AN ABSTRACT OF THE DISSERTATION OF

Ryan M. Arlitt for the degree of Doctor of Philosophy in Mechanical Engineering presented on September 11, 2015.

Title: Understanding Designer Mental Models to Support Computer Directed Analogical Design.

Abstract approved:

Robert B. Stone

Analysis of alternative concepts has a significant impact on design project outcomes, and yet many design teams fail to consider a significantly broad range of conceptual solutions. Within the realm of conceptual design exists a technique called design by analogy (DBA) – the practice of reapplying old solutions to new problems. DBA mitigates the effort required to generate a large field of candidate concepts by leveraging existing knowledge from a wide variety of domains, making it an attractive approach toward improving design outcomes. Unfortunately, DBA is challenging in the absence of expert knowledge. Designers need computational support in order to effectively identify a large number of high-quality analogical connections across a wide variety of domains. With this challenge in mind, **the goal of this dissertation is to improve the body of knowledge regarding computational support for design by analogy.** More specifically, this body of work includes five manuscripts. Manuscript 0 presents a review of several function-related design abstractions, including their impacts on education and industry. Manuscript 1 studies analogy retrieval in a novel design context and catalogs the types of abstract similarity (including function) commonly used to form analogies. Manuscript 2 examines a scalable approach to capturing analogy-relevant design knowledge to support large-scale analogy searching. Manuscripts 3 and 4 examine and modify a technique from *de novo* drug design for quickly indexing and retrieving design analogies. Manuscript 3 examines the domain independence of the technique, and manuscript 4 develops it as a large-scale design analogy search method. The body of work contributes to a greater understanding of (1) the abstractions used by designers during conceptual design, (2) the use of human computation to support conceptual design activities, and (3) large scale solution screening using a variety of mixed design abstractions. This understanding advances the creation of tools that enable designers to consider a wide range of conceptual solutions in spite of lacking domain expertise.

Understanding Designer Mental Models to Support Computer Directed Analogical Design

by
Ryan M. Arlitt

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented September 11, 2015
Commencement June 2016

Doctor of Philosophy dissertation of Ryan M. Arlitt presented on September 11, 2015

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial, and Manufacturing Engineering

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Ryan M. Arlitt, Author

ACKNOWLEDGEMENTS

The author wishes to express sincere appreciation to all those who provided him with the opportunities and support that contributed to his successes during his graduate studies. You know who you are.

CONTRIBUTION OF AUTHORS

Dr. Tumer wrote the description of the Function Failure Identification and Propagation framework in manuscript 0. Dr. Sen assisted with experimental design and data collection for manuscript 1. Anthony Nix assisted with data analysis and writing of manuscript 1. Tim Foglesong assisted with an alternate method of analysis for the research presented in manuscript 1, though these efforts did not lead to a publication. Sebastian Immel assisted with programming, task design, data collection, and writing for manuscript 2. Friederich Berthelsdorf assisted with task design, data collection, and writing for manuscript 2. Charlie Manion assisted with graph fragment mining for manuscript 3. Dr. Tumer and Dr. Campbell assisted with planning and writing for manuscript 3.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

Table Page

Introduction

Design is fundamentally distinct from scientific pursuits. While science involves the search for absolute truths, design seeks answers that are *good enough*. Conceptual design in particular can be described as the transformation of a problem space into a solution space – it is the designer's job to generate plausible mappings between these two domains. A variety of solutions is sought at this stage of design – when the solution space is still large and uncertainty is still high.

As a consequence of this large solution space, designers employ a unique set of mental shortcuts to find one or more satisfactory solutions to a problem quickly and easily. One such shortcut is the design analogy – a mapping between two domains that can reveal nonobvious connections between a problem and an existing solution. In drawing analogies, a designer leverages existing knowledge in new contexts, thus providing a tool to efficiently explain newly encountered artifacts, evaluate existing designs, and synthesize new designs. Analogy has proven its value as a design heuristic, but analogy formation is limited by a designer's own narrow experiences and knowledge models. In order to address this challenge, **the focus of this dissertation is on improving designers' analogizing capabilities by providing intuitive computational support for conceptual design analogy formation**.

Analogy

There has been a great deal of research about analogy across various fields. The first manuscript discusses analogy research as it pertains to design, but the key points as they pertain to the entire dissertation are discussed here.

Hofstadter [1] explains analogy as the process of understanding multiple “conceptual skeletons” at the right levels of abstraction and retrieving them according to their “ports of access.” These ports represent the handles by which a concept is later retrievable. In a design context, experts are more proficient at retrieving analogies than are novices [2]. One plausible explanation for this is experts’ more mature knowledge models – the “ports of access” for an expert designer are more likely to be governed by meaningful design knowledge, such as function. In contrast, a novice is more likely to retrieve a spurious analogy based on less meaningful types of similarity.

Gentner’s structure mapping theory of analogy explains this principle more precisely [3]. According to structure mapping, every conceptual domain can be represented as a network of concepts and the relationships between those concepts. If one domain’s relationships (content and structure) can be mapped onto another domain’s relationship content and structure, then there is a strong analogical relationship between those domains – hence the name “structure mapping.” Gentner uses a comparison between the Rutherford model of the atom and a planetary system to explain the theory [3]. In this example, an electron *revolves around* a nucleus while a planet *revolves around* the sun. A nucleus is *more massive than* an electron while the sun is *more massive than* a planet. The content of the relationships and their arrangement amongst their domains’ entities are both mappable from one domain to the other – thus there is a strong analogy between the Rutherford model and a planetary system.

Another important principle of structure mapping is that high-level causal relationships dominate analogy formation. If a relationship or set of relationships is known to cause another relationship, that *causal relationship* is of high importance in retrieving an analogy. In a design context, the concept of a causal relationship can be understood as a design abstraction – such as requirements, functions, or operating conditions – that play an important causal role in the synthesis of an artifact. If two systems share similar requirements or functions, then they are likely to have strong analogical similarity.

These two principles of mappable relationships and causal relationships are widely accepted as key concepts that govern analogy formation [4], and they form the main theoretical foundation for the work presented in this dissertation.

Analogy and Abstraction

Two important facilitators for design analogy are representation and abstraction. Representation refers to how concepts are modeled, while abstraction refers to the specificity with which those concepts are modeled. The two are closely related. For example, structure mapping deals with the representation of conceptual domains that lead to analogy formation. Multiple shared relationships between entities across domains indicate a valid analogical mapping. Revisiting Gentner’s example, the electron, nucleus, planet, and sun are entities; while *revolves around* is a relationship. This single highly abstract relationship encompasses a network of many lower order relationships involving sizes, distances, and physical laws. In other words, this single abstract relationship between entities in a domain serves the same role in finding analogies as a network of many low order relationships.

This abstraction provides a convenient heuristic for finding design analogies. Instead of mapping a complex network containing many relationships, a designer can rely upon a few simple abstract relationships to achieve the same goals. In the design domain, abstractions such as function and working principles are commonly used. Historically, research on product function abstraction and (to a lesser extent) component abstraction has furthered the goal of standardizing analogical mapping (e.g., [5, 6]). This dissertation studies and augments design analogies under this assumption: that large networks of mappable causal relations within a domain are interchangeable with design-relevant categories of abstract similarity. Under this assumption it is valuable to consider many types of abstract similarity in order to (1) increase the breadth of potential analogies retrieved and (2) increase the quality of analogies retrieved by mapping multiple similarity types. In doing so, it is possible to support a design-by-analogy methodology that considers a variety of design information simultaneously – including key requirements, functions, and physical constraints.

How to Read this Dissertation

This dissertation consists of five manuscripts. It is not necessary to read them in any particular order, although manuscript 0 is a literature survey and presents a good primer for the field. Table 1 summarizes the main topics of each manuscript and the key design capability that each one supports. The following sections summarize each manuscript by their main contributions and how they are related to each other.

Table 1. Chapter Topic Guide

Chapter	Topics	New Design Capability
Introduction (This Chapter)	• Context and problem • Contributions • Manuscript Relationships	
Manuscript 0 Impacts of Function-Related Research on Education and Industry	• Historical context • Function abstraction • Component abstraction • Failure abstraction	
Manuscript 1 Discovery of Mental Metadata Used for Analogy Formation in Function-Based Design	• Designers' analogy forming processes • Natural abstractions used to create design analogies • Single analogies versus compound analogies	Search design knowledge libraries in new ways according to flow behavior properties
Manuscript 2 The Biology Phenomenon Categorizer: A Human Computation Framework in Support of Biologically Inspired Design	• Extracting concept maps from natural language • Human computation • Games with a purpose	Use natural language data to support conceptual reasoning in computational design
Manuscript 3 Using Molecular Fingerprints to Infer Functional Similarity in Engineered Systems	• Molecular fingerprinting • Inferring abstraction relationships	Screen large design solution spaces within a digital design and manufacturing framework
Manuscript 4 Analogy Fingerprinting: Fast Analogy Search Inspired by Drug Design	• Molecular fingerprinting • Fast design concept matching	Evaluate large conceptual design solution spaces within a digital design and manufacturing framework
Conclusions	• Annotated conclusions from each manuscript • General conclusions	

Manuscript Descriptions by Contribution

This dissertation represents an effort to improve understanding of design analogy representation and abstraction in order to facilitate heuristic support for conceptual design analogy formation. The work consists of five manuscripts that encompass three overarching research contributions.

The very first manuscript establishes the background context for the dissertation's main thrust of design analogy research. It does not present any original research, hence the label "Manuscript 0" in Table 1.

The first contribution comes from manuscript one, which includes the various types of abstract similarity that designers were observed using to form design analogies. The study uses controlled experiments and protocol analysis to observe several types of similarity that are predictive of analogical relationships in design. The results show that designers often draw analogies based on not just what an artifact does (functions), but also on the things that an artifact interacts with (flows). Additionally, this study found no difference between the types of abstract information used to create single and compound analogies. Understanding these types and structures of commonly used information informs computational models for design decision support.

The second contribution comes from manuscript two, which discusses a scalable human computation method for capturing design-relevant analogy information, and is adapted from work aimed at capturing common sense knowledge. Current computational support for design by analogy suffers from problems of either information richness or scope. Tools with rich information lack breadth, while tools with high information breadth lack information richness. The second manuscript addresses the challenge of obtaining a large quantity of design information in a form that is easily computable in a context where (1) a large information library is needed and (2) natural language processing methods are insufficiently accurate. More specifically, the manuscript presents and evaluates a Game with a Purpose model used to capture relevant design analogy abstraction information within the context of biologically inspired design (a specific type of design by analogy). The approach captures this information in the form of restricted-vocabulary concept maps, where the restricted vocabulary captures design-relevant abstract relationships (e.g., functions and flows). Scalably capturing conceptual information in this computable form contributes to large scale screening of potential analogies.

The third contribution is the adaptation of a molecular search algorithm for use in design contexts where information is representable as a graph, and is presented in the third and fourth manuscripts. Specifically, the third publication demonstrates the applicability molecular fingerprinting to characterizing electromechanical products' functionality based on their component graphs. Results show a strong predictive connection between two important design abstractions in existing systems – functions and component classes – using a representation that enables efficient large-scale screening of solution candidates.

The fourth manuscript adapts this same algorithm for use with concept maps – such as those gathered using the human computation approach described in the second manuscript. This publication introduces the Analogy Fingerprinting algorithm and demonstrates its applicability for automatically retrieving good analogies. The Analogy Fingerprinting algorithm supports a process in which a designer creates a concept map of the most important facets of a design problem and modifies it to represent a conceptual solution. The designer can then use this concept map to computationally retrieve a breath of design analogies based the extent to which its abstract relationship structures map onto a library of solution candidates.

Relationships Between Manuscripts

This section outlines the main conceptual relationships between the four manuscripts, which are also summarized in Figure 1 under the categories of understanding, preparing, and matching design abstractions.

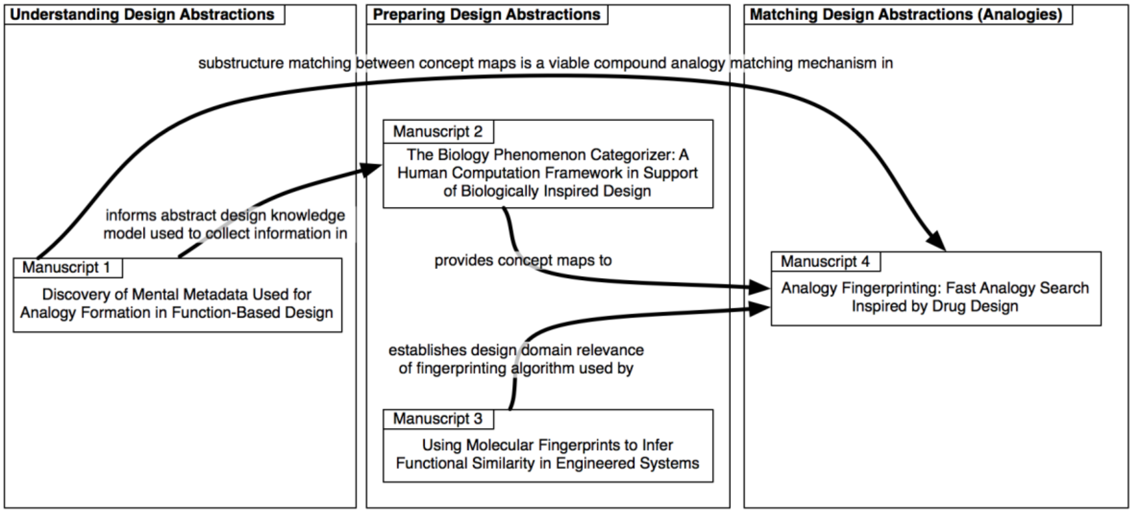

Understanding Design Abstractions
Preparing Design Abstractions
Matching Design Abstractions (Analogies)
substructure matching between concept maps is a viable compound analogy matching mechanism in
Manuscript 2
The Biology Phenomenon Categorizer: A Human Computation Framework in Support of Biologically Inspired Design
informs abstract design knowledge model used to collect information in
provides concept maps to
Manuscript 1
Discovery of Mental Metadata Used for Analogy Formation in Function-Based Design
Manuscript 4
Analogy Fingerprinting: Fast Analogy Search Inspired by Drug Design
establishes design domain relevance of fingerprinting algorithm used by
Manuscript 3
Using Molecular Fingerprints to Infer Functional Similarity in Engineered Systems

Figure 1. Conceptual Relationships between the Four Manuscripts

The key results of the first manuscript include the observation of many types of abstract similarity used in analogy formation, and lack of any observed difference in these similarity types between single and compound analogies.

Manuscript two uses a fixed relationship taxonomy to collect knowledge models, and the observed variety of similarity abstractions from manuscript one informs the creation of a diverse taxonomy to describe concept relationships. The types of concept maps produced in manuscript two are used directly by manuscript four to support computational analogy search.

Manuscript three observes a strong function-component correlation when using the path-based molecular fingerprinting algorithm from drug design. This manuscript establishes molecular fingerprinting as a useful technique for describing abstract design information in new design domains, and sets the stage for manuscript four to describe concept maps with the same algorithm.

Manuscript four presents a simple similarity measure to detect analogies between concept map fingerprints. This measure can retrieve full analogies to an entire problem as well as partial analogies to a subproblem. Manuscript one observed no significant difference between the similarity abstractions that designers use to draw analogies, regardless of whether they are single analogies or compound analogies (i.e., the composition of multiple analogies into a single design). This result from manuscript one supports using the techniques in manuscript four to perform simultaneous retrieval of both types of analogies from a single knowledge representation.

Impacts of Function Related Research on Education and Industry

Ryan M. Arlitt, Robert B. Stone, and Irem Y. Tumer

Impact of Design Research on Industrial Practice

Amaresh Chakrabarti and Udo Lindemann, Eds

Switzerland: Springer, 2016, pp. 77-99.

Abstract

Designers have long understood that a device must function well in order to satisfy its users, but only relatively recently has function been studied formally and extensively. The corresponding function-based paradigm focuses on abstracting what a system *does* separately from what it *is*. Within this paradigm, it is important to communicate abstract functions in a consistent manner, without binding them to their embodiments. This chapter discusses two recent outcomes in function-based design research, their impacts on education and industry, and the authors' observations regarding their adoption into practice. The first of these outcomes is an information schema for capturing design artifact knowledge, which includes a standardized function taxonomy. The information schema provides guidance for teaching functional thinking, and also supports basic computational design techniques during conceptual design. The second research outcome is a conceptual linking between functions and failure modes, enabling new types of failure analysis techniques in early design. Both research outcomes are likely still in the early stages of impacting practice, but evidence points toward the most immediate impacts occurring during education. While industry is typically more reserved regarding the details of their design practices, the chapter also presents several instances of practical interest in function-based design approaches.

Historical Context

The Internet boom of the 1990s improved the feasibility of engineering partnerships across large distances. As a result, design of complex engineering systems became an increasingly collaborative task among designers or design teams that were physically, geographically, and temporally distributed. The complexity of these products meant that a single designer or design team could no longer manage the complete product development effort. Additionally, developing products without sufficient expertise in a broad set of disciplines resulted in extended product development cycles, higher development costs, and quality problems. This shift toward increasingly knowledge-intensive and collaborative design increased the importance of computational design frameworks to support the representation and use of general knowledge among distributed designers [7].

Around this time, Product Data Management (PDM) systems hit their stride as an effective way to manage engineering data, such as computer-aided design (CAD) drawings. By organizing product component data, PDM systems improved communication, shortened production times, and reduced costs. However, designers were no longer merely exchanging geometric data (as supported by these PDM systems), but more general knowledge about design and design process, including specifications, design rules, constraints, rationale, etc. As such, merely providing access to schematics and CAD models was no longer sufficient. In order to support reuse of engineering knowledge, a representation was needed to convey additional information that answers not only “what?” questions about a design, but also “how?” and “why?” questions. Mappings from form to function had often been pointed to as an example of the kind of information that is needed for effective reuse of design knowledge, but were absent from traditional CAD models.

Early attempts at cataloging function were not entirely suitable for design repositories, being either extremely domain-specific or extremely general. For example, Collins et al. [8] developed a helicopter-specific list of 105 unique mechanical functions to accurately archive helicopter failure information. This approach is useful for cataloging and retrieving helicopter failure information, but is not generalizable to other types of systems. More generally, Pahl and Beitz [9] provide a highly abstracted vocabulary containing five functions and three flows (function operands), and Hundal [10] develops six abstract function classes, each containing more specific functions. The Theory of Inventive Problem Solving (TIPS or TRIZ), published by Altshuller in 1984 [11], describes all mechanical design with a set of 30 functions. TRIZ was developed through a survey of over 2 million patents, pointing to a high level of validity. Malmqvist et al. [12] noted that the TRIZ vocabulary would benefit from a structured function hierarchy using the Pahl and Beitz functions. A further review of function classification at the time can be found in Hubka and Eder [13].

To address the functional issues in PDM systems, the National Institute of Standards and Technology (NIST) held a workshop to identify basic research and industry needs for their Design Repository Project. This emerging research area of design repositories was aimed at making use of research in knowledge-based design to facilitate the representation, capture, sharing, and reuse (search and retrieval) of corporate design knowledge [14]. Importantly, while there was widespread use of functional decomposition at this time, there was no standard language for describing function [14]. Within such decompositions, whether for function or architecture, no standard existed concerning levels of abstraction. Specific needs identified at the workshop included: (1) a need for representation of function in CAD, in addition to geometry, (2) a need for a fixed representation scheme for modeling function, (3) a need for a commonly agreed set of functions performed by mechanical systems, and (4) a need for representations that are both human-interpretable and machine-interpretable [14]. To meet these needs, a collaborative research effort between NIST and academia was formulated to investigate the underlying framework for creating design repositories, including representation of design function, product architectures, and form; and notably lead to the development of a design repository data schema containing generalized function and component abstractions.

The remainder of this chapter will discuss two related research outcomes. The first is the aforementioned design repository information schema designed to address the needs identified by NIST, and the second is a relationship between functions and system failures. Each outcome is summarized, and followed by a discussion of their impacts in practice.

Research Outcome: A Design Repository Information Schema

The first research outcome, an information schema for describing artifacts in a design repository system, was formulated to enable designers to store and retrieve design knowledge at various levels of abstraction, from form (components, sub-assemblies and assemblies) to architecture description to function. The different levels of abstraction provide innovative ways to approach design. This information schema includes a function description language called The Functional Basis, a taxonomy of electromechanical components, and basic matrix representations that afford computational concept generation.

The Functional Basis

A systematic approach to functional modeling (e.g., [7, 9, 13]) generally has the designer decomposing a product's overall function into subfunctions until each subfunction is small and easily solved. Unfortunately, knowing when a function is *small and easily solved* can be quite ambiguous. As such, one of the key issues motivating the development of a consistent functional vocabulary was to provide guidance on when to stop decomposition. General function vocabularies (e.g., [9-11]), while applicable to a wide variety of domains, lack the detail to provide guidance on decomposition depth. In contrast, domain-specific function taxonomies (like Collins' helicopter-specific taxonomy [8]) are not useful outside of their fields.

The NIST Function Taxonomy and the Functional Basis were separate parallel efforts undertaken to address this disconnect between function abstraction layers. Both projects sought to create a general function taxonomy with high validity by unifying past research. To support this goal, the taxonomies were unified into a single reconciled Functional Basis (Table 2 and Table 3). The reconciled Functional Basis represents a general standard function taxonomy that describes the electromechanical design space at multiple levels of abstraction. This reconciled Functional Basis contains a set of functions (action verbs) and flows (nouns), to be used together as verb-noun pairs in a functional model. The function and flow sets both provide three levels of decomposition guidance. These levels are called primary, secondary, and tertiary; and they correspond to the level's degree of abstraction. A fourth column called *correspondents* offers synonyms to define and contextualize each function and flow. *Italicized correspondents* occur in multiple functions, indicating slightly different usages or senses of the word.

Table 2. Functional Basis Flows [15]

Class (Primary)	Secondary	Tertiary	Correspondents
Material	Human		Hand, foot, head
	Gas		Homogeneous
	Liquid		Incompressible, compressible, homogeneous,
	Solid	Object	Rigid-body, elastic-body, widget
		Particulate	
		Composite	
	Plasma		
	Mixture	Gas-gas	
		Liquid-liquid	
		Solid-solid	Aggregate
		Solid-Liquid	
		Liquid-Gas	
		Solid-Gas	
		Solid-Liquid-Gas	
		Colloidal	Aerosol
Signal	Status	Auditory	Tone, word
		Olfactory	
		Tactile	Temperature, pressure, roughness
		Taste	
		Visual	Position, displacement
	Control	Analog	Oscillatory
		Discrete	Binary
Energy	Human		
	Acoustic		
	Biological		
	Chemical		
	Electrical		
	Electromagnetic	Optical	
		Solar	
	Hydraulic		
	Magnetic		
	Mechanical	Rotational	
		Translational	
	Pneumatic		
	Radioactive/Nuclear		
	Thermal		
Overall increasing degree of specification →			

Table 3. Functional Basis Functions [15]

Class (Primary)	Secondary	Tertiary	Correspondents
Branch	Separate		Isolate, sever, disjoin
		Divide	Detach, isolate, release, sort, split, disconnect, subtract
		Extract	Refine, filter, purify, percolate, strain, *clear*
		Remove	Cut, drill, lathe, polish, sand
	Distribute		Diffuse, dispel, disperse, dissipate, diverge, scatter
Channel	Import		Form entrance, *allow*, input, *capture*
	Export		Dispose, eject, *emit*, empty, *remove*, destroy, eliminate
	Transfer		Carry, deliver
		Transport	Advance, lift, move
		Transmit	Conduct, convey
	Guide		Direct, shift, steer, straighten, switch
		Translate	Move, relocate
		Rotate	Spin, turn
		Allow DOF	*Constrain*, unfasten, unlock
Connect	Couple		Associate, connect
		Join	Assemble, fasten
		Link	Attach
	Mix		Add, blend, coalesce, combine, pack
Control Magnitude	Actuate		Enable, initiate, start, turn-on
	Regulate		Control, equalize, limit, maintain
		Increase	*Allow*, open
		Decrease	Close, delay, interrupt
	Change		Adjust, modulate, *clear*, demodulate, invert, normalize, rectify, reset, scale, vary, modify
		Increment	Amplify, enhance, magnify, multiply
		Decrement	Attenuate, dampen, reduce
		Shape	Compact, compress, crush, pierce, deform, form
		Condition	Prepare, adapt, treat
	Stop		End, halt, pause, interrupt, restrain
		Prevent	Disable, turn-off
		Inhibit	Shield, insulate, protect, resist
Convert	Convert		Condense, create, decode, differentiate, digitize, encode, evaporate, generate, integrate, liquefy, *process*, solidify, transform
Provision	Store		Accumulate
		Contain	*Capture*, enclose
		Collect	Absorb, consume, fill, reserve
	Supply		Provide, replenish, retrieve
Signal	Sense		Feel, determine
		Detect	Discern, perceive, recognize
		Measure	Identify, *locate*
	Indicate		Announce, show, denote, record, register
		Track	Mark, time
		Display	*Emit*, expose, select
	Process		Compare, calculate, check
Support	Stabilize		Steady
	Secure		*Constrain*, hold, place, fix
	Position		Align, *locate*, orient
Overall increasing degree of specification →			

In forward design, a designer can use the Functional Basis to iteratively decompose a functional model from a single black box function. To maximize form-independence and promote a wide search of the solution space, the first iteration is generally performed at the primary level. Subsequent iterations contain increasingly specific functions at the secondary and tertiary levels, until the designer shifts to component selection or domain-specific terminology. In general, decomposition to the secondary level is a good target due to its high information content [16]. In reverse engineering, the Functional Basis offers a way to consistently catalog products based on functions performed by those products, subassemblies, components, etc.

For example, a vise grip (Figure 2) can be described with the black box model in Figure 3. The black box model captures incoming and outgoing material, energy, and signal flows. Here, the vise grip's overall function is to *secure material*. *Mechanical* energy*,* *Hand* and *Object* materials, and a *Not Clamped* signal flow into the system. The same flows also exit the system after operation, except the system visually signals that the object is now *Clamped*. The functional model in Figure 4 provides a higher resolution functional view of the same system using Functional Basis terminology. As with natural language functional models, there are multiple correct ways to describe the system's function (e.g., the signal flow could be treated differently or omitted entirely), but the standard terminology enables meaningful comparison between multiple models.

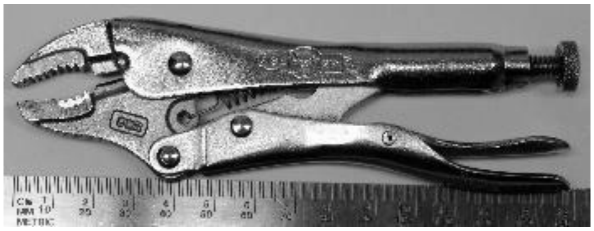


Figure 2. Vise Grip

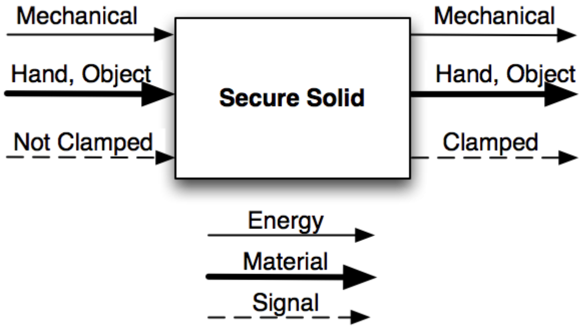


Figure 3. Vise Grip Black Box Model

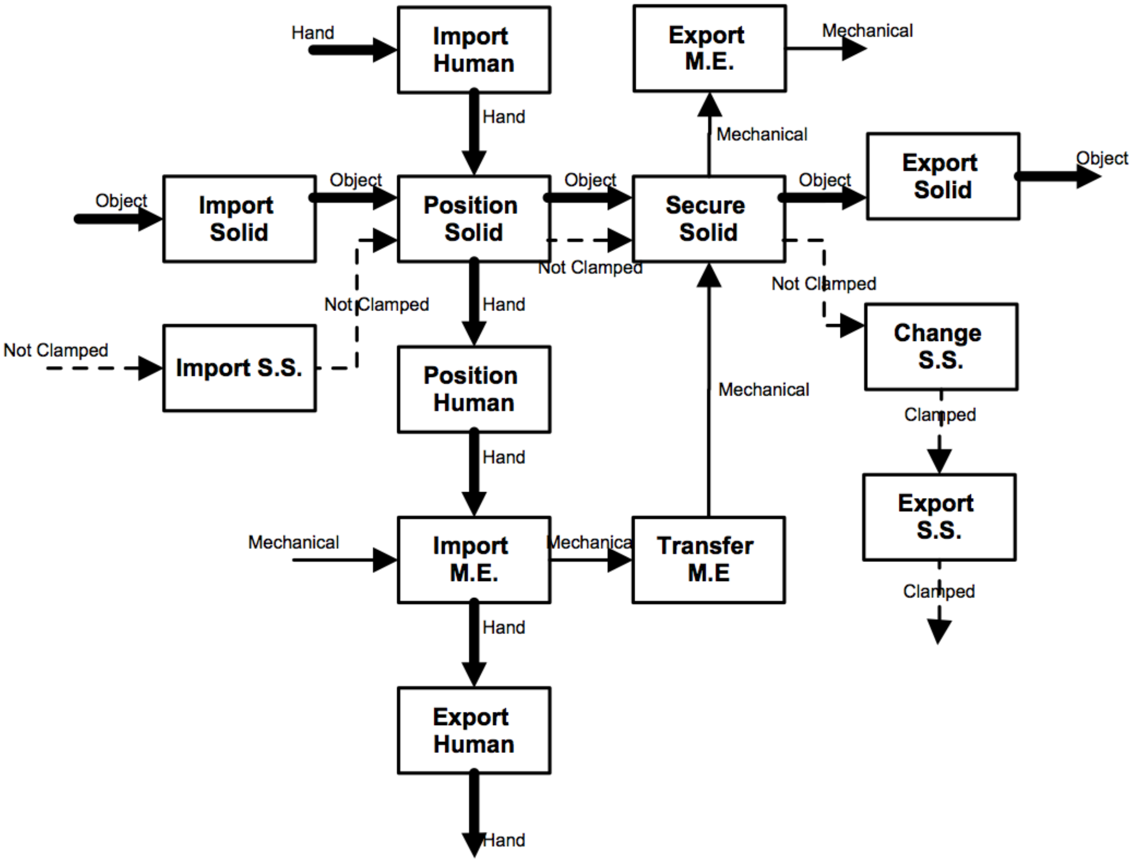


Figure 4. Vise Grip Functional Model

Several studies point to high validity of the Functional Basis. On grounds of theoretical validity, the Functional Basis is built upon extensive past work, subsuming the function taxonomies of Pahl and Beitz, Hundal, and Altshuller, as shown in Figure 5.

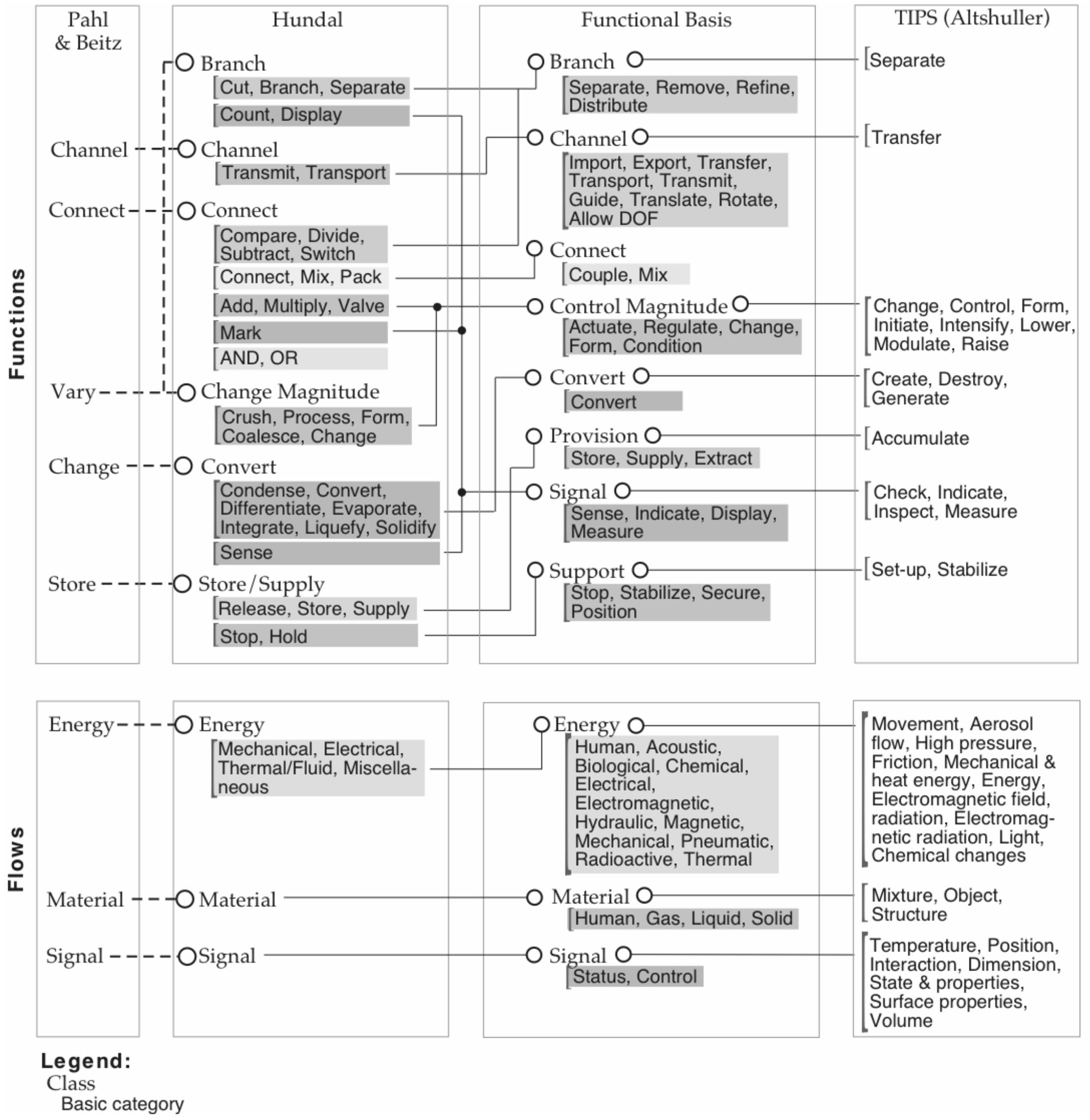

Pahl & Beitz
Hundal
Functional Basis
TIPS (Altshuller)
Functions
Branch
Cut, Branch, Separate
Count, Display
Channel
Channel
Transmit, Transport
Connect
Connect
Compare, Divide, Subtract, Switch
Connect, Mix, Pack
Add, Multiply, Valve
Mark
AND, OR
Vary
Change Magnitude
Crush, Process, Form, Coalesce, Change
Change
Convert
Condense, Convert, Differentiate, Evaporate, Integrate, Liquefy, Solidify
Sense
Store
Store/Supply
Release, Store, Supply
Stop, Hold
Branch
Separate, Remove, Refine, Distribute
Channel
Import, Export, Transfer, Transport, Transmit, Guide, Translate, Rotate, Allow DOF
Connect
Couple, Mix
Control Magnitude
Actuate, Regulate, Change, Form, Condition
Convert
Convert
Provision
Store, Supply, Extract
Signal
Sense, Indicate, Display, Measure
Support
Stop, Stabilize, Secure, Position
Separate
Transfer
Change, Control, Form, Initiate, Intensify, Lower, Modulate, Raise
Create, Destroy, Generate
Accumulate
Check, Indicate, Inspect, Measure
Set-up, Stabilize
Flows
Energy
Energy
Mechanical, Electrical, Thermal/Fluid, Miscellaneous
Energy
Human, Acoustic, Biological, Chemical, Electrical, Electromagnetic, Hydraulic, Magnetic, Mechanical, Pneumatic, Radioactive, Thermal
Movement, Aerosol flow, High pressure, Friction, Mechanical & heat energy, Energy, Electromagnetic field, radiation, Electromagnetic radiation, Light, Chemical changes
Material
Material
Material
Human, Gas, Liquid, Solid
Mixture, Object, Structure
Signal
Signal
Signal
Status, Control
Temperature, Position, Interaction, Dimension, State & properties, Surface properties, Volume
Legend:
Class
Basic category

Figure 5. The Functional Basis Compared to other Function Taxonomies [17]

More pragmatically, a study by Ahmed and Wallace [18] found that 90% of the functions described by a group of practicing aerospace engineering designers could be described by the Functional Basis, with two thirds of those function descriptions matching a Functional Basis term exactly. This study suggests that the Functional Basis has good validity in an industry engineering design context. Further, a study by Kurfman et al. [19] found that a directed approach to functional model creation using functional basis terminology produced more uniform functional models than an undirected approach. Finally, an information-theoretic study of the Functional Basis demonstrates that the information content of function terms increases from primary to secondary levels, while the jump from secondary to tertiary provides marginal benefits [16].

This function terminology is the first of several standard vocabularies and representations that are embodied in a design repository. Combined with these other standard vocabularies, the Functional Basis facilitates forward design activities including automated concept generation and early detection of potential failure modes.

The Component Taxonomy

Similarly to the Functional Basis, the electromechanical component taxonomy provides abstract categories for components in order to support a consistent knowledge vocabulary. General component terms are accompanied by synonyms and definitions, and are organized according to the functions that the components generally perform (Table 4).

Table 4. Component Taxonomy Excerpt

Primary Component Classification	Secondary Component Classification	Component Term	Component Subset	Synonyms	Definition
Branchers	Separators	...			
	Distributors	...			
Channelers	Importers/ Exporters	...			
	Transferors	Carousel			A device used to move material in a continuous circular path.
		Conveyor			A device used to move material in a linear path.
		Electric Conductor		*lead*	A device used to transmit electrical energy from one component to another.
			Electric Wire		An electric conductor in the form of a thin, flexible thread or rod.
			Electric Plate		An electric conductor in the form of a thin, flat sheet or strip.
		Electric Socket			A device in the form of a receptacle that transmits electrical energy via a detachable connection with an electric plug.
		Electric Plug			A device in the form of a plug that transmits electrical energy via a detachable connection with an electric socket.
		Belt		*strap, girdle, band, restraint*	A device shaped as an endless loop of flexible material between two rotating shafts or pulleys used to transmit mechanical energy.
		...			
	Guiders	Hinge		*pivot, axis, pin, hold down, jam, post, peg, dowel*	A device that allows rigidly connected materials to rotate relative to each other about an axis, such as the revolution of a lid, valve, gate or door, etc.
		Diode			A semiconductor device which allows current to flow in only one direction.
		...			
Connectors	Couplers				
	Mixers				
...	...	...			

As was the case with function, this taxonomy was formulated with the goals of standardizing electromechanical component terminology and enabling automated design tools [20], while being as complete and exclusive (i.e., low redundancy between terms) as possible. Because components are more concrete than functions, the component taxonomy is easier to use as a framework for domain-specific adaptation. Unlike function, technological progress results in new types of components. As a consequence, a general classification of components can always be updated, but the vast majority of components in the taxonomy form a stable core capable of describing most products.

Matrix Representations

Given these consistent abstractions for functions and components, several types of matrix representations are possible. These matrices reveal interesting similarities (functions) between apparently dissimilar physical solutions, and enable automated design tools. The matrix representations support simple mechanisms for propagating abstract functions forward into more physical domains.

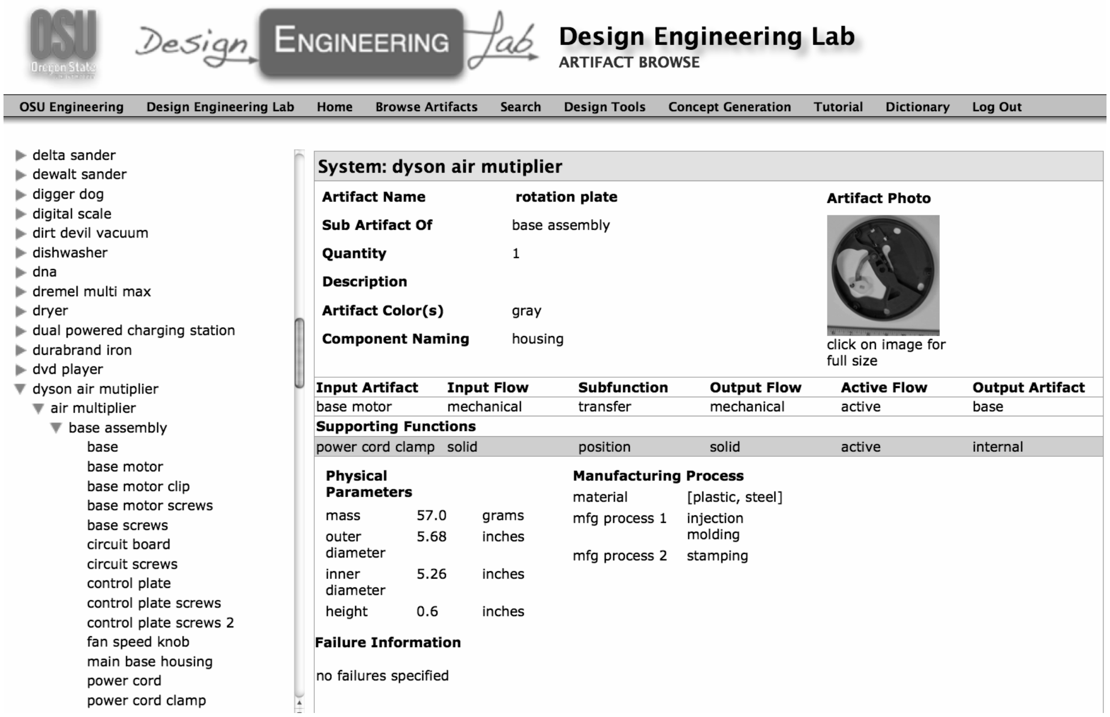
The first of these, called the Function Component Matrix (FCM), relates a product's subfunctions to the components that perform those functions. One axis lists functions, and the other axis lists components. Each matrix cell contains an integer representing the number of times a component has solved a given function, and FCMs can be created for an individual product or a set of products. Individual product FCMs can be combined via matrix addition. Consistent FCMs are made possible by the standardized terminologies of the Functional Basis and Component Taxonomy.

The Design Structure Matrix (DSM) catalogs the internal physical connectivity of a design. Several types of DSM exist, but a simple variety catalogs binary yes/no connections between components in a system. Both axes in this two dimensional matrix contain a row/column for each component, allowing pairwise comparisons between every pair of components in a system. A DSM can represent connections between specific individual artifacts inside a product or connections between components. Again, the standardized component terminology enables meaningful comparison and combination of separate DSMs.

Broadly, these representations enable tools that provide guidance from general abstract function description to domain-specific component selection. For instance, after aggregating a large number of FCMs representing historical product data, a designer can query the matrix for the desired functions to generate a large number of potential component solution candidates. These solution candidates take the form of morphological matrices wherein multiple potential solutions are given for each subfunction. This enables designers without expert knowledge to examine alternatives that they may not have otherwise considered.

Impacts of Design Repository Information Schema

The initial driver behind much of this work was to enable design repositories, and many of these results are appropriately embodied in a design repository (hereafter referred to as “The Design Repository”). The Design Repository represents an influential research outcome in that it broadly demonstrates the value of capturing and reusing product knowledge according its function. These vocabularies and techniques are used to capture knowledge about (at the time of writing) 184 reverse engineered electromechanical products. Products in the repository are decomposed to multiple levels of abstraction, including function data for components, subassemblies, and assemblies. Key artifact information, including function and component data, is stored using standard vocabulary. Figure 6 shows a typical artifact entry in The Design Repository. The rotation plate in the figure is a *housing* component in the Dyson Air Multiplier system, and it performs the function *transfer mechanical energy* from the *base motor* artifact to the *base* artifact.



Design Engineering Lab
ARTIFACT BROWSE

System: dyson air mutiplier

Artifact Name	**rotation plate**
Sub Artifact Of	base assembly
Quantity	1
Description	
Artifact Color(s)	gray
Component Naming	housing

Artifact Photo

Input Artifact	Input Flow	Subfunction	Output Flow	Active Flow	Output Artifact
base motor	mechanical	transfer	mechanical	active	base
Supporting Functions					
power cord clamp	solid	position	solid	active	internal

Physical Parameters

mass	57.0	grams
outer diameter	5.68	inches
inner diameter	5.26	inches
height	0.6	inches

Manufacturing Process

material	[plastic, steel]
mfg process 1	injection molding
mfg process 2	stamping

Failure Information

no failures specified

Figure 6. Rotation Plate Artifact in The Design Repository

Using The Design Repository, designers can store and retrieve design knowledge at these various abstraction levels, providing innovative ways to approach design. However, in addition to supporting a repository of design knowledge, the repository information schema has also had less tangible (but no less significant) impacts in both education and industry.

Education Impacts

To date, dozens of medium and large engineering schools in the U.S. introduce functional modeling in their undergraduate and graduate curriculum and use the Functional Basis as a language for expressing functionality. Owing to its small vocabulary, the Functional Basis guides students around common pitfalls associated with learning to create functional models. Some common pitfalls include references to specific components or forms, modeling the product as a flow through itself, or violating verb-object norms. Invalid functions (e.g., function descriptions that imply an embodiment) are more difficult to express when using Functional Basis terminology as opposed to natural language, which leads students to identify more product subfunctions [21] and increases repeatability in functional model creation [19]. For instance, the function-flow format of the Functional Basis encourages verb-object function descriptions (e.g, “rotate” becomes “transfer rotational energy”), and solution-centric function descriptions must be reconsidered to exclude references to form (e.g., “unlatch spring” becomes “actuate mechanical energy”).

In a separate but related effort, the Biomimicry 3.8 Institute has recognized function as valuable tool for organizing biological strategies in their AskNature database, which is used in classrooms around the world to teach and promote biologically inspired design (BID). The group has developed a biology-specific function taxonomy in order to help designers easily answer the question “How would nature do X?” Easily interpretable function categories in this taxonomy are the key to supporting the search process.

While the design repository research discussed prior did not directly influence these efforts, they illustrate an important parallel. The topic of biologically inspired design is widely studied in universities, but its application in practice remains limited. A series of BID workshops has brought together a community of researchers in order to address this issue by investigating ways to facilitate BID in a practical context. Function-based taxonomies represent a promising framework for mining and cataloging biological strategies, as seen in AskNature, to increase the ease of applying BID techniques. Progress in this area is still early, but several industry representatives have expressed interest in the outcomes of these workshops. More generally, such workshops may serve the dual roles of addressing research challenges and gaining critical industry support.

Industry Impacts

In industrial practice, Ford Motor Company participated in efforts to utilize the functional basis in its design efforts dating back to the late 1990s and early 2000s. A new program in Design for Six Sigma uses the functional basis as a method of developing critical and repeatable “transfer functions” to create robust designs. Informal reports indicate that functional modeling has been received with great enthusiasm, and the results show that the functional basis can model the large-scale systems developed by Ford.

Also in the automotive industry, General Motors engaged in research related to functional recall of prior components for reuse in their advanced design teams in the 2000s [22]. One area of interest included using function as a way to link customer need statements to appropriate vehicle related performance metrics that supported both Design for Six Sigma and requirements flowdown activities. The Functional Basis was presented to GM employees and utilized for these activities.

In a case of practical research application in an academic setting, a method for generating behavior models from functional models was applied to a Formula SAE car. This function-based behavioral modeling method [23] contains the steps (1) functional modeling, (2) state identification, (3) behavioral model element identification, (4) model solution, and (5) model iteration; and allows a designer to simulate system performance based on a functional model and the historical connectivity between functions and behavior equations. A full-vehicle dynamic simulation model of a Formula SAE car was created, providing a test and evaluation platform for the team to inform vehicle tire selection [24].

A project sponsored by the National Center for Defense Robotics extends functional modeling techniques to model product and process together [25]. The technique was used to model two vehicle decontamination processes: (1) the United States Army Nuclear, Biological, and Chemical (NBC) decontamination system and (2) the Kärcher TEP 90 decontamination procedure. The research assessed automation potential by calculating functional similarity between separate stations in each process, and showed that a single automated solution could likely accomplish the tasks of these multiple decontamination stations.

Guidelines and Platform behind transfer to Practice

The chief mode for moving this design research outcome into practice has been through training young engineers. The Design Repository, its related tools, and its data schema are used as a framework for teaching functional thinking in undergraduate engineering coursework. This approach has been used to teach the basics of functional modeling, and demonstrate its utility, using automated concept generation tools. These tools hide the historical data and matrix math from users while providing inspiration for multiple different concept variants.

For example, FunctionCAD [26] is a functional modeling environment that can enforce Functional Basis terminology and integrate directly with the Design Repository tools described in prior sections. A major goal driving the development of FunctionCAD was to ease students into functional thinking. Because of the extra effort associated with learning the function-based formalism, engineering students commonly opt to use natural language function terms instead of Functional Basis terms. The payoff for using a structured language is not immediately evident. FunctionCAD is a product of the design repository research that can experientially demonstrate this payoff without a lengthy learning process. For instance, a student using FunctionCAD might create a new functional model, export and load the file into the Design Repository concept generator, and retrieve a morphological matrix for that functional model. The tool's interface clearly indicates the available function terms, and can enforce other rules such as conservation of mass and energy. This demonstrates one added benefit of using the Functional Basis while imposing minimal obstacles on the designer.

A key takeaway observed from deploying tools like FunctionCAD is that software usability can have a severe impact on learning and acceptance of conceptual design techniques. A poor implementation can actually be worse than nothing at all. In order to maximize the effectiveness of research dissemination, especially when students are a target audience, the implementation must be stable and easy to use. When using prototype software as a teaching tool, students were observed becoming frustrated with bugs, missing features, and other usability issues. As a result, some students discounted the underlying approach as troublesome and ineffective. This effect has been observed with prototype versions FunctionCAD and Design Repository concept generator tools.

More generally, the effort required to learn and adopt new research findings is a barrier to their acceptance into practice. Tools like FunctionCAD are designed to minimize that effort while demonstrating the utility of the research findings. When usability issues decrease ease-of-use, such tools can become no more effective than teaching the methods directly.

It follows that usability and polish should be highly ranked requirements when such tools are anticipated to have a significant effect on training activities. Similarly, researchers should try to consider usability heuristics when producing research artifacts for outreach purposes.

A related contributor to the success of the Functional Basis as a teaching tool is its ease of adoption. Its function vocabulary balances natural language, physics-based, and teleological views of function. This balance affords descriptive power, simplicity, and flexibility. Similar attributes can be seen in other commonly accepted design tools, including TRIZ and Failure Modes and Effects Analysis (FMEA). These tools are simple and flexible enough for anyone to learn, and powerful enough to solve practical problems. It follows that design researchers should aim to condense research outcomes into simple and flexible packages. In short, our experiences using Design

Repository tools in the classroom indicate that usability and adaptability should be top priorities when formulating a design research outcome as a training tool.

Research Outcome: Function Failure Relationship

The second research outcome discussed in this chapter concerns the relationship between functions and failure. Failure, put simply, occurs when a system becomes unable to perform its intended function. The failure state manifests as unintended behavior. This conceptual linking between functions and failures has led to a number of tangible research products with the potential to influence practice. The research described in this section falls into one of two categories: component level function-failure approaches and system level function-failure approaches.

Component Level Failures

At an individual component level, failures are often the result of loading exceeding material limits. The material limits are ultimately a function of variation in the manufacturing process while the loading can be described by the component's performance equations. If this variation is specified up front, then that variation can be propagated back through the performance equations. This enables a designer to define the component form such that failure is avoided even in the presence of manufacturing variation. Taking this one step further, if components are linked to function then a designer can predict what components to use and what failure modes are possible well before any components are fabricated.

Motivated by the success of the prior taxonomy research, and the need to perform failure analysis as effectively as possible, a research effort in this area produced a general electromechanical failure mode taxonomy. The helicopter-specific failure taxonomy of Collins et al. [8], which formed the groundwork for a matrix-based failure lookup tool, also provides the basis for the electromechanical failure mode taxonomy. The end result is a taxonomy of updated mechanical failure modes [27] and new electrical failure modes [28] (Table 5). This abstract failure mode categorization enables earlier consideration of failure modes in the design process by enabling an FCM-style relationship between function and failure.

Table 5. Failure Mode Taxonomy Excerpt [27]

Primary Identifier	Failure Mode	Definition
Corrosion	…	…
Creep	…	…
Ductile Deformation (Ductile Material)	Brinelling	A static force induced permanent surface discontinuity of significant size occurring between two curved surfaces in contact as a result of local yielding of one or both mating members.
	Force induced elastic deformation	Occurs when the imposed operational loads or temperatures in a machine member result in elastic (recoverable) deformation such that the machine can no longer satisfactorily perform its intended function.
	Yielding	Occurs when the imposed operational loads or motions in a ductile machine member result in plastic (unrecoverable) deformation such that the machine can no longer satisfactorily perform its intended function.
Fatigue (Fluctuating loads or deformation)	High cycle fatigue	The sudden separation of a machine part into two or more pieces occurring when loads or deformations are of such magnitude that more than 10,000 cycles are required to produce failure.
	Impact fatigue	Failure of a machine member by the nucleation and propagation of a fatigue crack that occurs as a result of repetitive impact loading.
…	…	…

The Function Failure Design Method (FFDM) uses this standard failure mode taxonomy, along with historical failure data, to algorithmically predict failure modes from a design’s functions [29]. A binary function-component matrix relates functions to components, and a second matrix relates components to quantity of observed failures for each failure mode. Multiplying the two matrices gives the failure mode frequency for each function. The EF matrix can be generated for a single product, or for an entire database of functions, components, and failure modes. A designer can use this matrix of function-failure correlations to revise the functional model, inform component selection, and rank concept generator results.

The Function Failure Rate Design Method (FFRDM) extends the FFDM knowledge base by adding approximately 36,700 failures from Failure Mode/Mechanism Distributions 1997 (FMD-97) and Nonelectric Parts Reliability Data 1995 (NPRD-95). These additions improve the validity of the failure mode knowledge base, and using failure rate data from these documents instead of relative raw frequency improves the validity of FFDM's likelihood predictions.

In a separate parallel effort, the Risk in Early Design Method (RED) [30] extends FFDM to translate function and failure information into categorized risk elements. RED uses a set of risk-attitude heuristics to select from different types of likelihood and consequence equations. RED communicates risks according to their likelihood and severity in the form of a risk fever chart (Figure 7), commonly used to display risk elements in various companies, including NASA and Boeing. In this chart, all system risks are plotted according to their likelihood and consequence, providing the designer with a visual snapshot of the overall system risk.

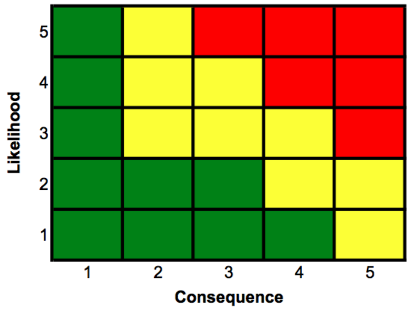


Figure 7. Risk Fever Chart

System Level Failures

A systems-view product of the function-failure relationship in early design is the Function Failure Identification and Propagation (FFIP) framework [31]. FFIP was introduced as a design-stage method for reasoning about failures based on the mapping between components, functions, and nominal and off-nominal behavior. The goal of the FFIP method is to identify failure propagation paths through the functional model by mapping component failure states to function 'health'. This approach uses simulation to determine fault propagation and fault effect, thus providing the designer with the possibility of analyzing component and interaction failures and reasoning about their effects on the rest of the system. The two main advantages of the FFIP method are: 1) a functional abstraction which allows it to be used in complex systems employing both software and physical components; and, 2) a simulation-based approach allowing analysis of multiple and cascading faults.

An FFIP analysis begins with a functional representation of a system and utilizes the mapping of functions to components in a component structural representation. A system simulation is built following the structural representation. The nominal and faulty behavior of generic components is stored as state machines in a component library. Each state represents a behavioral mode of the component where the qualitative intervals (high, low, etc.) of the input flow attributes are converted to output flow attributes. For example, in the nominal mode of a fuel line the input flow level of fuel is the same as the output. However, in the blockage fault mode, the output flow level is reduced to zero. Finally, the approach introduces a Function Failure Logic (FFL) reasoner which relates the input and output attributes of the component simulation to the expected change for the function mapped to those components. The result of an FFIP analysis is an evaluation of the health status of each function in the system. There are four potential health states for a function, as defined in Figure 8. These states are based on the concept that a function is the expression of the designer's intent describing the actions that affect the flows of energy, material and signal in the system.

1. Healthy: The function affects the flow as intended
2. Degraded: The function affects the flow differently than intended
3. Lost: The function does not affect the flow
4. No Flow: There is no flow for the function to act on (usually due to an upstream failure)

Figure 8. FFIP Function Health States

Impacts of Function Failure Research

The failure analysis tools commonly used in industry (e.g., Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA)) rely on expert knowledge to identify failure modes. For example, Team X at NASA's Jet Propulsion Laboratory (JPL) is an expert team used to create conceptual designs of space missions. The design activity itself takes place in a setting that promotes constant communication, and a risk expert on the team solicits potential risks from subsystem chairs. This reliance on experts to identify failures can serve as a design process bottleneck.

Eliminating this expert knowledge bottleneck was a major motivator driving function-failure research. The function-failure abstraction provides the means for a novice engineer to reuse expert knowledge for failure prediction. For instance, the failure modes, likelihood values, and severity values generated by RED can prepopulate an FMEA table. This approach provides a secondary baseline to compliment a traditionally generated FMEA (based on tribal knowledge of similar projects), and can be created without expert involvement. Additionally, connecting failures back to functions reduces FMEA’s reliance on physical component selections. This disentangling of form and function enables designers to begin FMEA earlier in the design process, reducing schedule pressures on failure identification.

In one attempt to apply the function-failure relationship in practice, the failure mode taxonomy was used to label failures described in JPL’s Problem/Failure Reporting (P/FR) database [32]. In general, the authors found that the database contained insufficient detail about the spacecraft systems and their failures to create a confident failure mode mapping. When additional information was available from individual reports and expert interactions, high-confidence failure mappings were created for 69 out of 86 (80%) of failure modes. A key takeaway from this work is that in order to make use of function-failure relationship design tools in practice, practitioners would need to capture additional information about failure events. In this case the tools don’t fit smoothly into existing practices, posing an obvious but important barrier to their adoption.

As indicated in the earlier section on the Functional Basis, the automotive industry (in these authors’ case that was General Motors) has shown interest in the usage of function-to-failure correlations that grew out of the FFDM work. The primary interest (in the mid 2000s) was for cataloging historical failure information to support FMEAs for new vehicle systems. The function-failure correlations made possible by the specification of functional and failure taxonomies were considered a framework by which in-house knowledge could be formulated and retained despite employee turn over.

In the realm of defense, the US Air Force investigated functional modeling as a platform for supporting counterterrorism operations [33]. The researchers demonstrated how to identify the most vulnerable functions in the model through injecting failures, tracing each failure’s propagation, and measuring function sensitivity. This failure propagation through a functional model closely parallels the FFIP methodology. As an example, a model of Improvised Explosive Device (IED) incidents was created using Functional Basis terminology. Faults were injected to demonstrate which functions in an example IED creation and use scenario are the most vulnerable to disruption. Due to the sensitive nature of this domain, the full extent of the research impact is unknown.

FFIP has been adopted in multiple projects in a variety of domains. At NASA projects, FFIP was morphed into Functional Fault Analysis to break down a system architecture [34] and analyze how faults propagate through aerospace systems. In this case FFIP demonstrates the value of function-failure linking in relatively practical terms, lending to the adoption and adaptation of its basic underlying principles. FFIP has also been applied to the design of nuclear power plants, led by a group at Aalto University in Finland, who have been consulting with the Radiation and Nuclear Safety Authority (STUK) of Finland [31] as to the applicability of the approach in future designs.

Finally, as a consequence of the complexity of modern vehicles, the Defense Advanced Research Projects Agency (DARPA) has invested in novel methods for design and verification of complex systems through their Adaptive Vehicle Make (AVM) program. FFIP was included as part of a model-based design effort led by

Palo Alto Research Center under DARPA funding to establish "correct-by-construction" design prior to prototyping [35]. Sustained interest in model-based design points toward the abstract function-failure relationship as having a fundamental impact on future design activities. A company that has formed through this project, CyDesign has commercialized portions of this approach.

Both FFIP and FFDM are part of a graduate course at Oregon State University that teaches various methods of failure and risk analysis. Students who have graduated from Oregon State University with this training have every intention to introduce these methods as the next generation failure and risk analysis tools into the reliability engineering practices with their current employers, which include NuScale, Xerox, Daimler, and Raytheon.

Conclusions

The Functional Basis, its utilization as a building block of the Design Repository, and the function-to-failure mappings have made impacts in education and in the practice of industry. In the education arena, we are likely still in the early stages of seeing the results as the concept of functional decomposition as a key activity in design process continues to take root in the US engineering education landscape. Early data (it is largely anecdotal at this point) leads the authors to conclude that the abstraction that is possible through the Functional Basis pays dividends in better designed products [36] and more critical thinking by students in the engineering design courses. While the outcome is generally a better result, the qualitative data indicates that grappling with abstraction is at times a mentally stressful activity – particularly during the first few encounters with the approach. With repetition the abstraction-making potential of using the Functional Basis during the conceptual design process becomes more natural and easier to implement for student engineers.

Considering the impact of the work on industry practice, the use of function has gained ground over the past decade. While industry is typically tight-lipped as to what makes up the “secret sauce” of their success, the authors speculate that based on our interactions there has been measurable acceptance of function-based methods within the design teams of US industry. As noted in our conclusions regarding educational practice, the abstraction-making potential of the Functional Basis and the function-failure approaches take some intentional practice to master. It therefore likely takes a supervisory champion to push these activities into the standard operating procedures at a given company. In general, we have seen at a minimum interest and preliminary use at automotive, aerospace and product innovation companies as well as national labs and Department of Defense agencies.

Summary

These research contributions have made their way into practice in different ways and at different rates, though the full extent of their impacts is difficult to measure. Education and training activities provide direct bottom-up influence, though tracing the impacts caused by newly trained engineers is challenging. The effects of such training may not manifest for years, and cultural inertia within established organizations can present barriers to acceptance of new design techniques.

In contrast, direct collaboration with industry provides top-down influence. This arena affords more immediate impact, but requires buy-in from key people in the organization. In this respect, small startups represent a compromise between receptiveness to new ideas and capacity to impact practice. In all likelihood, the continued combination of top-down and bottom-up techniques is necessary to produce noticeable change in practice.

In both of these arenas, our experiences indicate that the research outcomes must possess demonstrable utility by providing direct solutions to practical problems in an easy-to-use manner. Simplicity and flexibility of the core research contribution are critical to facilitate the transition into practice, such that interested stakeholders can adopt and adapt the research outcomes with low effort.

Discovery of Mental Metadata Used for Analogy Formation in Function-Based Design

Ryan M. Arlitt, Chiradeep Sen, Anthony Nix, and Robert B. Stone

Proceedings of the ASME 2015 International Design Engineering Technical

Conferences & Computers and Information in Engineering Conference

IDETC/CIE 2015

August 2-5, 2015, Boston, USA

IDETC2015-46963

Abstract

Applying previous solutions to solve new problems is a core aspect of design. In this context, analogies provide a mechanism to reapply previous solutions in new ways, but analogy formation is limited by a designer's knowledge. One approach toward improving a designer's analogy-forming capabilities is to provide an easy-to-use computational means of retrieving a wide breadth of relevant analogies. This work aims to answer what types of similarity are commonly used to draw design analogies, and whether some types of similarity are used more frequently in compound analogy versus single analogy. In this study, an experiment was performed to observe and document the types of information that designers found useful when forming analogies during conceptual design. A categorization of this information is sought in order to inform (1) the types of similarity data to store in an intuitive design-by-analogy database and (2) the form that a search query should take. The experiment consists of a design task and a follow up interview. Ten mechanical engineering graduate students specializing in design participated. These participants were interviewed, and their internal knowledge queries were encoded to reflect their objectives, thought process detail, direction of reasoning, and subject behavior type. Each conceptual design is cataloged according to whether it represents a compound analogy, a single analogy, or no analogy. The results show little difference between the types of information used in compound versus single analogy. Function, flow, and form information were all observed during analogy formation, indicating that all three types of information should play a role in a design-by-analogy database, regardless of generative goal. Notably, flow behavior was a commonly observed type of abstract

similarity across domains. This points to the value of capturing flow behavior abstraction in engineering analogy databases.

Introduction

A growing number of engineering design research efforts advocate the reuse of prior knowledge to support more informed design decisions. Related to this need is the hemorrhaging of corporate engineering knowledge as practicing design engineers retire or change careers. In both instances, there is a clear need to specify and archive the appropriate types of existing design knowledge – including contextual information that often eludes CAD drawings and design documentation. Such a specification would inform not only what types of information to capture, but what types of information to retrieve.

Paralleling this success with the domain of early engineering design reveals the crux of the challenge in early-stage design automation: the general lack of sufficient formalized knowledge about the elements of information (representation) and processes (reasoning algorithms) involved in the design process. Design, especially early-stage design, is a highly human-centric, creative activity. The process of synthesizing a solution from needs follows complex mechanisms, a complete understanding of which will require advances in the fields of human cognition and intelligence, computability theory, data structures and algorithms, and human and computer-aided formal reasoning. This research takes a step toward this level of understanding by cataloging the types of information elements transferred during design analogizing.

It has been shown that designers of all experience levels use analogy, but expert designers do so more effectively [2]. This overall understanding as it applies to engineering design has inspired many research projects in the areas of case-based reasoning and design-by-analogy, with applications in any domain or organization that relies on past experiences to inform design decisions. Experiments also show that in the early synthesis process, function-based thinking helps to broaden the solution search space. Cross-domain similarities between abstractions such as function provide a convenient shortcut for finding analogical connections. This provides a motivation for finding types of abstract similarity used to make analogies in original design. Database-driven design-by-analogy has the potential to facilitate analogy formation across domains by providing a designer with plausible analogy candidates generated in a large variety of ways.

Background

This section describes related work in function-based design, case-based reasoning, design by analogy, and repositories of design knowledge.

Function-Based Design

Functional analysis is a well-established design approach [37-40] wherein product function is separated from product form. System models created at the functional analysis stage consist of both functions and flows. Functions are modeled as nodes that operate on flows through the system, while flows are categorized as the materials, energies, and signals that flow between function nodes. Together these elements can be used to describe how a system interacts with its environment. Functional modeling allows designers to represent and discuss systems before a solution has been determined. In functional modeling, a standardized set of function terminology leads to repeatable and meaningful system descriptions [15]. The reconciled Functional Basis, a standard set of function and flow terms [15], has been shown to be an effective language for describing systems [41].

Similarly, the Component Taxonomy represents an effort to generalize components in the same way that the Functional Basis generalizes functions and flows. This taxonomy serves as a framework for the “archival, search and reuse” of component information [42]. Together these terminologies enable a generalized way of describing system function and internal structure, and make up the language used to describe products within a historical product information repository.

Case-Based Reasoning and Design by Analogy

Formally or informally, designers often reference and base their conceptual designs on previous solutions [43-46], commonly referred to as *case-based reasoning* or *design by analogy* (DBA). These types of formalizations are important because they support computer-assisted analogizing, and analogies are often difficult to retrieve from memory [47, 48]. Further, expert designers typically form more analogies than novice designers [2], pointing toward the value of DBA. This difference in approach may be due to expert designers’ chiefly schema-driven approach to analogy formation, as opposed to novices’ case-driven approach [2]. The schema-driven approach, due to its higher level of abstraction, necessarily allows for a wider range of analogies to be formed than does the case-driven approach. Difficulties that novices face when applying analogies include (1) failure to encode their experiences well, (2) trouble creating mental links between “components that play the same role", and (3) insufficient experience to acquire the relevant knowledge [48]. Computer-directed analogy formation can potentially provide support for all three types of difficulty by (1) providing smarter means of creating relationships, (2) capturing and recalling the important types of relationships between entities, and (3) augmenting a designer's experiences with additional information.

One class of analogy formation techniques relies on fostering creativity to produce analogies. For example, Synectics [49] aims to support creative analogy formation using direct analogies, personal analogies, symbolic analogies (i.e., metaphors), and fantasy analogies [50]. These types of creative analogies are not considered in this paper.

By the structure mapping definition of analogy, for a design process to be analogical, the knowledge transferred from a source case to the target problem must pertain to some *relation* between objects and not just an attribute of an object [3, 51]). Structure mapping is context-independent, which makes it an appropriate framework for studying analogy content. The definition of analogy used in this paper is any comparison between two domains that share similar structures of entities and relationships, with causal relations increasing the strength of analogy. By this definition, an analogy's strength falls somewhere on a continuum, and is based on the quantity and hierarchy of these relationships. A comparison that has surface similarity without deeper relationship alignment (e.g., a red apple and a red car) is not an analogous comparison. Additionally, if two things share large amounts of structural and literal similarity, then they are directly related rather than analogous (e.g., the hub-spoke-rim structure of bicycle wheels and cart wheels). While similarity between abstractions (e.g., functional similarity) is not directly indicative of a structure-mapping style of analogy, it does suggest a number of shared causal relationships. This motivates the goal to find different types of abstract similarity that are used to form analogies during conceptual design.

In contrast, many theories of analogy depend on knowledge content (e.g., [44, 52-54]). These theories focus on the content of knowledge that makes analogical transfer feasible by describing different types of analogies along the dimensions of *Why*, *What*, *How*, and *When* [55]. Most present computational theories of analogical design are content theories [55]. One common method explaining how within-domain analogies are formed is the *case-based method* (or the direct transfer model) [56] wherein knowledge is transferred without intermediate abstraction. A common method for cross-domain analogical transfer in computational design is the *schema-based model* [57] in which knowledge is transferred from a source case to a target problem by abstracting a solution schema. The IDeAL system [54, 58] is one of many implementations of a schema based model for conceptual design. The problem transformation model [52, 53, 59, 60] proposes that initial failures cause designers to reframe the problem [61], ultimately leading to a successful analogy to the new problem. This concept of problem transformation is related to the compound analogy model [62], which describes how designers combine multiple analogies to explore the problem and solution space.

Representations of Design Knowledge

In terms of existing representations, general content models of the early design process provide a rich range of categorizations. The Function-Behavior-Structure

(FBS) model [63] represents design knowledge in terms of structure, behavior predicted from structure, function, expected behavior, and design description. The design process in this model is a translation from function to design description. The problem mapping framework [64, 65] describes problem exploration in terms of requirements, issues, functions, behaviors, and artifacts.

More recently, biologically inspired design has been the focus of many analogical design studies due to the apparent effectiveness of design analogies made between very different domains. One example in this area includes the Design by Analogy to Nature Engine (DANE) [66], which uses the Structure-Behavior-Function [67] language to represent biological systems. The SR.BID problem schema and the Four-Box method for problem formulation and analogy formation [68] capture the operational environment, function, specifications, and performance criteria in a domain. IDEA-INSPIRE, another tool for finding biological analogies [69], uses the SAPPhIRE model [69] to capture information about engineered and biological artifacts. This model organizes information about a system's associated actions, state changes, physical phenomena, physical effects, inputs, organs (properties and conditions of a system), and parts.

Repositories of Design Knowledge

Over the past several decades, researchers have addressed knowledge-based design information systems and their associated product representations to support automation of some aspect of engineering design – typically the recall of past designs to mimic directly or inspire indirectly

Various efforts have sought to leverage the information in design repositories (e.g., [70-72]), which capture information in various data schemas to drive concept generation (e.g., [73-75]). These types of efforts support schema based analogy at various levels of abstraction, providing multiple ways to approach design. Functional schema matching is particularly useful in this context because it models a causal relationship at a level above physical components – it is abstract enough to promote meaningful matches, but not so abstract that it is unintuitive to the casual user.

The infrastructure supporting these efforts is an information ontology (e.g., [76, 77]) that describes what types of design information can be stored, the relationship of those elements, and the extensibility of including new and additional types of design information. This taxonomy allows for artifacts to be grouped into well-defined, but abstract, categories.

The Design Repository is an example of an analogical database that works at the schema level as opposed to the case level. Products within the repository are broken into components, and these components are tagged with general function [15] and component [42] information (general schemas) that enables the user to locate components with the desired functionality. The overall goal of the current work is to identify missing elements for a database such as the Design Repository whose capture may lead to better analogy formation support.

Research Approach

This section describes the approach for capturing analogy formation events. The process begins with an experiment designed to collect descriptions of newly designed concepts and previously observed systems. Next these descriptions are chunked and organized according to which concept they describe. After chunking, chains of premises, questions, and answers are extracted or inferred from the text in each chunk. Each question is then coded to characterize internal knowledge queries, and the types of codes involved in each concept formulation chain are used to characterize different types of analogy. These types of analogy are then compared with the direction of reasoning and type of similarity observed. Two raters performed the chunking, creating premise-question-answer chains, and question coding independently; reconciling the outcomes of each step before proceeding to the next.

Data Collection

In the experiment, ten graduate-level mechanical engineering students with a design focus were tasked with identifying the functionality required for a given design problem, and then generating solution concepts to solve it. A novel product design task, in this case the design of a towel-ironing and folding machine, was created in order to mitigate fixation on preconceived solutions. The problem domain is also familiar enough that reasoning about domain principles and related solutions is possible. The concept generation stage was followed by an interview. The subject was recorded throughout the entirety of the experiment, using both a video camera and a pen capable of recording writing and audio. The interview recordings were the only data of interest, but the entire experiment was recorded to create a consistent environment.

In task one, the participant was given a short written design brief, with instructions to identify some high-level actions of the device (Figure 9). Task one was essentially a training task; the purpose of which was to stimulate subjects to think about functions, without strongly biasing them toward any single solution. This task engages the participant in emulating an expert’s approach of considering general schema-based solutions, rather than the less successful case-driven analogies [2]. In task two participants were simply prompted to generate conceptual solutions for the towel-folding problem. This task was not timed; participants were allowed to continue until they were satisfied.

Design Problem:
Design an automatic towel-ironing machine for use in hotels. The purpose of this device is to press wrinkled towels and fold them. You are free to choose the degree of automation. At this stage of the project, there is no restriction on the types and quantities of resources consumed/emitted. However, the hotel has a desire to minimize waste and consumption of energy and materials. The design team is informed that typical hotels have the following resources already available: hot water lines, cold water lines, steam lines, and compressed air lines.

You are a member of the design team. Your tasks are as follows:

Task1:
Identify some main/high-level actions that the device will perform. Use utensils and media given for this purpose.

Figure 9. Experiment Design Prompt and Task 1

Following task two the experimenters interviewed the participant. The interview was conducted in two stages. In the first stage, participants explained their concepts and narrated their design choices. During this stage, interviewers noted organic comparisons between design features and other objects (potential analogies). In the second stage, interviewers asked questions about these design features and the objects to which they were compared in an effort to have the subject discuss the two in greater depth. This stage of the interview was similar in style to an articulated use interview for customer needs gathering [39] in that it was largely interviewee-driven. This style improves external validity of responses, but does not result in well-formed data. Specific questions were not prepared beforehand, though typical questions at this stage include “where have you seen that before?” and “what made you think of that?" Responses to these types of questions form the basis for inferring internal knowledge queries that took place in the course of analogy formation. This approach was selected over think-aloud because it elicits extra details about the designer's mental models of both the source and target domains.

Figure 10 contains sample data for a single analogy formation event produced by one subject. The bolded terms indicate potential questions or answers. This subject's sketch data and notes for tasks 1 and 2 contained three design concepts and covered approximately 1.25 pages of 8.5"×11" paper. The interview lasted 24 minutes and the resulting transcript spanned 8 pages of text.

Next thing was, similar, in that it has a **heat exchanger** again with the **steam**, except for this would be **two**, **weighted**, like a **flywheel-like weights**, like a **roller**, so that could either be **powered manually** or more likely with an **electric motor**. So the **wrinkly towel** would have to be **guided in**, and in theory the **flat towel** would **come out the back** side.

…

Yeah, so this would be a **roller**. In my head these were **12 inches in diameter** or something, **heavy rollers**, uh. And this would be more of an automated process right? Where the **wrinkly towel goes in**, and then these **tubes** again, so those would be **steam fed heat exchangers** to **heat it up**, and then that would **come out of the other end**.

…

If I was thinking actually of like a **steam roller**, but there's no heat there. I was just thinking of the **big rollers** for there **pressure**-wise. Like a **weighted roller**. Of course they don't have any heat there, I just used the **heat-exchanger** on it. I can't think of anything I've seen before that has that, even now I'm thinking about it and I can't think of anything.

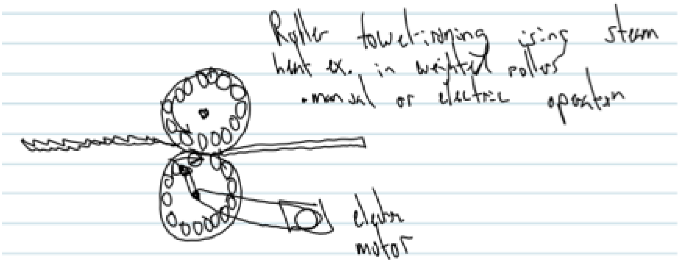

Roller towel-ironing using steam
heat ex. in weighted rollers
• manual or electric operation
electr motor

Figure 10. Subject Speech and Sketch Describing a Single Solution Analogy Event

Interview Transcription

After data collection, the audio-visual recording of the post-experiment interviews was transcribed into text. Table 6 shows portions of the transcript from the interview of a participant. The letter "I" denotes speech by an interviewer and "S" signifies speech by the subject. In this exchange, the experimenters request elaboration on concepts previously described by the subject. This example will be carried through the remainder of this section to aid explanation of the transcript analysis.

Table 6. Excerpt from an Experiment Participant's Interview Transcript

I	You talked about the folding assistant jig, you mentioned a hard boiled egg slicer, you mentioned a robot is boring, beyond that point did you think about the robot?
S	I figured it would probably be, if you were going to do it, it would probably be similar to how the jig would work except it would be automated.
I	And then you made very interesting comments. Origami, and easier to fold it if there's a second set of hands, and you can throw the towel on a bar. That it'll probably just fold itself on the bar. So the question would be, what made you think of origami? Why would that come up here?
S	Uh, well, I started by thinking how do I fold towels, and when I couldn't think of any more ideas I started thinking what other things do I fold? You fold paper for origami. I'm terrible at origami. But how can I make origami easier for myself? And that's having the instructions right there. It would be like, where to fold it.

Chunking

Chunking begins with the narrative part of the interview, when the subject explains each idea without interruption. If the subject sketched their ideas, the coder identifies and extracts text all pertaining to each concept sketch. If a concept is not embodied, but culminates in a final idea (e.g., "pulling on the edges to make sure it stays flat"), then instead text about that idea is identified. This is repeated for the question-and-answer portion of the interview, grouping together text that describes the same sketch/idea. All text describing a concept is later used to determine the type of analogy formation (if any) that occurred during that concept's formulation. Within each of these chunks, the coder identifies all sub-solutions or sub-features, which then aids in the next step: organizing the transcribed data in premises, questions, and answers.

Identification of Premises, Questions, and Answers in Interview Transcripts

For this task a “recognize and apply” cognitive model is assumed based on past studies of case-based reasoning [48]. In this context, recognition occurs when the designer identifies a relevant premise or answer to a question, while application occurs when the designer applies an answer to form a solution or a new question. For example, a subject may begin with the premise: “folding devices already exist,” which leads to the question “what else (existing device) performs the folding operation?” This question may then lead one to recognize that a letter folder performs this operation, forming both an answer and a new premise. From this premise, a new question may arise: “can I take what I know about letter folders and apply that to a towel folder?”

In this step, each transcript is examined in search of patterns where the thought process could be described as an interaction between three elements within the participant’s answers to the interview: (1) premises, (2) questions, and (3) answers, or in short: PQA chains. A premise is defined here as an assertion based on current understanding. A premise leads to a question when a knowledge gap is identified. A question leads to an answer when relevant information is pulled from long-term memory into working memory. In prescribing this model, the authors do not argue that analogy formation follows an ordered process of premise-question-answer; only that such a representation makes it easier to capture and categorize the internal knowledge queries.

Two coders created one PQA chain for each identifiable solution or feature, working backwards based on the participant's statements. Each PQA chain ends with a known solution because solutions are the easily observable goals of concept generation. After creating all of the PQA chains for a participant, the two coders reconciled these chains into a single agreed-upon set.

Table 7. Excerpt from a PQA Table

Premise	Question	Answer
I fold towels	**how do I fold towels?** […] and when I couldn't think of anymore ideas [...]	**[null]** when **I couldn't think of anymore ideas** [...] I started thinking what other things do I fold?
Other things are folded, besides towels	I started thinking **what other things do I fold?**	**You fold paper for origami.** I'm terrible at origami.
You fold paper for origami. I'm terrible at origami.	But **how can I make origami easier** for myself?	And that's **having the instructions** right there. It would be like, where to fold it.

Table 7 shows portions of a participant's PQA Table (Premise-Question-Answer Table). It describes a train of thought formed by the participant as follows: the participant started with one premise that other things than towels are also folded, and examining those things could help in finding analogies. This premise led to a specific question that takes the form of a query within his long-term memory for things that are folded. Once this question was "asked" mentally, the participant's cognitive processes returned the answer: origami. The purpose of the PQA table is to capture the plausible evolution of premise-question-answer tuples through the analogy-building process. These chains provide the basis for identifying the types of internal memory queries made during analogy formation.

In this table, all cells except those with italicized text contain exact strings from the transcript. The boldfaced portion of those strings indicate the phrases that qualify as premise, question, or answer in each case. The italicized text in the first two premises indicate phrases that were not uttered directly by the participant, but would make a rational candidate for the premise, based on the question and answer that followed. For example, the question "How do I fold towels" would be void if it was not premised first that "I fold towels."

Following categorization, questions are classified by their objective, direction of reasoning, response process, and behavior type.

Coding of Questions

Internal questions asked by the designer are encoded based on a coding scheme designed to categorize the types of information requests that designers make in an engineering organization [78]. This scheme was itself derived after examination of several other studies into designer information requests (e.g., [57, 79-81]). These past works studied the types of information used by designers during a design process. Because the scope of this experiment is focused only on internal requests made without going to an external source, the coding scheme is modified. As a result, the scheme used in this study only considers the objective, thought process detail, direction of reasoning (e.g., function to function, function to form, etc.), and level of abstraction. These categories are summarized in Table 8.

Table 8. Question Codes

Category	Code	Type
Objective	D1	Information
	D2	Confirmation
	D3	Comparison
	D4	Constructive Generation
	D5	Explanatory Generation
	D6	Analysis
	D7	Evaluation
Direction of Reasoning	H1	Flow to Flow
	H2	Flow to Function
	H3	Flow to Form
	H4	Function to Flow
	H5	Function to Function
	H6	Function to Form
	H7	Form to Flow
	H8	Form to Function
	H9	Form to Form
	H10	None
Response Process	F1	Retrieval-Recognition
	F2	Reasoning
	F3	Deliberation
Behavior	I1	Intended
	I2	Predicted
	I3	Observed
	I4	Procedural
	I5	No Value

The Objective category shows the various categories for objectives of the information request. This table has been preserved from the previous scheme, but results of the current experiments show evidence of only four goals at this stage of design: Information, Constructive Generation, Explanatory Generation, and Evaluation. Information is a simple retrieval operation that does not specify an objective. It answers the question, “what?” The goal of Constructive Generation is to create a new concept, and it answers questions of the form, “how could X do Y?” Explanatory Generation requests seek to create an explanation of an existing solution. It answers questions of the form, “how does X do Y?” The goal of Evaluation is to determine whether a solution is good enough. Evaluation answers the question “is X satisfactory?”

The Direction of Reasoning category captures types of information under consideration before and after a question is asked. A question’s corresponding premise and answer in a PQA table inform this categorization. Because this study occurs in the concept generation domain, function-based design classifications allow more specific categorization than the original coding schema. An exhaustive list allows for any transition between flows, functions, and forms. Function in this context includes any activity that satisfies a need, but also encompasses behavior that may not satisfy a need. The two are combined in order to mitigate issues of coder inference about the intent behind a behavior. Flow is defined as anything that the system is interacting with, but is not part of the design product. This includes both physical flows (e.g., towel) and energy flows (e.g., heat). Form in this context is defined as a solution embodiment. Typically this is an entire concept or a feature of a concept. Form also includes solutions that are not embodied in designed artifacts, such as paying people to fold the towels. In a function-based design context, the expected transition is from function to form.

The Response Process category allows capture of the type of thought process. The two main processes accounted for are Retrieval and Reasoning. Reasoning accounts for questions that require some inferences to be made, while Retrieval accounts for information requests that return a single piece of information. Deliberation accounts for questions for which the answer is a network of several premises, answers, and arguments.

The Behavior Type category has not been altered from the source material [78], and includes Intended, Predicted, Observed, and Procedural behavior. With memory as the only resource available to the designer, there is little difference between a predicted behavior and an observed behavior. Both describe a query that returns something that behaves in the manner expected by the designer. At the cognitive level, it is difficult to reliably differentiate between behaviors that have been directly observed and behaviors that the designer has inferred from incomplete knowledge. This is further complicated when one considers that these inferences may have occurred before storage in long-term memory. As a result, cases that deal with an expected behavior of an observed form have been classified as observed behavior. One exception to this rule is the case of mental simulation to evaluate a newly designed conceptual form. In this case, behavior is predicted from a candidate structure in order to identify issues with the design.

Following through with the example from Table 7, there exist three questions. The first question, “how do I fold towels?” represents a request to recognize and retrieve (F1) information (D1) that is already stored in memory. It starts with a function (folding), and requests information on how that task takes place (H6). This request seeks a form matching a behavior that has already been observed by the designer (I3). The second question, “what other things do I fold?” is again an information retrieval request (D1F1). It seeks a flow that is similar to a towel (H1). The request deals with the behavior of flows that have been previously observed by the designer (I3). The third question, “how can I make origami easier?” is an attempt to generate a solution (D4). To do so requires reasoning beyond simply querying the solution from memory

(F2). This request takes an intended behavior (I1), making origami easier, and tries to formulate a solution (H6).

The two coders coded each question independently one subject at a time, and then reconciled their codes to arrive at an agreed upon set. Cohen's Kappa values for inter rater reliability [82] are calculated for Objective, Direction of Reasoning, Response Process, and Behavior as 0.60, 0.51, 0.55, and 0.62 respectively. According to a commonly used guideline [83] 0.41-0.60 corresponds to moderate agreement while 0.61-0.80 corresponds to substantial agreement. If these values are calculated for only the second half of the ten subjects analyzed, agreement over Objective increases to 0.83 ("almost perfect" agreement [83]) while the others remain relatively unchanged (0.57, 0.52, and 0.64 respectively).

Results and Discussion

This section presents a summary of the different questions observed, followed by a categorization of the different types of concept generation processes, an analysis of the direction of reasoning, a discussion of the first concept generated by each participant, and the types of similarity observed during analogy formation.

Summary of Questions

Ten participants generated a total of 75 concepts, leading to 237 PQA tuples and 58 different types of code assignments. The most commonly assigned codes appear in Table 9.

The most common objectives include constructive generation and explanatory generation, while information and evaluation were used less frequently. Evaluation is used primarily in the context of predicting the behavior of a new concept in order to identify problems (D7F3I2). The coders did not select analysis (D6) to capture any objectives, instead capturing mental analysis as explanatory generation, constructive generation, or evaluation depending on the goals of the analysis. Confirmation and comparison were not observed.

Common directions of reasoning include flow-to-form, function-to-form, and form-to-form, though all other directions were observed at least once. These are discussed in depth in the Direction of Reasoning section.

Table 9. Commonly Observed Question Codes

Code	Interpretation	Number Observed
D5H6F1I3	Recall observed forms that perform a function	45
D4H9F2I1	Synthesize new form from old form	31
D5H9F1I3	Recall observed forms similar to observed forms	17
D4H6F2I1	Synthesize form directly from function	12
D5H3F1I3	Recall observed forms with similar flows	10
D4H3F2I1	Synthesize form from similar flow	7
D4H9F1I3	Synthesize form with directly reused parts	6
D4H10F2I1	Create a solution with intended behavior- most commonly form from issue	5
D4H39F2I1	Synthesize form from both flow and form	4
D5H7F1I3	Recall observed flows operated on by a form	4
D7H8F3I2	Evaluate a new concept’s ability to perform a function	4

Response process was at times challenging to categorize based on the available information. While the difference between retrieval-recognition (F1) and reasoning (F2) is clear, the difference between reasoning (F2) and deliberation (F3) is more ambiguous. In this study, the deliberation response process was used to categorize questions which either (1) had multifaceted answers, (2) were unanswerable with available information, or (3) required mental simulation. Both of these situations contain additional underlying questions that were not uncovered during the interview. This distinction may arise due to the granularity of the observations produced by the experimental design rather than two significantly different types of thought processes. Aurisicchio et al. [78] report that the area of greatest disagreement in applying their coding scheme was related to this distinction between reasoning and deliberation, and the findings from this study have reproduced this ambiguity.

The observed types of behavior were mainly Intended (I1) and Observed (I3). Intended behavior mainly captures the functionality of a newly generated concept, while observed captures the behavior and function of a previously observed system. Predicted (I2) captures predicted behavior (irrespective of functional intent) from newly generated concepts. Predicted behavior was rarely captured except in cases of mentally simulating a newly design concept.

Categorizing Types of Analogy

Starting with the codes in each concept it was determined whether an analogy was involved. If so, the level of detail was categorized. The objective and behavior categories were used for this task. A combination of explanatory generation and observed behavior (D5I3) or information and observed behavior (D1I3) indicates a reference to a previously observed solution. A combination of constructive generation and intended behavior (D4I1) indicates that a new concept has been generated. The categories and categorization rules are summarized in Table 10.

Concepts with at least two previously observed solutions and two newly generated solutions are said to be compound analogies. By contrast, a concept with at least two new solutions but no reference to observed solutions is evidence of compound design: multiple new solutions but no references to other domains. Concepts in the middle - those involving multiple observed systems and exactly one new solution - are tagged as weak compound analogy. The rationale for the "weak" compound analogy label is that this category contains a mixture of compound analogy and schema-based analogy, but is predominantly compound analogies. Some compound analogies did not have an intermediate idea stated aloud, and some schema-based analogies had multiple similar thoughts about the same source idea.

Table 10. Rules for Categorizing Concept Generation Process

Category	Rule	Interpretation
Compound Analogy	(D5I3 + D1I3) >= 2 & D4I1 >= 2	At least two observed systems and at least two newly generated ideas.
Weak Compound Analogy	(D5I3 + D1I3) >= 2 & D4I1 == 1	At least two observed systems and exactly one newly generated idea.
Compound Design	(D5I3 + D1I3) == 0 & D4I1>=2	At least two newly generated ideas and no reference to observed systems.
Direct Reuse	(D5I3 + D1I3) >= 1 & D4I1 == 0	At least one observed system and no reference to a newly generated idea.
Schema Analogy	(D5I3 + D1I3) >= 1 & D4I1 >= 1	At least one observed system and at least one newly generated idea.

A compound analogy involves the interplay between problem and solution domains [62]. An analogy solves a sub-problem, and then uncovers a new problem. A more complete definition for compound analogy might include an evaluation step as well (D7), but the experiment was not designed to consistently elicit this level of depth for each concept. Five compound processes included evaluation steps.

The remaining two categories are direct reuse and schema-based analogy. Direct reuse takes place when a previously observed system is the final concept (e.g., "pay workers to fold the towels"). Schema-based analogy takes place when a new original design with inferred behavior is generated based on an observed system. If a concept meets the criteria for both schema-based analogy and compound analogy, it is categorized as compound analogy.

These concept categorizations enable analyses comparing the generative goals of a designer to the types of reasoning and similarity used.

Direction of Reasoning

Next, direction of reasoning for each category is examined. There is no clearly preferred direction of reasoning for any specific analogy type (Table 11, left). However, a comparison between concepts where analogies were used versus those where they were not used (Table 11, right) indicates that at least one direction of reasoning is correlated with the use of analogy ($p = 0.0017 < 0.05$ using Fisher's Exact Test). In order to identify the specific significant categories, ten post-hoc tests are conducted on ten 2x2 contingency tables comparing each direction of reasoning against the sums of the remaining nine categories. This reveals the form-to-form and "none" categories as significant for $p < 0.005$ after a Bonferroni correction.

This result correlates with the expectation that analogies are chiefly drawn between existing forms. Likewise it is not surprising that uncategorized processes are usually not involved with analogy. The "none" code was commonly used when a comparison to a previously seen idea could not be identified – which by definition is required for analogy. For example, one subject started with a concept, posed the question "how can I automate this?" and then added motors. This instance was coded as D4-H10-F2-I1 – a newly constructed concept with no direction of reasoning from premise to answer, designed using a reasoning thought process to produce an intended behavior.

Table 11. Direction of Reasoning for each Process Type

		Strong Compound Analogy	Weak Compound Analogy	Compound Design	Schema Analogy (abstraction)	Direct Reuse (no abstraction)	No Classification	With Analogy	Without Analogy
Direction of Reasoning	**Flow to Flow**	1	3	0	1	3	0	5	3
	Flow to Function	0	0	0	1	1	0	1	1
	Flow to Form	8	6	3	4	3	1	18	7
	Function to Flow	0	1	0	0	1	0	1	1
	Function to Function	0	0	0	2	2	0	2	2
	Function to Form	17	8	3	11	22	3	36	28
	Form to Flow	3	2	1	3	3	0	8	4
	Form to Function	3	0	3	1	0	1	4	4
	Form to Form	30	7	4	19	8	2	56	14
	None	0	1	4	4	2	8	5	14

First Concept

While it is challenging to determine the interrelationships between concepts, the first concepts of each participant are relatively easy to compare. The design prompt of pressing and folding a towel is closely related to familiar systems and processes like using a hand iron to press clothing or folding a towel by hand. These are systems with high literal similarity to the problem, and their reapplication is evidence of direct transfer (as opposed to schema-based analogy). Of the nine participants that generated concepts for removing wrinkles, four participants' first concept is either two flat plates pressing together or reusing an existing pressing machine. Three first concepts used a steam room, and the remaining two used rollers.

Seven of these nine participants used a flat pressing surface within the first two concepts. This is evidence of early preliminary direct transfer: analogical abstraction is not necessary because literal similarity between the problem and existing solutions is high. Hand irons apply heat, pressure, and steam to a piece of fabric. The fabric is pressed between a flat metal surface and a flat ironing board. While ironing towels is likely not something that most people have done, it is easy to infer that what works for a thin piece of cloth could be directly applied or modified slightly to work for a thick piece of cloth as well. As one participant said of a clothing steamer, "the material I saw was very delicate kind of material, but with a tougher kind of material I think you will need a heavier jet."

Only two of the five process types were observed in the pool of first concepts. Four of these concepts were created by direct reuse and six were created by compound analogy. This could indicate different goals: while direct reuse quickly increases the breadth of concepts generated, compound analogy increases the detail of a concept. The group that started with direct use generated an average of 9.75 concepts, while the group that started with compositional analogy generated an average of 6 concepts. Only one participant did not use direct reuse for any of their concepts. While there exists an observed difference in concept generation goals, there is no correlation between the types of similarity used to generate analogical connections and the depth of the search.

Types of Similarity

Analogies played a role in 35 of the 75 concepts. The types of similarity that were referenced in connecting the source and target domains are cataloged as shown in Table 12. Each analogy source is documented, and then reference text from that analogy's chunk is searched for statements connecting the source domain to the target problem. The types of similarity referenced in the quoted text lead to the development of eight similarity categories (Table 13).

These types of similarity are separated into compound and single analogy processes in order to determine whether the depth of analogy process affects the useful types of similarity. The first four categories relate to the material, behavior (irrespective of intent), shape, and function of a form. Working principles in both domains were occasionally used, while the behavior of flows in the source and target systems was more commonly used. Descriptor similarity ("iron") and process similarity ("if I am ironing my clothes, what would I do?") were each observed only once. The total number of observations for each type of similarity is shown in Table 13.

The compound category includes both strong and weak compound analogies, while the single analogy category includes only schema-based analogies. The results show no discernable difference between the types of similarity used to form compound and single analogies (p = 0.74 using Fisher's Exact Test), though each type of information is a candidate for capture and reuse in database-driven DBA.

Table 12. Examples of Observed Analogies and their Similarity References

Example	Types of Similarity	Reasoning (quoted from interviews)
Bike Tire Knobs	Similar material, shape, and motion of a form	*I thought of the bike tire because I figured the conveyor belt would be some sort of rubber, and the only rubber thing I know that's circular and turns that has knobs is a bike tire. So, so yeah. That's why I have the knobs.*
Barbeque Lid	Similar material, motion, and shape of a form	*And when I looked, this wide shape for some reason, when I was thinking of metal, and this was a metal barbeque with a metal top, and it's just the first thing that popped into my head.*
Origami	Similar function of form and behavior of flow	*I started thinking what other things do I fold? You fold paper for origami.*
Cold Rolling Steel	Similar function of form, different behavior of flow	*Like, cold rolling steel. So when you… when it gets flatter, thinner…*
Vacuum Packing	Similar shape and function of a form	*I saw that, like I was watching TV or something, and I saw somebody doing [vacuum packing]. […] when I started drawing out the thing and I thought about the tracks and I thought oh, we could just do it like that.*
George Foreman Grill	Similar working principles of a form	*So, because it gets hot, and I guess I think of like making, grilled cheese sandwiches on the George Foreman grill. [...] And you put it in, so obviously there's a measureable amount of heat and pressure. I guess those were the things I was thinking that would flatten a towel.*
Falling Water	Similar flow behavior	*[…] keyword is 'falling' [gestures air quotes]; towel is falling, water is falling.*
Lint Roller	Similar process and similar flow behavior	*Basically, I was thinking 'If I am ironing my clothes, what I would do?'*

Table 13. Types of Similarity Observed in Compound and Single Analogies

Type of Similarity	Type of Analogy		
	Compound	Single	Total
Material of a Form	1	1	2
Behavior of a Form	4	4	8
Shape of a Form	3	0	3
Function of a Form	7	7	14
Working Principles	3	3	6
Behavior of a Flow	8	10	18
Descriptor	0	1	1
Process	0	1	1

Conclusions and Future Work

As other work has shown, schema-based similarity of shared functions can improve retrieval of design analogies, so other types of abstract similarity should likewise facilitate computational analogy retrieval. The results of this work provide evidence suggesting a variety of abstractions to support schema-based design analogies. The **key findings** of this work include: (1) that flow behavior was observed as a commonly used type of abstract similarity for drawing analogical connections, and (2) that there was no significant difference in the types of similarity used to inform compound and single analogies.

Notably, while flow behavior was exhibited as a common connection between domains in this study, a flow behavior abstraction to support database-driven DBA does not currently exist. This study also resulted in several inconclusive observations about common types of internal knowledge queries, the frequencies of various concept generation categories, and the prevalence of direct reuse as a preliminary design strategy. Potential areas for future work include studying these areas in more detail, mining the collected data for further correlations, and investigating the relationship of concept quality to concept generation process, direction of reasoning, and the presence or absence of analogy.

In general terms, these results provide insights into the types of high-value mental shortcuts and processes that commonly facilitate analogy formation. As they relate to industry, the results inform the creation of CAD tools and knowledge management techniques to help novice designers see conceptual connections between institutional design knowledge and existing design challenges. This may be useful not only for helping novice designers to perform more like experts, but for helping companies interested in developing a dynamic and innovative product lineup to explore nonobvious cross-domain solutions and strategies.

The results of this study show no significant difference between the types of similarity used to draw compound and single analogies. This result suggests that the value of different types of similarity with respect to forming analogical connections is independent of the generative goal. Whether expanding the breadth of the concept pool (single analogy) or improving concept fidelity and problem understanding (compound analogy), different types of abstract similarity are equally useful.

With respect to the direction of reasoning, function-to-form, form-to-form and flow-to-form reasoning all occurred frequently in analogy formation. This suggests that function, flow, and form information should all play roles in similarity-based analogy retrieval. Of these, similarity between the behaviors of flows through a system is both prevalent in the results and missing from the categorizations in the reviewed literature.

For example, common analogies observed in this study include paper-processing devices (e.g., printers, printing presses, and junk mail folders) and sheet metal rollers. One way of abstracting this problem is by function. A search query of “shape material” could be used to retrieve metal rollers, but likely not printers. While printers change the shape of paper, the design intent of a printer has little to do with this behavior. The starting and ending shapes are also the same; so state-based methods may also have difficulty detecting this behavior.

Paper shares more literal similarity with towels than with sheet metal, yet all three undergo processes that could be used to flatten or fold something. There are properties of paper and sheet metal that relate to their “flattenability;” their emergent behavior under specific conditions. Many designers in this study inferred from these properties (and from observed behaviors of paper and metal) that paper and sheet metal are sufficiently similar to cloth that similar mechanisms will produce similar flattening behavior in both.

The results of this study suggest that a designer could leverage flow behavior abstraction (e.g., “foldability” and “flattenability”) to search for systems that interact with things possessing desired (e.g., towel-like) behavioral properties. More generally, abstracting the behavioral properties of flows (in addition to system functions) can be a valuable approach to finding analogical connections, especially when the analogy search is guided computationally. This approach provides a simple search heuristic to improve the quantity of potential high quality analogies for a designer to consider. Because analogy is a major component of design, this will improve design outcomes.

The Biology Phenomenon Categorizer: A Human Computation Framework in Support of Biologically Inspired Design

Ryan M. Arlitt, Sebastian R. Immel, Friederich A. Berthelsdorf, and Robert B. Stone

Journal of Mechanical Design

136(11):111105

Abstract

Locating relevant biological analogies is a challenge that lies at the heart of practicing biologically inspired design. Current computer-assisted biologically inspired design tools require human-in-the-loop synthesis of biology knowledge. Either a biology expert must synthesize information into a standard form, or a designer must interpret and assess biological strategies. These approaches limit knowledge breadth and tool usefulness respectively. The work presented in this paper applies the technique of human computation, a historically successful approach for information retrieval problems where both breadth and accuracy are required, to address a similar problem in biologically inspired design. The broad goals of this work are to distribute the knowledge synthesis step to a large number of non-expert humans, and to capture that synthesized knowledge in a format that can support analogical reasoning between designed systems and biological systems. To that end, this paper presents a novel human computation game and accompanying information model for collecting computable descriptions of biological strategies, an assessment of the quality of these descriptions gathered from experimental data, and a brief evaluation of the game's entertainment value. Two successive prototypes of The Biology Phenomenon Categorizer (BioP-C); a cooperative, asymmetric, online game; were each deployed in a small engineering graduate class in order to collect assertions about the biological phenomenon of cell division. Through the act of playing, students formed assertions describing key concepts within textual passages. These assertions are assessed for their correctness, and these assessments are used to identify directly measurable correctness indicators. The results show that the number

of hints in a game session is negatively correlated with assertion correctness. Further, BioP-C assertions are rated as significantly more correct than randomly generated assertions in both prototype tests, demonstrating the method's potential for gathering accurate information. Tests on these two different BioP-C prototypes produce average assertion correctness assessments of 3.19 and 2.98 on a five point Likert scale. Filtering assertions on the optimal number of game session hints within each prototype test increases these mean values to 3.64 and 3.36. The median assertion correctness scores are similarly increased from 3.00 and 3.00 in both datasets to 4.08 and 3.50. Players of the game expressed that the fundamental anonymous interactions were enjoyable, but the difficulty of the game can harm the experience. These results indicate that a human computation approach has the potential to solve the problem of low information breadth currently faced by biologically inspired design databases.

Introduction

Designers often base conceptual designs on previously known information [43-45, 84], and cross-domain analogies have proven to provide meaningful inspiration to design problems. However, designers do not typically possess biology knowledge in the depth and breadth required to discover applicable analogies to their engineering problems. As a consequence, the usual difficulties encountered in forming analogies [48] are highly exacerbated in biologically inspired design. Biologist-designer collaboration is one solution, but has the obvious drawback of relying on immediate access to a biologist. A design team will not always have a biology expert, and the knowledge of any single expert is unlikely to be applicable to a wide variety of design problems. In contrast, computational techniques have the potential to leverage vast quantities of existing biology knowledge to provide design inspiration.

At present, computational approaches to inspiration tend to be either natural language processing (NLP) or database driven. Database approaches require well-formed biology phenomenon knowledge, relying on expert human synthesis to correctly form the knowledge so that it can be repurposed as a design strategy. The expert is able to contribute their depth of knowledge to the database, but the number of contributing experts limits the breadth of knowledge in the database. In contrast, natural language processing approaches can quickly parse large text corpora, extracting biology knowledge as they go. These approaches generally use a combination of linguistic heuristics and statistical techniques to determine the most likely interpretations for strings of words [85]. However, humans remain superior at handling natural language (e.g., [86]). As a consequence, NLP approaches tend to trade information quality for raw information breadth.

Given that these approaches suffer from opposing challenges, there exists an opportunity to synergize complimentary elements from both. The main contribution of this paper is an attempt to address this conflict between breadth and quality. Specifically, the paper introduces a scalable method for populating a biologically inspired design database in a way that could be used by future computational tools to find design analogies across domains. The secondary contribution of the paper is an evaluation of the method.

This paper presents a novel approach to populating a biology knowledge database: The Biology Phenomenon Categorizer (BioP-C). BioP-C is an effort to combine the superior reasoning capabilities of humans with the raw processing power of computational approaches to biology knowledge categorization. BioP-C takes the form of a game that, as a side effect of players' participation, collects externally valid assertions about biological phenomena and strategies. These assertions are organized in a general and computable format that is designed to facilitate meaningful design-to-biology analogizing. The main goals of this paper are (1) to present the BioP-C game and information schema, and (2) to evaluate the validity of preliminary data collected using this framework. Validation activities include an evaluation of individual BioP-C assertions and an exploratory investigation of how these assertions' correctness might be improved.

The remainder of the paper contains the following sections: Related Work, Approach, Results and Discussion, Example, and Conclusions. The Approach section includes discussion of the BioP-C framework, player experience, and several validation approaches. The Results and Discussion section addresses the outcomes of each validation activity. The Example section presents the envisioned usage of BioP-C data for providing design inspiration. The paper concludes with a brief synthesis of the results and identifies areas for future work.

Background

This section discusses related research in design by analogy, biologically inspired design, human computation, and the dual efforts of Open Mind Common Sense and ConceptNet for collecting and organizing commonsense knowledge.

Analogy in Design

Biologically inspired design is a subset of design by analogy, and research on design by analogy has created a number of approaches to facilitate conceptual design analogizing. Among these are the IDeAL system [54, 58] which supports computer aided conceptual design with a data schema model, a functional similarity metric for finding design analogies [5], the WordTree design by analogy method [87], and Latent Semantic Indexing (LSI) approaches for finding analogous patents [88] and functional models [6, 89]. The IDeAL system, and similar prescriptive models of design knowledge, support analogizing between systems by enabling direct matching between values stored as the various data types. Relatedly, the functional similarity metric assesses the functional distance between products (and thus the degree of analogy to some extent), based on restricted-terminology functional models. In contrast to the prescribed schemas of database driven methods, the WordTree and LSI approaches rely on descriptive techniques. These techniques highlight existing conceptual connections rather than prescribing a new formalism. WordTree uses WordNet's [90] term connectivity to provide inspiration for potential analogies, while LSI and related techniques rely on large quantities of information and dimensionality reduction techniques to increase the likelihood of matches between noisy but related documents.

There exist numerous content models of design knowledge that could support analogy formation (e.g., [15, 91]), as well as many models of biology knowledge. Connecting the two domains is a challenge because it requires abstraction to a level shared by both biological and engineered systems. The Engineering-to-Biology

Thesaurus [92] represents an effort to connect these domains on the functional level, but connecting non-functional abstraction levels remains challenging. A general method is required to support broad conceptual connectivity across domains. Structure mapping [3], the theory that two domains are most strongly analogically related when they share similar structures of relationships between domain elements, is sufficiently general to facilitate cross-domain transfer. Structure mapping forms the theoretical underpinning of the approach presented in this paper.

Biologically Inspired Design

Existing biologically inspired design tools, which consist of both database and natural language processing approaches, serve as benchmarks for BioP-C. These existing tools include AskNature, the Design by Analogy to Nature Engine (DANE), Idea/Inspire, and two natural language search tools.

AskNature (asknature.org) and DANE [93, 94] are two examples of database approaches with excellent information depth. AskNature uses a catalog of strategy pages arranged by a function-based biomimicry taxonomy, with each page describing how a problem is solved by a specific biological phenomenon. DANE uses the Structure-Behavior-Function (SBF) data schema [91], providing additional conceptual representations of the systems, which can add value to a designer seeking understanding and inspiration from a biological solution. An evaluation of DANE in a classroom environment indicated that the students found the richness and multiple representations to be useful, but the lack of content was extremely limiting [95]. Conversely, AskNature's larger content database and relatively high quality of information offered good initial strategies, but the lack of detailed information made those strategies difficult to evaluate [95]. In both AskNature and DANE, human curation is the bottleneck preventing the databases from growing rapidly. In DANE, only the curators can provide additional information to the database. In AskNature, although users can submit new strategies, a curator must approve them before the strategy is searchable.

Idea/Inspire [69] is another database tool for supporting biologically inspired design that uses the SAPPhIRE (State change-Action-Part-Phenomenon-Input-oRgan-Effect) data schema [69], enabling direct connections across engineering and biology domains. Idea/Inspire contains over 100 plant and animal phenomena, and provides inspiration based on a user's verb/noun/adjective keyword search. Idea/Inspire represents a moderately large and relatively detailed database for assisting biologically inspired design, but by its nature suffers from the same curation problem as DANE and AskNature.

Natural language processing (NLP) techniques [85] provide an alternative approach. NLP approaches have the power to be descriptive rather than prescriptive in nature, and benefit from increased breadth of search over database approaches. In an evaluation of a basic text search tool in a classroom environment, designers had difficulty locating relevant inspiration in the search results, pointing toward a need for high quality term matching and filtering [95]. Improving upon this basic functionality, the BID Lab search tool [96, 97] uses language heuristics to address the filtering issue, locating analogies in a biology textbook by using part of speech patterns to identify causally related functions. Structure mapping theory supports causal function relationships as strong indicators of good analogy [3], but extracting these relationships using language heuristics can be sensitive to the writing style in the source text. For example, the technique of stylometry is used to identify authors based on “writer invariant” features of text. Syntactic features, such as part of speech patterns, have been experimentally verified for this purpose [98]. Applying the same part of speech patterns across different sources risks introducing variation into search results, thus returning extraneous information that the designer must filter manually.

Human Computation and Games with a Purpose

Human computation is a method that solves large problems by breaking them into subproblems that can be solved by many people. Luis von Ahn’s CAPTCHA [99] was initiated as a method for determining humans from spam bots to prevent automated scanning and collecting of information. CAPTCHAs require that the user identify a word that has been visually distorted so that optical character recognition (OCR) software cannot recognize it. Its successor, reCAPTCHA [100], was created to harness CAPTCHAs as a form of human computation for the purpose of deciphering scanned text that is unrecognizable by software. Other examples of human computation include classifying galaxy shapes [101] and identifying features on Mars [102]. Human computation approaches are inherently scalable because distributing the subtask to many participants is trivial.

A non-trivial portion of human computation is incentivizing users, as the task may solve a problem (e.g., protecting websites from spam bots), but does not add value for those completing the task. Games with a purpose (GWAPs) provide a solution to this problem by providing users with entertainment while harvesting information. Examples of GWAPs include the ESP Game for image tagging [103], Verbosity for common sense fact acquisition [104], Tag-a-Tune for music tagging [105], and Jinx for word sense disambiguation [106]. Generally, these games are two-player cooperative games where players are paired with an anonymous partner. Players are given no means of communication outside of the game itself, as any communication could be a method of subverting the game and invalidating the gathered data.

The game design of a GWAP is shaped by the desired information. In a symmetric game, both players perform the same task, and verification is established by agreement between responses. Because both players must perform the entire task, symmetric games are generally limited to simple tasks. In an asymmetric game, each player performs a separate task, with data coming from ‘hints’ generated by one of the players, and validation coming from the other player, who uses the ‘hint’ to select the correct response. Asymmetric games generally have more complex information requirements.

For example, the ESP Game is a symmetric output-agreement game [107] that collects image tags. Player pairs type descriptions of the same image, as quickly as possible, until they type the same word. Players’ scores increase as they improve their speed. This scoring function incentivizes players to produce a high quantity of image tags, and correctness is confirmed via tag frequency over multiple games on the same image. Another symmetric game (Tag-a-Tune) collects labels for audio clips. Because the task of labeling music styles naturally produces less convergence than labeling images, Tag-a-Tune uses an input-agreement mechanic [107]. In games using this mechanic, players must infer whether they are working from the same source information based on the output that their partner is producing.

Asymmetric games can provide the sophisticated agreement mechanics required by more complex tasks. Each partner is given a different role in the game, which is accompanied by a different task. For example, the GWAP Verbosity is designed to collect commonsense facts from players. Players are assigned to the role of either Narrator or Guesser. The Narrator is given a secret word, and must used restricted-form templates to provide hints about the word (e.g., “It contains a keyboard” as a hint for “laptop”), without using any words from a restricted list. The templates result in well-formed assertions about the key word, while the word restrictions promote a breadth of new assertions. Assertions are rated on both a time-dependent scoring function and the frequency with which they are independently created. Verbosity’s fact quality was evaluated by asking six human raters to evaluate whether the statements in a 200 assertion sample were true [108], and a similar validation approach is used in this paper.

Another GWAP, called “Foldit” (fold.it), has players solve 3D puzzles. These puzzles are surrogate representations of protein folding problems in computational biology. Players maximize their score by folding a simulated protein structure into low energy states, which contributes to solving the related protein-folding problem [109]. Foldit has led to meaningful results in the protein-folding domain where experts alone were unsuccessful [110, 111]. This game exemplifies a class of single-player GWAPs that use a built-in scoring function based on descriptive models with high external validity. Foldit demonstrates that relatively few people are necessary to reformulate a problem requiring high cognitive effort into one that non-experts can solve.

ConceptNet and Open Mind Common Sense

Open Mind Common Sense (OMCS) [112] is a platform for gathering common sense knowledge from the general public. A series of online tasks gathered common sense facts (e.g., “baseball is a sport”) from visitors to the OMCS website, accumulating over one million common sense facts. Early versions of OMCS contained several different types of information gathering activities, but template activities (similar to those later used in Verbosity) produced the highest quality of information [112]. Later versions of OMCS used a set of templates created from previously generated natural language facts. BioP-C uses a similar template style to organize information. In this paper, the most highly rated facts on the OMCS website are used to benchmark BioP-C’s assertion quality.

ConceptNet is a related project that consists of a semantic network containing these OMCS facts [113], and enables some basic reasoning using these assertions [114]. In ConceptNet, concepts are graph nodes and relationships are directional edges. BioP-C uses a similar structure, leaving open the possibility for future merging with ConceptNet data. Additionally, dimensionality reduction (e.g., singular value decomposition) has been used to effectively find analogous concepts in ConceptNet [115]. Such a technique reduces the variation created by natural language concepts and relationships, improving general network connectivity and increasing the number of suggested analogies. Another rank reduction technique for finding design analogies uses LSI (which is itself based on singular value decomposition) on patent documents [6, 88]. Improvements to the basic LSI algorithm include probabilistic LSI [116], which improves precision; and latent Dirichlet allocation, which improves topic mixtures [117].

Approach

The research approach is described next, broadly organized into two sections. The Framework section details the general framework, the method for collecting source text, the BioP-C game, and the limited relationship set used as both gameplay elements and a representation of the collected knowledge. The Validation Approach section discusses validation efforts including descriptions of how trial data was collected, an approach to assessing the correctness of BioP-C's assertions, and an assessment of the game's entertainment value.

Framework

The knowledge-gathering process begins with collecting small chunks of domain-relevant text, and ends with a representation of the relationships within and between the atomic concepts in these texts.

A set of roughly paragraph-sized texts is collected from reliable sources within the domain in which a network of relations is desired. This is ideally an automated process. For example, in the prototypes presented later in the paper, a collection of research paper abstracts was collected on the subject of cell division using a semi-automatic process. This collection of texts is stored in a database and is presented to the players of the online game BioP-C. In each round of the game, one of these paragraphs serves as the main element on the playing field.

During each round, players generate assertions from the source text as a byproduct of gameplay (e.g. "**cell** *has* **cell membrane**"). Session data is recorded to assess assertion correctness, improve gameplay mechanics, and identify undesirable player dynamics. Once a set of assertions is generated, it is added to the collective information network. This is represented as a graph, with concept tokens (e.g. **cell**), as nodes, and relationships (e.g. *has*) as edges.

A search algorithm can traverse this graph for the purpose of finding analogies – related concepts with similar relationship structures. Liu and Singh [114] demonstrate a spreading activation approach for searching ConceptNet, which is convenient for applying assertion confidence (stored as edge weights) to finding analogy candidates. The spreading activation algorithm establishes baseline similarity by propagating edge weights outward from a single source node. Weights naturally decay as they spread further from the source, but weights on merging paths are additive. This technique finds nodes that are strongly connected through many different paths in the graph, which means that related nodes possess a high degree of shared information. Matches at this stage reflect an unknown combination of simple attribute similarity and analogical similarity. Further subgraph matching around these potential matches can be evaluated for deeper relationship alignment, which more closely reflects true analogy. The BioP-C graph closely reflects the design of ConceptNet, enabling support for similar search techniques, while also permitting compatibility between the two information structures.

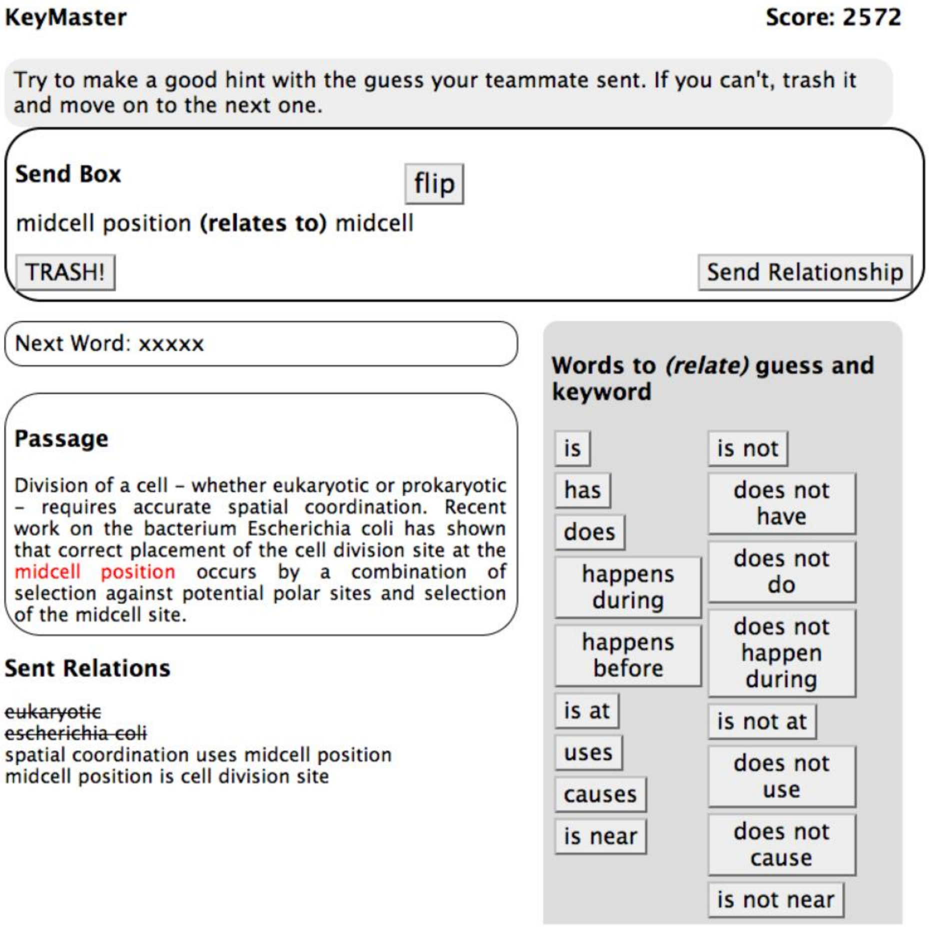
The BioP-C Game

The BioP-C game enables collection of assertions about biological phenomena. It should be noted that there have been three iterations of this game. The first (v0.1) was a single player game not previously reported on. The second (v0.2) was used to generate a significant amount of usable data and feedback from players, and provided one set of data for the validation. The third version (v0.3) is a refinement of v0.2 with mostly cosmetic improvements, and it provided a second set of data for this study. BioP-C v0.3 is described here, and changes between versions are explained where relevant.

Two players, called the Codebreaker and the Keymaster, engage in an anonymous asymmetric online game. The players share knowledge of a paragraph (in this study, an abstract from a set of biology papers related to cell division). The game begins with the Keymaster picking a word or contiguous phrase from this paragraph. This selection is referred to as “the keyword.” The Codebreaker then begins selecting words or phrases from the shared paragraph, attempting to guess the keyword. No other communication is possible.

For every guess the Codebreaker makes, the Keymaster can respond with a relationship to the keyword from a list on the right-hand side of their screen. They have nine possible relationship types, which are matched to the guess in the form “**A** *relates to* **B**.” The placement of the guess and the keyword is up to the Keymaster. Eight of these nine relationships have a corresponding negated relationship (e.g. *is* and *is not*), and the Keymaster also has the option to discard guesses that are unrelated to the keyword. Keyword negation and discarding are new to v0.3. Players in v0.2 had the option of ignoring a guess, which is not possible in v0.3.

For example, consider the phrase “…correct placement of the cell division site at the midcell position…” The Keymaster selected the keyword “**midcell position**,” and the Codebreaker responds with a guess of “**cell division site**” The Keymaster could then choose to respond with the phrase "**midcell position** *is* **cell division site**" (as shown in Figure 11), or perhaps "**cell division site** *is at* **midcell position**." The Keymaster's response is saved to the database as a player-made assertion, and then sent back (with the keyword removed) to the Codebreaker to inform future guesses as shown in Figure 12. In the case of these example responses, the Codebreaker would see "**?????** *is* **cell division site**" and "**cell division site** *is at* **?????**" respectively. This process continues until the Codebreaker guesses correctly or the players' shared score reaches zero, ending the game.



KeyMaster **Score: 2572**

Try to make a good hint with the guess your teammate sent. If you can't, trash it and move on to the next one.

Send Box

midcell position **(relates to)** midcell

Next Word: xxxxx

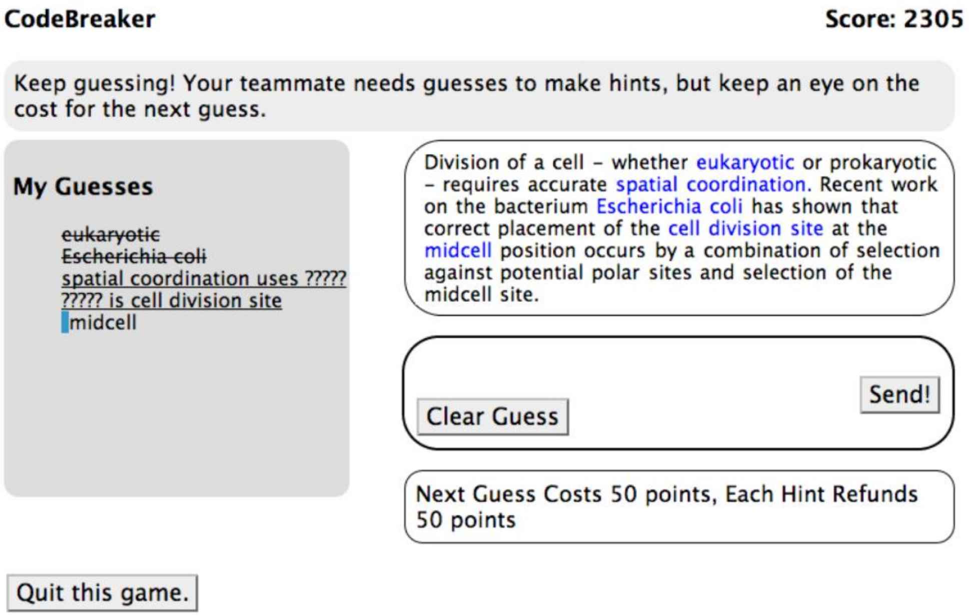
Passage

Division of a cell – whether eukaryotic or prokaryotic – requires accurate spatial coordination. Recent work on the bacterium Escherichia coli has shown that correct placement of the cell division site at the midcell position occurs by a combination of selection against potential polar sites and selection of the midcell site.

Sent Relations

~~eukaryotic~~
~~escherichia coli~~
spatial coordination uses midcell position
midcell position is cell division site

Figure 11. BioP-C v0.3 Keymaster view (paragraph sourced from [118]



Keep guessing! Your teammate needs guesses to make hints, but keep an eye on the cost for the next guess.

My Guesses

~~eukaryotic~~
~~Escherichia coli~~
spatial coordination uses ?????
????? is cell division site
midcell

Division of a cell – whether eukaryotic or prokaryotic – requires accurate spatial coordination. Recent work on the bacterium Escherichia coli has shown that correct placement of the cell division site at the midcell position occurs by a combination of selection against potential polar sites and selection of the midcell site.

Next Guess Costs 50 points, Each Hint Refunds 50 points

Figure 12. BioP-C v0.3 Codebreaker view (paragraph sourced from [118])

The scoring system in BioP-C v0.3 is uses a combination of game duration and number of hints, and starts players with a shared pool of points that slowly decrements over time. This shared score also decrements by a variable amount with every guess made by the Codebreaker, and increases with every hint formed by the Keymaster. The game ends when a score reaches zero, which disincentivizes mass random guessing. Players' scores in BioP-C v0.2 simply decremented after each guess, and these scores had no impact on ending the game.

It is important to note that these interactions are asynchronous. As the Keymaster is selecting a relationship which they feel best accommodates the guess they have been given, the Codebreaker can send additional guesses. The Keymaster only sees the current guess and the next guess in the queue if it exists.

The information captured from these exchanges between players is stored on a directional multigraph (i.e., edges have directionality and multiple edges can exist between the same pair of nodes) that captures atomic concepts on nodes and relationships on edges. Edge types follow a taxonomy generalized from that used in the Open Mind Common Sense (OMCS) [112] project to catalog common sense facts in any domain. In BioP-C these relationships are intended to capture general physical, spatial, functional, temporal, and causal connections between concepts. This type of knowledge organization supports extremely general schema-based analogizing approaches, and supplements them with connectionist techniques (i.e., techniques based purely on graph connectivity independent of knowledge content).

Implementation

In the BioP-C prototypes, data is stored in a Postgres SQL database containing source paragraphs, user-selected words and relationships, and session data for each game. A JavaScript powered front-end employs a simple short-polling design pattern, and interfaces with the database using the Django web framework. Additionally, the jQuery JavaScript API was used to minimize the need to accommodate specific browsers.

Relationship Set

There is a tension between the entertainment value of BioP-C as a GWAP and the desire for high precision representation of biology concepts. Existing taxonomies were considered (e.g., the Functional Basis [15]), however the multitude of potential terms (in addition to learning their precise definitions) could easily overload players, making these existing choices a poor game design option at this stage. In the current version of the game, the emphasis has been on enjoyable gameplay. To further this goal, the twenty-two concept relationships identified in [40] have been condensed into a list of nine: *is*, *has*, *causes*, *happens before*, *happens during*, *is at*, *is near*, *does*, and *uses.* These words represent a compromise that provides a relatively simple user interface.

Initial play testing confirmed that having the complete set of twenty-two ConceptNet relationships visible was unwieldy and confusing. Distinctions between ConceptNet relationships can be quite subtle, and even in the case of clearly distinct relationships, twenty-two unique options exceeds the number that a player could be reasonably expected to deal with, motivating the consolidation of relationships.

The goal of this consolidation was to reduce cognitive load, especially for new players. By restricting the number of choices a player can make for a given relationship to a minimum while still addressing every category of relationship identified by OMCS, ambiguous cases where different people might make different choices are also minimized. To accomplish this, the authors began with an arbitrary goal of having seven relationships that could be directly mapped to the OMCS set. The current set represents numerous iterations and evaluations of the design implications thereof. The nine relationships chosen conform to the criteria that they are simple English, and every OMCS word can be mapped to at least one of them. In response to player feedback, BioP-C v0.3 introduced a negated relationship for each positive relationship (e.g., *does not* for *does*). The *happens before* relationship is a unique case because it is reversible by flipping its direction. The negated relationships are shown in Figure 11.

The precision of this mapping varies somewhat, but the overall generality promotes human-driven abstraction that supports general analogizing. OMCS has twenty-two relationships in eight categories. BioP-C has nine relationships in 8 categories. The category of "General Relationship" is considered to be a superclass of all the relationships used in the game, especially the relationships *is*, *uses*, and *does*.

Table 14 shows the mapping of the BioP-C relationships to the OMCS relationships. It can be clearly seen that there is not a one-to-one correlation between the two sets. The BioP-C relationships were chosen to be as general as possible while maintaining separation between categories. This set of connecting words was chosen with the ideal goal that there should be one relationship that was clearly the best for any given situation. This meant restricting the possible relationships, but not so much that a user would not find themselves unable to select a reasonable word to connect a keyword and guess.

The relationships *is at* and *is near* represent spatial relationships. As in English, *is at* can also express a temporal relationship. The relationships *happens before* and *happens during* are used to round out the time relationships. Causal relationships are represented by *causes.* This is an example of where a colloquial word is used instead of a more explicit relation such as “EffectOf,” thus sacrificing some precision to be more relatable to a user.

The word *uses* represents what might be called a catch-all word, whose purpose is to cover any number of complex relationships. This showcases the tradeoffs inherent in designing a word game for both correctness and fun. By combining a large number of potential relationships into a single word, the cognitive load on the player is reduced. By the same token, the precision for the game is diminished.

Table 14. BioP-C relationship mappings to OMCS relationships

Category	BioP-C Relationships	OMCS Relationships
General Relationship	All	ConceptuallyRelatedTo
		ThematicKLine
		SuperThematicKLine
Things *(Structure)*	is, has	IsA
		PropertyOf
		PartOf
		MadeOf
		DefinedAs
Agents *(Function/Behavior)*	uses, does	CapableOf
Events *(Function/Behavior)*	happens before, happens during, is at	PrerequisiteEventOf
		FirstSubeventOf
		SubeventOf
		LastSubeventOf
Spatial	is at, is near	LocationOf
Causal *(Behavior)*	causes	EffectOf
		DesirousEffectOf
Functional *(Function/Black Box Function)*	does	UsedFor
		CapableOfReceivingAction
Affective *(Function/Black Box Function)*	uses	MotivationOf
		DesireOf

As functional relationships play an important role in design analogy formation, the *does* relationship is specifically intended to describe function-to-form connections. Such a relationship supports a simple back-edge matching approach to finding functional analogies.

The structure words *has* and *is* exemplify the limits of the current set of BioP-C relationships. These two words can express a wide range of structure relationships, but OMCS has five structure words, making it capable of more subtlety. In future versions of BioP-C, it will be desirable to capture these more nuanced relationships. One solution would be to have a sliding scale of lexical depth correlated to difficulty level. Players also have the option to reverse the directionality of relationships (i.e., "**A** *relates to* **B**" vs. "**B** *relates to* **A**"), which improves the descriptive power of this small set without introducing more relationship types.

Gathering Source Text

It is a considerable challenge to automatically locate and reliably interpret passages describing biological phenomena. Reducing the problem to that of simply locating such passages is much simpler, but still nontrivial. BioP-C v0.3 uses journal abstracts located in ScienceDirect's [119] "Agricultural and Biological Sciences" category with the keywords "cell division." Abstracts were selected because they are available in high quantity, contain succinct but significant biological information, contain inherent links for further information if required, and have the general reliability of a peer-reviewed source.

Research using the Engineering-to-Biology Thesaurus [120] has demonstrated that biology keywords are more likely to cause inspiration if they are common vocabulary, rather than technical jargon. Eq. 1, used by Glier et al. to rank word commonness in several different domains [120], gives a normalized score for word readability. In Eq. 1, N_i gives the normalized readability score of word i where N_i is the frequency of word i in the text and N is the vector of frequencies for all words in the text. The 5000 most common words from the Corpus of Contemporary American English [121] are used to establish readability. Additionally, passage length is considered, with shorter passages ranking more highly. The logarithmic terms normalize scores between 0 and 1, which is convenient when distributing weighting factors in calculating the overall passage rankings. Eq. 2 ranks passages based on length and the frequency of their words in common vocabulary, using each passage's median word readability score and normalized inverse passage length. In Eq. 2, R_j gives the overall readability score of passage j where $w_{\widehat{N}}$ is the readability weighting factor, $\widehat{N_j}$ is the vector of all individual word readability scores in passage j, w_n is the passage length weighting factor, n_j is the number of words in passage j, and n is a vector containing the number of words in each passage. The ranking scheme arising from these two equations prioritizes short passages with a low ratio of jargon to common vocabulary. The validation activities presented in this article use equal weightings of 0.5 for both readability and passage length.

$$\widehat{N_\iota} = 1 - \frac{\ln(N_i + 1) - \ln(\max(N))}{\ln(\min(N) + 1) - \ln(\max(N))} \qquad \text{Eq. 1}$$

$$R_j = w_{\widehat{N}} * median(\widehat{N_j}) + w_n \left[1 - \frac{n_j - \min(n)}{\max(n) - \min(n)}\right] \qquad \text{Eq. 2}$$

This technique still benefits from including a human rater to remove miscategorized, non-biology, and opinion papers. However, because this can occur after ranking, human curation is limited to the highly ranked abstracts. The abstract ranking algorithm was informally validated by human raters, who ranked a randomized subset of abstracts about cell division into the same high/low ranks as did the algorithm. Further development and formal evaluation of the passage-ranking algorithm are areas for future work, as the current reliance on human curation is unsatisfactory.

Validation Approach

Broadly, the goal of validation activities in this paper is to address the question: “Does the GWAP approach embodied by BioP-C represent a feasible approach for populating a biologically inspired design database with broad and accurate information?” In order to conduct this evaluation, the two prototypes were given to two different graduate-level design classes. BioP-C v0.2 was presented in a product design class and BioP-C v0.3 was presented in a biologically inspired design class, both in the mechanical engineering department. Both game tests were preceded by a lecture explaining the basic goals of BioP-C as well as how to play. This lecture had the dual purposes of (1) teaching the students how to play and (2) replicating the ecological conditions faced by many GWAPs, where ethical concerns require that players be informed of how their data will be used. Students’ performance and participation in the activities had no impact on their grades in the course. All activities were conducted over the course of approximately one hour during a regularly scheduled lecture time.

This arrangement produced a large number of distinct assertions with which to test basic hypotheses regarding their correctness. While the number of participants in each trial was relatively low, each player produced multiple assertions based on a variety of complex factors. The player dynamics that emerge from interactions between the Keymaster and Codebreaker are influenced by player personalities, passage text, and evolving game history. Because of the high complexity arising from the combination of these interdependent and continuously changing factors, multiple assertions created by the sample player can be treated as unique data samples. This is in contrast to many design studies wherein subjects produce at most one data point per treatment group.

In the first test, 11 students and one instructor generated 105 assertions in 28 game sessions with some baseline relatedness to cell division. An additional 135 assertions regarding cell division were collected from 18 students across 38 game sessions in the second test. For this study, two random subsets of 50 assertions were sampled from each test session. A technical error during assertion retrieval from the database invalidated 27 assertions from the BioP-C v0.2 test that were used to create the correctness assessment survey. As a result, the BioP-C v0.2 assessment uses a sample size of 23 assertions per treatment group, while the BioP-C v0.3 assessment uses 50 assertions per treatment group.

The BioP-C v0.3 game improves on BioP-C v0.2 by improving player matchmaking, adding a basic timer-based scoring system, allowing the Keymaster to discard unrelated guesses, and enabling the Keymaster to form hints involving negation (e.g., "**preprophase** *is not* **critical cell volume**"). While these game design details are different between tests, the underlying mechanisms for collecting assertions are the same, and thus both sets of tests are informative about the overall method.

Two factors are assessed using this data: (1) individual assertion correctness and (2) players' enjoyment of the game. Correctness addresses the factual accuracy of statements produced by the users, and is the product of human assessment of assertion truth. This approach aims to provide a bottom-up view of the technique's validity. The correctness of individual assertions (factor 1) is measured by comparing the assertions' relative truth against groups of low-correctness and high-correctness statements. Additionally, the correctness values are used to check for correlations with directly measureable session data. A general assessment of entertainment value (factor 2) addresses the likelihood that people will play the game, and helps to identify areas for improvement. The first factor addresses the game's ability to collect accurate information. The second factor addresses the likelihood of obtaining a broad player base and thus broad information.

Relative Assertion Correctness

The correctness of BioP-C's assertions is assessed by comparing them versus assertions at the theoretical upper and lower bounds of correctness. This assessment takes the form of a 150 Likert item survey. Each item in the survey is an assertion. For each assertion, the rater indicates their level of agreement with whether the assertion is true on a five point Likert scale, where a response of five indicates "strongly agree," four indicates "agree," three indicates "neither agree nor disagree," two indicates "disagree," and one indicates "strongly disagree." A response of three suggests rater uncertainty about assertion truth due to factors including ambiguity and conditional truth of the assertion. Each assertion comes from one of three equal-sized groups: BioP-C assertions, random nonsense assertions, and high quality assertions from Open Mind Common Sense. Surveys were completed digitally and the presentation order of survey items was randomized for each respondent. In order to mitigate rater fatigue, a response of "NA" was presented as an option to indicate "I don't know." Two Wilcoxon signed-rank tests address whether mean Likert ratings for each assertion are significantly different between (1) BioP-C assertions versus random nonsense and (2) BioP-C assertions versus Open Mind Common Sense assertions.

For the lower bound of correctness, nonsense assertions were randomly generated from concept tokens identified by BioP-C players. To generate a single statement, two tokens were randomly selected and paired with a random relationship type from the BioP-C relationship taxonomy. This algorithm generated statements like: (1) "**dinosaurs** *does* **genes**," (2) "**control** *has* **changes in cell shape**," and (3) "**biochemical events** *happens during* **growth."** Clearly most of these randomly generated statements are nonsense, but the algorithm occasionally produced an apparently true statement, as in the third statement above. The purpose of comparing BioP-C's assertions to these random statements is to assess the quality of relationships selected by players, independent of the quality of tokenization, as compared to random chance. Using BioP-C tokens and relationships rather than those from a different vocabulary produces a conservative test because the nonsense assertions are stylistically similar to BioP-C assertions.

For the upper bound of correctness, the top fifty assertions were taken from the OMCS website, as voted upon by site visitors. These assertions represent the theoretical upper bound of correctness for short natural language statements generated via human computation. Representative assertions from this set include: (1) "**baseball** *is* **a sport**," (2) "*an activity* **a dog** *can do is* **bark**," and (3) "**a book** *can be made of* **paper."** While OMCS assertions have more expressive capability, the general shared **token** *relationship* **token** style enables direct comparison with between BioP-C and

OMCS. The purpose of comparing BioP-C’s assertions to high quality statements is to assess the relative validity of BioP-C’s information gathering mechanisms.

Correctness Indicator Correlations

In an effort to streamline identification of assertion correctness, correlations are assessed between the subjective correctness ratings and quantitative session data. A significant correlation between directly measurable quantities and correctness would provide a way to automatically assign confidence values to facts (independent of raw assertion frequency). Two correlations with correctness were tested: (C1) the number of hints given in a single round and (C2) the length of the source passage. For C1 it is hypothesized that a large number of hints is indicative of the hints themselves being inadequate. Good hints should lead players to the solution in fewer iterations than bad hints. For C2, it is hypothesized that players have more difficulty synthesizing longer passages. Because providing hints requires the player to synthesize the passage, increasing the difficulty of synthesis should lower the quality of hints.

Entertainment Value Assessment

The entertainment value of the game is important because it contributes to a large player base, and thus more data. As a consequence, it is important to assess the entertainment value of the prototype game to identify ways to improve the player experience. In this study, a brief survey was given to players after each prototype test. The responses to these questions are used to assess the game’s entertainment value and identify improvement opportunities.

Results and Discussion

The Results and Discussion section contains results for relative assertion correctness, correctness indicator correlations, and assessment of entertainment value.

Relative Assertion Correctness

In order to establish the relative correctness of BioP-C v0.2 and BioP-C v0.3 assertions, twelve human raters (six per dataset) ranked their general agreement with assertions from Open Mind Common Sense (OMCS), BioP-C, and a group of randomly generated nonsense assertions. The BioP-C v0.2 survey contained 50 assertions from each treatment group, but a technical error necessitated reducing this number to 23. The BioP-C v0.3 survey contained 50 new assertions from BioP-C v0.3, but the same OMCS and nonsense assertions. Random-ordered versions of these relative correctness surveys were given to two different sets of six mechanical engineering graduate students. The surveys were untimed, and the respondents were given instructions to respond with “NA” if they could not decide how to respond.

The missing data from “NA” responses complicates the analysis, and so statistical tests were conducted in two different ways: (1) by ignoring “NA” entries when averaging responses to each assertion, and (2) by replacing each respondent’s missing data with the imputed mean for that respondent. Approach (1) weights individual ratings differently based on the number of raters that respond with “NA” for an assertion, while (2) tends to pull ratings to the center. No single assertion was rated as “NA” by all raters, indicating that some raters chose to assign neutral or low scores to ambiguous phrases. The nonsense category used to benchmark BioP-C’s performance captures the raters' tendencies for scoring ambiguously constructed statements as moderately low correctness.

Two different Wilcoxon signed-rank tests, each comparing BioP-C assertion correctness scores against those of the nonsense and OMCS statements, are both statistically significant ($p < 0.05$) for both datasets using both approaches to handling missing data. Using approach (1), the p-values for the BioP-C v0.2 dataset are 0.003 for BioP-C vs. Nonsense and 2.13e-4 for BioP-C vs. OMCS. For the BioP-C v0.3 dataset, these values are 0.022 and 1.69e-9 respectively. Using mean imputation, these values are 2.07e-4 and 5.55e-5 for v0.2, and 2.89e-4 and 1.15e-9 for v0.3. A Bonferroni correction for two tests per dataset modifies the significance level from α=0.05 down to α=0.025, which does not change these findings. The result of this relative comparison indicates that BioP-C's statement correctness is better than random, but does not approach the correctness of the most highly agreed upon OMCS statements. The first approach to handling missing data produced the most conservative p-values. A comparison of average scores for each group is shown for BioP-C v0.2 in Figure 13 and BioP-C v0.3 in Figure 14.

The raw quantities of "NA" responses in each treatment group correlate inversely with assertion correctness, which speaks to the difficulty of the rating task. The OMCS treatment group assertions received 0% and 1% "NA" ratings as a percentage of the overall number of ratings in the BioP-C v0.2 and v0.3 surveys respectively. The BioP-C assertion groups received 25% and 34% "NA" responses, and the nonsense group received 33% and 45% "NA" responses. Based on informal conversations with the raters, the "NA" ratings were typically used in response to two situations:

assertion ambiguity and rater's incomplete biology knowledge. The higher prevalence of "NA" in the nonsense group indicates that denoting ambiguity was its primary usage mode.

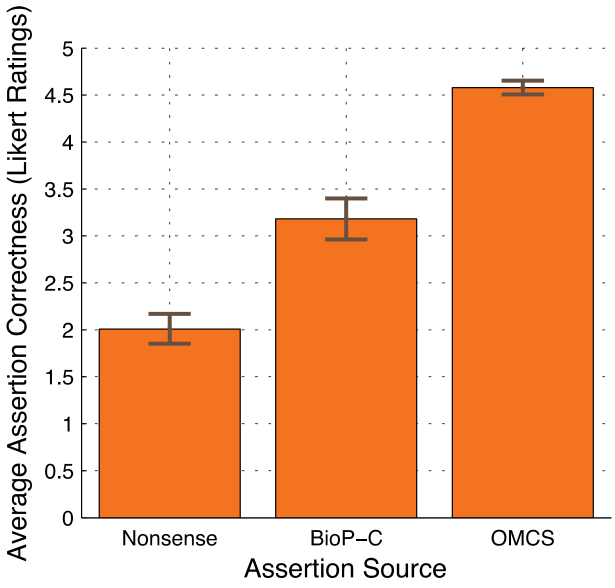


Figure 13. Relative assertion correctness for BioP-C v0.2 (N=23)

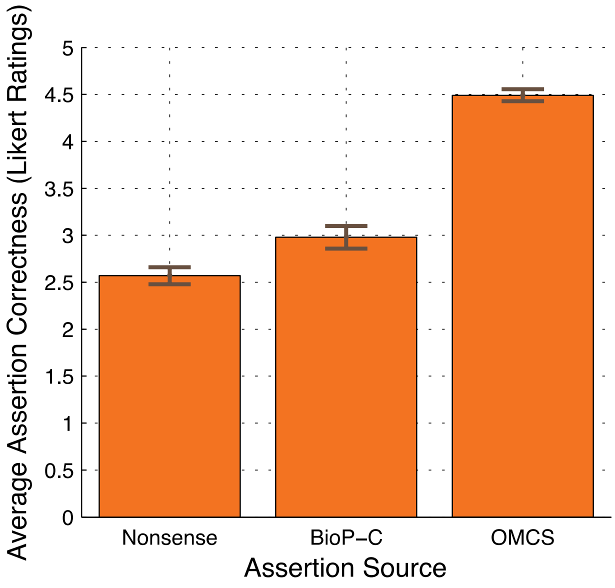


Figure 14. Relative assertion correctness for BioP-C v0.3 (N=50)

Comparing these nonsense assertions against the BioP-C assertions rather than interpreting the raw Likert scores of each group blocks the noise resulting from raters' preferences about how to handle ambiguous statements. Future assessments of this kind will include multiple types of "NA" responses in order to better capture the impacts of ambiguity and rater knowledge on correctness ratings.

While BioP-C's outperformance of nonsense data is a meaningful result, its assertion correctness is also a significant distance from the theoretical maximum. A number of factors contribute to this outcome.

First, the abstraction imposed by the small relationship schema limits expressive capability. Aggregate statement correctness is reduced as a direct consequence. In contrast, this abstraction improves graph connectivity, which supports analogy formation. For example, the relationship *does* is intended to capture function-to-form relationships. It is simple to search for such abstracted relationships on a given function node in the network in order to return potential forms that solve the function.

Second, these tests compare unfiltered BioP-C assertions against heavily filtered OMCS assertions. The average correctness of the BioP-C assertion group is harmed by apparently unclear assertions, such as "**`site`** *`does`* **`accomplished`**," which lower the average value of the information. The presence of such assertions highlights the importance of filtering and weighting mechanisms to prioritize information that is likely to be correct. There are two primary mechanisms for filtering these nonsense assertions. The first is to weight BioP-C graph edges based on the number of times an assertion is made. The second is to identify session data predictors of whether a statement is likely to be true. These mechanisms can be employed in combination with game mechanic refinements that improve unfiltered assertion correctness.

Notably, the average unfiltered BioP-C scores for both tests are near three on the five-point Likert response scale. This clearly indicates that BioP-C does not consistently produce statements that are unambiguously true in the absence of context. However, the significant difference in ratings between the BioP-C and Nonsense assertion groups suggests that BioP-C produces *some* valuable information. This combination of high and low quality information is an expected outcome of a relatively complex human computation task. The Correctness Indicator Correlations section discusses a filtering technique for isolating this valuable information without involving human raters.

Correctness Indicator Correlations

In order to identify potential correctness indicators from directly measureable session data, two correlations with assertion correctness were tested. The first test examines the number of hints in the assertion's source session, and the second test examines the number of words in the assertion's source passage. Both tests are conducted on the combined data set of 73 correctness ratings from BioP-C v0.2 and BioP-C v0.3.

Pearson's correlation coefficient is calculated for both dataset pairs. Additionally, a t-test is performed to determine the statistical significance of each linear correlation. The correlation between number of hints and correctness gives Pearson's $\rho = -0.298$ and $p = 0.011$ (Figure 15), indicating that there is a statistically significant moderate negative correlation between the number of hints in a game session and assertion correctness (for α = 0.05). A separate study of 70 randomly sampled data points from BioP-C v0.2 suggested this same correlation, but fell just short of reaching statistical significance [122]. For passage length and correctness, Pearson's ρ = -0.030 and *p* = 0.803 (Figure 16), indicating no correlation between passage length and assertion correctness. While results from the separate study of 70 BioP-C v0.2 assertions suggested a significant positive correlation between passage length and correctness (longer passages predicted higher correctness [122]), the study presented here suggests no such correlation.

As expected, a lower number of hints in a game session can serve as a predictor for higher assertion correctness. This result can be trivially applied to the datasets from the relative assessments to filter out all sessions with a large number of hints. An optimal cutoff number has not been determined for mass scale implementation, and so this section presents the best cutoff for both sets of assertions. In the BioP-C v0.2 dataset, filtering out sessions with more than three hints reduces the number of assertions from 23 to 12, and increases the group's mean correctness value to 3.64 (from 3.19). The corresponding median value is increased to 4.08 (from 3.00). A Wilcoxon signed-rank test against a random sample of 12 nonsense assertions indicates that this filtered subset of BioP-C assertions differs from the nonsense assertions by a statistically significant amount (*p* = 0.026). The BioP-C v0.3 dataset's correctness is most improved by filtering out sessions with greater than six hints, which reduces the set size from 50 to 17 assertions. The corresponding mean and median values are increased to 3.36 and 3.50 respectively (from 2.98 and 3.00), and a Wilcoxon signed-rank test indicates a statistically significant difference in correctness from a randomly sampled set of 17 nonsense assertions (p = 0.017). This evidence serves to demonstrate the potential of using this type of session data to filter assertions. Further studies of this variety may help to identify additional filtering criteria and optimal cutoff values.

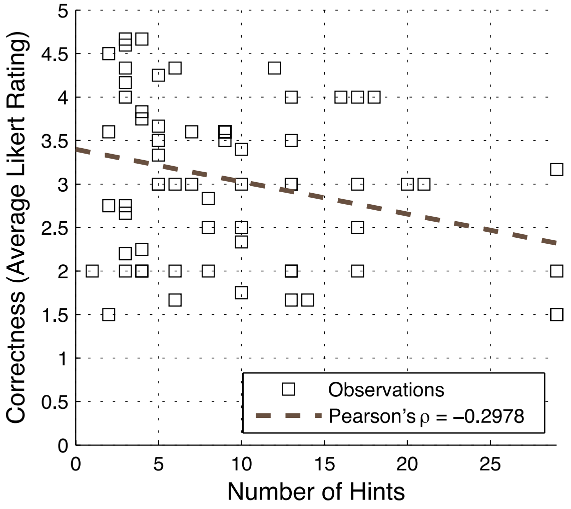


Figure 15. BioP-C assertion correctness versus number of hints in source game (p = 0.0105)

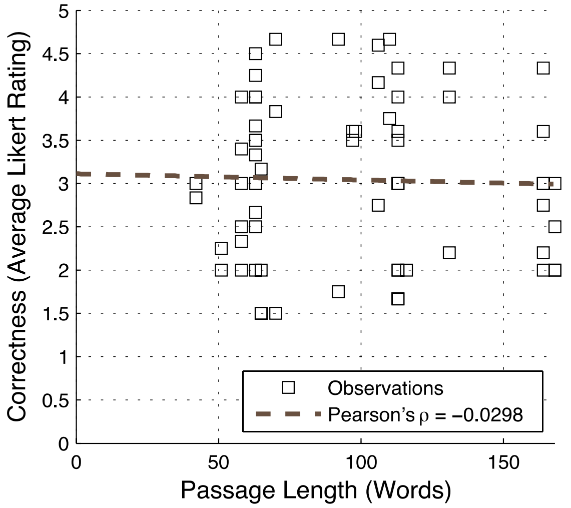


Figure 16. BioP-C assertion correctness versus source passage length (p = 0.8027)

A potential limitation of this analysis is that individual player data was not collected across multiple game sessions. It is likely that players have innately different skill levels, which should translate directly into higher assertion correctness. These unmeasured skill levels are confounded with the passage length and number of hints in a game session. For example, a single session with a given player pair produced nine different assertions. Each of these assertions has a different correctness value associated with it, but all of these assertions came from the same game session. In this case, all of the different correctness scores would be associated with the same player skill level and same passage length of 92 words. As a consequence, individual player skill represents a potential noise factor in these tests as well as a potentially beneficial correctness predictor to be studied in future work.

Entertainment Value Assessment

In order to elicit feedback about the player experience, a short open-ended survey was given to both groups of students after their participation. Subjectively, this feedback leads the authors to conclude that while these prototypes were perceived as more fun than an early prototype that lacked game elements, BioP-C in its current state is not seriously competitive with existing entertainment options.

The feedback from this survey highlights some key attributes of the BioP-C player experience. Most notably, a frequent comment from both tests indicates that players enjoyed trying to work together with their teammate. These emergent interactions between players are a necessary component of the data collection mechanism, so it is advantageous that this mechanism is not in conflict with entertainment value. Multiple players also commented that the entertainment value of the game depends on the quality of the hints; players that form high quality hints and guesses are fun partners, while players that produce low quality guesses and hints also harm their partner's experience. This suggests that a player population will self-select for high-performing players, while low-performing players will quickly abandon the game. Foldit mitigates the issue of game difficulty by providing a series of simple tutorial puzzles to gradually introduce the player to game mechanics. A similar technique could be used in BioP-C to form a core of high-performing players that allows the population to grow.

This dependency between assertion quality and fun also highlights the need to improve the player experience. The quality of assertions depends heavily on how easily understood the passage is, and players in both prototype tests expressed that some passages were difficult to understand. While an effort was made to select passages with minimal jargon, challenging terminology inevitability exists in scientific paper abstracts. To mitigate this issue, multiple players suggested an additional user interface element to provide dictionary definitions of highlighted words in a passage. This feedback also indicates that improvements to the passage-screening algorithm are a worthwhile approach to increasing the game's entertainment value.

Players also requested several other features including (1) relationship negation, (2) more relationship choices, and (3) a full-featured scoring system.

The capability to express negation was requested several times (e.g., **`plant`** *`is not`* **`animal)`** during the BioP-C v0.2 test. In the case of two unrelated concepts, the expected BioP-C v0.2 behavior was that players would choose to ignore bad guesses entirely, not producing an assertion. Player feedback clearly indicates that this is not satisfying behavior, prompting the inclusion of negative relationships in BioP-C v0.3. ConceptNet is already capable of expressing negation, so this feature does not prevent future integration with BioP-C data.

Multiple players in both tests requested more relationship choices. At present, adding more detailed relationships would harm the generality of connections in the network, which promotes a wide breadth of analogical matches. However, more precise relationships are useful for analogy filtering. Future work will investigate approaches to dynamically present more detailed relationship choices to players based on past assertions about a given concept or group of concepts. Such an approach will require baseline concept information to be effective, but should improve specificity. Further, once baseline graph connectivity is established using general relationships, detailed natural language relationships can be gathered to increase the potential power of analogical matching. In terms of structure mapping using shared relationships, a positive match on multiple player-specified relationships would be a stronger indicator of analogy than the generic relationships used in the current version of BioP-C. Just as OMCS uses a variety of tasks to gather different types of information into the same network, it may be worthwhile to develop an additional BioP-C game for capturing detailed relationship information.

While a more feature-filled scoring system was less frequently requested, observations during BioP-C v0.2 testing support scoring improvements to promote behaviors that produce better assertions. A good scoring system forms the core of player incentives by promoting high quality information as well as player satisfaction.

In the BioP-C v0.2 test, players' scores simply started at ten and decremented once for every guess. This system incentivized high quality hints and guesses, but promoted slow games with periods of inactivity. An effective technique used in prior GWAPs (e.g., [104, 106]) requiring low cognitive effort is to base player scores on the speed of responses. Correctness in these cases is determined based on time taken to get the correct solution, taking advantage of the correlation between correctness and speed. Because game speed is correlated with a low number of actions, it is reasonable to formulate a scoring metric that is strongly based on the number of hints formed in a session. These factors influenced the development of the scoring system in BioP-C v0.3. As additional session data predictors of correctness are discovered, the scoring system can be altered to promote these other behaviors as well.

Example

This section uses data from the unpublished BioP-C v0.1 prototype to present an example of how the BioP-C network can facilitate conceptual design. This early version used different game mechanisms, and lacked two-player validation, but the data is useful for demonstrating the envisioned usage of BioP-C for supporting design inspiration tools. The prototype used a relationship taxonomy closely based on OMCS [40], and the relationships between concepts have been modified for this example to reflect the current relationship taxonomy. This example is demonstrative only; a design tool has not yet been developed.

BioP-C v0.1 presented players with three different passages describing xylem: a tube-like structure in plants used to convey water upward from the roots. All three passages were sourced from DANE, like the one shown in the excerpt below. The bolded words are those that users selected to form some of the assertions in the following example.

> This model describes the process of water **transport** from the plant roots to the leaves via the **xylem**. Transportation occurs as a result of tension in the upper **xylem** created from osmosis moving water into the leaves. The movement of one water molecule affects those below it as a result of cohesion, creating tension. Tension is transmitted through the length of the **xylem** through cohesion forces between water molecules, which are stacked in the very thin tubes of the **xylem**. To a lesser extent root pressure and adhesion between water molecules and xylem **cell** walls also plays a role, not covered here. [123]

Figure 17 shows a subgraph of the nodes surrounding “**xylem**,” generated from a small subset of these responses, with edges filtered to remove relationships asserted less than twice. In addition to this initial filtering, most of the irrelevant nodes have been pruned for simplicity of presentation.

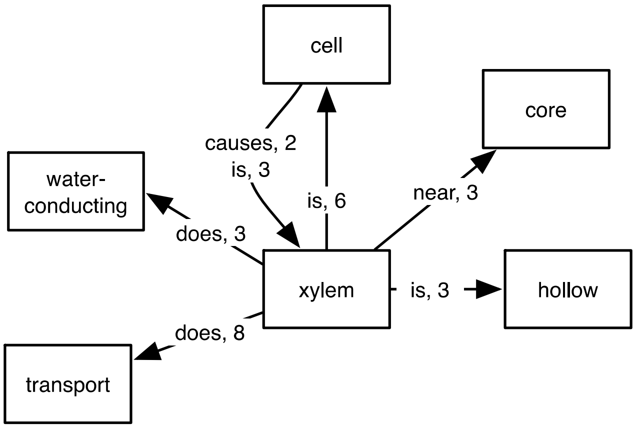


Figure 17. Subgraph of data collected from prototype test

The data structure is a directional multigraph (edges have directionality, and multiple edges can exist between the same two nodes), with each edge possessing a ‘relationship type’ and ‘weight’ attribute. These attributes track how frequently players have indicated a relationship between two nodes. For example, if three players have asserted “**xylem** *is* **hollow**,” then the nodes **xylem** and **hollow** are connected with an edge possessing a relationship type of *is* and a weight of *3*.

As a demonstration of how a designer might use this information, consider a conceptual redesign of an oil pump. The designer describes the design problem or existing system using assertions of the same style as those generated by the game. Because the designer makes these assertions, all edge weights are set to the maximum to reflect the highest level of validity. If the designer asserts that a pump is hollow, is made of metal, has an impeller, and transports oil, then these assertions would be represented as shown in Figure 18. After merging these assertions with the existing knowledge network, the (pruned) search space looks like Figure 19. The edge weighting and graph merge algorithms offered in this example are quite simple, but a design tool need not be limited to those presented here. For instance, the designer could specify confidence values on edges, and natural language processing techniques such as stemming and lemmatization could control graph merge aggressiveness.

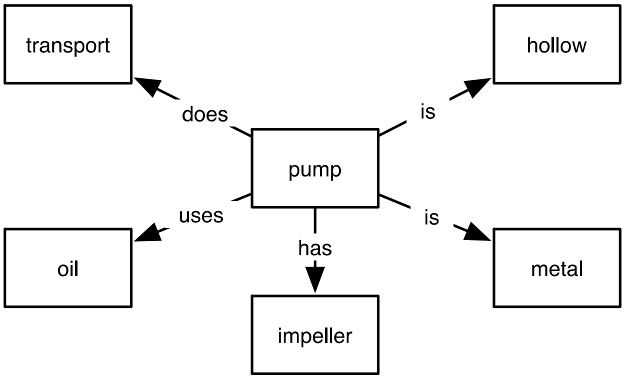


Figure 18. Example of a simple redesign encoding

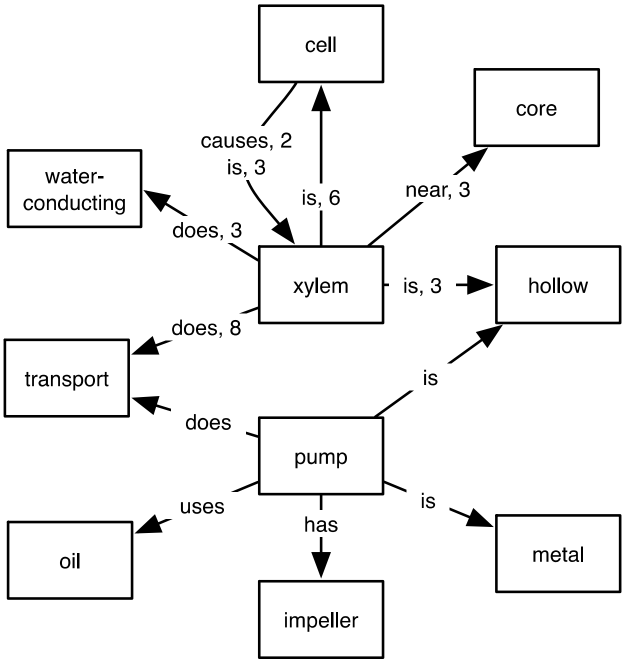


Figure 19. Search space connecting “xylem” to “pump”

A spreading activation search using normalized edge weights, with designer-entered assertions weighted to 1 and an arbitrary decay rate of 0.9, finds **xylem** as a candidate match due to the multiple strong pathways between **pump** and **xylem** **(**Figure 20). Secondary matching reveals the shared features “*does* **transport**” and “*is* **hollow**.” Back edge matching on the *“does”* relationship is especially meaningful in this context because it implies a function shared by two embodiments. Based on these results, an inspiration tool could suggest xylem as a potential analogy, and also provide references to the source documents that generated the assertions containing **xylem**. From this point the designer investigates whether the mechanism embodied by xylem (capillary action) is appropriate for conveying oil. This algorithm reflects the “many are called but few are chosen” (MAC/FAC) model of similarity-based retrieval [124]. The spreading activation stage of the search finds potential matches, and secondary matching filters these candidates.

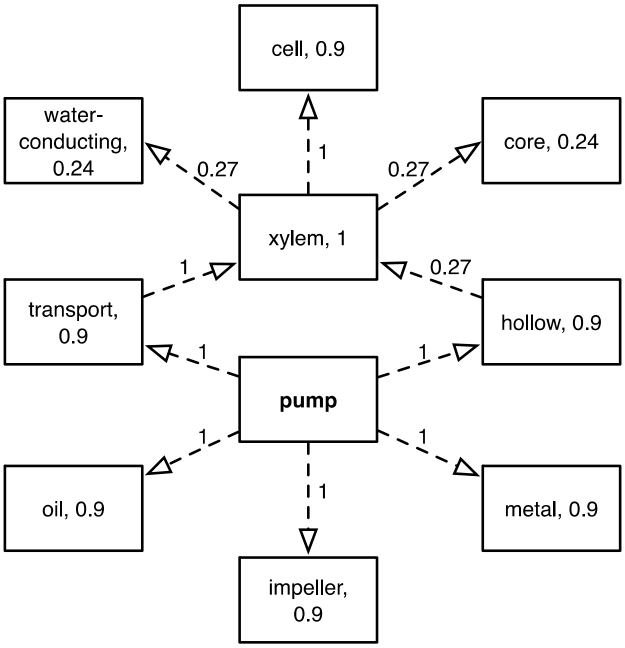


Figure 20. Spreading activation from "pump" finds "xylem"

Deeper relationship matching is less straightforward. While the game presented in this paper emphasizes graph connectivity and abstraction across domains to enable large numbers of matches, natural language relationships would further increase match quality. An obvious solution to this problem is to develop a second complimentary game to elicit natural language relationships between domain concepts (as in OMCS). Pragmatically, it may also be valuable to match distributions of abstract relationship types between nodes as evidence for first-order relationship matching.

Additionally, the graph can trivially be represented as an adjacency matrix to support "bag of words" style approaches (such as LSI) to inferring similarity. A third dimension captures relationship typing, where each two-dimensional slice of the matrix represents connections of a single relationship type. Summing on the third dimension returns a standard two-dimensional adjacency matrix containing untyped edge weights between nodes. While LSI uses weighting algorithms that rely on term uniqueness to determine word weights, BioP-C can provide word weights based on human judgments. This property could be used to supplement an LSI-based design inspiration tool by providing alternative term weightings.

Conclusions and Future Work

This paper presents and assesses a GWAP for collecting computable knowledge about biological phenomena for the purpose of aiding biologically inspired design. Specifically, this work assesses the external validity of individual assertions. Humans assess these assertions for correctness, and these ratings are used to identify potential directly measureable indicators of high correctness assertions. Additionally, this paper identifies factors affecting the game's entertainment value, and potential design features to address shortcomings in this area. The results of this study suggest that a GWAP approach has strong potential to collect valid biology knowledge into a semantic network format that can support biologically inspired design tools.

Notably, the correctness of unfiltered BioP-C assertions was rated as significantly better than random and significantly worse than the theoretical maximum, indicating that some of the information produced by BioP-C is correct. Additionally, a statistically significant negative correlation was found between statement correctness and the number of hints created in a game session, which supports a simple and effective filtering operation. Raters' agreement with whether BioP-C assertions are true tends to fall between "neither agree nor disagree" and "agree," indicating that many of the assertions are ambiguous. This highlights a limitation of the work, and suggests that future work is needed to (1) identify additional behaviors that indicate assertion correctness and (2) refine the game design to encourage these desirable behaviors.

The game presented in this paper uses a limited set of general relationships, but better classification will be possible as confidence in general relationship types grows. For example, more detailed relationships can be defined as subclasses of the high level relationships based on existing taxonomies of biology and engineering knowledge. Any number of strategies could support this change. These strategies might include additional mechanisms within the current game design, such as dynamic limitation of available game relationships based on BioP-C's previously collected data. Alternatively, separate game environments could support filtering existing assertions and gathering player-specified relationships.

The validation of this work has revealed the delicate design tradeoff between entertainment value and information quality. Complicated tasks produce better data, but there exists a complexity threshold past which players will not enjoy the game. There may exist a Pareto frontier representing the non-dominated set of tradeoffs indicating the limits of what can be learned from a human algorithm in this context, but it is unlikely that BioP-C v0.3 has reached this point. In order to understand the potential of this approach, future work in this area should aim to quantify this tradeoff, establish where these limits exist, and supply heuristics relating game design to information requirements.

Using Molecular Fingerprinting to Infer Functional Similarity in Engineered Systems

Ryan M. Arlitt, Charles Manion, Robert B. Stone, Matthew Campbell, and Irem Tumer

Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2015
August 2-5, 2015, Boston, USA
IDETC2015-46888

Abstract

Design of new and advanced materials with shape-shifting or origami-like capabilities is an area that bears a strong similarity to the design of electromechanical products yet has not leveraged such systematic approaches. In this paper, computational methods to design Metal Organic Responsive Frameworks (MORFs) – which are a theoretical type of material that can change their shape and porosity in response to light – are investigated. However, it is a significant challenge to computationally identify MORFs that are both feasible and useful, i.e., systemic invention (as opposed to discovery) of new MORFs. The proposed framework utilizes the typical product design process to iteratively generate new candidates, evaluate their properties, and then guide the generation of the next set of candidates. A materials designer could then leverage this knowledge to generate structures or substructures with specific functional goals in mind. In this paper an approach to inferring functional similarity of systems using structural information – based on both drug design and database-driven product design – is evaluated. The results demonstrate an observable correlation between structural fingerprints of electromechanical products and electromechanical function. This evidence, combined with the well-established similar property principle in drug design, supports the usage of molecular fingerprinting for providing high-level functional guidance in a MORF design framework based on purely structural information.

Introduction

The discovery of new materials can have a transformative impact in a wide range of applications, but trial and error plays a major role in this process. This paper works toward realizing a systemic design-based approach that combines computational exploration of a materials design space with human intuition in order to generate dynamic materials that are both feasible and useful. More specifically, this work focuses on facilitating the computational creation of a new class of photoresponsive materials called Metal Organic Responsive Frameworks (MORFs). One potential application for these materials is hydrogen storage. Porosity of a MORF could be dynamically increased to import hydrogen and decreased to supply hydrogen. However, the actual design and creation of a variety of MORFs is a relatively unexplored problem; it is not well understood which types of structures will be feasible or useful. It is this task of solution space exploration for which principled design techniques can be of use.

MORFs represent a theoretical class of materials that can change their shape and porosity in response to light. MORFs combine Metal Organic Frameworks (MOFs) with photo-isomerizing molecules to produce materials that behave as stochastic linkages. While there are thousands of known MOF structures and many photoisomerizing units that can be incorporated into them, the MORF design space is governed by complex constraints. The linker components that connect the framework together must remain connected as they fold. Further, the order in which linkers fold cannot be precisely controlled, so the framework must be sufficiently compliant. These constraints are applied at a small scale under forces that are not readily apparent, which increases the complexity of behaviors that MORFs may exhibit. As a consequence, it is especially challenging to design new MORFs that are both feasible and useful. Many of these challenges exist in other molecule design domains as well, such as in drug design.

The goal of this research is to produce a computational design framework that leads to the invention of useful MORFs. In general, this process begins with a database of nodes, linkers, and combination rules. Nodes and linkers are the basic building blocks of MOFs, and the combination rules describe chemically valid ways to combine atoms. Candidate MORFs are rapidly generated using these rules and building blocks, and then screened for feasibility. The goals during initial feasibility screening are to use computationally inexpensive techniques to predict whether the candidate MORF’s properties are close to the design target, and whether the photoisomers in that MORF will be able to fold. For example, one of these preliminary screens might address whether the change in linker stiffness is acceptably close to what is required. The results from this process serve to update a predictive model relating MORF structures to their behaviors. In addition to this closed loop computational exploration of the solution space, MORF candidates are also screened for functional usefulness. This is a challenging problem because usefulness arises from a combination of dynamic behavior and human needs. Assessing usefulness requires expert judgment, but it is infeasible for a human judge to examine every candidate. To assist with this process, candidate MORFs with similar behavior are clustered, and then domain experts assess exemplar candidates from each cluster to determine whether they are likely to meet a need.

MORF design is distinct from most molecular design problems because its functionality comes from dynamic motion rather than the static positioning of atoms.

This shared goal of dynamic functionality also connects the problem of MORF design with the problem of electromechanical design. Much as in electromechanical design, the goal is to produce a structure that serves a particular function, but it is much more challenging to precisely capture a molecule's usefulness than to capture its behavior or structure. In spite of MORFs being an unexplored area, this connection allows existing electromechanical system data to serve as a platform for making judgments about the efficacy of various computational techniques in the MORF domain.

This paper contributes to the creation of a MORF design framework by demonstrating that an explicit representation of function is not needed. Instead, a structural representation from drug design can be used, and function can be inferred from this representation. This structural representation has two important properties. First, it is relatively efficient to calculate, which makes it conducive to efficient solution space exploration. Second, it is significantly correlated with dynamic function.

Background

The following subsections discuss related work in both engineering design and drug design. These background sections serve not only to present the related literature from these fields, but also to relate the important concepts back to engineering design and MORF design. The key points from this section are as follows.

Drug design techniques are closely related to computational design synthesis techniques; both provide a computational framework for exploring a solution space. Within this type of framework, Quantitative Structure-Activity Relationship modeling (QSAR modeling) and structural similarity comparison provide mechanisms for guiding the solution search and evaluating solution candidates. Fragment-based approaches, which capture and apply important substructures of solutions, provide a method to reduce the size and complexity of the MORF search space. The concept of similarity between candidate solutions plays an important role in both computational design synthesis and drug design, and so this section discusses vector representations of systems that enable efficient similarity screening. Of the available representations in drug design, the structural fingerprint representation is selected for evaluation in the context of dynamic systems based on its past successes in predicting the (static) functionality of drug molecules. This section highlights the compatibility of structural representations in the electromechanical and molecular domains, and argues that this compatibility enables electromechanical product information to be used as a test bed for demonstrating the value of structural fingerprints in the MORF domain.

Summary of Challenges

There are several challenges involved in realizing a MORF generation framework. First, it is impossible to enumerate the full search space of candidate MORFs. Second, it is challenging to characterize MORF structures' functionality. It is easier (though still challenging) to characterize MORF structures by their behavior, and easier still to characterize structures using the structural information directly. Third, there is no efficient means to evaluate candidate MORFs via existing models (such as the Quantitative Structure-Activity Relationship models discussed later in this section) or similarity to benchmark MORFs.

To address these challenges, a representation is needed that (1) enables efficient substructure search and similarity calculation, (2) is predictive of system functionality, and (3) is in a form that facilitates QSAR model development. Structural fingerprints address challenges (1) and (3), but it must be demonstrated that this representation is predictive of system functionality. More details about these points are presented in the remaining background sections.

Computational Design Synthesis

In the domain of computational design synthesis, generative grammars are an approach to capturing design knowledge in a manner that can be reapplied to computationally generate new designs. Such rules have been successfully applied to a number of engineering design areas including electromechanical products [125] and sheet metal parts [126]. A major advantage of grammar rules is that they enable rapid generation of many solution candidates, altering design from a problem of generating concepts to a problem of evaluating concepts. Generative grammars are adopted in the MORF design framework as a means to explore the extremely large chemical space. These grammars consist of a fixed set of rules to grow a wide variety of graphs from a starting seed. This approach takes advantage of the fact that both engineered systems and molecules can be represented as undirected labeled graphs. In these graphs, nodes represent components or atoms, while edges represent physical connections between components or bonds between atoms.

A closely related work in the design domain uses grammar rules to generate a large space of candidate design topologies based on a functional black box [127]. Human judgment is required to make the final evaluation on the quality of these design topologies, but the number of solutions is too large for a human to evaluate every topology manually. To address this issue, k-means clustering is used to group similar topologies [128]. These groupings could then be used to facilitate human-in-the-loop exploration and evaluation of the solution space.

In order to support this clustering, distance values between each pair of Design Structure Matrices (DSMs) [129] in the referenced study are calculated by taking the matrix difference and determining the Euclidian norm of the result. These DSMs have exactly one row and column for each type of component (where the types of components are specified by a component taxonomy). Aggregating components in this way ensures dimensional consistency between all DSMs and enables valid similarity comparisons, but loses structural information when a system contains more than one of any component type. This loss of information makes it impossible to distinguish between systems with identical types of components but different topologies. For example, different types of gearboxes are made of similar types of components, but the ways in which their gears are connected has a major impact on the types of motion and applications for which each gearbox is suitable. In applications where it is important to differentiate between types of dynamic behavior (such as electromechanical product design or MORF design), it is important to capture this type of topology information.

The Similar Property Principle

In drug design, the Similar Property Principle [130] says that drug molecules with similar structures tend to have similar properties and biological activity. This generally valid assumption supports approaches that infer functional clusters from much more readily available structural information. The principle holds true in drug design for large ranges of molecules spanning a wide variety of structures [131-133], in spite of the presence of “activity cliffs” [134] where small structural changes result in large changes in activity. As a result, it can be said that structure-based approaches to drug design are probabilistic in nature. While the shared properties of any two structurally similar molecules are uncertain, molecules with high structural similarity are more likely to share properties [131].

Function based engineering design relies on a similar assumption: that similar components generally do similar things (e.g., electric motors usually rotate things, screws usually couple things together). As a result, automated design of electromechanical products can also be described as probabilistic. A key difference is that a drug’s functionality (biological activity) is a direct result of its atoms’ static positioning, while functionality in electromechanical design typically includes dynamic motion. MORF design shares characteristics with both domains. While MORF design takes place at the same scale as drug design, the desired outcome of dynamic functionality is more similar to function based engineering design. Because of these similarities, it is valuable to consider drug design techniques for creating new nanoscale dynamic systems.

Quantitative Structure-Activity Relationship (QSAR)

A Quantitative Structure-Activity Relationship model (QSAR model) is often central to the discovery of new drugs. This type of regression model aims to relate molecular structure to biological activity – a major measure of performance in drug design. While biological activity is a common target, desirable molecular properties are also commonly mapped to structure (Quantitative Structure-*Property* Relationships), and similar techniques could be used to map structure to groups of dynamic MORF behaviors. Typically QSAR models are created using empirically validated structure-activity correlations or (less commonly) simulation results. Partial least squares regression is a common method to create these relationships [135]. In contrast, some QSAR methods correlate substructure fragments with biological activity rather than considering each atom individually (e.g., Fragment-Based QSAR [136] and Hologram QSAR [137]). Such approaches serve to narrow the search space when performant fragments are known, all without leaving the computational design loop. The work presented in this paper aims to support the creation of both traditional and fragment-based QSAR methods for MORF design.

While QSAR is useful for correlating structure to directly measurable properties, the goal of this project is to relate MORF structures to behavior and function categories. And while a drug's biological function is easily quantifiable, quantifying behaviors in the unexplored MORF domain is more challenging. This motivates the need to demonstrate a correlation between structure and behavior – in domains where functions involve dynamic motion – using structural representations that have been effective for drug screening. Evidence for this correlation in the electromechanical consumer product domain (as shown in the Results section) supports development of QSAR-style models to predict dynamic behavior or function classifications based purely on structure. These results apply to both the electromechanical product and MORF domains.

Fragment-Based Approaches

Atoms are the basic building blocks of molecules, so it is reasonable to use atoms themselves as the building blocks in a drug design database. Unfortunately, computationally generating new molecules from atoms quickly results in combinatorial explosion [138]. One technique for managing this search space size is to capture fragments (i.e., molecular substructures) rather than atoms, similar to how a product designer might select an electric motor rather than an armature and stator. This reduces the size of the chemical search space from roughly 10^{70} molecules to 10^{16} molecules [138]. Frequently used fragment types in fragment-based drug design include rings, functional groups, or particularly unique or interesting patterns within a domain [139].

For MORF design, one key task will be capturing fragments that are unique to distinct functional groups. Such fragments could then be used to seed the generation of new structures with similar properties, which has the effect of simplifying the search space. The results section will show an example of fragment mining in the consumer product domain using a group of vacuum cleaners. This example is equally applicable to molecule design because systems in both domains can be represented as undirected graphs.

Fingerprints

The challenge of performing drug design *in silico* requires a representation that contains a large amount of information in a small space. Linear bit vectors describing 2D properties of molecules are not only very fast at substructure screening [139], but have been shown to be superior to 3D descriptors for distinguishing compounds' biological activity [133]. Structural keys [138] are an early type of bit vector representation that capture structural features (such as important fragments) by assigning a bit index to each important feature. Unfortunately, structural keys containing anything more than 1D descriptors are time consuming to use. Not only must a new structural key mapping be designed for each new application, but generating a bit vector for each molecule is inefficient.

Hashed fingerprints address these issues while preserving the bit vector representation that enables efficient screening. Much like structural keys, hashed fingerprints represent structural information (node labels and connectivity) in a form that promotes fast similarity screening. A major advantage of structural fingerprints is that features do not need to be prescribed. Instead, each molecule's structural information is hashed into a fixed length bit vector using the same algorithm and hashing functions for every molecule. Additionally, because every structure generates a fixed length bit vector, this representation is conducive to matrix methods for developing predictive QSAR-style models (e.g., partial least squares). In virtual screening applications, fingerprints have been found to be equal or superior to using chemical graphs directly [140].

A commonly used style of fingerprint is the Daylight fingerprint [139]. This type of fingerprint relies on a fixed vocabulary of node labels. In drug design and MORF design, these are atoms. In engineering design, a component taxonomy (e.g., [141]) provides a similar set of node labels. It is worth noting that more recent fingerprinting methods use the property information of nodes (atom properties) rather than their labels (atom names), which increases the chances of finding different structures with similar behavior (e.g., [142]). While advantageous, this type of information is more challenging to abstract to the domain of electromechanical components. Such abstraction is left to future work.

Drug design relies heavily on the assumption that similar structures will have similar biological activity. A key difference between drug molecules and MORFs is the type of functionality. A drug's behavior is based on the types of sites that it binds to in the human body, which is determined by the static structure of atoms in the molecule. In contrast, a MORF's behavior is determined by the type and configuration of nodes and linkers, which determine how the MORF will move. This paper addresses whether this structural fingerprinting representation, which has been empirically validated over years of use for predicting biological activity, is also correlated with dynamic function. Such evidence is needed to support the use of structural fingerprinting for both automated MORF design and automated engineered product design.

Chemical Similarity

While fingerprints were originally designed based on the Bloom filter [143] as a way to support screening operations, they are commonly also used to calculate pairwise similarity between molecules for clustering. The most commonly preferred similarity measure in drug design is the Jaccard coefficient (Eq. 3). The Jaccard coefficient divides the total quantity of vector indices where both vectors have *true* bits by the number of vector indexes where at least one bit is *true.* Indices where neither vector has a *true* bit are ignored. This measure has proven its effectiveness in a wide range of drug design applications [144], and it is for this reason that the Jaccard coefficient is used in this study.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad \text{Eq. 3}$$

One common usage for these similarity measures is to perform chemical clustering in order to manage the complexity of a search. For example, a representative subset of candidates from each cluster may be selected for further simulation and study [145]. In the MORF generation framework, such clustering will support both feasibility and usefulness screening by creating a reduced list of representative molecules for intensive molecular simulations.

While no clustering algorithm is clearly superior in all cases [146], Ward's method [147] and the Jarvis-Patrick method [148] represent two successful and commonly used chemical clustering algorithms [146]. The electromechanical products in this study are clustered according to Ward's method. Comparing the efficacy of different clustering algorithms is not considered in this paper.

Design Repository

Much as in drug design, externally valid function and structure data is the best way to evaluate structure-function correlations. The Design Repository [74] provides a convenient dataset to evaluate this correlation between structural fingerprints of engineered products and those products' dynamic functionality. The Design Repository contains 184 products that have been reverse engineered both structurally and functionally. Most of these products are in the electromechanical domain, and for this study all products outside of this domain (e.g., biological systems) are omitted. This study makes use of DSMs for each of these products. Components in each DSM are labeled according to their component taxonomy types [141], and components are not aggregated in order to preserve all connectivity information.

Methodology

This section demonstrates that an explicit representation of function is not needed to support electromechanical product generation, but that distinctly different dynamic function groups can be inferred from a structural representation that facilitates efficient solution space search. Because MORFs possess dynamic functions (as do electromechanical products), and because the MORF domain is similar to the drug design domain where fingerprinting approaches have already been successful in supporting molecular search, these results also demonstrate that the approach is valid in the MORF domain. The methodology begins with creating structural fingerprints for every product in the Design Repository. Pairwise similarity between every two products is then calculated, and the significance of several predefined functional groups of products is evaluated.

Fingerprinting and Similarity

Each product DSM is converted into a Daylight-style fingerprint: a uniform length (2^{16}) bit vector where small patterns of bits correspond to the presence of substructures. Given a set of path lengths, the algorithm extracts all paths of each length. Figure 21 shows an example of this process in which path lengths of one, two, and three are extracted from a small starting structure.

The node labels A, B, and C could represent atoms in a molecule or components in an engineered product. For example, if this graph represented a system with three interlocking pieces of housing and no other components, these labels would be

Housing, Housing, and Housing. The actual system graphs analyzed from the Design Repository are much larger and more sparsely connected than the one in this example, with an average size of approximately 18 nodes.

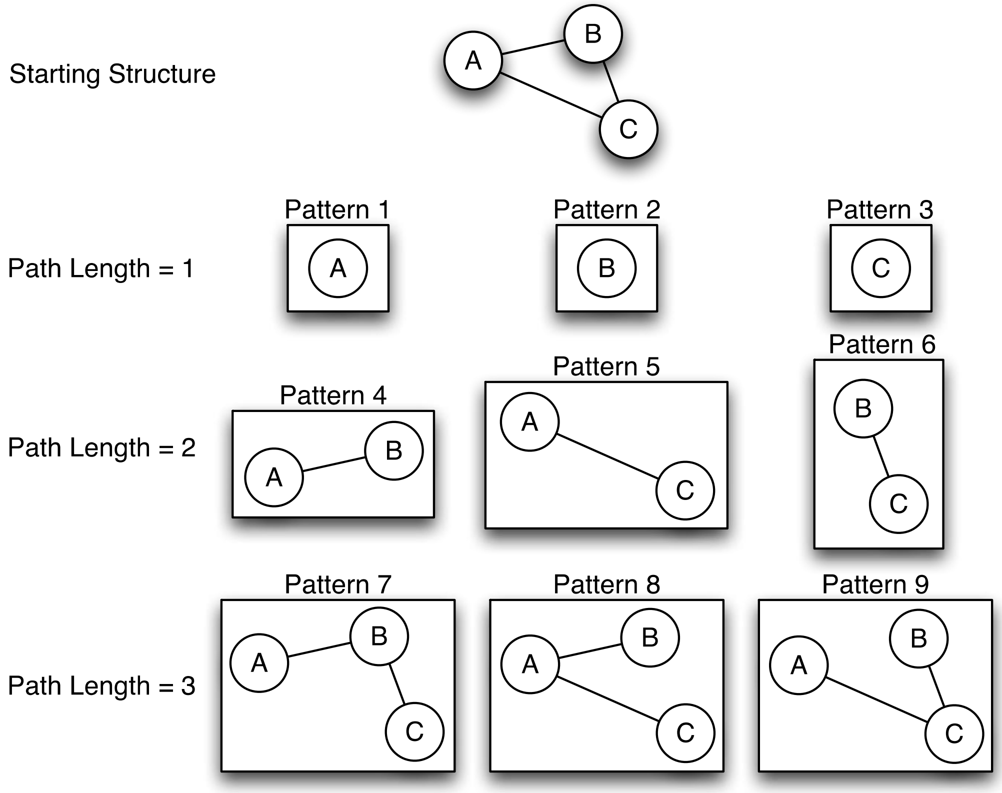


Figure 21. Extracting All Patterns of Length 1, 2, and 3 from a Simple Starting Structure

Path lengths of one and four are used in this study; adding additional path lengths creates more patterns, which leads to prohibitively high-density fingerprints for very large systems. Two of these large systems were removed prior to analysis, resulting in a maximum bit density of 0.28 and an average bit density of 0.03. This density can be improved for large systems by using larger fingerprints (e.g., 2^{32}).

Each component (graph node) in each of these paths is given a canonical label using a version of Morgan's algorithm [149]. This canonical labeling allows the detection of graph isomorphisms across subgraphs, regardless of orientation, by assigning the same canonical labels to all isomorphic graphs. This prevents Pattern 8 in Figure 21 from being extracted as ABC in some systems and CBA in other systems. Morgan's algorithm recursively calculates graph invariants for each node until all nodes have a different score or a convergence failure is detected. If the algorithm does not completely converge (i.e., there are ties), then a tiebreaker is needed. In the case of these electromechanical product DSMs, the tiebreak order is determined by alphabetical order of all possible component labels in the Design Repository's component taxonomy. In molecular fingerprinting, the atomic number of nodes and the bond order of edges can both be used. A DSM is created for each of these paths, and the nodes are sorted in the newly created canonical order.

Next, each path is hashed to a pattern of four bits in the fingerprint vector using a cyclic redundancy check (CRC) [150], a common fingerprinting hash function. The input to the hash is based on the node labels and their connectivity. CRC is advantageous because it can hash variable length patterns into fixed length outputs. The four CRC generator polynomials used to perform the hashing are 0x8d95, 0x8fdb, 0x968b, and 0x9eb2 based on Koopman's search for the optimal general purpose CRC polynomials in error detection applications [151]. The specific hashing functions are not as important as the concept that every pattern in every system (molecule or engineered product) is hashed in the exact same way. Any given substructure path will always correspond to the same set of bits in a fingerprint, which means that systems with large amounts of shared substructure will have similar fingerprints. In this study, each unique path in each DSM is hashed to a unique pattern of bits in its fingerprint.

For example, Figure 22 shows the patterns 1 and 8 from Figure 21 being hashed into a short 2^4 bit fingerprint using three arbitrary hash functions. The result of the three hashes is a uniquely identifying set of bits in the fingerprint. The combination of bits 1, 7, and 8 uniquely identify Pattern 1, while the bits 5, 7, and 12 uniquely identify Pattern 8. The 7th bit contains a collision, an incidental overlap of two patterns. Collisions introduce an acceptable level of error in exchange for the capability to efficiently search and match substructures. Optimizing the number of hash functions and the length of the fingerprint can help to mitigate the negative effects of collisions.

The resulting fingerprints can be used directly to support screening operations (i.e., efficient library search for desired subgraphs), or pairwise distances between molecules can be calculated using the Jaccard coefficient. This distance measure gives the structural similarity between systems (electromechanical products or molecules), and can serve as the basis for a clustering algorithm [152].

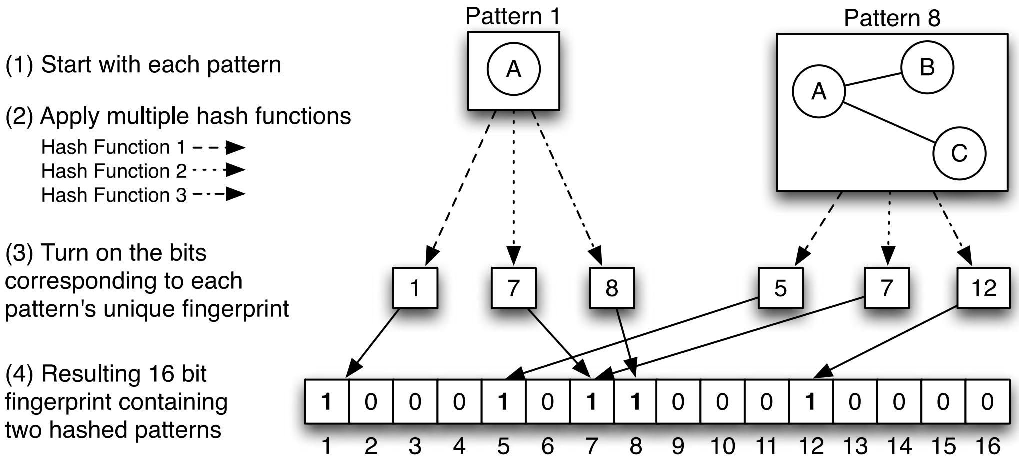

Pattern 1
Pattern 8
(1) Start with each pattern
(2) Apply multiple hash functions
Hash Function 1
Hash Function 2
Hash Function 3
(3) Turn on the bits corresponding to each pattern's unique fingerprint
(4) Resulting 16 bit fingerprint containing two hashed patterns
A
A
B
C
1
7
8
5
7
12
1 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Figure 22. Hashing Each Pattern into a Fixed Length Fingerprint

Evaluation

In order to evaluate the validity of Jaccard distance measures constructed from structural fingerprints of electromechanical products, the Mann-Whitney Rank Sum test is used. This test answers the question of whether the internal similarity for a given cluster of functionally similar products is significantly higher than the remaining population's similarity to that set of products. In other words, for each set of functionally similar products, does the distance measure rate them as significantly more similar to each other than to the rest of the products in the repository? More generally, this question addresses whether the distance measure is valid for grouping systems with similar dynamic functionality.

To answer this question, the following steps are taken. First, the full pairwise distance matrix between all products is calculated. Second, for each product, distance measures are converted to distance ranks. Third, for each set of functionally similar products (identified by human raters *a priori*), the distances between products in the functional group and all products in the data set are summed. This produces a single measure of the product's overall similarity to the products in the predefined cluster.

Table 15 shows this process for the functional group of beverage makers. The row labels include every product in the repository, while the column labels include only the six beverage makers. Each row of distance scores has been converted into similarity ranks. Each cell contains the rank value of a product's similarity to one of the products in the prescribed function cluster. In the event of a tie, the average rank is taken.

For example, the product "black 12 cup deluxe coffee" (labeled as product A) is most similar to itself, second most similar to "black 12 cup economy coffee," and 36th most similar to "black 4 cup regular coffee." By comparison, the "b and d drill attachment" is 120th, 106th, and 88.5th most similar to these first three coffee makers. The total rank score is the sum of these individual values, which captures each product's overall similarity to the six beverage makers. A low total rank score indicates high overall similarity to the products in the cluster. The smallest value in each row is excluded from this sum in order to control for the impact of comparing products within the cluster to themselves. For example, the similarity rank of 1 is excluded for "black 12 cup deluxe coffee," as is the rank of 88.5 for the "b and d drill attachment." This tends to increase the conservativeness of the test because it reduces the outside-of-cluster rank scores more than the within-cluster rank scores.

Table 15. Product Rank Scoring Example

	Product Name	**A**	**B**	**C**	**D**	**E**	**F**	**Total Rank Score**
A	black 12 cup deluxe coffee	1	2	36	12	8	5	63
B	black 12 cup economy coffee	3	1	21	17	19	2	62
C	black 4 cup regular coffee	14	18	1	17	2	27	78
D	mr coffee iced tea maker	21	37	41.5	1	13	35	147.5
E	white 12 cup regular	8	34	6	9	1	26	83
F	white 4 cup economy coffee	3	2	28	14	11	1	58
-	b and d drill attachment	120	106	88.5	129	111	109	663.5
-	b and d dustbuster	106	75.5	75.5	81	72	79	489
-	b and d jigsaw	29.5	69	76	18	12	45	249.5
	…	…	…	…	…	…	…	…

Lastly, the Mann-Whitney Rank Sum test is performed to compare the total rank scores of products within the cluster to the total rank scores of products outside of the cluster. This test is used instead of a t-test because these rank scores are not normally distributed. This approach evaluates the quality of the distance measure independently of any specific clustering algorithm.

Results and Discussion

After removing all products without component DSM data, 143 products remain. The function groups and products in Table 16 were selected prior to any analysis. Each function group is characterized by a high level shared function, with the exception of two product families of power tools. The rationale for including these product families is that they generally share subfunctions such as modular docking capability or rotating a tool head.

The significance of these product groupings is assessed using both the structural fingerprinting representation discussed previously and a functional similarity metric that uses latent semantic indexing to emphasize uniquely identified functions in a product [89]. This benchmark metric does not account for functional model topology; only the types and quantities of functions are observed.

As shown in Table 17, the structural fingerprints perform as well as or better than the function similarity metric for grouping systems with similar functions. In structural fingerprinting, every grouping is significantly differentiated from the rest of the product population ($p < 0.05$). The functional similarity metric detects significant groupings for only the beverage makers, saws, and Black and Decker products.

Table 16. Human-Specified Function Groups

Group Name	Group Members
Beverage Makers	'black 12 cup deluxe coffee' 'black 12 cup economy coffee' 'black 4 cup regular coffee' 'mr coffee iced tea maker' 'white 12 cup regular' 'white 4 cup economy coffee'
Drills	'b and d drill attachment' 'delta drill' 'firestorm drill' 'mac cordless dril-driver' 'skil drill'
Toothbrushes	'colgate motion toothbrush' 'crest toothbrush' 'oral b toothbrush'
Saws	'b and d circular saw attachment' 'b and d jigsaw' 'b and d jigsaw attachment' 'delta circular saw' 'delta jigsaw' 'firestorm circular saw' 'firestorm saber saw' 'skil circular saw' 'skil jigsaw' 'versapak circular saw'
Sanders	'b and d palm sander' 'b and d sander attachment' 'delta sander' 'dewalt sander' 'random orbital sander' 'versapak sander'
Black and Decker Products	'b and d can opener' 'b and d circular saw attachment' 'b and d drill attachment' 'b and d dustbuster' 'b and d jigsaw' 'b and d jigsaw attachment' 'b and d mini router attachment' 'b and d palm sander' 'b and d power pack' 'b and d rice cooker' 'b and d sander attachment' 'b and d screwdriver' 'b and d sliceright'
Firestorm Products	'firestorm battery' 'firestorm circular saw' 'firestorm drill' 'firestorm flashlight' 'firestorm saber saw' 'firestorm screwdriver'
Vacuum Cleaners	'bissell hand vac' 'blowervac' 'bugvac' 'dirt devil vacuum' 'neato robotics vacuum cleaner' 'shopvac'

Each significant result indicates that a cluster's internal similarity is significantly higher than its similarity to the rest of the products in the Design Repository. This outcome validates the fingerprinting representation for correctly detecting high similarity among each set of functionally similar electromechanical products. Repeating this result for many sets of products validates fingerprinting as an accurate means of detecting functional similarity in the Design Repository. This result is extrapolated as evidence that features in structural fingerprinting are correlated to function in other sets of electromechanical products and dynamic molecules.

Table 17. Mann-Whitney Rank Sum Test Results

	Product Group	**p-values**	
		Functional Similarity	**Structural Fingerprint Similarity**
Functional Groups	Beverage Makers (6)	*1.5009e-04*	*3.9035e-05*
	Drills (5)	0.1611	*0.0317*
	Toothbrushes (3)	0.2538	*0.0033*
	Saws (10)	*0.0319*	*7.4516e-06*
	Sanders (6)	0.3648	*0.0358*
	Vacuum Cleaners (6)	0.1162	*0.0011*
Product Families	Black and Decker Products (13)	*1.5214e-03*	*2.2914e-05*
	Firestorm Products (6)	0.0879	*0.0120*

Clustering

A clustering algorithm is needed to convert the full pairwise similarity matrix into clusters of components with high internal similarity. This paper does not compare the efficacy of different clustering algorithms, though the results of hierarchical agglomerative clustering using Ward's method with an arbitrary cutoff of 20 clusters are presented here for discussion. A sample of these clusters is given in Table 18.

Many of these clusters have a clear interpretation. For example, cluster 6 contains several vacuum cleaners (including one that was mistakenly omitted from the human-generated list of vacuums), a hair dryer, and several power tools. All of these products except for the power pack contain an electric motor assembly, and most of these assemblies drive fans or blades. The functionality of this group is split between guiding fluids and separating materials.

Cluster 14 also has a clear interpretation. All products except the bumble ball and the water pump contain motorized blade assemblies for separating materials. Upon closer inspection, the impeller inside the water pump is (erroneously) modeled as a motorized blade, accounting for its similarity to the other products in its group.

Notably, some of the large clusters are difficult to interpret. For example, cluster 4 contains power tools, kitchenware, and a few digital products. Most of these products share subassemblies that control fluid flow, but little else. Difficulty interpreting clusters is a common weakness of clustering algorithms, though it is encouraging that many clusters in this example have a clear interpretation.

In contrast to these large clusters, the interpretations of clusters 8, 10, and 15 are very clear. These clusters respectively contain two coffee makers, two electric toothbrushes, and two nail drivers that use air pressure along with (erroneously) a tube cutter. Upon closer inspection, many of the components in the tube cutter and mini air nailer were not assigned component taxonomy labels in the Design Repository, leading to small DSMs, and resulting in very sparse fingerprints with a few incidentally matched bits. This highlights an important (though unsurprising) weakness: systems with very little information are difficult to characterize and differentiate from other systems. These results suggest that this approach works best for graphs of similar sizes above a minimum size threshold.

Table 18. Example Clusters using Ward's Method

Cluster #4 air hawg toy plane alcohawk digital alcohol detector apple usb mouse b and d can opener b and d circular saw attachment b and d jigsaw attachment cotton candy machine craftsman nextec multi tool dirt devil vacuum dremel multi max first shot nerf gun health o meter digital scale skil drill skil jigsaw snowcone maker stir chef tractor sprinkler walker	Cluster #6 b and d dustbuster b and d power pack b and d sliceright bissell hand vac bugvac delta nail gun dewalt sander eyeglass cleaner firestorm circular saw firestorm drill hair trimmer supermax hair dryer versapak circular saw
Cluster #8 black 12 cup economy coffee white 4 cup economy coffee	Cluster #10 colgate motion toothbrush oral b toothbrush
Cluster #14 b and d jigsaw delta jigsaw juice extractor mini bumble ball presto salad shooter vibrating razor water pump	Cluster #15 bosch brad nailer grip right mini air nailer ridgid tube cutter

More generally, this clustering example demonstrates that in the absence of any functional information, it is possible to group systems with similar functions. However, any given clustering algorithm introduces additional ambiguity, and it must be tuned for a specific domain. In the context of a MORF design problem, this enables techniques such as similarity-based sampling and fragment mining that maximize the value of each computationally expensive simulation and facilitate the development of predictive models.

Fragment mining

After clusters of similar candidates have been identified, they can be mined for fragments to (1) populate a database of starting seeds for use in the generative phase and (2) inform the development of a QSAR-style model. After testing the vacuum cluster for validity, fragment substructure mining for largest common subgraph was performed using Subdue [153]: a graph based unsupervised learning system. Mining each cluster separately greatly reduces the computational expense of finding common subgraphs. Figure 23 shows a common substructure in the vacuum group, and such fragments may be useful seeds for the automated design of new vacuum cleaners. The presence of shared substructures in a cluster is guaranteed because fingerprint similarity detects groups with high amounts of shared structural information, but determining what those substructures are requires this additional step. The expected size of such fragments is directly proportional to the size of the total population and inversely proportional to the size of its parent cluster. Fragments that come from small clusters will be larger (and frequently less valid) than fragments that come from large clusters because small clusters are created using stricter similarity cutoffs. Increasing the population size will also increase fragment size and validity (at additional computational expense) because it increases the number of similar candidates. Finding the optimal population and cluster cutoffs will be a matter of tuning a specific algorithm for a specific context.

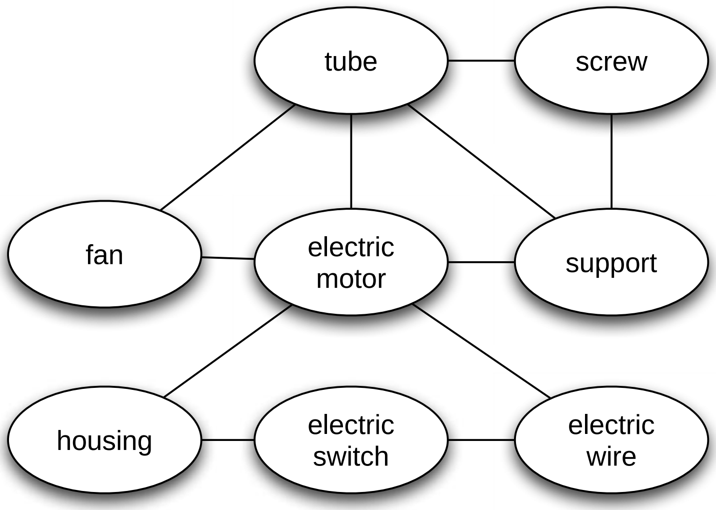


Figure 23. Vacuum Cleaner Structural Fragment

In the case of vacuum cleaners, this common substructure could be entered into a library to provide reference structural patterns for removing debris with air pressure. Such reference patterns could serve as starting seeds for new candidate generation, or be used to automatically evaluate new candidates (e.g., Fragment-Based QSAR). In the case of MORF design, an identical process can be used to extract structural fragments from clusters with interesting simulated behavioral characteristics.

Application to MORF Design

The envisioned MORF generation framework is broadly divided into two stages as shown in Figure 24. The first stage consists of undirected exploration to generate a coarse predictive model relating MORF behavior to structure. The second stage is a materials design task involving a domain expert and a set of functional requirements. It should be emphasized that this framework is a work in progress, and does not exist in working form at the time of publication.

The exploratory stage begins with generating a large number of candidate structures using graph grammar rules. Next, these candidates are screened for feasibility using computationally inexpensive metrics such as synthetic accessibility and activation energy. Third, molecules are fingerprinted and clustered as described in the electromechanical product example. Fourth, a representative subset of molecules is selected using similarity-based sampling. These molecules undergo more expensive molecular dynamics simulations. In addition to more accurate feasibility information, the outcomes of these simulations will provide simple behavioral measures such as changes in unit cell volume, overall changes in dimensions, work performed, and the ways in which the photoisomerizing forces interact with the geometric constraining forces. These behavior parameters are then correlated with structural fingerprints to produce a QSAR-style regression model. Additionally, common substructures are extracted from groups of molecules that have similar performance values across several behavior metrics. These substructures provide necessary information for constructing a fragment based QSAR-style of model. As the MORF domain matures and fragments are frequently used to serve the same behaviors and functions, this approach will facilitate the creation of a component taxonomy of MORF fragments.

From the results of the exploratory stage, a Quantitative Structure-Behavior Relationship model (QSBR model) can be constructed and updated, with the goal of relating structures and key substructural fragments to the behaviors that they produce. At this stage it is not known whether a traditional regression model based on structures in their entirety, a fragment-based model, or a combination of the two will be most effective. Armed with this QSBR model, a materials designer would be able to start with a set of specific functional requirements, transform these requirements into desired behaviors, and then computationally explore structures in the solution space that are most likely to satisfy these behaviors. The results of this exploration will serve to update the predictive models and guide the selection of candidates to synthesize.

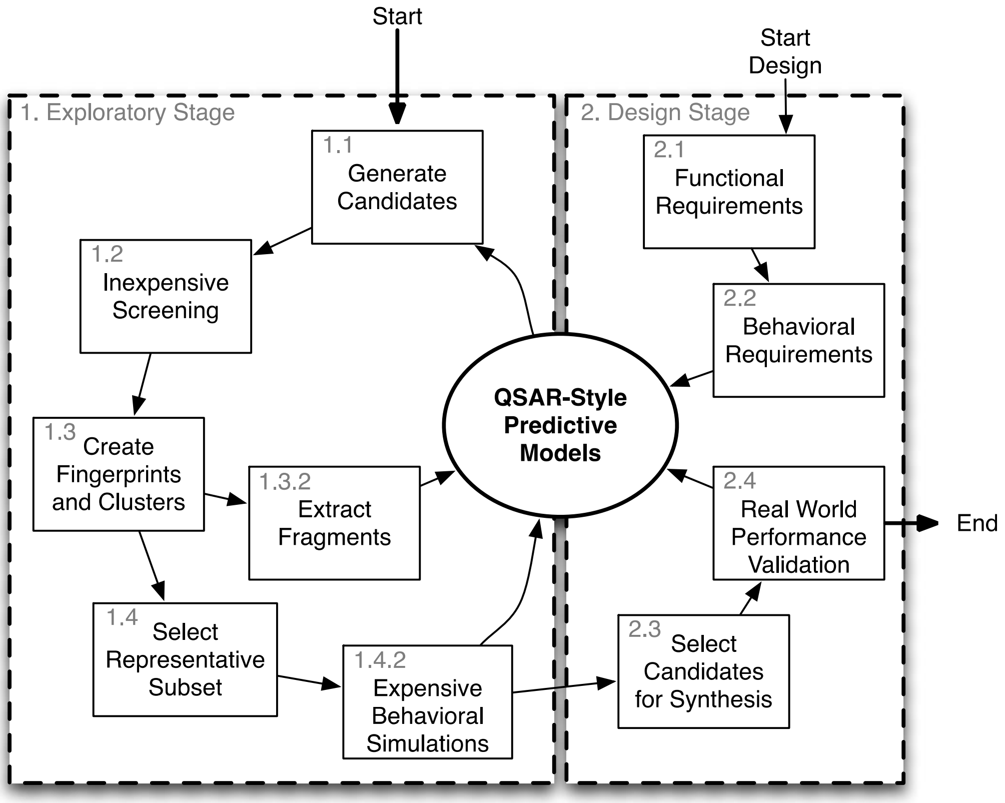

Start
Start Design
1. Exploratory Stage
2. Design Stage
1.1 Generate Candidates
1.2 Inexpensive Screening
1.3 Create Fingerprints and Clusters
1.3.2 Extract Fragments
1.4 Select Representative Subset
1.4.2 Expensive Behavioral Simulations
QSAR-Style Predictive Models
2.1 Functional Requirements
2.2 Behavioral Requirements
2.4 Real World Performance Validation
2.3 Select Candidates for Synthesis
End

Figure 24. Envisioned MORF Generation Framework

Conclusions and Future Work

This paper explored a method to facilitate efficient solution space exploration using the structural fingerprinting representation. It was demonstrated that structural fingerprints of electromechanical products are correlated with their functions. Solution candidates were clustered using these fingerprints, and representative fragments were extracted from the functionally distinct cluster of vacuum cleaners. These fragments represent structural backbones of solutions with different functionality. Together these results suggest a feasible method for exploring a solution space of dynamic functionality.

Efficient exploration of the solution space is a major challenge in both molecular design and engineering design. Creating a computational framework for MORF generation requires techniques for efficiently exploring the solution space in a way that is predictive of MORF functionality. This study contributes to this goal by demonstrating that structural fingerprinting, which is already known to be a facilitator of efficient solution space exploration, is also correlated with labels of dynamic function.

The results demonstrate a correlation between structural fingerprints of eight groups of electromechanical products and those products' shared functions. This correlation provides evidence that structural fingerprints are a viable representation for inferring clusters of systems with distinctly different functionality. This result suggests that fingerprints could be valuable to support the automated design of both MORFs and electromechanical systems.

In library design tools such as the Design Repository, fingerprints will provide a fast way to search for products based on a desired component substructure without performing costly subgraph search. The Design Repository also contains functional models, and these functional models can also be represented as graphs. Fingerprints of functional models will similarly enable efficient search for products with a desired set of function chains. Additionally, this approach is scalable to very large design libraries. All that is required is a tool for consistently generating (1) fingerprints of products in the repository, and (2) fingerprints for substructure search queries.

In automated design tasks wherein a large number of candidate solutions are created, fingerprinting and clustering provide tools to reduce the complexity of evaluation as well as to guide the algorithms that generate solution candidates. Given that function is correlated with structural fingerprints, clustering similar products reduces the complexity of two different types of concept evaluation. For evaluation tasks that require expensive computation, representative solutions from each cluster can be evaluated, just as in drug design. For evaluation tasks that require human interpretation, clustering can reduce the size of the search space and facilitate interactive exploration as described in [128]. Using fingerprints to support these clustering operations eliminates the need to specify a dictionary of important features while preserving topological information. Additionally, common substructure fragments in these well-performing clusters can be used as seeds for generating additional concept variants in the same solution neighborhood.

While these results suggest the value of a fingerprinting approach to support computational search, they require verification and validation in the MORF design context. In future work, the fingerprinting approach presented here will be applied in the MORF generation framework to catalog feasible candidate structures. These fingerprints will serve as the basis for calculating similarity, forming clusters, and selecting representative candidates for molecular dynamics simulations during a search. The most feasible results from each run will be used to inform a predictive QSAR-style model that relates molecule structure to important behavior characteristics in the MORF domain. During usefulness screening, these representations will likewise support similarity screening, clustering, and the eventual visualization of each cluster’s behavioral characteristics. Further, different clustering algorithms must be assessed and tuned for their ability to produce meaningfully different groups of candidate structures in this context. Fragment mining from these clusters will create new seeds with known behavior properties in order to generate new candidates with similar behavior. After this framework is implemented it will be possible to conduct experiments with respect to verifying, validating, and tuning the search process.

Acknowledgements

The authors would like to acknowledge the useful inputs and discussions by Alex Greaney, Brady Gibbons, Bryan Maack, and Laura Oliveira from Oregon State University; as well as Jeffrey Rack from Ohio University.

This research is funded by the W.M. Keck Foundation. All views expressed in this article are those of the authors and do not necessarily represent the views of the W.M. Keck Foundation.

Analogy Fingerprinting: Large Scale Analogy Search Inspired by Drug Design

Ryan M. Arlitt, and Robert B. Stone

In preparation for submission to Design Studies

Abstract

This manuscript presents the Analogy Fingerprinting algorithm for quickly matching a large number of strong analogies to a given design problem. Analogy Fingerprinting uses path-based molecular fingerprinting from drug design to enable large-scale matching according to the structure mapping theory of analogy. The body of this article presents the algorithm, including its relationships to analogy theory and molecule search, and its potential impact on design. The paper concludes with a brief validation experiment that demonstrates the effectiveness of Analogy Fingerprinting for retrieving good analogies when combined with one of the three similarity measures examined.

Introduction

A key challenge for software that aims to automatically generate good design analogies is to surprise the designer with high quality ideas that had not yet been considered. In order to meet these requirements, it is necessary to search a large solution space. The alternative is to manually specify a narrow search space, which by definition also reduces the novelty associated with an analogy from that space. For example, a designer is unlikely to ask for every possible analogy between their problem and the domain of carpentry unless they already suspect a strong conceptual connection. If the designer is confident that carpentry contains a good analogy, then half of the analogy matching challenge is already solved.

This type of design analogy search is challenging to perform on this massive scale. The search must span many different sources of information, and consequently this information will be in a wide variety of forms. For example, a typical catalog of design information might include an artifact’s overall and constituent functions, constituent components, dimensions, constraints, use case information, and/or various types of requirements. The challenge associated with analogical matching based on these information categories is that, outside of a small subset of potential analogical solutions that exist in engineering databases, most potential analogies are not explicitly described in neatly organized categories. Instead, potential solutions are often described in natural language, in sources such as in scientific publications, textbooks, and patents. (Although one may argue – convincingly – that the language used in patents is hardly “natural,” it is not neatly organized in the manner of a design database.) As a consequence, there is a need for algorithms that are capable of distilling key conceptual relationships from this noisy data – and then performing matching – on a large scale.

A concept map [154] is a type of information representation that models a set of concepts and the relationships between them. Concept maps provide a convenient representation for performing large scale analogical matching because – while they have the necessary relationship structure – they can be generated from a wide variety of sources. Representations such as natural language descriptions, databases, and technical specifications can all be represented as a set of key concepts and the relationships between them. For example, the entity-relationship style of most databases requires little-to-no processing to convert into a concept map, and there exist a variety of methods for converting natural language passages into concept maps (e.g., natural language processing and human computation [96, 112, 155]). This work assumes that a scalable method or set of methods has been used to create large library of concept maps from this zoo of potential information sources.

The Analogy Fingerprinting algorithm presented in this paper leverages the molecular fingerprinting algorithm from drug design to address the challenge of high-speed analogy matching on a large scale. Given a set of potential analogy candidates encoded according to their internal conceptual relationships (i.e., concept maps), the algorithm matches a design problem to its most plausible analogies. In terms of matching, Analogy Fingerprinting attempts to satisfy the structure mapping [3] criterion – that an analogy between two domains is strong because of matching relationships between entities in those domains. In simple terms, this means that there is a mapping between the domains' relationship structures.

The remainder of this paper is structured as follows. First, a brief overview of the related work discusses analogy formation and relevant drug design techniques. The next section describes the Analogy Fingerprinting algorithm and its application to design. The following sections present an explanatory example of the Analogy Fingerprinting, results of the effectiveness of several similarity measures, and a brief discussion of the method's validity.

Background

This section describes related concepts and work in the areas of analogy, graph theory, molecular fingerprinting, and fingerprint similarity.

Analogy

Designers often base new solutions on old solutions using a process called design by analogy [43-45, 84]. The principle that analogy takes place between networks of concepts and the relationships in each network is widely agreed upon [4]. For example, Gentner's structure mapping theory [3] states that analogical mapping occurs between a source domain and a target domain. Each domain is represented as a structured set of concepts and relationships. An analogy is formed when relationships can be mapped from source to target. The strength of the analogy increases as the number and structural matching of the relationships increase. If one relationship causes another, this increases the strength of the relationship further. An example given by Gentner [3] draws an analogy between the solar system and the Rutherford model of the atom. An electron revolves around a nucleus, while a planet revolves around the sun. An electron is less massive than a nucleus while a planet is less massive than the sun. Each of these relationships strengthens the alignment between the planetary domain and the atomic domain. The Analogy Fingerprinting algorithm presented in this paper matches design analogies based on the structural alignment of such relationships.

Computationally finding such mappings is important because, while analogy formation is indicative of expert designer performance [2], it is challenging for a designer to effectively index and retrieve a large number of potential analogy candidates [48].

Graph Theory

A fundamental concept that supports the Analogy Fingerprinting algorithm is that many very different types of information can be represented as a graph. Briefly, a graph is a set of information comprised of nodes (also called vertices) and edges (also called arcs). Relevant to this work are molecular graphs and concept map graphs. A molecular graph describes the bonds (represented as edges) connecting different atoms (represented as nodes). A concept map graph represents the relationship (represented as edges) between different concepts (represented as nodes). One of the challenges solved in this paper is that of converting an algorithm designed to operate on molecular graphs to one that works on concept map graphs.

Another challenge solved by Analogy Fingerprinting relates to the speed of search. Graph substructure search is an NP-complete problem [139]. In contrast, fingerprint search has linear time complexity [139]. The majority of computation in a fingerprint search is done as a preprocessing step – converting the graph into a fingerprint. As a consequence, a designer (both human and computational) can examine a significantly larger size of the search space on demand than by using graph search directly.

Molecular Fingerprinting

Molecular fingerprinting is an algorithm designed to aid in the rapid screening of drug molecules in drug design applications.

What problems it solves

The molecular fingerprinting algorithm addresses several problems in computational drug design. First, it takes the complex information stored in a molecular graph and transforms it into a representation that can be screened rapidly – it is more efficient to search for substructures in a fingerprint than in a molecular graph. Second, these fingerprints can be used to rapidly calculate structural similarity between two molecules. Third, a population of fingerprints and performance data can be formulated into a predictive model that relates the presence or absence of substructures to their various types of performance. Such models are commonly called Quantitative Structure-Activity Relationship Models or Quantitative Structure-Property Relationship Models (QSAR or QSPR). The algorithm presented in this paper leverages the first two of these approaches.

How it works

There are several types of fingerprinting algorithms, but they all use a hashing function to map substructure data into bits in a fixed length vector. Structural key fingerprints rely on a prescribed set of substructure types; the presence or absence of each substructure is captured in a bit position that corresponds to that substructure [139]. In situations where it is impractical to select a set of graph features beforehand, an alternative approach is to generate an impromptu set of features as fingerprinting occurs. Path-based fingerprints do not rely on a prescribed vocabulary, instead hashing all paths of a specific length (typically from 2-7) into a set of bits in a fixed-length binary vector [139]. For example, Figure 25 shows a single graph substructure being hashed into a single bit. For a molecular substructure search query involving path-based fingerprints, the search query is fingerprinted and then compared to every fingerprint in a searchable set. The search returns every molecule that contains all of the same bits as the search query. Other applications use fingerprints to calculate similarity between molecules. Other types of fingerprinting algorithms, such as 3D molecular fingerprints, are outside the scope of this work.

A common approach to generating the fingerprint is to first represent each path as a hashable entity. Molecules can be represented in a myriad of ways; one of which is the SMILES string [156]. SMILES strings capture complex molecular graph information into compact strings, which can then be hashed into a fixed-length binary vector. Each path is hashed into several bits using a Cyclic Redundancy Check (CRC) – an algorithm that was originally designed to ensure that a digital message has not been altered [150]. CRC is desirable because it hashes into a fixed length (which is necessary for comparing two different fingerprints) and can be calculated quickly. In brief, a CRC uses a generator polynomial and a binary representation of the hashable item to repeatedly perform polynomial division. The remainder after performing this process serves as that item's hash. A set of multiple generator polynomials can be used to create unique pattern of bits for each path.

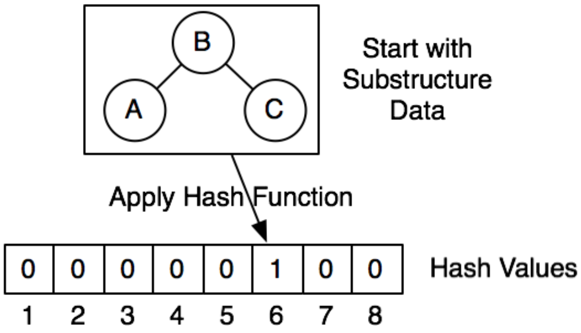


Figure 25. Simple Structure Hashing

The process of hashing into multiple bits stems from Bloom filters [143] – probabilistic data structures which were initially designed to speed up hard drive access times by checking for the absence of a desired file's unique signature. Hashing each entity into multiple bits generates a pattern with a much higher probability of being unique than with a single hash, thus reducing the likelihood of a false positive. In other words, using multiple hashes reduces the probability of collisions between two different pieces of data by reducing the likelihood that all of their hashes are the same. An optimal number of hashes results in the lowest probability of total collisions between two patterns, and is dependent on the number of entities being hashed and the size of the bit vector.

The Bloom filter approach aligns perfectly with the goals of molecular search. Both approaches aim to quickly narrow the field of candidate hard drive locations/molecules, and both are guaranteed to return every match (along with a false positive rate dictated by the probability of collisions).

Why it works

A key advantage of path-based fingerprinting is its content-agnostic approach to representing substructures. A fingerprint can be created for any graph, regardless of content and domain. Such an algorithm can index and search any information that can be represented as a graph, but this is not the only advantage of the approach.

In addition to their content-independence, path-based fingerprints provide a built-in weighting scheme based on fragment size – large fragments are weighted more heavily than small fragments. This is largely due to the way in which they capture the overwriting substructure of molecules. For example, ethanol (Figure 26) can be represented using the SMILES string CCO (two Carbons and an Oxygen connected in a linear path) based on its hydrogen-depleted molecular graph (Figure 27). A simple path-based fingerprint of ethanol would capture the paths for CC, CO, and CCO – three bits of information rather than one (Figure 28). A search query for the path CCO would match on all three of these bits (100% match), while a query for CC or CO would match only one out of three (33% match).

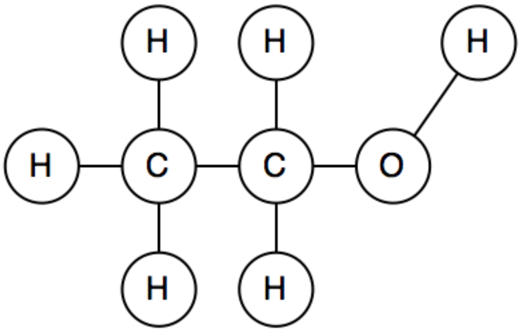


Figure 26. Ethanol Molecular Graph

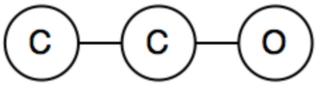


Figure 27. Hydrogen-Depleted Ethanol Molecular Graph used to produce SMILES String

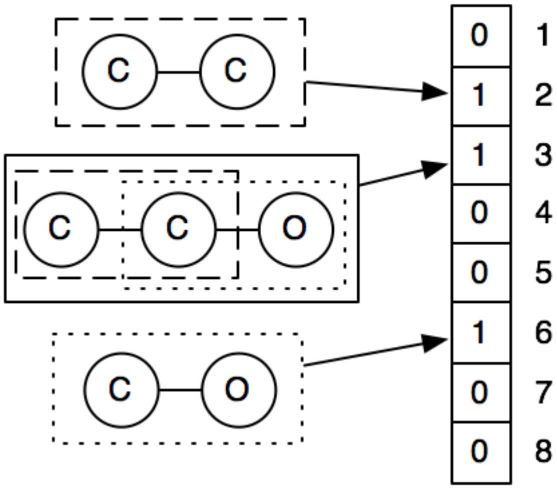


Figure 28. Ethanol Graph Path Hashes using a Single Fictional Hashing Function

As the size of the path increases, so does the amount of information it captures. Every new bit carries with it the possibility for a successful match or a failed match. The overall effect is that a path-based fingerprint weights large substructures more heavily than small substructures. Eq. 4 describes this relationship for path-based fingerprints of linear substructures starting at length two and going up to the length of the substructure, where L is the length of the substructure's path and *B(L)* is the total number of bits that the substructure has in its fingerprint.

$$B_{path}(L) = \frac{L^2 - L}{2} \qquad \text{Eq. 4}$$

Ring substructures are weighted more heavily. Cyclic substructures allow more unique paths using the same number of nodes. For example, the path ABCD contains exactly one four-node path. A ring made from the four nodes ABCD contains four unique paths: ABCD, BCDA, CDAB, and DABC. Directionality is irrelevant – the subgraph described by ABCD is the same as the subgraph described by DCBA. Ring substructures follow the relationship given in Eq. 5

$$B_{ring}(L) = L^2 - L \qquad \text{Eq. 5}$$

As a consequence, ring substructures are favored more heavily than linear substructures in a path-based fingerprint.

Similarity Measures

Given two fingerprints, it is often desirable to calculate their similarity. Drug design provides many different similarity measures for this purpose, but this work will examine just two: Jaccard similarity and Russell/Rao similarity. Additionally, a third approach to calculating similarity – the Membership similarity measure – is developed and tested based on the specific requirements of analogy matching.

Table 19 presents convenient abbreviations for describing similarity measures. Each similarity measure is described in terms of shared presence or absence of bits in each fingerprint.

Table 19. Similarity Measure Contingency Table Shorthand

	Fingerprint B Bits Present	Fingerprint B Bits Absent
Fingerprint A Bits Present	a	b
Fingerprint A Bits Absent	c	d

For example, "a" in Table 19 represents the total quantity of bits shared by both Fingerprint A and Fingerprint B, while "c" represents the total quantity of bits that are absent from Fingerprint A but present in Fingerprint B.

Jaccard similarity (Eq. 6) [157] is commonly used in drug design applications [144], and has been demonstrated as a strong general-purpose similarity measure when compared against other metrics [140]. This measure counts the ratio of shared bits to the total number of bits in both fingerprints. Bit positions that are absent from both fingerprints are ignored. The main drawbacks of Jaccard similarity are its tendency to give low similarity scores when the query fingerprint is very small, and its bias toward specific similarity values [144].

$$J(A,B) = \frac{a}{a+b+c} \qquad \text{Eq. 6}$$

A second similarity measure, the Russell and Rao similarity measure (Eq. 7) [158] is also tested in this work due to its favorable performance in molecular search [159].

$$R(A,B) = \frac{a}{a+b+c+d} \qquad \text{Eq. 7}$$

A third similarity measure – related more closely to screening rather than calculating similarity – is also tested. The initial design goal of fingerprinting was to quickly screen a library of molecules for the presence or absence of a given substructure, as in a Bloom filter. This type of search has two outcomes; either a candidate matches all bits in a query (indicating that the substructure is very likely to be present in its entirety) or it doesn't (indicating that the substructure is definitely not present). The remaining population of matching candidates is very likely to contain the substructure, though this is not guaranteed due to the probabilistic nature of fingerprint searching (i.e., the potential presence of collisions).

The Membership similarity measure builds on this idea to determine how closely a given search query matches this 100% membership criterion. Membership similarity of a candidate *C* inside of a search query *Q* is defined according to Eq. 8. A query whose every bit is accounted for in a search candidate is a 100% match for a substructure in that candidate. The results section shows that the membership similarity measure is the most performant of the three measures examined in this study.

$$M(C,Q) = \frac{a}{a+b} \qquad \text{Eq. 8}$$

Notably, this metric is not a true similarity measure because M(C,Q) and M(Q,C) will produce different results. However, it is useful for comparing the extent to which members in a set of candidates satisfy the Bloom filter style of membership criterion beyond a simple pass/fail. In the context of design, it is useful for finding the analogy candidates that most closely address all or part of a given problem.

The Analogy Fingerprinting Algorithm

The Analogy Fingerprinting Algorithm section begins by discussing the algorithm's goals, its relationship to molecular fingerprinting, its application to design, its impact on design. The core of the section describes the algorithm's mechanics and presents an explanatory example.

Goals

The goal of Analogy Fingerprinting is to provide a fast way to retrieve high-quality analogies by matching their relationship structures to a given problem domain. Put simply, the algorithm is designed to match design problems with relevant analogies. Because conceptual relationships play a key role in analogy formation, the algorithm operates on concept map representations of design information. A concept map describing a design problem's main concepts – whether those concepts are requirements, desired functionality, constraints, or any other key information – serves as a search query. Each existing solution (i.e., potential analogy) is similarly represented as a concept map of the key domain concepts that allow an artifact to solve a problem. If a design problem shares many key conceptual relationships with a potential solution – regardless of what type of information comprises those relationships – then the solution satisfies the structure mapping criterion for a good analogy.

Relationship to Molecular Fingerprinting

Molecular fingerprinting is an approach to encoding the structural information of molecules into a form that is easily searchable. The path-based approach to fingerprinting is content-agnostic; the interesting types of structural information need not be specified beforehand. In path-based molecular fingerprinting, the types and connectivity of atoms in a molecular graph are hashed into fingerprint bits. This approach takes advantage of the finite number of possible atoms.

Concept maps differ from molecular graphs in that (1) edges are labeled, (2) edge labels are important while node labels are not important, (3) multiple edges between nodes are possible, (4) edges are directional, and (5) there is no generally agreed-upon taxonomy of labels (in this case this means that there is no standard taxonomy of relationships between concepts).

To accommodate these differences, the following changes are made to the fingerprinting algorithm. (1,2) Edge labels are promoted to labels of their preceding nodes. Node labels are ignored for fingerprinting the analogical structure map of a concept map, but can be used as part of a separate process to assess semantic similarity between concept maps. (3) In the case that multiple edge labels are promoted into the same node, that node is duplicated once for each new label. All edge connections are identical for all of these duplicate nodes. (4) In addition to simplifying the promotion of edge labels to node labels, edge directionality is captured by extracting paths following only outward edges. (5) Natural language edge labels are promoted into a predetermined relationship classification prior to fingerprinting.

Application to Design

In the design literature, a key principle of conceptual design is to consider many different alternatives before selecting a concept to develop (e.g., [38, 160]). Techniques like brainstorming, C-Sketch [161], mind mapping [162], and many more are all used to address this goal. Any technique that can provide a *wide variety* of conceptual design inspiration that is *relevant* to a problem, while also minimizing the designer's time investment, is a valuable conceptual design tool. Analogy Fingerprinting supports this process with a large library of concept map fingerprints. A designer can then make rapid queries for existing systems – both engineered and natural – that possess an analogical alignment to the problem at hand.

In order to do so, the designer formulates their design problem space as a concept map. This process is non-trivial; it requires a good understanding of the problem. The designer must decide on the most important information elements across many different abstract categories (e.g., requirements, functionality, constraints, known issues, and more), and then must model the conceptual relationships between them. Including superfluous information in this model can change the result because any fingerprint search weighs all information evenly, regardless of content. As a consequence, the designer must consider the impact of each addition to the model. Constructing several models can mitigate the impact of model construction details.

A second challenge is similar to one faced in functional modeling. A valuable functional model describes what something should do, not what it shouldn't do. For example, a roof should “stop liquid” rather than “not leak.” To perform an analogy fingerprint search, the designer should construct a concept map in the desired state. For example, a concept map can describe a design situation in which a leaky roof should be replaced with a non-leaky roof. This concept map should contain an assertion like “roof stops rain” rather than “rain bypasses roof” or “water enters dwelling.” In contrast, if this leakiness is a required operational constraint (maybe the designer wants to use the rain to water indoor plants), then “roof stops rain,” “rain bypasses roof,” and “water enters dwelling” might all be included as important functional requirements.

This concept map serves as a search query. This search query is fingerprinted and compared to fingerprints of every potential analogy in the knowledgebase using a binary similarity measure. This produces a similarity ranking of existing systems to be presented to the user.

Design Impact

This style of approach – in which a designer specifies conceptual interrelationships between many types of design specifications (e.g., requirements, functions, constraints) – has the potential to improve design outcomes in several ways.

The first advantage is an improvement in the breadth of high-quality solutions considered. The algorithm represents analogical structure in a way that is quickly and easily searched. Matching conceptual structure maps leads to high-quality analogies, while the high speed of search improves the breadth of analogies that may be considered.

The second advantage is that fewer design iterations are needed to converge to a valid solution. A typical engineering design process thrives on iteration. Each iteration costs resources, but provides valuable information. For example, scrum [163] and similar design processes produce many quick prototypes in short sprints. These design sprints reduce uncertainty of design outcomes – such as requirement satisfaction and subsystem interactions – that are difficult to predict. In contrast, a major advantage of a strong design analogy is its inherent validity – the analogous object already solves a relevant problem, and its behaviors and challenges are generally understood based on real-world performance. In contrast, a new design from first principles lacks the validity from this built-in real-world testing. The algorithm facilitates analogical matching on the conceptual features that are most important to the designer. This matching provides conceptual suggestions that inherently require less testing and iteration because the validity of the base system is already known.

The third advantage relates to handling complexity. There is a heuristic in complex systems architecting that suggests "doing the hard part first" [164]. Every design decision reduces the decision space for every subsequent design decision. The goal behind doing the hard part first is to mitigate the compound difficulties of solving difficult challenges in the face of restrictions imposed by past design decisions. This approach provides a framework for a designer to identify and confront difficult challenges across multiple levels of abstraction and subsystems early in the design process. An appropriate analogy or set of analogies that address these difficult challenges can be identified early in a design process.

How the Algorithm Works

Each concept map is a directional multigraph; a graph where edges are directional and parallel edges are allowed. Each edge is labeled with a relationship. Each edge relationship must be described using a fixed vocabulary. For example, the Functional Basis [15] and Component Taxonomy [42] provide restricted vocabularies for describing electromechanical functions and components. The periodic table provides a restricted vocabulary for describing atoms. ConceptNet [113] and BioP-C [155] provide restricted vocabularies for describing general relationships.

An alternative option (and the one taken in the example presented later in this paper) is to use WordNet [90] to promote natural language relationship labels into a high-level taxonomy of relationships. This algorithm repeatedly promotes the root word of each relationship label into its hypernym until the label is so general that it cannot be promoted again. In the case of multiple possible hypernyms, the algorithm selects the one that is most frequently used. For example, the verb "transform" has seven possible word senses. The first of these is selected and promoted to its hypernym of "change." This word sense of "change" does not have any further hypernyms, and so "change" is the final label given to this relationship. The verb "convert" follows a similar path to the same top hypernym. The end effect is that two concept maps – one containing the assertion "X changes Y" and another containing the assertion "A converts B" – will match each other on this feature. It is also possible to represent concept maps at lower levels of abstraction by first promoting every edge to its top hypernym and then backtracking by a fixed number of hops.

This generalized approach to normalizing concept maps provides an advantage that is synergistic with path-based fingerprinting. Path-based fingerprinting circumvents the need for a predefined set of patterns, just as hypernym promotion bypasses the need to predefine a fixed relationship vocabulary.

Starting with this restricted vocabulary concept map, several transformations must occur before a fingerprinting algorithm can be applied (summarized in Figure 29). In the case of molecular fingerprinting, edges are unlabeled and undirected. Molecular fingerprinting captures the connectivity between the various types of atoms. In order to apply a similar fingerprinting algorithm to concept maps; the directionality, potential parallelism, and edge labels must be represented in a manner that makes it possible to capture every chain of relationships. The end goal is to preserve the graph's connectivity such that extracting every path of a fixed set of lengths captures the full variety of relationship structures in a concept map.

Starting with a concept map with a restricted vocabulary of relationships, each edge's relationships label is shifted onto its predecessor node. As in structure mapping, relationships between concepts are important, while the concepts themselves are not. Each node with multiple outward edges (and thus more than one of these new edge label properties) is duplicated, complete with all of its inward and outward edges. Next, all paths of lengths two through five are extracted from the graph. The path extraction step preserves directionality information by following only outward edges when creating paths. No single node can be in a path more than once. Similarly, a node and its duplicate are not permitted to appear in the same path. Each of these paths represents a single point of information in a potential match. If two concept maps share a large percentage of paths constructed in this manner, then those concept maps have high analogical similarity according to the structure mapping definition.

1. Begin with a concept map represented as a directional multigraph
2. Convert edge labels (relationships) to a fixed vocabulary
3. Promote each edge label to an attributes of its preceding node
4. For each node with multiple outward edges, split that node and duplicate all inward edges
5. Extract all paths of lengths 2 to 5 from the concept map graph
6. Hash each of these paths into a set of bits in the fingerprint

Figure 29. Analogy Fingerprinting Algorithm

Following this path extraction, fingerprint hashing can continue in a manner identical to molecular fingerprinting. All paths of lengths 2-5 are extracted from the graph, and only outward edges are traced in order to preserve directionality information. Each of these paths is hashed into a 2^{32} bit fingerprint using 14 different cyclic redundancy check [150] polynomials to create a pattern of 14 different bits for each path.

The selection of these numbers of path lengths and hash functions was done based on tuning on a limited data set rather than rigorous study, but it is trivial to show that a reasonable fingerprint size and number of hash functions can be used to index a very large number of concept maps with a very small probability of false positives. The optimal number of hash functions to minimize the error rate is given by Eq. 9 [165], where *m* is the filter size and *n* is the number of elements being indexed.

$$k = \frac{m}{n} \ln(2) \qquad \text{Eq. 9}$$

Assuming 100 million concept maps are being indexed into a 2^{32} bit fingerprint, the optimal number of hash functions is 29 – resulting in a 1.1e-9 probability of false positives. Even with 14 hash functions, this probability is estimated at a negligible 1.7e-8, given by Eq. 10 [166].

$$p \approx \left(1 - e^{-kn/m}\right)^k \qquad \text{Eq. 10}$$

Every available concept map is fingerprinted in this manner to create a quickly searchable analogy library. A designer can then formulate a concept map, fingerprint it using the same algorithm, and quickly retrieve relevant analogies.

Example

This section presents a simple example of the algorithm using the analogy between the planetary model of the atom and the planets themselves.

Figure 30 (left) shows a concept map of the planetary domain, and Figure 30 (right) shows a concept map of the atomic domain.

The first step in the algorithm is to promote every relationship into a restricted relationship vocabulary, as shown in Figure 31. This example generates this vocabulary automatically and organically, using WordNet to promote every relationship into its most probable top-level hypernym. Each relationship is represented by a WordNet Synset: a data type that includes the word, its part of speech, and its word sense.

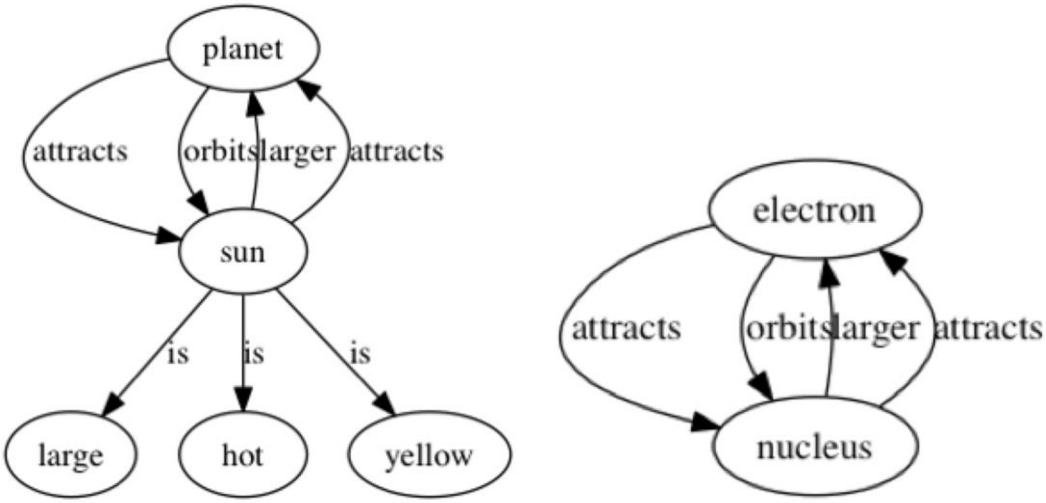


Figure 30. Concept Maps for Planetary Domain and Atomic Domain

For example, the relationship "attracts" is automatically promoted to Synset "move.v.02." This Synset refers to the verb "move," defined in WordNet as, "[to] cause to move or shift into a new position or place, both in a concrete and in an abstract sense." In contrast, "move.v.01" is defined as, "[to] change location; move, travel, or proceed, also metaphorically." The verb "attracts" clearly refers to an entity causing another entity to move rather than an entity's own movement, and the word sense "move.v.02" correctly captures this distinction.

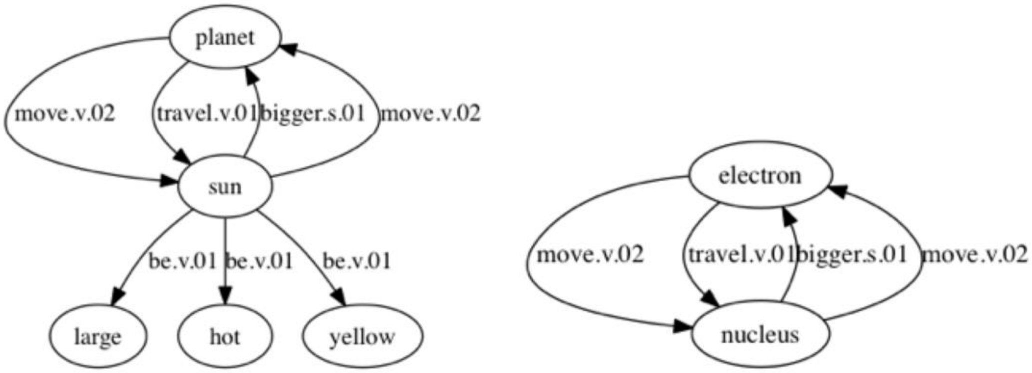


Figure 31. Hypernyms of all Concept Map Relationships

The next step is to remove relationships that express property information, as shown in Figure 32. According to structure mapping theory, relationship information is significantly more important than attribute information. As a simple heuristic, the algorithm removes any relationships whose top hypernym is “be.v.01.” This specific sense of the word “be” – which expresses that something is a property or attribute of something else – is defined in WordNet as, “have the quality of being; (copula, used with an adjective or a predicate noun).

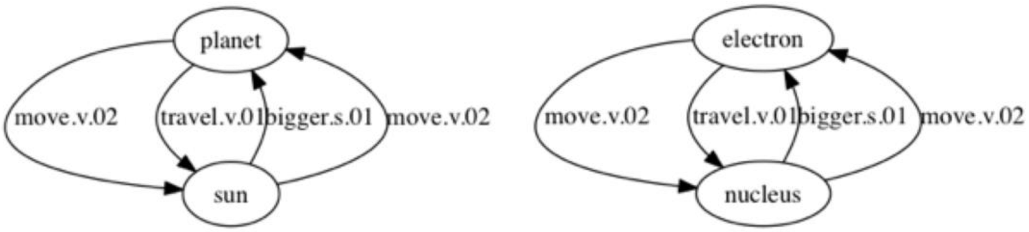


Figure 32. Concept Maps without Attribute Information

After a concept map’s relationship vocabulary has been normalized and its property descriptors have been pruned, each relationship is promoted onto its preceding node. For example, in Figure 32 (left), the relationships “move.v.02” and “travel.v.01” flow from planet to sun. Both of these are promoted onto the preceding node, replacing “planet” with a list of all outgoing relationships. After doing so, due to the simplicity of the example, the two concept maps are identical.

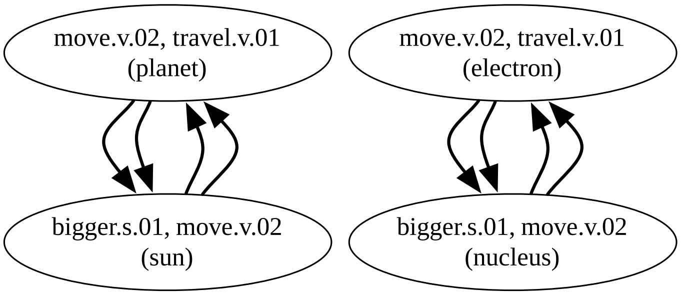


Figure 33. Concept Maps with Relationships Promoted onto Nodes

Next, each node with multiple relationship labels is duplicated while preserving its old connections. For example, the top node in Figure 33 (left) – formerly the "planet" node – is split into a "move.v.02" node and a "travel.v.01" node as shown in Figure 34 (left). Both of these nodes maintain their connections to the node containing "bigger.s.01" and "move.v.02" (formerly "sun"). This node is also split, resulting in the graphs shown in Figure 34. The old labels on each of these nodes are shown in this figure in order to clarify the path extraction phase, but these node labels from the original graph are not actually used by the algorithm.

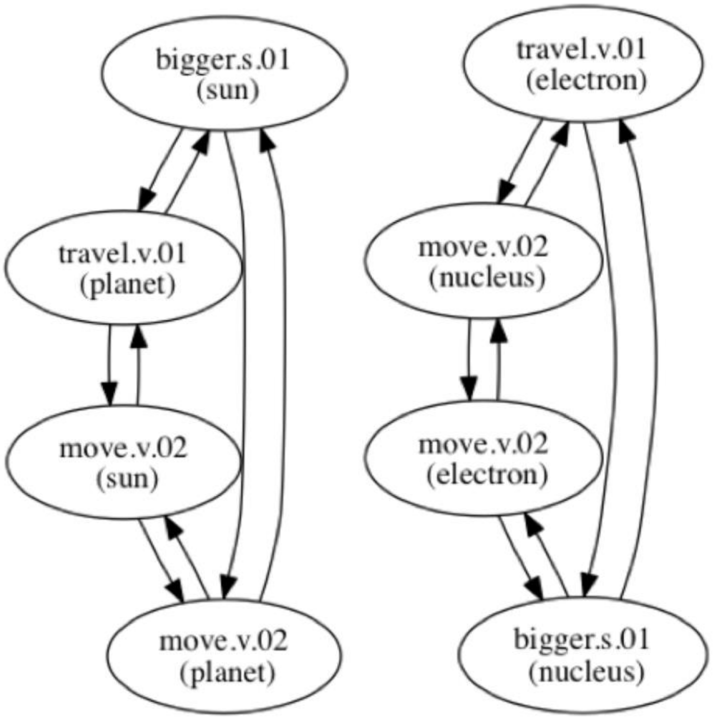


Figure 34. Concept Maps After Splitting Nodes

The next step is to extract all nonrepeating paths within a specified range of lengths. Nodes that have the same old label cannot be repeated. Because there are only two nodes (electron and nucleus), the graph does not contain any paths greater than length two. All of these paths are shown in Figure 35. Next, a string representation of each of these paths is hashed into a unique pattern of bits. Figure 35 depicts the hash results by showing the pattern of bit positions in a 32-bit vector that corresponds to each path. For this example, 14 different hash seeds are used to produce a unique signature of 14 bits for each path.

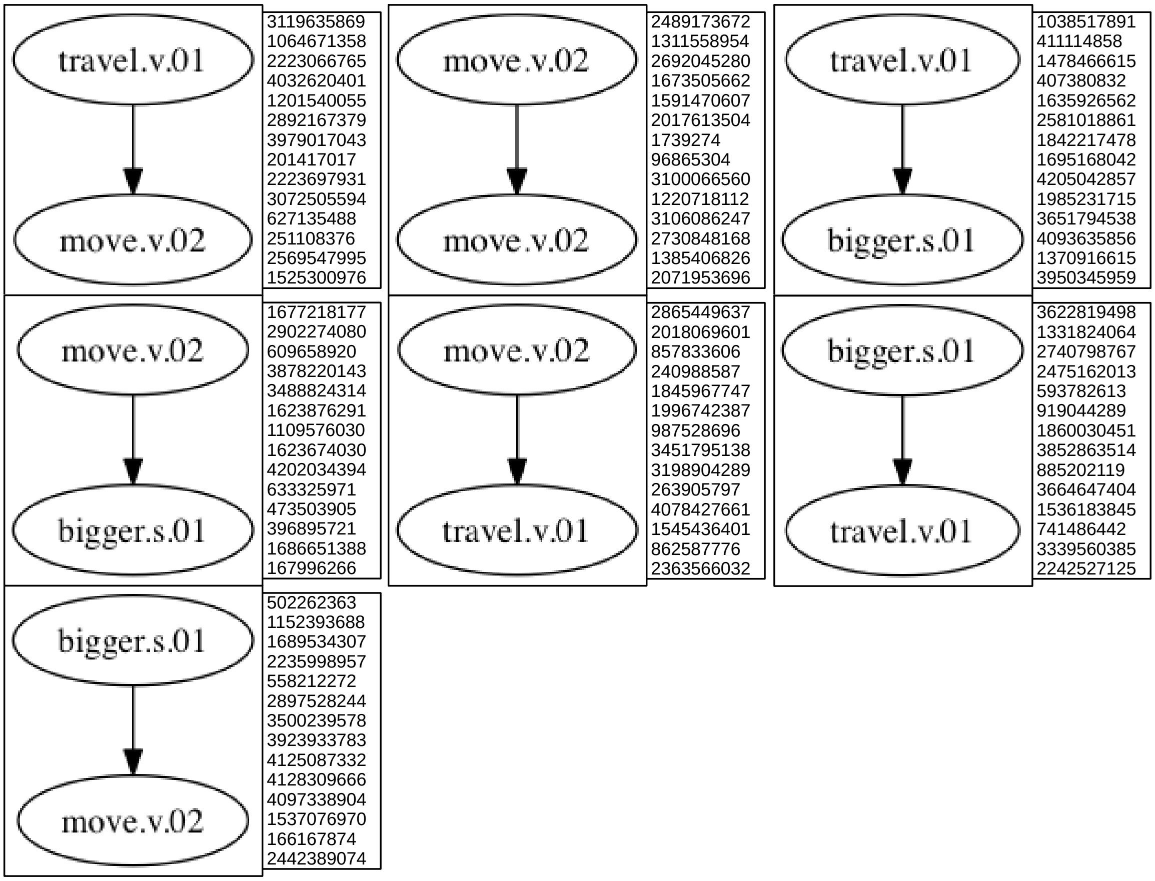


Figure 35. All Paths for Planetary Domain Concept Map

Assumptions

The Assumptions section discusses the assumed content and abstraction of concept maps used by the Analogy Fingerprinting algorithm.

What’s in a Concept Map?

An implicit (and important) assumption in this method is that every concept map contains the most meaningful information related to the concept of interest. Each concept map is modeled at a level that is most conducive to describing the phenomenon of interest. I.e., while every conceptual domain exists at many levels of abstraction, it is described according to its most salient features.

If a concept is modeled in great detail at low level, then it is assumed that this description is the most meaningful. Search queries for relevant analogies will describe the design situation at the desired abstraction level. If a query and a potential analogy are modeled at different levels of abstraction, then it is assumed that the potential analogy is not the focus of the current search query.

For example, a passage describing photosynthesis might contain relationships between various mechanisms that comprise the process. A more general passage about plant survival characteristics may only mention photosynthesis as part of a larger goal. The general passage would be more likely to match a survival problem, while the photosynthesis passage would be more likely to match an energy problem.

Mixed Abstraction

An issue with the graph-based approach to representing knowledge models is the difficulty of expressing high order relationships between groups of concepts. For example, an entire process (e.g., annealing) may be treated as either a group of distinct steps or as a single atomic unit. For some reasoning tasks it may be useful to say that the entire process produces the end result, while for others it may be more useful to understand that the final step leads to the end result. Annealing is a heat treatment process wherein a metal is heated to a specific temperature and then cooled at a specific rate. This process can be used to relax internal stresses in a metal. The overall process of annealing and the internal subprocess of cooling could both be modeled as leading to the end result. A comprehensive method for analogy matching must accommodate all levels of abstraction.

While the current work does not address this challenge, the concept map representation is capable of doing so. A single node in a concept map is loosely typed; it can represent any type of information, and can even contain entire subgraphs. Inserting such a node into a concept map and relating it to the other concepts in a graph can express high-order relationships across mixed levels of abstraction. The challenge of recursively fingerprinting and retrieving such nested relationships across mixed levels of abstraction is left to future work.

Validation Experiment

As is often the case in design research, new techniques are seldom introduced and validated simultaneously – true validation requires long term *in situ* testing, observation, and ultimately acceptance by educators and practitioners. The Validation Experiment section describes an experiment that addresses the value of Analogy Fingerprinting in a single design situation. The section is broadly divided into two subsections – the first subsection describes the construction of the experiment and the second subsection describes the analyses and results.

Approximating a Design Analogy Situation

Validation of the algorithm was performed with students in a graduate-level biologically-inspired design course. The students were presented with a design prompt and each student constructed a concept map of a desired solution. These concept maps are used as queries for testing the efficacy of matching analogy fingerprints. The design prompt describes a design context in which three separate problems exist relating to temperature sensitivity, vulnerability to predators, and energy efficiency. After creating these concept maps, the students were introduced to existing biologically-inspired design approaches that deal with the type of weakly-specified information that appears in concept maps.

Context

A research organization owns capsules containing scientific instrumentation for measuring climate data across the world. They want to deploy these capsules in the arctic tundra of the North Pole. Unfortunately, their current design suffers from the challenges of the arctic tundra climate:

- Some of the instruments in the capsule are sensitive to low temperatures.
- Polar bears have been known to destroy these capsules. The scientists speculate that one of the electromagnetic signals emitted by the instruments is attracting the bears.
- Each capsule is accompanied by a solar array that powers all of its instruments. Unfortunately, sunlight in the tundra is insufficient for large portions of the year, causing many of these capsules to become inactive.

Problem Statement

Given this context, create a concept map of what an artifact that solves the problem should be. A concept map captures the important concepts in a domain, as well as how those concepts are related to each other. You should consider the structural and functional aspects of a solution, but you are not limited to just these. Other possible aspects include constraints, user needs, or reasonable assumptions.

Figure 36. Design Prompt for Concept Map Creation

Figure 37 shows an example of a query concept map. As with the other concept maps in the data set, the experimenter manually condensed multiple-word relationships into single-word relationships (e.g., “quantified by” becomes “quantifies” and the directionality changes). Additionally, part of speech tags were added to support automatic hypernym promotion, and adverb phrases such as “is in” (e.g., “capsule is in ground”) are reduced to the root adverb (e.g., “in”) and tagged as adverbs accordingly.

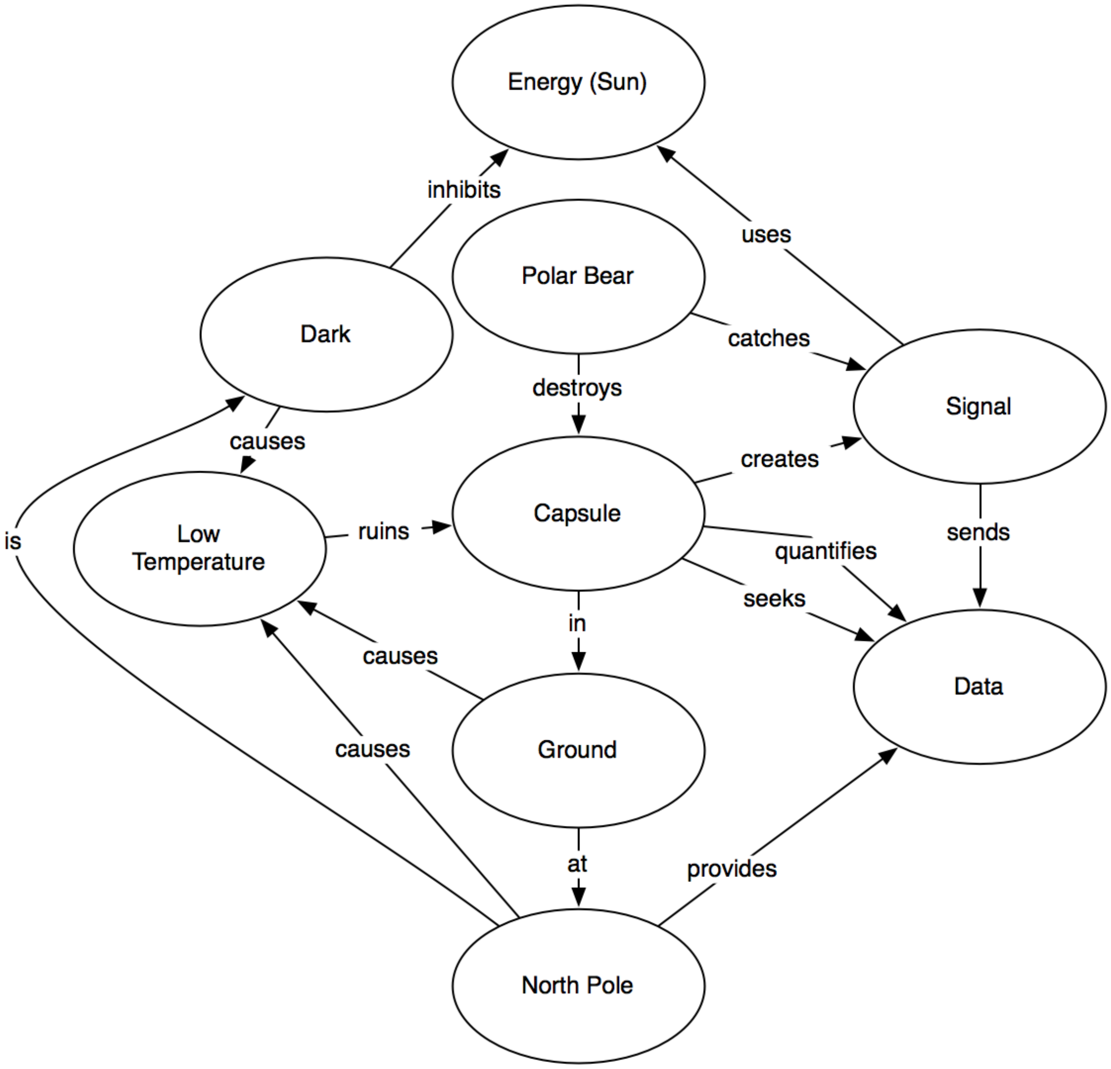


Figure 37. Sample Query Concept Map

These concept maps were fingerprinted and used as search queries against two sets of seven concept maps that were labeled *a priori* as either good analogies or bad analogies. The good analogies were selected based on the biological systems that the students suggested as good analogies after the concept mapping activity. The bad analogies were selected from the biology domain by the first author. The only criteria for the bad analogies were that they are available in the AskNature database of biological systems [167] and that the biological system of interest has little conceptual overlap with any of the subproblems given in the design prompt (in the judgment of the author). The author created all of the concept maps in both sets manually using the steps enumerated in Figure 38.

1. Identify and read a passage describing the biological system.
2. Identify the main strategy described in the passage.
3. Identify the most meaningful words and phrases in the passage.
4. Using the most meaningful words and phrases as nodes, construct relationships to describe the content of the strategy.

Figure 38. Concept Map Creation Algorithm

Table 20 lists the resulting analogy candidates, with examples of bad and good analogy candidates in Figure 39 and Figure 40 respectively. It should be noted that an automatic algorithm for converting natural language to concept maps would produce a more authentic test of analogy fingerprints, and creating these concept maps manually somewhat idealizes the assessment. As a consequence of this idealization, the current experiment assesses analogy fingerprint matching without the noise introduced by imperfections in any specific NLP or human computation algorithm.

Table 20. Good and Bad Analogy Candidates

Good Analogies	**Bad Analogies**
Heliotropic Flowers	Ant and Anti-Bacterial Fungus Relationship
Photosynthesis	Ant Nest Rebuilding Behavior
Polar Bear Fur	Biopolymer Fermentation
Porcupine Quills	Caddisfly Glue
Porpoise Blubber	Jiggled Mud Construction Technique
Snake Dens	Sponge Growth
Whale Blubber	Vine Structure

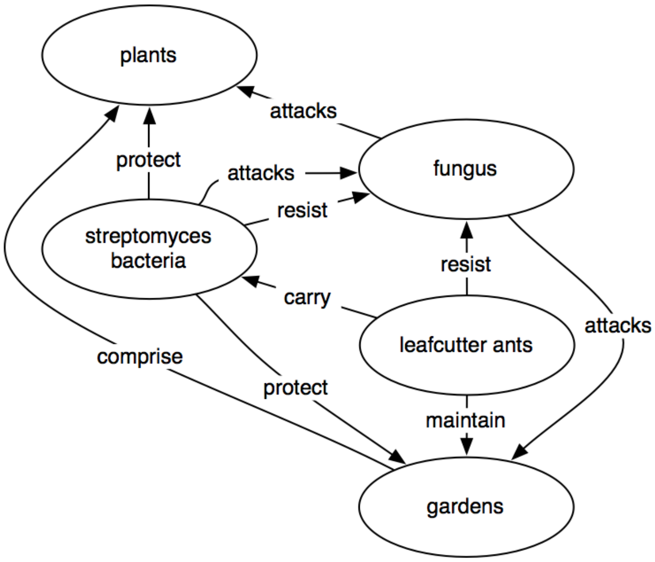


Figure 39. Bad Analogy Sample – Leafcutter Ants' Symbiotic Relationship with Streptomyces Bacteria

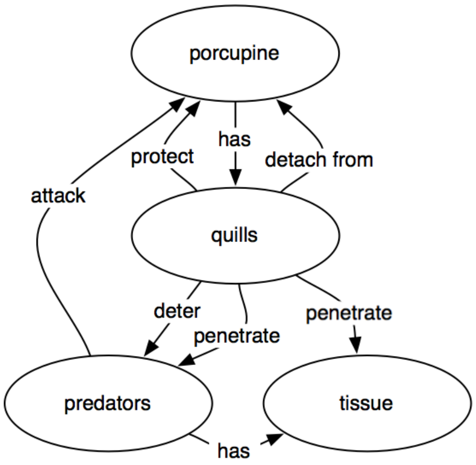


Figure 40. Good Analogy Sample – Porcupine Quills

After removing concept maps created by nonnative English speakers (which would introduce unwanted noise into the results), ten concept maps produced by six students remain. The density, number of nodes, and number of edges for each of these concept maps are shown in Figure 41, where density d is defined according to Eq. 11, with m representing the number of edges and n representing the number of nodes. The averages for each of these quantities are 0.1800, 9.8 nodes, and 13.2 edges respectively. By comparison, these average values for the good analogy set are 0.2995, 6.6 nodes, and 8.6 edges; and for the bad analogy set these values are 0.2968, 6.4 nodes, and 9.3 edges. Three Mann-Whitney U Tests comparing the good and bad analogy sets return p-values of 0.80, 1.0, and 0.54 for these three graph descriptors, suggesting no obvious differences between the construction of the good and bad analogy test sets.

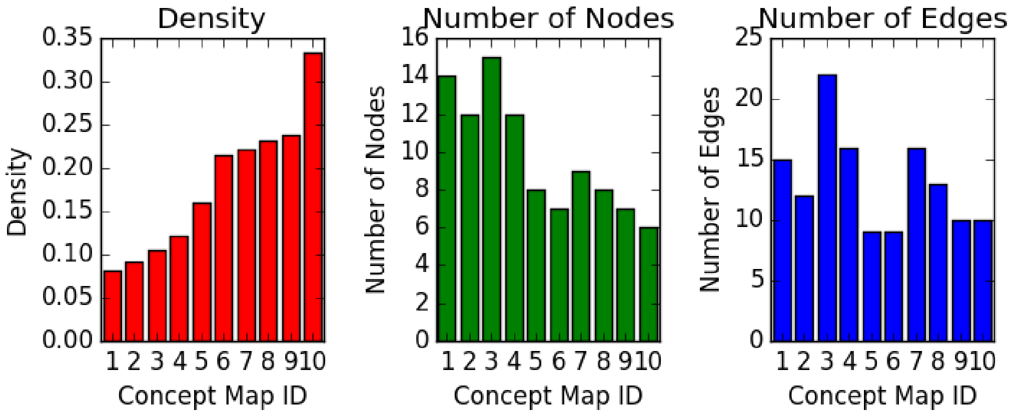


Figure 41. Summary Graph Descriptors for every Query Concept Map

$$d = \frac{m}{n(n-1)} \qquad \text{Eq. 11}$$

Notably, no single concept map in the test set created by the students addresses all three of the subproblems described in the design prompt, while each potential “good” analogy addresses exactly one subproblem. For example, the concept map in Figure 37 contains the assertion “polar bear destroys capsule.” This phrase adequately summarizes the problem, but does not contain any design intent to prevent the capsule’s destruction (e.g., “protective device deters polar bear”). As a consequence, for any given query from this test set, some of the “good” analogies are actually bad matches. The end result is that the “good” group contains some false positives for every search query, increasing the conservativeness of any test designed to evaluate the retrieval rate of the algorithm.

Analogy Fingerprinting Effectiveness – Analyses and Results

In order to test the quality of results retrieved from analogy fingerprints, each of these concept maps was fingerprinted using the Analogy Fingerprinting algorithm. Next, using each of the ten human-generated concept maps as a query, similarity scores were calculated between each query and each potential analogy. A higher similarity score indicates a better match between query and potential analogy source than does a low similarity score. A one-tailed Mann-Whitney U Test was used to compare the ranks of the two groups, addressing the question of whether the good analogies rank significantly higher than the bad analogies. More precisely, this test rank-orders every similarity score, and then determines whether the mean of the ranks from the good analogy group is significantly different from the mean of the ranks from the bad analogy group. This nonparametric test is appropriate because it makes no assumptions about normality. The results of this test for each similarity measure are presented in Table 21.

Table 21. Results of Mann-Whitney U Test for Several Similarity Measures

Similarity Measure	**p-value**
Jaccard	0.2202
Membership	*0.0414*
Russell/Rao	0.2736

The results in Table 21 suggest that the two traditional similarity measures (Jaccard and Russell/Rao) do not adequately prioritize the good analogies. However, a single analogy is rarely a complete solution to a problem. Instead, an analogy (or part of an analogy) often solves one sub-problem. This phenomenon, known as compound analogy, was observed and documented in the context of biologically-inspired design [168]. In the context of a concept map, a compound analogy is an analogical system that slots into a part (i.e., subgraph) of the overall problem's concept map. In light of this knowledge, it is reasonable to suspect that measuring the degree to which a potential analogy fits into a query map is an effective means of performing analogy matching. The Membership similarity measure addresses this type of atomic solution-to-problem matching. The p-value below the significance level of $\alpha = 0.05$ for the Membership similarity measure supports the hypothesis that Membership similarity is an effective means for retrieving atomic analogies to match a problem. A one-tailed Welch's t-test (which does not assume equal variances within both sets of similarity values) further supports the significant result for Membership similarity (p=0.0344).

In the context of this experiment, these results show that Membership similarity ranks good analogies higher than bad analogies. The results also suggest that applying Analogy Fingerprinting to a novel design task enables retrieval of good analogies when the Membership similarity measure is used. Future work is needed to determine the algorithm's performance under various conditions including different concept map generation algorithms, different information sources, different domains, larger sample sizes, and additional types of similarity measures.

A second view of the results is given by calculating precision (Eq. 12) – a commonly used measure for assessing information retrieval algorithms [85] that provides a simple but useful way to assess such an algorithm's performance. Precision is defined as the ratio of true positives (*tp)* to the total sum of true positives and false positives (*fp)*.

$$Precision = \frac{tp}{tp + fp} \qquad \text{Eq. 12}$$

In order for precision to be meaningful in the context of this experiment, it is necessary to specify a cutoff to distinguish between analogies that are retrieved and not retrieved. *Precision at K* (P@K) [169] calculates precision for the top *K* results in a given information retrieval task, and provides an easily interpretable way to assess search results that are meant for a human user. Figure 42 plots the P@K for all values of *K* based on Membership similarity scores. In the case of ties, false positives are ranked above true positives. By its nature, P@K scores are most meaningful for low values of K, where K corresponds to the number of results that a user might be expected to examine before accepting a good result or constructing a new query. The plot also shows the level at which the algorithm has no discriminatory power. In this experiment there are an equal number of good and bad analogy candidates, which means that a precision of 0.5 is no better than random.

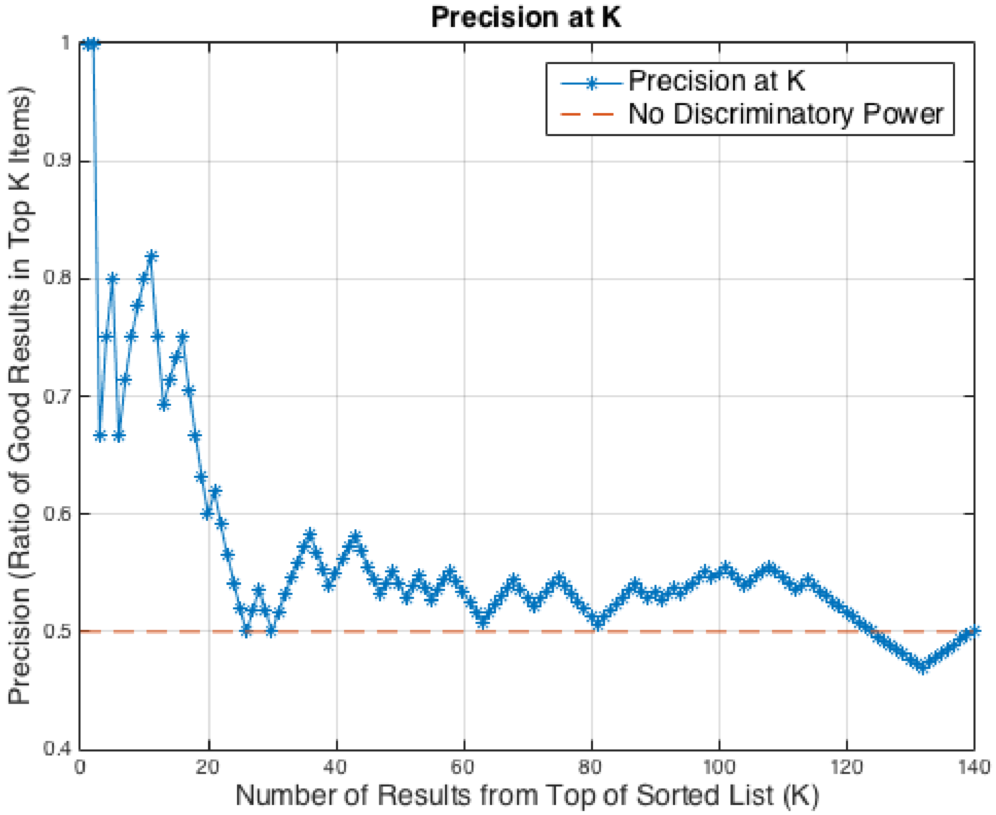


Figure 42. Precision at K for Membership Similarity

The plot shows that Analogy Fingerprinting has good precision for small values of *K*, which is desirable for a search algorithm with human-interpreted search results. Furthermore, the top 15-20 results have acceptable precision, with subsequent results providing insufficient confidence of high quality. For example, assuming a single user created every concept map query in the test set, and then performed an Analogy Fingerprint search against the "good" and "bad" analogy search groups, this plot shows that 80% of the top ten search results would be relevant analogies. Ultimately, these results imply that while there are many combinations of queries and good analogies that do not return high similarity scores, the topmost results of an Analogy Fingerprinting query are likely to be of high quality. The conditions under which the algorithm fails to match high-quality analogies (e.g., the subproblem corresponding to a good analogy is not adequately described in a query) – as well as mitigation strategies for each – are left to future work.

Conclusions

This paper presented the Analogy Fingerprinting algorithm for indexing concept maps. Given a fingerprinted library of engineered and natural systems, Analogy Fingerprinting enables fast searching to find the most relevant design analogies in the library. The speed comes directly from the well-understood properties of Bloom filters, while structure mapping theory and the significant results obtained in the validation experiment support the relevance of retrieved analogies using this method. As a consequence, Analogical Fingerprinting may provide designers with an effective option to overcome common analogy-forming shortcomings – such as poorly encoded knowledge, insufficient experience, and difficulty creating mental links across domains [48] – while reaping the benefits of considering a wide range of potential conceptual solutions. Considering many alternatives is a core principle of conceptual design that is known to improve design outcomes, and it follows that Analogy Fingerprinting represents a fast way for designers to improve their design outcomes.

Conclusions

Holistically, these manuscripts examine information abstractions in the context of design by analogy. The first manuscript finds new (currently poorly recognized) ways to abstract design information to support computational analogy search. The second manuscript demonstrates human computation and Games with a Purpose (GWAPs) as viable and scalable approaches to gathering abstract design information. The third and fourth manuscripts evaluate and extend an algorithm for characterizing graph-based system abstractions. The third manuscript demonstrates the algorithm's value for searching large quantities of design structure information, while the fourth manuscript demonstrates its usage in large-scale analogy search.

Manuscript 1 Conclusions

Manuscript 1 examines the similarity abstractions that designers use when creating conceptual design analogies. As other work has shown, abstract functional similarity can improve retrieval of design analogies. This work examines other types of abstract similarity that can facilitate computational analogy retrieval. The results of this work provide evidence suggesting a variety of abstractions to support schema-based design analogies. The **key findings** include: (1) that flow behavior is a commonly used type of abstract similarity for drawing analogical connections, and (2) that there was no significant difference detected in the types of similarity used to inform compound and single analogies. Notably, while flow behavior was exhibited as a common connection between domains in this study, there exists no established flow behavior abstraction to support database-driven DBA.

In general terms, these results provide insights into the types of high-value mental shortcuts and processes that commonly facilitate analogy formation. As they relate to industry, the results inform the creation of CAD tools and knowledge management techniques to help novice designers see conceptual connections between institutional design knowledge and existing design challenges. This may be useful not only for helping novice designers to perform more like experts, but also for helping companies interested in developing a dynamic and innovative product lineup to explore nonobvious cross-domain solutions and strategies.

The results of this study show no significant difference between the types of similarity used to draw compound and single analogies. This suggests that the value of different types of similarity with respect to forming analogical connections is independent of the generative goal. Whether expanding the breadth of the concept pool (single analogy) or improving concept fidelity and problem understanding (compound analogy), different types of abstract similarity are equally useful.

This study also resulted in several inconclusive observations about common types of internal knowledge queries, the frequencies of various concept generation categories, and the prevalence of direct reuse as a preliminary design strategy. Potential areas for future work include studying these areas in more detail; mining the collected data for further correlations; and investigating the relationship of concept quality to concept generation process, direction of reasoning, and the presence or absence of analogy.

All three directions of analogical reasoning occurred frequently in this study: function-to-form, form-to-form and flow-to-form. This suggests that function, flow, and form information should all play roles in similarity-based analogy retrieval. Of these, similarity between flows through a system is both prevalent in the results and missing from the categorizations in the reviewed literature.

For example, common analogies observed in the results of this study include paper-processing devices (e.g., printers, printing presses, and junk mail folders) and sheet metal rollers. One way of abstracting this problem is by function. A search query of “shape material” could be used to retrieve metal rollers, but likely not printers. While printers change the shape of paper, the design intent of a printer has little to do with this behavior. The starting and ending shapes are also the same; so state-based methods may also have difficulty detecting this behavioral similarity.

Paper shares more literal similarity with towels than with sheet metal, yet all three undergo processes that could be used to flatten or fold something. There are properties of paper and sheet metal that relate to their “flattenability;” their emergent behavior under specific conditions. Many designers in this study inferred from these properties (and from observed behaviors of paper and metal) that paper and sheet metal are sufficiently similar to cloth that similar mechanisms will produce similar flattening behavior in both.

The results of this study suggest that a designer could leverage flow behavior abstraction (e.g., “foldability” and “flattenability”) to search for systems that interact with things possessing desired (e.g., towel-like) behavioral properties. More generally, abstracting the behavioral properties of flows (in addition to system functions) can be a valuable approach to finding analogical connections, especially when the analogy search is guided computationally. This approach provides a simple search heuristic to improve the quantity of potential high quality analogies for a designer to consider. Because analogy is a major component of design, this will improve design outcomes.

Manuscript 1 Impact

The observed importance of flow information agrees with the semantic network view of knowledge definition; that concepts are best defined by their related concepts. This result suggests new ways to catalog and mine design data. Similarity searching within existing libraries can be performed not just on the artifacts themselves, but also on the flows that interact with those artifacts. This practice will improve the outcomes of computational design support that relies on similarity searching, such as in knowledge management systems.

The indiscernible difference between similarity types in single and compound analogies suggests that the same tools and techniques can be used to support both types of design approaches. More generally, this result supports the notion that single and compound analogies follow the same cognitive mechanisms. This result impacts developers of analogical support tools and researchers who study analogy. The former will benefit from the understanding that separate capabilities are not required to support both types of analogy, while the latter will benefit from this new knowledge to design studies of compound analogy.

These conclusions support the value of mixed-abstraction concept maps (containing relationships to describe functions, forms, and flows) as a means to support analogical matching. Unfortunately, computational support for analogy matching requires a large library of candidates, and such concept maps are not readily available. In order to address this issue, manuscript 2 assesses a scalable technique for generating a large set of concept maps.

Manuscript 2 Conclusions

Manuscript 2 presents and assesses a Game with a Purpose (GWAP) for collecting computable knowledge about biological phenomena for the purpose of aiding biologically inspired design. The assessment addresses the external validity of individual assertions collected by the GWAP. Humans assess these assertions for correctness, and these ratings are used to identify potential directly measureable indicators of high correctness assertions. Additionally, manuscript 2 identifies factors affecting the game's entertainment value and potential design features to address shortcomings in this area. The results of this study suggest that a GWAP approach has strong potential to collect valid biology knowledge into a semantic network format that can support biologically inspired design tools. More generally, the manuscript demonstrates that GWAPs represent a viable technique to support information retrieval tasks in design research and practice.

Notably, the correctness of unfiltered BioP-C assertions was rated as significantly better than random and significantly worse than the theoretical maximum, indicating that some of the information produced by BioP-C is correct. Additionally, a statistically significant negative correlation was found between statement correctness and the number of hints created in a game session, which supports a simple and effective filtering operation. Raters' agreement with whether BioP-C assertions are true tends to fall between "neither agree nor disagree" and "agree," pointing to the ambiguity of many assertions collected using the current framework. This highlights a limitation of the current work, and suggests that future work is needed to (1) identify additional player behaviors and session metadata that indicate assertion correctness and (2) refine the game design to encourage these desirable behaviors.

The game itself uses a limited set of general relationships, but better classification will be possible as confidence in general relationship types grows. For example, more detailed relationships can be defined as subclasses of the high level relationships based on existing taxonomies of biology and engineering knowledge. Any number of strategies could support this change. These strategies might include additional mechanisms within the current game design, such as dynamic limitation of available game relationships based on BioP-C's previously collected data. Alternatively, separate game environments could support filtering existing assertions and gathering player-specified relationships.

The validation of this work has revealed the delicate design tradeoff between entertainment value and information quality. Complicated tasks produce better data, but there exists a complexity threshold past which players will not enjoy the game. There may exist a Pareto frontier representing the non-dominated set of tradeoffs indicating the limits of what can be learned from a human algorithm in this context, but it is unlikely that BioP-C v0.3 has reached this point. In order to understand the potential of this approach, future work in this area should aim to quantify this tradeoff, establish where these limits exist, and supply heuristics relating game design to information requirements.

Manuscript 2 Impact

The research presented in manuscript 2 supports human computation and games with a purpose as viable means to address the scalability issue of processing natural language when the accuracy of natural language processing techniques is insufficient. The key results of this publication are the following:

1. Human computation provides a viable alternative method to natural language processing for collecting design information in a computable form.
2. Assertion quality can be inferred from the metadata produced during information collection.

The direct impact of this work comes as a proof-of-concept for a scalable method to collecting design abstractions from natural language. The feasibility of this capability enables computational design support that relies on the types of knowledge – such as concept maps – that can be collected from natural language text.

More broadly, human computation and GWAPs have the potential to aid design research in other ways. For example, online surveys are a common method to gather human inputs modeling consumer preference (e.g., [170]). These surveys have the disadvantage of requiring an external incentive, such as monetary compensation. In addition to the obvious financial drawback of collecting data in this way, the incentive does not necessarily align with the researchers' goal of producing high-quality data. While some participants may be intrinsically motivated, and researchers have techniques for screening out bad data, this situation is far from ideal. If there exists a method that can improve the intrinsic motivation of research participants – while producing very little garbage data – it is worthwhile to pursue development of that method. GWAPs and human computation present an opportunity to collect high quality design research data without compromising the need for large sample sizes. One obvious challenge in this context is the high effort required to develop a human computation task. Such a task (or library of tasks) would need to be extensible to a wide variety of research goals – much as the content of a survey can be changed in order to study different topics – in order to represent a feasible alternative to existing data collection methods.

Assuming that there exists a scalable method for producing design abstraction relations (like that shown in manuscript 2), the question arises of how best to use this information. In other words, in what ways can a large library of design information improve the process of design? One answer to this question lies in the field of drug design – a field that uses large libraries of abstracted structure data to help design new drugs. Manuscript 3 evaluates one such drug design technique in a context of electromechanical design abstractions, and extrapolates its value into other design domains.

Manuscript 3 Conclusions

Manuscript 3 explores a method to facilitate efficient solution space exploration using the structural fingerprinting representation. It is demonstrated that structural fingerprints of electromechanical products are predictive of their functions. Solution candidates are clustered using these fingerprints, and representative fragments are extracted from the functionally distinct cluster of vacuum cleaners. These fragments represent structural backbones of solutions with different functionality. Together these results suggest a feasible method for exploring a solution space of dynamic functionality.

Efficient exploration of the solution space is a major challenge in both molecular design and engineering design. This work evaluates the viability of molecular fingerprinting for the purpose of describing systems poorly understood design domains. One such domain is that of Metal Organic Responsive Frameworks (MORFs), a theoretical type of shapeshifting material that changes shape stochastically in response to light. Creating a computational framework for on-demand invention of new MORFs requires techniques for efficiently exploring the solution space in a way that is predictive of MORF functionality. This study contributes to this goal by demonstrating that structural fingerprinting, which is already known to be a facilitator of efficient solution space exploration, is also predictive of functionality that results from dynamic behavior.

The results demonstrate a correlation between structural fingerprints of eight groups of electromechanical products and those products' shared functions. This correlation provides evidence that structural fingerprints are a viable representation for inferring clusters of systems with distinctly different functionality. This result suggests that fingerprints could be valuable to support the automated design of not only electromechanical systems, but also systems in poorly understood domains – such as MORFs.

Furthermore, in library design tools such as the Design Repository, fingerprints will provide a fast way to search for products based on a desired component substructure without performing costly subgraph searches. The Design Repository also contains functional models, and these too can be represented as graphs. Fingerprints of these functional models will similarly enable efficient search for products with a desired set of function chains. Additionally, this approach is scalable to very large design libraries. All that is required is a tool for consistently generating (1) fingerprints of products in the repository, and (2) fingerprints for substructure search queries.

In automated design tasks wherein a large number of candidate solutions are created, fingerprinting and clustering provide tools to reduce the complexity of evaluation as well as to guide the algorithms that generate solution candidates. Given that functional similarity can be predicted from structural fingerprints, clustering similar products reduces the complexity of two different types of concept evaluation. For evaluation tasks that require expensive computation, representative solutions from each cluster can be evaluated, just as in drug design. For evaluation tasks that require human interpretation, clustering can reduce the size of the search space and facilitate interactive exploration (e.g., as described in [128]). Using fingerprints to support these clustering operations eliminates the need to specify a dictionary of important features while preserving topological information. Additionally, common substructure fragments in these well-performing clusters can be used as seeds for generating additional concept variants in the same solution neighborhood.

While these results suggest the value of a fingerprinting approach to support computational search, they require verification and validation in the MORF design context. In future work, the fingerprinting approach presented here will be applied in the MORF generation framework to catalog feasible candidate structures. These fingerprints will serve as the basis for calculating similarity, forming clusters, and selecting representative candidates for expensive simulations during a search. The most feasible results from each run will be used to inform a predictive QSAR-style model that relates molecule structure to important behavior characteristics in the MORF domain. During usefulness screening, these representations will likewise support similarity screening, clustering, and the eventual visualization of each cluster's behavioral characteristics. Further, different clustering algorithms must be assessed and tuned for their ability to produce meaningfully different groups of candidate structures in this context. Fragment mining from these clusters will create new seeds with known behavior properties in order to generate new candidates with similar behavior. After this framework is implemented it will be possible to conduct experiments with respect to verifying, validating, and tuning the search process.

Manuscript 3 Impact

The results of manuscript 3 demonstrate a viable high-throughput strategy for relating two abstraction layers (in this case, function and form). The easily computed structural information is used to infer less easily computed behavioral information, which supports high-throughput functional screening. This strategy is well validated in drug design, and manuscript 3 demonstrates its value in other domains. Researchers in the area of computational design synthesis will be able to leverage these results to design search algorithms that enable rapid large-scale solution space exploration. Such algorithms are needed to support the development of a rapid digital design and manufacturing infrastructure – especially in the area of complex systems design, where search spaces are very large.

The domain independence of this search algorithm begs the question: in what other ways can this algorithm provide design value? Manuscript 4 addresses this question by extending the high volume screening strategy in manuscript 3 into the domain of conceptual analogy matching.

Manuscript 4 Conclusions

Manuscript 4 presents the Analogy Fingerprinting algorithm for indexing concept maps. Given a fingerprinted library of engineered and natural systems, Analogy Fingerprinting enables fast searching to find the most relevant design analogies in the library. The speed comes directly from the well-understood properties of Bloom filters, while structure mapping theory and the significant results obtained in the validation experiment support the relevance of retrieved analogies using this algorithm. As a consequence, Analogy Fingerprinting can provide designers with an effective option to overcome common analogy-forming shortcomings – such as poorly encoded knowledge, insufficient experience, and difficulty creating mental links across domains [48] – while reaping the benefits of considering a wide range of conceptual solutions. Considering many alternatives is a core principle of conceptual design that is known to improve design outcomes, and it follows that Analogy Fingerprinting represents a fast way for designers to improve their design outcomes.

Manuscript 4 Impact

This manuscript contributes to the development of an intuitive design by analogy method. Novice designers' analogy retrieval abilities are limited because their mental models are poorly organized and narrow in scope. This definition also includes expert designers in cases when specialized knowledge of conceptually distant fields (such as biology) is required to draw a design analogy, and these distant fields can be valuable analogy sources. This dissertation supports using a mixture of weakly typed information (in the form of mixed-abstraction concept maps) and large scale computational matching to support designers with narrow and poorly organized mental models. Additionally, Baya [171] finds that fluid handling of “all types of information” is critical during conceptual design. Unlike a prescriptive framework of design knowledge (e.g., [171]), the Analogy Fingerprinting design method uses concept maps to computationally augment conceptual design analogy search. Thus, a conceptual design method that uses Analogy Fingerprinting is likely to provide valuable computational support during conceptual design.

Based on the results of the Analogy Fingerprinting validation, a revised Analogy Fingerprinting Design methodology might contain the following steps:

1. Generate a concept map of the most important concepts in the problem domain.
2. Insert and delete concepts and relationships from the concept map to model an abstract solution (e.g., remove unwanted functions or components, add new functions, or add new constraints).
3. Use the modified concept map as a query against an existing fingerprint library.
4. Repeat steps 1-3 until a satisfactory quantity of concepts has been collected.

The method could be further extended by implementing a semantic similarity check, which would enable secondary sorting based on surface similarity. Such a technique would provide a method of fine-tuning the analogical distance of results – analogies with high semantic similarity are likely to be from a conceptually similar domain, while analogies with low semantic similarity are likely to be from a conceptually different domain.

Analogy Fingerprinting represents a departure from the spreading activation algorithm used by Liu and Singh to find analogies within ConceptNet [113]. While the spreading activation approach benefits from the weighted edges aggregated from many inputs, it returns matches based on the strength of the overall connections (node weights and number of parallel paths) between two nodes – effectively a measure of similarity rather than analogy. A second matching algorithm is then needed to confirm the mapping of relationship structures. Analogy Fingerprinting forgoes the similarity search aspect of the analogy search, instead keeping each individual system description separate. The separate analogy fingerprints of each system description can be quickly assessed for their relationship structures' mappability onto any given query.

Key Contributions

To summarize, this dissertation presents three major thrusts focused on understanding design analogies, preparing analogy candidate information, and matching analogy candidates to design problems. Within these thrusts, this dissertation contributes to a greater understanding of

- the abstractions used by designers during conceptual design.
- the use of human computation to support conceptual design activities, specifically with respect to information gathering.

- large scale solution screening using a variety of weakly typed abstractions.

All of these contributions are made specifically within the context of design by analogy.

Impact on Design

The very first manuscript (manuscript 0) addresses the impact of function-based research on education and industry. A key takeaway from this manuscript is that in order to impact design practice, research outcomes must possess simplicity, flexibility, and direct applicability to a practical problem. The research presented in this dissertation culminates in the Analogy Fingerprinting method, which demonstrates all of these crucial aspects.

Simplicity

Analogy Fingerprinting enables a designer to find solution analogies using the easy-to-create descriptive framework of concept mapping.

Concept maps were created with the goal of documenting meaningful learning of science concepts in students [172]. They are also used to promote meaningful learning in new domains [173] – students document their own understanding of a domain, which allows the students and their teachers to examine and modify their knowledge structures. Concept maps are routinely created for domains in which the creator is not an expert, which supports concept mapping as an easy-to-use tool for describing design problems.

The simplicity of Analogy Fingerprinting comes from two places: (1) its usage of these easily created concept map representations as the human generated input, and (2) the direct computability of these concept maps to perform analogy search. As a consequence, the designer’s workflow consists of just two steps: (1) create a concept map of the design problem and (2) select feasible analogical solutions from the Analogy Fingerprinting search results. Given these factors Analogy Fingerprinting is simple enough to be used by designers of a wide variety of skills levels and knowledge sets.

Flexibility

Flexibility reflects a common characteristic of successful design tools – such as Failure Modes and Effects Analysis – that require very little training and can be adapted to a wide array of specific purposes. Flexibility is a strength of the Analogy Fingerprinting design method, especially with respect to the way in which designers formulate analogy queries. The concept mapping formalism has very few unbreakable rules, and the content of each concept map need only be restricted to natural language (i.e., symbolic concepts must be described in terms that can be communicated through speech – a very modest restriction). As a consequence, any designer should be able to create a concept map to address their problem domain with little training. This concept map can then be used to automatically generate valuable feedback about alternative conceptual solutions. Due to the low effort required to generate a list of alternative concepts, and the domain independence of the approach, the method is flexible enough to be applied to a wide variety of design processes.

More generally, the flexibility allowed by concept mapping enables an alternative approach to design. Traditional design approaches emphasize the importance of making design decisions independently of their expected implementation. This practice of maintaining solution independence during early design stages (e.g., requirements generation and functional decomposition) improves the likelihood of generating a higher quantity and quality of concepts [38, 160]. However, many designers exhibit a tendency to skip functional analysis exactly because most of the results will be poor, in spite of the fact that it increases the likelihood of finding better solutions [160]. Compounding this issue is the fact that many potential abstract solutions can be perceived as obviously violating physical constraints, which prevents their consideration. It can be challenging to consider this coupling between abstractions without allowing implementation details (e.g., component selection) to dominate abstract reasoning (e.g., functional analysis). As a consequence, it is valuable to provide methods that allow designers the freedom to maintain solution independence while considering multiple layers of abstraction.

Analogy Fingerprinting enables a designer to maintain this solution independence while still considering multiple types of abstraction at the same time. For instance, Analogy Fingerprinting enables search for analogous systems that contain both functional and compositional correspondents. If a design problem requires that a system must have the overall functionality of X, and the system must contain two subsystems that have subfunctions Y and Z, this information can be captured in a concept map and searched according to its Analogy Fingerprint. A successful search results in a design concept that matches these general functional and implementation details. This allows the designer to maintain solution abstraction in a concept search while increasing the likelihood of finding a high quality match. Assuming a sufficiently large concept library, the designer need not have any prior knowledge of the potential search results in order to find a match. This mitigates bias in the designer's reasoning during the early stages of design – when the analysis of alternative concepts matters most – while providing a basic framework to reduce the amount of iteration required to arrive at a high quality solution concept.

Application to an Existing Problem

This work applies directly to the problem of concept development and selection – a process that can lead to costly mistakes if not adequately performed. For example, a Government Accountability Office study of 32 Department of Defense acquisition projects found that most of them did not conduct a "robust assessment of alternatives." Within this sample, projects that did not examine a wide range of concepts were more likely to experience high cost or schedule growth [174]. It is because of this effect that many standard engineering design texts teach the importance of concept development and selection (e.g., [38, 40, 160, 175]). The work in this dissertation provides a directly applicable solution to one aspect of this problem – it enables designers to consider many alternatives for relatively low cognitive effort. Additionally, it enables a design team to consider alternative concepts that fall outside the realm of the team's breadth of expertise. In doing so, the research presented in this dissertation enables the creation of tools and methods that lead to improved design outcomes through the consideration of more alternatives. Given a population of design teams that uses such a tool – given widely accepted importance of conceptual design – it is likely that such teams will experience reduced cost and schedule overruns, and thus improved design outcomes.

Bibliography

[1] D. R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*. New York: Basic Books, Inc., 1979.

[2] L. J. Ball, T. C. Ormerod, and N. J. Morley, "Spontaneous Analogising in Engineering Design: A Comparative Analysis of Experts and Novices," *Design Studies*, vol. 25, pp. 495-508, 2004.

[3] D. Gentner, "Structure-Mapping: A Theoretical Framework for Analogy," *Cognitive Science,* vol. 7, pp. 155-170, 1983.

[4] K. J. Holyoak and R. G. Morrison, *The Oxford handbook of thinking and reasoning*: Oxford University Press, 2013.

[5] D. A. McAdams and K. L. Wood, "A Quantitative Similarity Metric for Design-by-Analogy," *Journal of Mechanical Design,* vol. 124, pp. 173-182, June 2002.

[6] K. R. Poppa, "Theory and application of vector space similarity measures in computer assisted conceptual design," Ph.D., Mechanical Engineering, Oregon State University, Corvallis, 2011.

[7] D. G. Ullman, *The Mechanical Design Process*. New York, NY: McGraw-Hill, 1997.

[8] J. A. Collins, B. T. Hagan, and H. M. Bratt, "The Failure-Experience Matrix: A Useful Design Tool," *Journal of Engineering for Industry,* vol. 98, pp. 1074-1079, 1976.

[9] G. Pahl and W. Beitz, *Engineering Design: A Systematic Approach*. London: Design Council, 1984.

[10] M. Hundal, "A Systematic Method for Developing Function Structures, Solutions and Concept Variants," *Mechanism and Machine Theory,* vol. 25, pp. 243-256, 1990.

[11] G. Altshuller, "Creativity as an exact science," 1984.

[12] J. Malmqvist, R. Axelsson, and M. Johansson, "A Comparative Analysis of the Theory of Inventive Problem Solving and the Systematic Approach of Pahl and Beitz," in *Proceedings of the 1996 ASME Design Engineering Technical Conferences*, Irvine, CA, 1996.

[13] V. Hubka and W. Ernst Eder, *Theory of Technical Systems*. Berlin: Springer-Verlag, 1984.

[14] S. Szykman, R. Sriram, and S. Smith, "Proceedings of the NIST Design Repository Workshop," National Institute of Standards and Technology, Gaithersburg, MDNovember 1996.

[15] J. Hirtz, R. Stone, D. McAdams, S. Szykman, and K. Wood, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts," *Research in Engineering Design,* vol. 13, pp. 65-82, 2002.

[16] C. Sen, B. W. Caldwell, J. D. Summers, and G. M. Mocko, "Evaluation of the functional basis using an information theoretic approach," *AI EDAM (Artificial Intelligence for Engineering Design, Analysis and Manufacturing),* vol. 24, p. 87, 2010.

[17] R. Stone and K. Wood, "Development of a Functional Basis for Design," *Journal of Mechanical Design,* vol. 122, pp. 359-370, 2000.

[18] S. Ahmed and K. Wallace, "Evaluating a Functional Basis," presented at the DETC/CIE, Chicago, Illinois, 2003.

[19] M. A. Kurfman, J. Rajan, R. B. Stone, K. L. Wood, and M. E. Stock, "Experimental Sudies Assessing the Repeatability of a Functional Modeling Derivation Method," *Journal of Mechanical Design,* vol. 125, pp. 682-693, 2003.

[20] T. Kurtoglu, M. Campbell, C. Bryant, R. Stone, and D. McAdams, "Deriving a Component Basis for Computational Functional Synthesis," in *International Conference on Engineering Design, ICED'05*, Melbourne, Australia, 2005.

[21] M. Kurfman, R. Stone, M. Van Wie, K. Wood, and K. Otto, "Theoretical Underpinnings of Functional Modeling: Preliminary Experimental Studies," in *Proceedings of DETC2000*, Balitmore, MD., 2000.

[22] R. L. Nagel, R. S. Hutcheson, R. Stone, D. McAdams, and J. Donndelinger, "Function Design Framework (FDF): Integrated Process and Function Modeling for Complex System Design," presented at the ASME International Design Engineering Technical Conferences, Brooklyn, NY, 2008.

[23] R. S. Hutcheson, D. A. McAdams, R. B. Stone, and I. Y. Tumer, "Function-based behavioral modeling," in *ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2007.

[24] R. S. Hutcheson, D. A. McAdams, R. B. Stone, and I. Y. Tumer, "Effect of Model Element Fidelity Within a Complex Function-Based Behavioral Model," in *2008 ASME Computers and Information in Engineering Conference, Integrated Systems Engineering Symposium, IDETC/CIE2008.*, New York City, NY, 2008.

[25] R. L. Nagel, R. B. Stone, and D. A. McAdams, "A Process Modeling Methodology for Automation of Manual and Time Dependent Processes," in *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2006.

[26] R. Nagel, J. L. Greer, R. B. Stone, and D. A. McAdams, "FunctionCAD: A Functional Modeling Application Based on the Function Design Framework," presented at the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, San Diego, CA, 2009.

[27] I. Y. Tumer, R. B. Stone, and D. G. Bell, "Requirements for a Failure Mode Taxonomy for Use in Conceptual Design," in *International Conference on Engineering Design*, Stockholm Sweden, 2003.

[28] S. J. Uder, R. B. Stone, and I. Y. Tumer, "Failure Analysis in Subsystem Design for Space Missions," in *ASME Design Engineering Technical Conferences, Design Theory and Methodology*, Salt Lake City, Utah, 2004.

[29] R. Stone, I. Tumer, and M. Van Wie, "The Function Failure Design Method," *Journal of Mechanical Design*, vol. 127, pp. 397-407, 2004.

[30] K. Grantham Lough, R. B. Stone, and I. Y. Tumer, "The Risk in Early Design Method (RED)," *Journal of Engineering Design*, vol. 18, 2007.

[31] S. Sierla, I. Tumer, N. Papakonstantinou, K. Koskinen, and D. Jensen, "Early integration of safety to the mechatronic system design process by the functional failure identification and propagation framework," *Mechatronics*, vol. 22, pp. 137-151, 2012.

[32] R. A. Roberts, I. Y. Tumer, R. B. Stone, and A. F. Brown, "A Function-based Exploration of JPL's Problem/Failure Reporting Database," in *International Mechanical Engineering Congress and Expo*, Washington, D.C. U.S.A., 2003.

[33] R. Nagel, J. L. Greer, R. B. Stone, and D. A. McAdams, "Fault Propagation and Sensitivity Analysis to Support Counterterrorism Activities," presented at the 19th Annual INCOSE International Symposium, Singapore, 2009.

[34] T. Kurtoglu, S. B. Johnson, E. Barszcz, J. R. Johnson, and P. I. Robinson, "Integrating system health management into the early design of aerospace systems using functional fault analysis," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, 2008, pp. 1-11.

[35] S. Uckun, T. Kurtoglu, P. Bunus, I. Y. Tumer, C. Hoyle, and D. Musliner, "Model-Based Systems Engineering for the Design and Dvelopment of Complex Aerospace Systems," in *SAE Aerotech Conference*, 2011.

[36] S. Oman, I. Y. Tumer, B. Gilchrist, R. Stone, A. Nix, and C. Rebhuhn, "Towards a Repository of Innovative Products to Enhance Engineering Creativity Education," in *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Chicago, Illinois, 2012.

[37] G. Pahl, W. Beitz, J. Feldhusen, and K. H. Grote, *Engineering Design: A Systematic Approach*, 3rd ed.: Springer Verlag, 2007.

[38] D. G. Ullman, *The Mechanical Design Process*, 4th ed. Boston: McGraw-Hill, 2010.

[39] K. Otto and K. Wood, *Product Design: Techniques in Reverse Engineering, Systematic Design, and New Product Development*. New York: Prentice-Hall, 2001.

[40] K. T. Ulrich and S. D. Eppinger, *Product Design and Development*, 3rd ed. Boston, MA: McGraw-Hill/Irwin, 2004.

[41] S. Ahmed and K. Wallace, "Evaluating a Functional Basis," in *Proceedings of DETC'03*, Chicago, IL, 2003.

[42] T. Kurtoglu, M. I. Campbell, C. B. Arnold, R. B. Stone, and D. A. Mcadams, "A component taxonomy as a framework for computational design synthesis," *Journal of Computing and Information Science in Engineering*, vol. 9, p. 011007, 2009.

[43] H. Casakin and G. Goldschmidt, "Expertise and the Use of Visual Analogy: Implications for Design Education," *Design Studies*, vol. 20, pp. 153-175, 1999.

[44] B. T. Christensen and C. D. Schunn, "The Relationship of Analogical Distance to Analogical Function and Pre-Inventive Structure: The Case of Engineering Design," *Memory and Cognition*, vol. 35, pp. 29-38, 2007.

[45] C. M. Eckert, M. Stacey, and C. Earl, "References to Past Designs," in *Studying Designers'05*, University of Provence, Aix-en-Provence, France, 2005, pp. 3-21.

[46] P. Leclercq and A. Heylighen, "5,8 Analogies Per Hour," in *Artificial Intelligence in Design'02*, J. S. Gero, Ed., ed Dordrecht, the Netherlands: Kluwer Academic Publishers, 2002, pp. 285-303.

[47] M. L. Gick and K. J. Holyoak, "Analogical Problem Solving," *Cognitive Psychology*, vol. 12, pp. 306-355, 1980.

[48] J. L. Kolodner, "Educational Implications of Analogy: A View from Case-Based Reasoning," *American Psychologist*, vol. 52, pp. 57-66, 1997.

[49] W. J. Gordon, "Synectics: The development of creative capacity," 1961.

[50] N. Cross, *Engineering design methods: strategies for product design*: John Wiley & Sons, 2008.

[51] B. Falkenhainer, K. D. Forbus, and D. Gentner, "The Structure-Mapping Engine: Algorithm and Examples," *Artificial Intelligence*, vol. 41, pp. 1-63, 1989/90.

[52] J. Clement, "Observed Methods for Generating Analogies in Scientific Problem Solving," *Cognitive Science*, vol. 12, pp. 563-586, 1988.

[53] J. Clement, *Creative Model Construction in Scientists and Students: The Role of Imagery, Analogy, and Mental Stimulation*. Dordrecht: Springer, 2008.

[54] A. Goel and S. Bhatta, "Design Patterns: An Unit of Analogical Transfer in Creative Design," *Advanced Engineering Informatics*, vol. 18, pp. 85-94, 2004.

[55] A. Goel, "Design, analogy and creativity," *IEEE Expert Intelligent Systems and Their Applications*, vol. 12, pp. 62-70, 1997.

[56] M. Maher and M. Balachandran, "Case-Based Reasoning in Design," 1995.

[57] M. Gick and K. J. Holyoak, "Schema Induction and Analogical Transfer," *Cognitive Psychology*, vol. 15, pp. 1-38, 1983.

[58] S. Bhatta and A. Goel, "Learning Generic Mechanisms for Innovative Design Adaptation," *Journal of Learning Sciences*, vol. 6, pp. 367-396, 1997.

[59] T. W. Griffith, N. J. Nersessian, and A. K. Goel, "The role of generic models in conceptual change," in *Proceedings of the eighteenth annual conference of the cognitive science society*, 1996, pp. 312-317.

[60] T. Griffith, N. Nersessian, and A. Goel, "Function-follows-form: generative modeling in scientific reasoning," in *Proc. 22nd Cognitive Science Conference*, 2000, pp. 196-201.

[61] N. Nersessian, *Creating scientific concepts*: MIT Press, 2008.

[62] S. S. Vattam, M. E. Helms, and A. K. Goel, "Compound analogical design: interaction between problem decomposition and analogical transfer in biologically inspired design," in *Design Computing and Cognition'08*, ed: Springer, 2008, pp. 377-396.

[63] J. S. Gero and U. Kannengiesser, "The Situated Function - Behaviour - Structure Framework," *Artificial Intelligence in Design*, pp. 89-104, 2002.

[64] M. Dinar, J. Shah, P. Langley, G. Hunt, and E. Campana, "A Structure for Representing Problem Formulation in Design," in *Proceedings of the 18th International Conference on Engineering Design (ICED11)*, 2011.

[65] A. Danielescu, M. Dinar, C. MacLellan, J. J. Shah, and P. Langley, "The Structure of Creative Design: What Problem Maps Can Tell Us About Problem Formulation and Creative Designers," presented at the DETC/CIE, Chicago, IL, 2012.

[66] C. Sen, "A formal representation of mechanical functions to support physics-based computational reasoning in early mechanical design," PhD, Mechanical Engineering, Clemson University, 2011.

[67] S. Kadar-Cabelli, "Purpose-Directed Analogy," in *Seventh annual conference of Cognitive Science Society*, 1985.

[68] M. Helms and A. K. Goel, "The Four-Box Method of Problem Specification and Analogy Evaluation in Biologically Inspired Design," in *ASME 2014 International Design Engineering*

Technical Conferences and Computers and Information in Engineering Conference, 2014, pp. V007T07A005-V007T07A005.

[69] A. Chakrabarti, P. Sarkar, B. Leelavathamma, and B. Nataraju, "A Functional Representation for Aiding Biomimetic and Artificial Inspiration of New Ideas," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* vol. 19, pp. 113-132, 2005.

[70] J. Summers, D. Maxwell, C. Camp, and A. Butler, "Features as an Abstraction for designer convenience in the Design of Complex Products," in *Proceedings of DETC-2000*, Baltimore, MD, 2000.

[71] W. C. Regli and V. A. Cicirello, "Managing Digital Libraries for Computer-Aided Design," *Computer Aided Design,* vol. 32, pp. 119-132, 2000.

[72] M. Bohm, R. Stone, and S. Szykman, "Enhancing Virtual Product Representations for Advanced Design Repository Systems," *Journal of Computer and Information Science in Engineering,* vol. 5, pp. 360-372, 2005.

[73] C. R. Bryant, D. A. McAdams, R. B. Stone, T. Kurtoglu, and M. Campbell, "A Validation Study of an Automated Concept Generator Design Tool," presented at the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Philadelphia, PA, 2006.

[74] M. R. Bohm, J. P. Vucovich, and R. B. Stone, "Using a Design Repository to Drive Concept Generation," *Journal of Computer and Information Science in Engineering,* vol. 8, 2008.

[75] S. Shooter, T. Simpson, S. Kumara, R. Stone, and J. Terpenny, "Toward a Multi-Agent Information Management Infrastructure for Product Family Planning and Mass Customisation," *International Journal for Mass Customisation,* vol. 1, pp. 134-155, 2005.

[76] M. Bohm, R. Stone, T. Simpson, and E. Steva, "Introduction of a Data Schema: To Support a Design Repository," *Computer-Aided Design,* vol. 40, pp. 801-811, 2008.

[77] M. R. Bohm, R. B. Stone, T. W. Simpson, and E. D. Steva, "Introduction of a Data Schema: The Inner Workings of a Design Repository," in *ASME International Design Engineering Technical Conferences*, Philadelphia, PA, 2006.

[78] M. Aurisicchio, R. H. Bracewell, and K. M. Wallace, "Characterising in Detail the Information Requests of Engineering Designers," presented at the IDETC/CIE, Philadelphia, Pennsylvania, 2006.

[79] T. Griffith, N. Nersessian, and A. Goel, "The Role of Generic Models in Conceptual Change," in *18th Cognitive Science Conference*, San Diego, 1996.

[80] T. Griffith, N. Nersessian, and A. Goel, "Function-Follows-Form: Generative Modeling in Scientific Reasoning," in *22nd Cognitive Science Conference*, 2000.

[81] V. Baya, "Information Handling Behaviour of Designers during Conceptual Design: Three Experiments," PhD, Mechanical Engineering, Stanford University, 1996.

[82] J. Cohen, "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement,* vol. 20, pp. 37-46, 1960.

[83] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics,* pp. 159-174, 1977.

[84] P. Leclercq and A. Heylighen, "5,8 Analogies per Hour," in *Artificial Intelligence in Design'02*, 2002, pp. 285-303.

[85] C. D. Manning and H. Schütze, *Foundation of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999.

[86] M. L. Mauldin, "Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition," in *12th National Conference on Artificial Intelligence (AAAI '94)*, Seattle, WA, 1994, pp. 16-21.

[87] J. Linsey, K. Wood, and A. Markman, "Increasing Innovation: Presentation and Evaluation of the Wordtree Design-by-Analogy Method," ed New York, 2008.

[88] K. Fu, J. Chan, J. Cagan, K. Kotovsky, C. Schunn, and K. Wood, "The Meaning of "Near" and "Far": The Impact of Structuring Design Databases and the Effect of Distance of Analogy on Design Output," *Journal of Mechanical Design,* vol. 135, 2013.

[89] K. Poppa, R. Arlitt, and R. Stone, "An Approach to Automated Concept Generation through Latent Semantic Indexing," in *Proceedings of the 2013 Industrial and Systems Engineering Research Conference*, San Juan, Puerto Rico, 2013.

[90] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, pp. 39-41, 1995.

[91] S. S. Vattam, A. K. Goel, S. Rugaber, C. E. Hmelo-Silver, R. Jordan, S. Gray, *et al.*, "Understanding complex natural systems by articulating structure-behavior-function models," *Educational Technology & Society*, vol. 14, pp. 66-81, 2011.

[92] J. K. Nagel, "A Thesaurus for Bioinspired Engineering Design," in *Biologically Inspired Design*, ed: Springer, 2014, pp. 63-94.

[93] S. Vattam, B. Wiltgen, M. Helms, A. K. Goel, and J. Yen, "DANE: fostering creativity in and through biologically inspired design," in *Design Creativity 2010*, ed: Springer, 2010, pp. 115-122.

[94] B. Wiltgen, S. Vattam, M. Helms, A. K. Goel, and J. Yen, "Learning Functional Models of Biological Systems for Biologically Inspired Design," in *11th IEEE International Conference on Advanced Learning Technologies (ICALT)*, Athens, GA, 2011, pp. 355-357.

[95] R. Arlitt, B. O'Halloran, J. Novak, R. Stone, and I. Tumer, "Applying Designer Feedback to Generate Requirements for an Intuitive Biologically Inspired Design Tool," in *International Mechanical Engineering Congress and Exposition*, Houston, TX, 2012.

[96] L. H. Shu, "A natural-language approach to biomimetic design," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 24, pp. 507-519, 2010.

[97] H. Cheong and L. Shu, "Retrieving causally related functions from natural-language text for biomimetic design," *Journal of Mechanical Design*, vol. 136, p. 081008, 2014.

[98] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for information Science and Technology*, vol. 60, pp. 538-556, 2009.

[99] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using hard AI problems for security," in *Advances in Cryptology, EUROCRYPT 2003*, ed Warsaw, Poland: Springer, 2003, pp. 294-311.

[100] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "recaptcha: Human-based character recognition via web security measures," *Science*, vol. 321, pp. 1465-1468, 2008.

[101] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, *et al.*, "Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey," *Monthly Notices of the Royal Astronomical Society*, vol. 389, pp. 1179-1189, 2008.

[102] M. V. Marc Mercuri, Tim Harris, Cassie Bowman. (2010). *Be a Martian*. Available: http://beamartian.jpl.nasa.gov/

[103] L. Von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004, pp. 319-326.

[104] L. Von Ahn, M. Kedia, and M. Blum, "Verbosity: a game for collecting common-sense facts," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, Montreal, Quebec, Canada, 2006, pp. 75-78.

[105] E. L. Law, L. Von Ahn, R. B. Dannenberg, and M. Crawford, "TagATune: A Game for Music and Sound Annotation," in *International Conference on Music Information Retrieval (ISMIR'07)*, Vienna, Austria, 2007, pp. 361-364.

[106] N. Seemakurty, J. Chu, L. Von Ahn, and A. Tomasic, "Word sense disambiguation via human computation," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, Washington, DC, 2010, pp. 60-63.

[107] A. J. Quinn and B. B. Bederson, "Human computation: a survey and taxonomy of a growing field," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, BC, Canada, 2011, pp. 1403-1412.

[108] L. Von Ahn, "Human Computation," Computer Science, Carnegie Mellon University, 2005.

[109] S. Cooper, A. Treuille, J. Barbero, A. Leaver-Fay, K. Tuite, F. Khatib, *et al.*, "The challenge of designing scientific discovery games," in *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, Monterey, CA, USA, 2010, pp. 40-47.

[110] F. Khatib, F. DiMaio, S. Cooper, M. Kazmierczyk, M. Gilski, S. Krzywda, *et al.*, "Crystal structure of a monomeric retroviral protease solved by protein folding game players," *Nat Struct Mol Biol*, vol. 18, pp. 1175-1177, 2011.

[111] C. B. Eiben, J. B. Siegel, J. B. Bale, S. Cooper, F. Khatib, B. W. Shen, *et al.*, "Increased Diels-Alderase activity through backbone remodeling guided by Foldit players," *Nat Biotech*, vol. 30, pp. 190-192, 2012.

[112] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu, "Open Mind Common Sense: Knowledge acquisition from the general public," in *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, Irvine, CA, 2002, pp. 1223-1237.

[113] H. Liu and P. Singh, "ConceptNet, A practical commonsense reasoning tool-kit," *BT Technology Journal,* vol. 22, pp. 211-226, October 2004.

[114] H. Liu and P. Singh, "Commonsense reasoning in and over natural language," in *Knowledge-based intelligent information and engineering systems*, ed Berlin - Heidelberg: Springer, 2004, pp. 293-306.

[115] R. Speer, C. Havasi, and H. Lieberman, "AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge," in *AAAI*, Chicago, Illinois, USA, 2008, pp. 548-553.

[116] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, Berkeley, CA, USA, 1999, pp. 50-57.

[117] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[118] S. M. Sullivan and J. R. Maddock, "Bacterial division: Finding the dividing line," *Current Biology,* vol. 10, pp. R249-R252, 2000.

[119] ScienceDirect [Online]. Available: http://www.sciencedirect.com/

[120] M. W. Glier, D. A. McAdams, and J. S. Linsey, "An Experimental Investigation of Analogy Formation using the Engineering-to-Biology Thesaurus," in *IDETC/CIE*, Portland, Oregon, 2013.

[121] M. Davies, "The Corpus of Contemporary American English as the first reliable monitor corpus of English," *Literary and linguistic computing,* vol. 25, pp. 447-464, 2010.

[122] R. Arlitt, F. Berthelsdorf, S. Immel, and R. Stone, "Using Human Computation to Assist Biologically Inspired Design: Evaluating a Game-with-a-Purpose," presented at the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Buffalo, NY, 2014.

[123] . *Design by Analogy to Nature Engine (DANE)*. Available: http://dilab.cc.gatech.edu/dane/

[124] K. D. Forbus, D. Gentner, and K. Law, "MAC/FAC: A Model of Similarity-Based Retrieval," *Cognitive Science,* vol. 19, pp. 141-205, 1995.

[125] T. Kurtoglu and M. I. Campbell, "Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping," *Journal of Engineering Design,* vol. 20, pp. 83-104, 2009/02/01 2009.

[126] J. Patel and M. I. Campbell, "An Approach to Automate and Optimize Concept Generation of Sheet Metal Parts by Topological and Parametric Decoupling," *Journal of Mechanical Design,* vol. 132, pp. 051001-051001, 2010.

[127] T. Kurtoglu, M. I. Campbell, J. Gonzalez, C. R. Bryant, R. B. Stone, and D. A. McAdams, "Capturing empirically derived design knowledge for creating conceptual design configurations," in *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2005, pp. 249-257.

[128] M. Kumar and M. Campbell, "Organizing a Design Space of Disparate Component Topologies," in *Design Computing and Cognition, 2010*, J. S. Gero, Ed., ed: Springer Netherlands, 2010, pp. 465-485.

[129] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: a review and new directions," *Engineering Management, IEEE Transactions on,* vol. 48, pp. 292-306, 2001.

[130] M. A. Johnson and G. M. Maggiora, "Concepts and applications of molecular similarity," 1990.

[131] Y. C. Martin, J. L. Kofron, and L. M. Traphagen, "Do structurally similar molecules have similar biological activity?," *Journal of medicinal chemistry,* vol. 45, pp. 4350-4358, 2002.

[132] J. Willett, *Similarity and clustering in chemical information systems*: John Wiley & Sons, Inc., 1987.

[133] R. D. Brown and Y. C. Martin, "Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection," *Journal of chemical information and computer sciences,* vol. 36, pp. 572-584, 1996.

[134] G. M. Maggiora, "On outliers and activity cliffs why QSAR often disappoints," *Journal of chemical information and modeling,* vol. 46, pp. 1535-1535, 2006.

[135] S. Wold, M. Sjöström, and L. Eriksson, "PLS-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems,* vol. 58, pp. 109-130, 2001.

[136] Q. S. Du, R. B. Huang, Y. T. Wei, Z. W. Pang, L. Q. Du, and K. C. Chou, "Fragment-based quantitative structure-activity relationship (FB-QSAR) for fragment-based drug design," *Journal of computational chemistry,* vol. 30, pp. 295-304, 2009.

[137] D. R. Lowis, "HQSAR: a new, highly predictive QSAR technique," *Tripos Technical Notes,* vol. 1, pp. 1-10, 1997.

[138] J.-L. Faulon and A. Bender, *Handbook of chemoinformatics algorithms*: CRC Press, 2010.

[139] "Fingerprints - Screening and Similarity," in *Daylight Theory Manual*, ed: Daylight Chemical Information Systems, 2011.

[140] J. W. Raymond and P. Willett, "Effectiveness of graph-based and fingerprint-based similarity measures for virtual screening of 2D chemical structure databases," *Journal of computer-aided molecular design,* vol. 16, pp. 59-71, 2002.

[141] T. Kurtoglu, M. Campbell, C. Bryant, R. Stone, and D. McAdams, "A Component Taxonomy as a Framework for Computational Design Synthesis," *Journal of Computers and Information Science in Engineering,* vol. 9, 2008.

[142] Y. C. Martin and S. Muchmore, "Beyond QSAR: lead hopping to different structures," *QSAR & Combinatorial Science,* vol. 28, pp. 797-801, 2009.

[143] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM,* vol. 13, pp. 422-426, 1970.

[144] P. Willett, "Similarity searching using 2D structural fingerprints," in *Chemoinformatics and Computational Chemical Biology*, ed: Springer, 2011, pp. 133-158.

[145] G. M. Downs and J. M. Barnard, "Clustering methods and their uses in computational chemistry," *Reviews in computational chemistry,* vol. 18, pp. 1-40, 2002.

[146] J. W. Raymond, C. J. Blankley, and P. Willett, "Comparison of chemical clustering methods using graph-and fingerprint-based similarity measures," *Journal of Molecular Graphics and Modelling,* vol. 21, pp. 421-433, 2003.

[147] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association,* vol. 58, pp. 236-244, 1963.

[148] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *Computers, IEEE Transactions on,* vol. 100, pp. 1025-1034, 1973.

[149] H. Morgan, "The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service," *Journal of Chemical Documentation,* vol. 5, pp. 107-113, 1965.

[150] W. W. Peterson and D. T. Brown, "Cyclic codes for error detection," *Proceedings of the IRE,* vol. 49, pp. 228-235, 1961.

[151] P. Koopman. (2014). *Best CRC Polynomials*. Available: http://users.ece.cmu.edu/~koopman/crc/

[152] P. Willett, J. M. Barnard, and G. M. Downs, "Chemical similarity searching," *Journal of chemical information and computer sciences,* vol. 38, pp. 983-996, 1998.

[153] N. S. Ketkar, L. B. Holder, and D. J. Cook, "Subdue: Compression-based frequent pattern discovery in graph data," in *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, 2005, pp. 71-76.

[154] J. D. Novak and A. J. Cañas, "The theory underlying concept maps and how to construct and use them," Florida Institute for Human and Machine Cognition Pensacola Fl2008.

[155] R. M. Arlitt, S. R. Immel, F. A. Berthelsdorf, and R. B. Stone, "The Biology Phenomenon Categorizer: A Human Computation Framework in Support of Biologically Inspired Design," *Journal of Mechanical Design,* vol. 136, p. 111105, 2014.

[156] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *Journal of Chemical Information and Computer Sciences,* vol. 28, pp. 31-36, 1988/02/01 1988.

[157] P. Jaccard, "THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1," *New Phytologist,* vol. 11, pp. 37-50, 1912.

[158] P. F. Russell and T. R. Rao, "On Habitat and Association of Species of Anopheline Larvae in South-eastern Madras," *Journal of the Malaria Institute of India,* vol. 3, pp. 153-178 pp., 1940.

[159] J. D. Holliday, C. Hu, and P. Willett, "Grouping of coefficients for the calculation of inter-molecular similarity and dissimilarity using 2D fragment bit-strings," *Combinatorial chemistry & high throughput screening,* vol. 5, pp. 155-166, 2002.

[160] K. Otto and K. L. Wood, *Product Design: Techniques in Reverse Engineering, Systematic Design, and New Product Development*. New York, New York: Prentice-Hall, 2001.

[161] J. J. Shah, N. VARGAS‐ HERNANDEZ, J. D. Summers, and S. Kulkarni, "Collaborative Sketching (C‐ Sketch)—An idea generation technique for engineering design," *The Journal of Creative Behavior,* vol. 35, pp. 168-198, 2001.

[162] J. Wycoff and B. Trade, "Mindmapping: Your personal guide to exploring creativity and problem-solving," 1991.

[163] K. Schwaber and J. Sutherland, "The scrum guide," *Scrum Alliance,* 2011.

[164] M. W. Maier and E. Rechtin, *The Art of Systems Architecting*, 2nd ed.: CRC Press, 2000.

[165] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and practice of bloom filters for distributed systems," *Communications Surveys & Tutorials, IEEE,* vol. 14, pp. 131-155, 2012.

[166] M. Mitzenmacher and E. Upfal, "Bloom Filters," in *Probability and computing: Randomized algorithms and probabilistic analysis*, ed: Cambridge University Press, 2005, pp. 107-112.

[167] (2015). *AskNature*. Available: http://www.asknature.org/

[168] S. S. Vattam, M. E. Helms, and A. K. Goel, "Compound Analogical Design: Interaction between Problem Decomposition and Analogical Transfer in Biologically Inspired Design," presented at the Design Computing and Cognition, Georgia Institute of Technology, Atlanta, USA 2008.

[169] C. D. Manning, P. Raghavan, and H. Schütze, "Evaluation in information retrieval," in *Introduction to information retrieval*. vol. 1, ed Cambridge, MA: Cambridge University Press, 2008.

[170] T. N. Reid, E. F. MacDonald, and P. Du, "Impact of Product Design Representation on Customer Judgment," *Journal of Mechanical Design,* vol. 135, p. 091008, 2013.

[171] V. Baya, "Information Handling Behavior of Designers During Conceptual Design: Three Experiments," PhD, Mechanical Engineering, Stanford University, 1996.

[172] J. D. Novak and D. Musonda, "A twelve-year longitudinal study of science concept learning," *American Educational Research Journal,* vol. 28, pp. 117-153, 1991.

[173] J. D. Novak and A. J. Cañas, "The theory underlying concept maps and how to construct and use them," 2008.

[174] "Defense Acquisitions: Many Analyses of Alternatives Have Not Provided a Robust Assessment of Weapon System Options," G. A. Office, Ed., ed, 2009.

[175] G. Pahl and W. Beitz, *Engineering Design: A Systematic Approach*: Springer Verlag, 1996.