content: element-wise and combined, and three ways of interpreting the vocabulary: fixed, used, and reduced. Additionally, a metric of information density of a vocabulary is presented. Further, these metrics are applied to the Functional Basis vocabulary to show that the hierarchy of its terms is not useful to support construction of function structures, as the secondary level has a much higher usefulness than the other two. This analysis provides a theoretical support to the empirical findings that the secondary level is used much more frequently than the other two levels [Caldwell et al., 2008].

However, the information metric cannot accurately estimate the usefulness of a model or a vocabulary unless a rigorous formalism is established for function modeling. In the absence of formalism, function structures are subject to representational inconsistencies, as shown in Section 3.4.4. As a result, any information score of a model would be inconclusive, as explained in Section 5.4.2. In analyzing the problem of formalism, a critical gap in the representation of artifact functionality is identified in this thesis: the lack of formalism in the topological construction of function structures. As a solution, a novel representation of function is presented, which defines a function in terms of a triple {*Name, In_List, Out_List*}, and accompanying topological rules. It has been shown that the new representation of functions can be applied to each verb in the Functional Basis vocabulary, essentially producing an evolved vocabulary that is isomorphic to the Functional Basis, yet, where every term is more expressive than the text-based description of verbs and nouns. Although the adequacy of the Functional Basis terms in constructing function structures has been challenged in previous research [Caldwell et al., 2008], it is noted that a vocabulary can be used to enforce consistency of term selection in function models. With this evolved version, this consistency can be extended to the model topology.

Notably, the information metric developed here is an indirect surrogate to the usefulness of function structures, as it measures the information captured *within* the models, rather than directly measuring what the designer can achieve by using it. Hence, the metric needs to be *externally* validated to test how closely it reflects the usefulness of models perceived by designers. This validation can be performed through a human-subject experiment, and determining if the function structures with higher information content are identified to be more useful to the designers. Such exercises can also reveal which entities of a function structure, such as the functions, flows, topology, or the vocabulary, is the largest contributors to the external usefulness of the model.

CHAPTER EIGHT: FUTURE RESEARCH DIRECTIONS

In order to develop a comprehensive representation of functions to support automated interpretation and reasoning, first the aspects of the domain that are valuable to the designer need to be identified to defined. In this chapter, eight outstanding issues are discussed that need to be formally represented and integrated with each other in order to further formalize the domain of artifact functionality.

8.1 Environmental Context of the Artifact

One limitation of the function structure representation is that it describes only the inner workings of the artifact but does not explicitly capture the interaction of the artifact with the environment. A recent study shows that the inclusion of environ-specific terms in a function structure makes it more interpretable to the human designer [J. Thomas et al., 2009]. These terms provide some contextual information about the product that is difficult to capture using controlled vocabularies. These additional parameters are called the environmental context of the product. Even if a representation is developed for this information, integrating that representation with the existing formalism of function structures remains as a challenge.

8.2 User Interaction with the Artifact

The user can be viewed as an entity in the artifact's environment, or as a separate entity, formulating the artifact-user-environment triple that describes the situatedness of the artifact in its surroundings. However, unlike the artifact or the environment, the user is a conscious agent empowered to choose the application of an artifact in a given environment. The affordance-based view of functionality is based upon this issue. Therefore, one way of addressing the interaction within the above-mentioned triple could be to develop a formal representation of affordances and integrate that with the remainder of the model (artifact and environment). However, due to the user's ability to choose, the positive and negative affordances are defined for the artifact-environment duality, rather than the artifact alone. This analysis illustrates the challenges and complexities involved in the modeling of user interaction in the unified model of functionality.

8.3 Function, Behavior, and Side Effects

Function and behavior of an artifact are related through its side effects. For example, the function (intended actions) of an incandescent lamp is to provide light, while the behavior (actual actions) includes both light and heat produced by the lamp. The difference between function and behavior is, therefore, the side effect: heat. To the designer, who wants to design a light-producing device, this side effect is undesired, as it leads to loss of efficiency. However, the side effects are sometimes utilized by the user to their benefit. For example, a lamp can be used to keep food warm in the display boxes of a cafeteria. Therefore, side effect can be desired or undesired, and differently viewed by the user or the designer. In each case, the perception is dependent not only on the artifact (lamp), but also on the environment (cafeteria, food).

8.4 Conservation of Mass and Energy

Mass and energy are conservable entities of the universe. Therefore, for logical consistency, the flows of material and energy across any sub-function or collection of sub-functions within a model must be conserved. By enforcing conservation the model can be reasoned upon for product efficiency. In the case of the incandescent lamp, if both light and heat energies at the output are explicitly modeled and the conservation of the input electrical energy is accounted for between these two outputs, the resulting model can support reasoning such as if the lamp is used as a light source, and has an efficiency of 45%, then it will be 100-45=55% efficient, when used as a heat source.

8.5 Representation of Signals

Signals are not physical entities, but information encoded in the state of an entity. For example, the needle of a magnetic compass itself is not North or South, it is a parameter related to the needle, namely, the *direction of the needle when suspended freely from its center of gravity* that represent North. Similarly, the light coming out of a traffic signal can be modeled as energy, as the conservation principles requires an energy output to account for the input electrical energy. However, it is a parameter of the light, namely *color* or *frequency,* that acts as the signal to a driver. By modeling the light as a signal, other signal-producing concepts, such as a colored flag, can be conceptualized. Therefore, both approaches of modeling the light, as energy or signal, have their benefits.

8.6 Logical Relations between Functions and Flows, and the States of the Artifact

Presently function structures only depict one state of the artifact at a time. However, many products operate on multiple states in time. For example, the function of a storage cell is "to store electrical energy" when it is charging, which changes to "to supply electrical energy" during discharging. These two states are connected by logical a relation, which in this case implies that both states cannot exist at the same time. By extending the representation to depict both states in the same model, the representation can be made more expressive.

8.7 <u>Representation of Flow Attributes</u>

In the hair dryer function structure of Figure 9, the adjective *hot* attached to the noun *air* indicates the necessity of capturing the states of a flow under transformation. In fact, other than the function *Convert,* which implies the transformation of one flow type to another, all other Functional Basis verbs imply a change of an attribute associated with the flow. For example, *Transfer* implies a change in flow location, and *Mix* between air and thermal energy implies a rise in temperature of the air. The flow attributes can be modeled using tuples like {voltage, current, cycle} for electrical energy, which completely defines the state of the energy. Similarly, for mechanical energy, the list {torque, speed, direction} can be used to define a state

8.8 <u>Scalability under Decomposition and Composition</u>

While developing representations for all the above aspects of functionality, the scalability of the model under composition and decomposition needs to be ensured. For example, under decomposition, the flows that cross the system boundary of the resulting decomposed model must be exactly the same flows that were input and output to the black-box model. Similarly, there should be a formal mechanism so that each sub-function identifies the other sub-functions that exchange flows with it as its environment, and by algebraically adding the environment for each sub-function, the environment for the black-box function must be obtained. At present, such consistency has to be manually enforced in function structures.

Ultimately, a unified representation of artifact functionality that addressed all of the above issues in a coherent fashion is sought. However, these issues and their inter-dependencies can make this problem complex. For practical usability, a consistent model with limited scope is more preferred than a broad model with inconsistent behavior. Therefore, the development of a unified model of artifact functionality should be approached incrementally, ensuring consistent behavior at each step. The ultimate motivation is to develop formal representations for all of the above issues and integrate them into a unified and consistent representation, which will support automated description, interpretation, and reasoning of product functions.

APPENDICES

Appendix A: Definition of Function Verbs within the Design Repository

The definitions of the Functional Basis verbs are shown in Table 16. In this table, the verbs are listed in the left column and the definitions of those verbs stored in the Design Repository are listed in the right column. For identification of the hierarchical levels, the primary verbs are marked with one dot, while the secondary and tertiary verbs are marked with two and three dots respectively.

Table 16: Definition of Functional Basis verbs within the Design Repository

Verb	Definition text within the Design Repository
• Branch	To cause a flow (material, energy, signal) to no longer be joined or mixed.
•• Separate	To isolate a flow (material, energy, signal) into distinct components. The separated components are distinct from the flow before separation, as well as each other.
•• Distribute	To cause a flow (material, energy, signal) to break up. The individual bits are similar to each other and the undistributed flow.
• Channel	To cause a flow (material, energy, signal) to move from one location to another location.
•• Import	To bring in a flow (material, energy, signal) from outside the system boundary.
•• Export	To send a flow (material, energy, signal) outside the system boundary.

Verb	Definition text within the Design Repository
•• Transfer	To shift, or convey, a flow (material, energy, signal) from one place to another.
••• Transport	To move a material from one place to another.
••• Transmit	To move an energy from one place to another.
•• Guide	To direct the course of a flow (material, energy, signal) along a specific path.
••• Translate	To fix the movement of a flow by a device into one linear direction.
••• Rotate	To fix the movement of a flow by a device around one axis.
••• Allow DOF	To control the movement of a flow by a force external to the device into one or more directions.
• Connect	To bring two or more flows (material, energy, signal) together.
•• Couple	To join or bring together flows (material, energy, signal) such that the members are still distinguishable from each other.
••• Join	To couple flows together in a predetermined manner.
••• Link	To couple flows together by means of an intermediary flow.
•• Mix	To combine two flows (material, energy, signal) into a single, uniform homogeneous mass.
• Control Magnitude	To alter or govern the size or amplitude of a flow (material, energy, signal).

Verb	Definition text within the Design Repository
•• Actuate	To commence the flow of energy, signal, or material in response to an imported control signal.
•• Regulate	To adjust the flow of energy, signal, or material in response to a control signal, such as a characteristic of a flow.
••• Increase	To enlarge a flow in response to a control signal.
••• Decrease	To reduce a flow in response to a control signal.
•• Change	To adjust the flow of energy, signal, or material in a predetermined and fixed manner
••• Increment	To enlarge a flow in a predetermined and fixed manner.
••• Decrement	To reduce a flow in a predetermined and fixed manner.
••• Shape	To mold or form a flow.
••• Condition	To render a flow appropriate for the desired use.
•• Stop	To cease, or prevent, the transfer of a flow (material, energy, signal).
••• Prevent	To keep a flow from happening.
••• Inhibit	To significantly restrain a flow, though a portion of the flow continues to be transferred.
• Convert	To change from one form of a flow (material, energy, signal) to another. For completeness, any type of flow conversion is valid. In practice, conversions such as convert electricity to

Verb	Definition text within the Design Repository
	torque will be more common than convert solid to optical energy.
•• Convert	To change from one form of a flow (material, energy, signal) to another. For completeness, any type of flow conversion is valid. In practice, conversions such as convert electricity to torque will be more common than convert solid to optical energy.
• Provision	To accumulate or provide a material or energy flow.
•• Store	To accumulate a flow.
••• Contain	To keep a flow within limits.
••• Collect	To bring a flow together into one place.
•• Supply	To provide a flow from storage.
• Signal	To provide information on a material, energy or signal flow as an output signal flow. The information providing flow passes through the function unchanged.
•• Sense	To perceive, or become aware, of a flow.
••• Detect	To discover information about a flow.
••• Measure	To determine the magnitude of a flow.
•• Indicate	To make something known to the user about a flow.
••• Track	To observe and record data from a flow.

Verb	Definition text within the Design Repository
••• Display	To reveal something about a flow to the mind or eye.
•• Process	To submit information to a particular treatment or method having a set number of operations or steps.
• Support	To firmly fix a material into a defined location, or secure an energy or signal into a specific course.
•• Stabilize	To prevent a flow from changing course or location.
•• Secure	To firmly fix a flow path.
•• Position	To place a flow (material, energy, signal) into a specific location or orientation.

Appendix B: Information Content of the Delta Jigsaw Function Structure using the Functional Basis Vocabulary

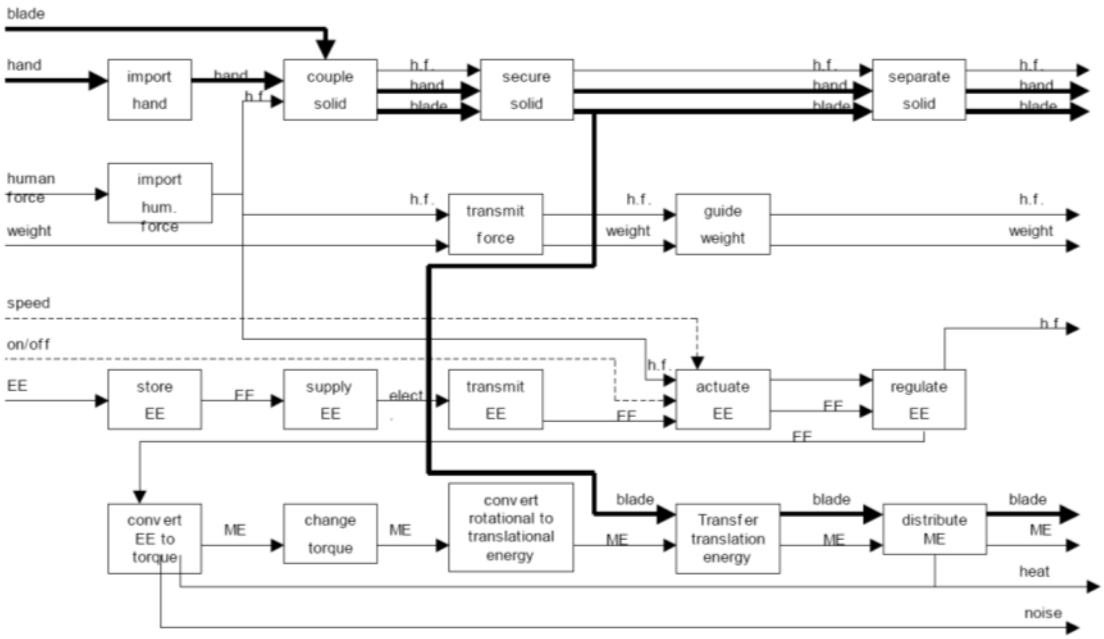


Figure 31: Function structure of the Delta jigsaw

Table 17: Results: Information content of the Delta jigsaw function structure

Noun Levels ↑	F	U	R	F	U	R	F	U	R	F	U	R
3	252	168	168							413	295	354
2	210	126	210				354	295	354			
1	84	84	84	236	236	236						
0				51	51	51	85	68	85	102	68	85
	0			1			2			3		

Verb Levels →

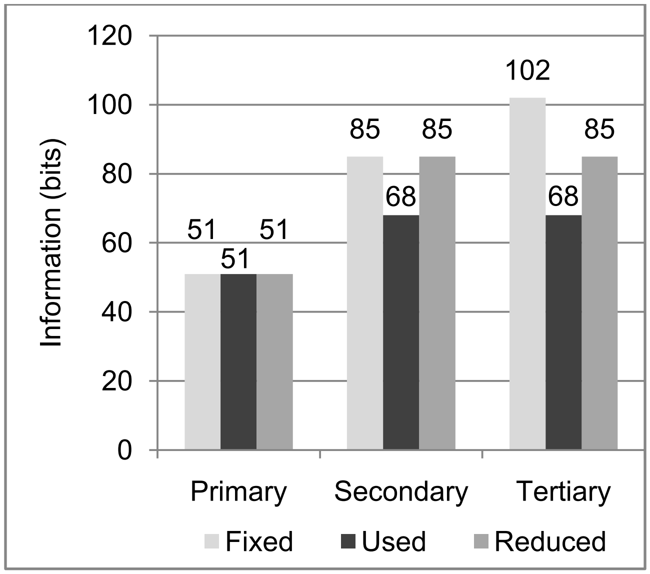


Figure 32: Delta jigsaw information content: Verbs only: M(1,0), M(2,0), M(3,0)

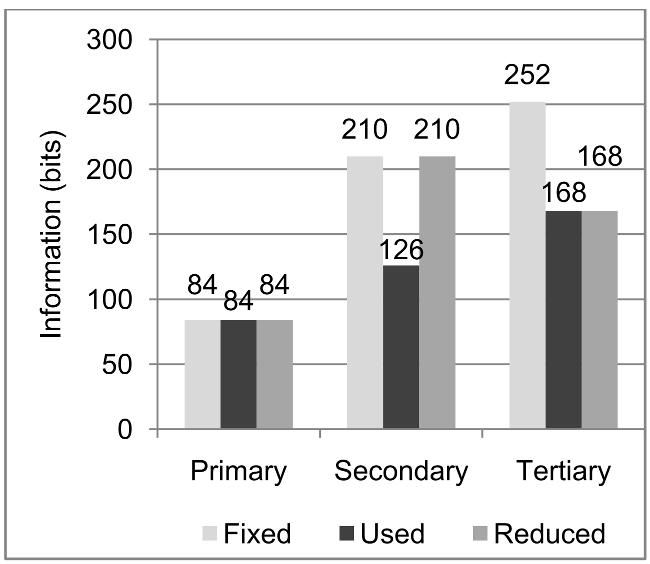


Figure 33: Delta jigsaw information content: Nouns only: M(0,1), M(0,2), M(0,3)

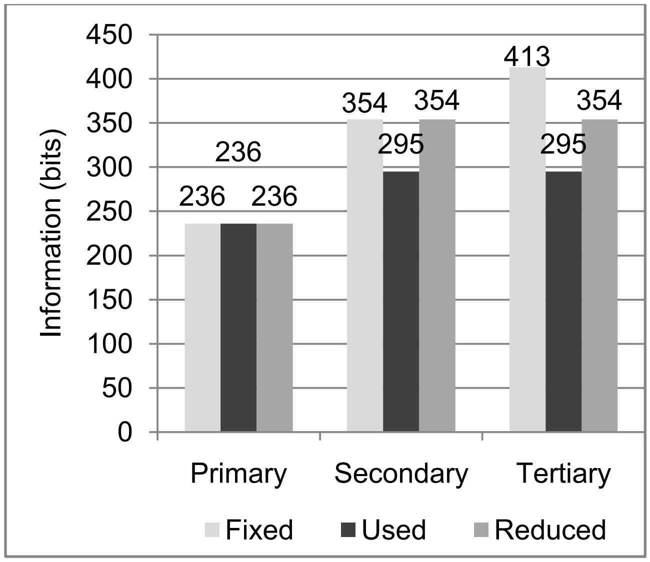


Figure 34: Delta jigsaw information content: Combined: M(1,1), M(2,2), M(3,3)

Appendix C: Information Content of the Brother Sewing Machine Function Structure using the Functional Basis Vocabulary

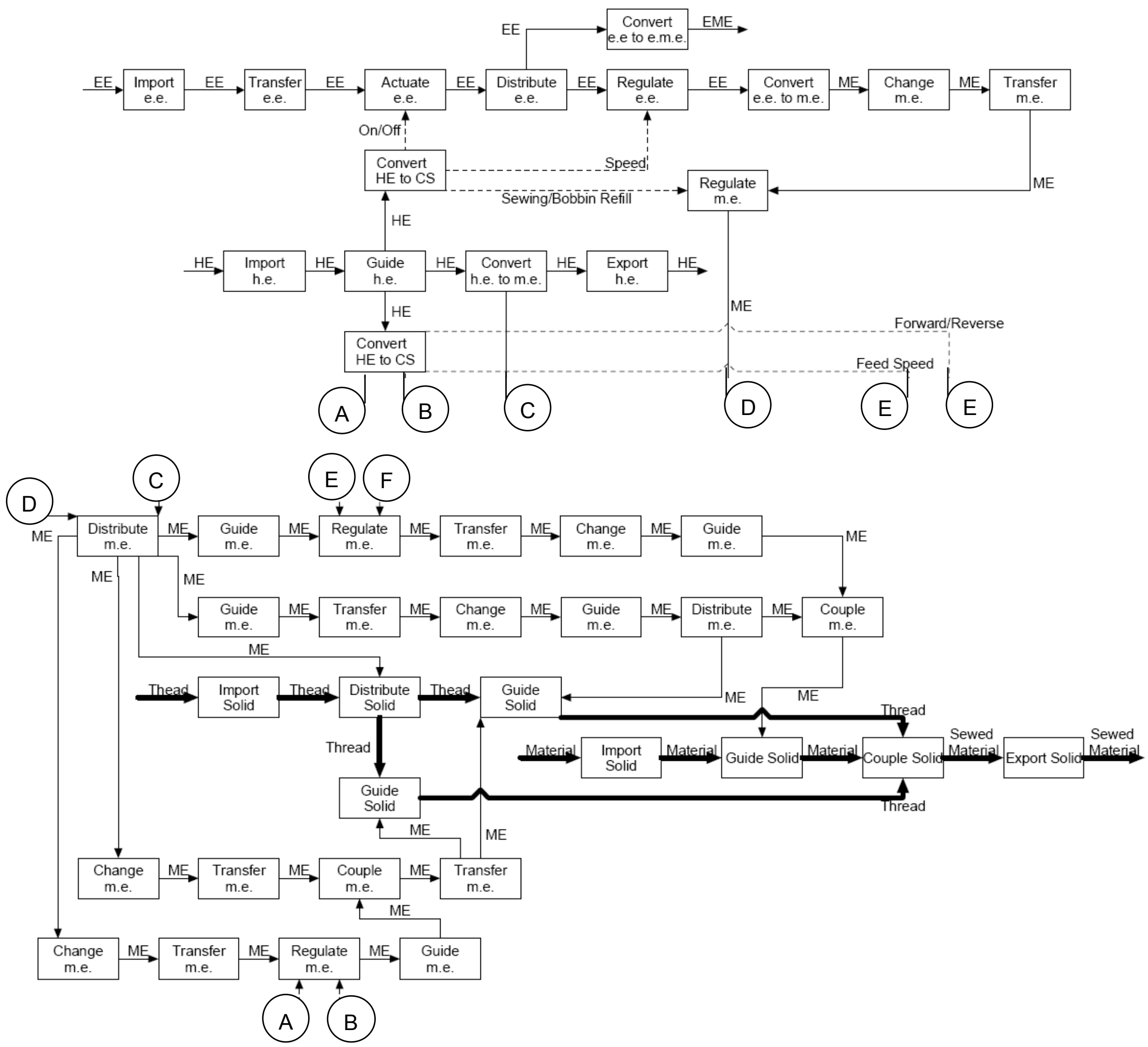


Figure 35: Function structure of Brother sewing machine

(Each numbered flow from the top half connects to the corresponding numbered flow in the bottom half)

Table 18: Results: Information content of Brother sewing machine function structure

Noun Levels →		F	U	R	F	U	R	F	U	R	F	U	R
	3	384	256	256							756	540	648
	2	320	192	320				648	540	648			
	1	128	128	128	432	324	432						
	0				132	132	132	220	176	176	264	176	220
		0			1			2			3		

Verb Levels →

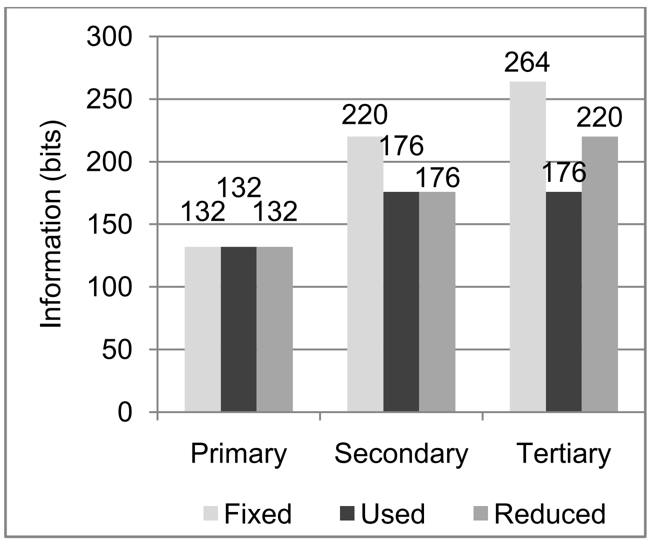


Figure 36: Brother sewing machine information content: Verbs only: M(1,0), M(2,0), M(3,0)

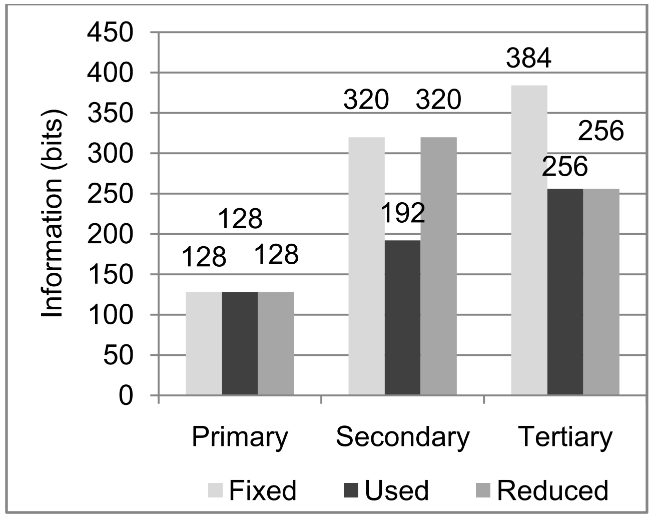


Figure 37: Brother sewing machine information content: Nouns only: M(0,1), M(0,2), M(0,3)

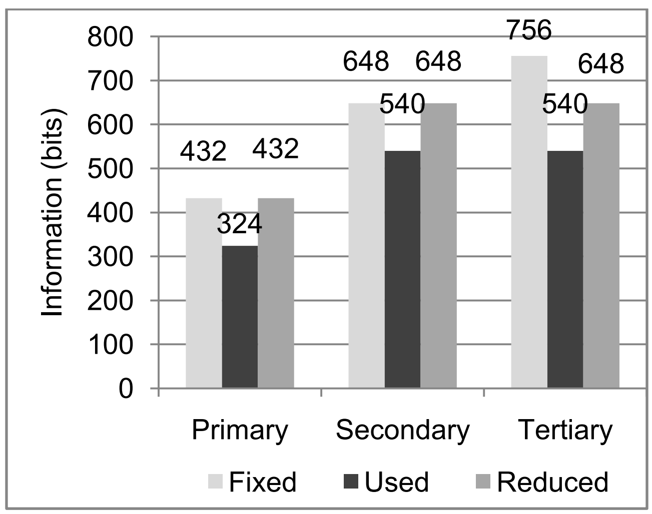


Figure 38: Brother sewing machine information content: Combined: M(1,1), M(2,2), M(3,3)

REFERENCES

Albers, A., Matthiesen, S., Thau, S., & Alink, T. (2008). Support of Design Engineering Activity through C&CM - Temporal Decomposition of Design Problems. *TMCE 2008 Symposium*. Izmir, Turkey.

Ameri, F., Summers, J.D., Mocko, G.M., & Porter, M. (2008). Engineering Design Complexity: An Experimental Study of Methods and Measures. *Research in Engineering Design 19(2-3)*, 161-179.

Arunajadai, S.G., Stone, R.B., & Tumer, I.Y. (2002). A Framework for Creating a Function-Based Design Tool for Failure Mode Identification. *ASME 2002 Design Engineering Technical Conference and Computers and Information in Engineering Conference*. Montreal, Canada.

Baader, F. (1996). A Formal Definition for the Expressive Power of Terminological Knowledge Representation Languages. *Journal of Logic and Computation 6(1)*, 33-54.

Baumgart, B.G. (1974). Geometric Modelling for Computer Vision. *Artificial Intelligence Report Number CS-463*. Stanford University Stanford, CA, USA.

Bhatta, S., Goel, A., & Prabhakar, S. (1994). Innovation in Analogical Design: A Model-Based Approach. *Third International Conference on Artificial Intelligence in Design (AID-94)*. Lausanne, Switzerland.

Bobrow, D.G. (1984). Qualitative Reasoning about Physical Systems: An Introduction. *Artificial Intelligence 24(1-3)*, 1-5.

Bohm, M.R., & Stone, R.B. (2004). Representing Functionality to Support Reuse: Conceptual and Supporting Functions. *24th Computers and Information in Engineering Conference*, Vol. 4, pp. 411-419. American Society of Mechanical Engineers, New York, NY 10016-5990, United States, Salt Lake City, UT, United States.

Bohm, M.R., Stone, R.B., Simpson, T.W., & Steva, E.D. (2006). Introduction of a Data Schema: The Inner Workings of a Design Repository. *IDETC/CIE 2006. ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Philadelphia, PA, USA.

Bohm, M.R., Stone, R.B., & Szykman, S. (2003). Enhancing Virtual Product Representations for Advanced Design Repository Systems. *23rd Computers and Information in Engineering Conference*, Vol. 1 B, pp. 635-645. American Society of Mechanical Engineers, New York, NY 10016-5990, United States, Chicago, IL, United States.

Bohm, M.R., Stone, R.B., & Szykman, S. (2005). Enhancing Virtual Product Representations for Advanced Design Repository Systems. *Journal of Computing and Information Science in Engineering 5(4)*, 360-372.

Brenner, S.C., & Scott, L.R. (2008). The Mathematical Theory of Finite Elements Methods. Springer Science & Business Media Inc., New York, NY, USA.

Bryant, C.R., McAdams, D.A., & Stone, R.B. (2006). A Validation Study of an Automated Concept Generator Design tool. *ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Philadelphia, Pennsylvania, USA.

Caldwell, B.W., & Mocko, G.M. (2007). Towards Rules for Functional Composition. *ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Brooklyn, New York, USA.

Caldwell, B.W., Sen, C., Mocko, G.M., Summers, J.D., & Fadel, G.M. (2008). Empirical Examination of the Functional Basis and Design Repository. *Third International Conference on Design Computing and Cognition*. Atlanta, Georgia, USA.

Carter, T.J. (2006). An Introduction to Information Theory and Entropy. Complex Systems Summer School, California State University, Stanislaus, CELLULAR AUTOMATON, USA. Available at: http://csustan.csustan.edu/~tom/SFI-CSSS/2006/info-lec.pdf

Chandrasekaran, B. (2005). Representing Function: Relating Functional Representation and Functional Modeling Research Streams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 19*, 65-74.

Chandrasekaran, B., & Josephson, J.R. (1997). Representing Function as Effect. *Fifth International Workshop on Advances in Functional Modeling of Complex Technical Systems*, (Modarres, M., Ed.), pp. 3-16. Paris, France.

Chandrasekaran, B., & Josephson, J.R. (2000). Function in Device Representation. *Engineering with computers 16*, 162-177.

Chen, P.P. (1976). The Entity-Relationship Model - Toward A Unified View of Data. *ACM Transactions on Database Systems 1(1)*, 9-36.

Chen, P.P. (1980). Entity-Relationship Approach to Systems Analysis and Design. *First International Conference on the Entity-Relationship Approach*. North-Holland.

Chen, P.P. (1981). Entity-Relationship Approach to Information Modeling and Analysis. *Second International Conference on the Entity-Relationship Approach*. Washington, DC, USA.

Collins, J.A., Hagan, B.T., & Bratt, H.M. (1976). Failure-Experience Matrix - A Useful Design Tool. *Journal of Engineering for Industry B 98(3)*, 1074-1079.

Coste-Marquis, S., Lang, J., Liberatore, P., & Marquis, P. (2004). Expressive Power and Succinctness of Propositional Languages for Preference Representation. *9th International Conference on Principles of Knowledge Representation and Reasoning (KR'04)*, pp. 203-212. Whistler, Canada.

Erden, M.S., Komoto, H., VanBeeK, T.J., D'Amelio, V., Echavarria, E., & Tomiyama, T. (2008). A Review of Function Modeling: Approaches and Applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 22*, 147-169.

Gero, J.S. (1990). Design Prototypes: A Knowledge Representation Schema for Design. *AI Magazine 11(4)*, 26-36.

Gero, J.S., & Kannengiesser, U. (2000). Towards a Situated Function-Behaviour-Structure Framework as the Basis for a Theory of Designing. *Workshop on Development and Application of Design Theories in AI in Design Research*. T. Smithers, Worcester, Massachusetts.

Gero, J.S., & Kannengiesser, U. (2002). The Situated Function-Behaviour-Structure Framework. *Artificial Intelligence in Design*, 89-104.

Goel, A.K., & Bhatta, S.R. (2004). Use of Design Patterns in Analogy-Based Design. *Advanced Engineering Informatics 18*, 85-94.

Hartley, R.V.L. (1928). Transmission of Information. *Bell System Technical Journal 7(1928)*, 535-563.

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

Hopcroft, J., Motwani, R., & Ullman, J. (2001). Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, New York, NY, USA.

J. Thomas, Sen, C., Summers, J.D., & Mocko, G.M. (2009). Investigation of Interpretability of Three Function Representations. *2009 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, San Diego, CA, USA (accepted).

Kernighan, B.W., & Ritchie, D.M. (2004). The C Programming Language. Person Education (Singapre) Pte. Ltd., Indian branch, New Delhi, India.

Keuneke, A.M. (1991). Device Representation - The Significance of Functional Knowledge. *IEEE Expert 6(2)*, 22-25.

Kirschman, C.F., & Fadel, G.M. (1998). Classifying Functions for Mechanical Design. *Journal of Mechanical Design 120(3)*, 475-482.

Kruse, R.L., & Ryba, A.J. (1999). Data Structures and Program Design in C++. Prentice-Hall, Inc., Upper Saddle River, New Jersey.

Kurfman, M.A., Stock, M.E., Stone, R.B., Rajan, J., & Wood, K.L. (2003). Experimental Studies Assessing the Repeatability of a Functional Modeling Derivation Method. *Transactions of the ASME 125*, 682-693.

Kurfman, M.A., Stone, R.B., Wie, M.V., Wood, K.L., & Otto, K.N. (2000). Theoretical Underpinnings of Functional Modeling: Preliminary Experimental Studies. *2000 ASME Design Engineering Technical Conference*. Baltimore, Maryland.

Kurtoglu, T. (2007). A Computational Approach to Innovative Conceptual Design. *Mechanical Engineering*, Vol. PhD, p. 155. University of Texas, Austin.

Kurtoglu, T., Campbell, M.I., Bryant, C.R., Stone, R.B., & McAdams, D.A. (2005a). Deriving a Component Basis for Computational Functional Synthesis. *International Conference on Engineering Design, ICED '05*. Melbourne.

Kurtoglu, T., Campbell, M.I., Gonzales, J., Bryant, C.R., & Stone, R.B. (2005b). Capturing Empirically Derived Design Knowledge for Creating Conceptual Design Configurations. *2005 ASME International Design Engineering and Technical Conference*. Long Beach, California, USA.

Linz, P. (2006). An Introduction to Formal Languages and Automata. Narosa Publishing House, New Delhi.

Luger, G.F. (2002). Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Addiaon-Wesley, Essex, England.

Maier, J.R.A. (2008). Rethinking Design Thoery. *Mechanical Engineering 130(9)*, 34-37.

Maier, J.R.A., & Fadel, G.M. (2001). Affordance: The Fundamental Concept in Engineering Design. *2001 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Pittsburgh, Pennsylvania, USA.

Maier, J.R.R., & Fadel, G.M. (2002). Comapring Function and Affordance as Bases for Design. *Proceedings of ASME 2002 Design Engineering Technical Conferences and Computer and Information in Engineering Conference*. Montreal, Canada.

Mateescu, A., & Salomaa, A. (1997). Aspects of Classical Language Theory. *Handbook of formal languages*, (Rozenberg, G., & Salomaa, A., Eds.), Vol. 1. Springer-Verlag, New York, NY, USA.

McAdams, D.A., & Wood, K. (2002). A Quantitative Similarity Metric for Design-by-Analogy. *Journal of Mechanical Design 124(June 2002)*, 173-182.

Minsky, M.L. (1965). Matter, Mind and Models. *IFIP Congress*, Vol. 1, pp. 45-49. Washington, D.C., USA.

Mocko, G.M., Summers, J.D., Fadel, G.M., Teegavarapu, S., Maier, J.R.A., & Ezhilan, T. (2007). A Modelling Scheme for Capturing and Analyzing Multi-Domain Design Information: A Hair Dryer Design Example. *16th International Conference on Engineering Design*. Paris, France.

Nebel, B. (2000). On the Compilability and Expressive Power of Propositional Planning Formalisms. *Journal of Artificial Intelligence Research 12*, 271-315.

Noon, R.K. (2001). Forensic Engineering Investigation. CRC Press LLC, Boca Raton, FL, USA.

Otto, K.N., & Wood, K.L. (2001). Product Design Techniques in Reverse Engineering and New Product Development. Prentice Hall, Upper Saddle River, NJ.

Pahl, G., Beitz, W., Feldhusen, J., & Grote, K.H. (2007). Engineering Design: A Systematic Approach. Springer-Verlag London Limited, London.

Radhakrishnan, R., & McAdams, D.A. (2005). A Methodology for Model Selection in Engineering Design. *Transactions of the ASME 127(May 2005)*, 378-387.

Rich, E., & Knight, K. (1991). Artificial Intelligence. McGraw-Hill, Inc., New York, NY, USA.

Rodenacker, W. (1971). Methodisches Konstruieren. Springer-Verlag, Berlin, Germany.

Sen, C., Caldwell, B.W., Summers, J.D., & Mocko, G.M. (2010). Evaluation of the Functional Basis using an Information Theoretic Approach. *Artificial Intelligence for Engineering Design, Analysis & Manufacturing 24(1)*.

Shannon, C.E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal 27*, 379-423 and 623-656.

Sridharan, P., & Campbell, M.I. (2004). A Grammar for Function Structures. *ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Salt Lake City, UT, USA.

Sridharan, P., & Campbell, M.I. (2005). A Study on the Grammatical Construction of Function Structures. *Artificial Intelligence for Engineering Design, Analysis & Manufacturing 19*, 139-160.

Stone, R.B., Tumer, I.Y., & Stock, M.E. (2005). Linking Product Functionality to Historic Failures to Improve Failure Analysis in Design. *Research in Engineering Design 16(2)*, 96-108.

Stone, R.B., & Wood, K.L. (2000). Development of a Functional Basis for Design. *Journal of Mechanical Design 122(4)*, 359-370.

Suh, N.P. (1990). The Principles of Design. Oxford University Press Inc., New York.

Summers, J.D. (2005). Expressiveness of the Design Exemplar. *ASME 2005 International Design Engineering & Technical Confernces & Computers and Information in Engineering Conferences*. Long Beach, California, USA.

Summers, J.D., & Ameri, F. (2008). An Algorithm for Assessing Design Complexity through a Connectivity View. *Proceedings of the TMCE 2008*. I. Horváth and Z. Rusák, Izmir, Turkey.

Summers, J.D., & Shah, J.J. (2003). Developing Measures of Complexity for Engineering Design. *ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Chicago, Illinois USA.

Summers, J.D., Vargas-Hernández, N., Zhao, Z., Shah, J.J., & Lacroix, Z. (2001). Comparative Study of Representation Structures for Modeling Function and Behavior of Mechanical Devices. *DETC2001: Computers in Engineering*. Pittsburgh, Pennsylvania.

Szykman, S., Racz, J.W., & Sriram, R.D. (1999). The Representation of Function in Computer-Based Design. *1999 ASME Design Engineering Technical Conferences*. Las Vegas, Nevada, USA.

Tumer, I.Y., & Stone, R.B. (2001). Analytical Methods to Evaluate Failure Potential during High-Risk Component Development. *2001 ASME Design Engineering Technical Conferences*. Pittsburgh, Pennsylvania, USA.

Ullman, D.G. (1992). The Mechanical Design Process. McGraw-Hill, New York.

Ulrich, K.T., & Eppinger, S.D. (2008). Product Design and Development. McGraw-Hill, New York, NY, USA.

Umeda, Y., & Tomiyama, T. (1995). FBS Modeling: Modeling Scheme of Function for Conceptual Design. *9th. International Workshop on Qualitative Reasoning*.

Vucovich, J., Bhardwaj, N., Ho, H.-H., Ramakrishna, M., Thakur, M., & Stone, R. (2006). Concept Generation Algorithms for Repository-Based Early Design. *ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Philadelphia, Pennsylvania, USA.

Winston, P.H. (1992). Artificial Intelligence. Addison-Wesley Publishing Company, Reading. Massachusettes, USA.

Woods, W.A. (1983). What's Important about Knowledge Representation. *Computer 16*.

Zeid, I. (2007). Mastering CAD/CAM. Tata McGraw-Hill Publishing Company Limited, New Delhi, India.