Symbol emergence in design

Abstract

Identifying a standard vocabulary and ontology is viewed as an important task for engineering design. While a number of high-level ontologies have been proposed, these are difficult to ground in terms of actual design instances, and manual definitions of the symbols are often incomplete and difficult to maintain. As an alternative, we propose an "infant designer" paradigm which abstracts patterns for the "functionally feasible regions" (FFR) while evaluating many individual configurations in the design space. These learned FFR patterns (which may arise due to minimal levels of functional acceptability, or from optimization) often embody dependency relationships among the design parameters, i.e. the good designs lie along lower-dimensional manifolds in the design parameter space. We show how such manifolds exist in several design situations; each combination of the original design parameters may be thought of as a "chunk"; the space of these chunks models only the "good designs". Next, we show how the patterns defined based on these chunks constitute image schemas, which may be implicit (e.g. the pattern for an FFR), or explicit (where the relationship is observable). These patterns or image schemas are incipient semantic model leading to symbols. We present examples of how such image schemas are arrived at in different design classes, and also demonstrate how some of them may be similar, or even invariant under design change.

1 Efforts towards standardizing the design vocabulary

Evolving a standardized vocabulary for design has emerged as an important focus in engineering design with a need for communicating between differing systems and design groups. Possible applications include developing design repositories [Bohm *et al.*, 2005; Nanda *et al.*, 2007], computer assisted conceptual design [Gorti and Sriram, 1996; Campbell *et al.*, 2000; Kurtoglu *et al.*, 2005], [Chakrabarti *et al.*, 2005; Gero and Fujii, 2000], etc. It is clear that vocabularies are structured, that is there are considerable relations between terms. Often, this is viewed as an ontology or as a structured relationship that captures a part of the semantics of these terms. One popular view of the engineering system considers the flow of energy, information, etc, and proceeds downward into detailed design. With its roots in value engineering ideas from the 1940s, these notions were seeded by the analysis in Pahl and Beitz [Pahl and Beitz, 1996] and a particularly influential study by Welch and Dixon [Richard and Dixon, 1994], leading to modern ontological models like the widely used *functional basis* model [Hirtz *et al.*, 2002] or implementations on ontology tools [Nanda *et al.*, 2007; Szykman *et al.*, 2001].

1.1 The semantics of design symbols

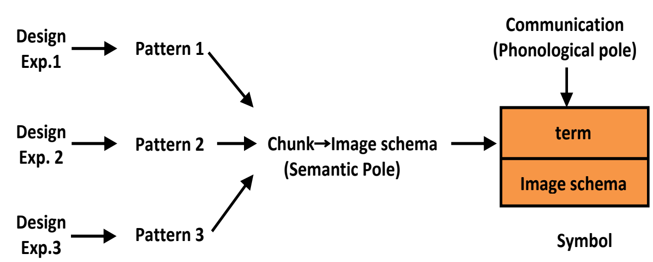


Figure 1: *Emergence of symbols based on experience*: Often the same abstract pattern (or *chunk*) appears in many experiences (e.g. the notion of "fit" for peg in hole, bolt in latch, plug in sink, etc.). If a chunk is valuable in compactly representing many situations, it has a higher likelihood of being communicated, thus acquiring a phonological pole and becoming a symbol. A symbol can then form other associations besides the initial chunk, all of which together constitute its semantic pole or *image schema*.

All these models define a number of symbols at different levels in the hierarchy. Unfortunately the term "symbol", as it is used in the logic and computational community is considerably different from its usage in cognitive linguistics and in everyday life. In the latter usage, symbols are imbued with meaning grounded on experience, whereas in the formal usage, it is merely a token constructed from some finite alphabet, and is related only to other terms. If we present an analogy, a blind man knows "red" is a different color from "blue" and "green" but his understanding of red is is dramatically different from that of a sighted person, because the semantic pole is not connected to direct experience.On the other hand, "symbol" has come to be understood in cognitive science (and also traditionally in linguistics, e.g. de Saussure ([De Saussure, 19161986]), as the tight binding of the of the psychological impression of the sound (the "phonological pole") with the mental image of the meaning (the *semantic pole*) [Langacker, 1986]. The mental image or image schema includes all sorts of associations and is somewhat different for each user, though social convention ensures a degree of overlap between mental images within the language community.

However, the notion of symbol is more far-reaching than communication. It turns out that to some extent, the symbols help divide up the world into chunks, and eventually, it may reflect changes in how we think.

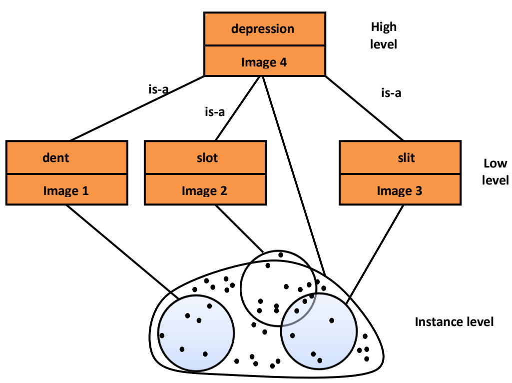


Figure 2: *Abstraction starts with ground instances*: Each symbol in this hierarchy has a term or label ("dent", "slot" or "slit") and a corresponding abstract pattern or "image schema". The image schema is used in identifying an instance as belonging to a symbol category, but also in composing symbols, and in interpreting higher abstractions. Primitive design ontologies are born *is-a* through usage; when instances already known as dents or slots are also labelled as "depressions" by a trusted user, the system learns the subclass relationship. This makes grounded instances available even for the more abstract symbols. Similarly, other relations e.g. "dents are generally undesirable" would also be learned through usage and become part of the image schema. The number of such associations for each symbol is often very large, and limiting these to a few user-determined definitions is a major contributor to brittleness in knowledge systems.

Symbols in CAD systems are not completely devoid of the semantic pole, but those symbols instantiated in the implemented CAD systems are mostly suitable to a particular domain but are not transferable to other existing or new domains. For example, a "slot" on a coin-machine vs a "slot" on a machining fixture both have similar physical and geometric symbol but these two instances are in different domains. These type of situations are almost impossible to program, since the range of usage very large, and may even be unbounded.

1.2 Bottom-Up Semantics in design

An alternative that has been proposed for modeling design concepts is to attempt to move more towards the human process, to learn symbols based on design experience[Gero and Fujii, 2000]. The human design process is a constant, motivated exploration of the design space, e.g. through sketching. All the while, the designer is focusing on the designs that are "good" in some functional sense, and eventually, some kinds of patterns emerge as the common characteristics of these designs. This is one sense in which sketches "talk back" to the designer [Goldschmidt, 2003]. These patterns result in constraints whereby many of the initial design variables can be combined, a process cognitively known as *chunking* [Gobet *et al.*, 2001].

In case of designing a padlock, we may learn that in a padlock, to balance the strength in its components, the shackle diameter increases roughly in proportion with body size. Thus these two parameters can then be brought down to a single chunk. In this way, expert designers based on their experience will come up with "good designs" [Gross, 1986] by choosing wide range of design variables, than would have beeen possible initially.

An early attempt at discovering patterns in the design space of shapes may be seen in relation to 2D shapes in the work of [Park and Gero, 1999]. [Moss *et al.*, 2004] have developed a system in which the M-agents consider the good designs and can be used to extract chunks which are added to memory and become part of the variable space. Similary a recent approach by [Sarkar *et al.*, 2008], which considers Singular Value Decomposition (SVD) on a co-occurance matrix of matrix of variables and constraints to identify the relations between different variable groups.

1.3 Emergence of design concepts

None of these proposals however learn the concepts underlying the symbol (the semantic pole) in a grounded manner, and therefore lack the flexibility of the human designer. By *grounded*, we refer to the progressive manner in which a human designer learns her concepts - the more abstract ones are based on earlier, concrete concepts, but are still presented through instances and in the end, many concepts are grounded in terms of a number of experiential instances.

In this work, we propose to take the first step towards building this grounded semantics, which we call the birth of symbols. What is a symbol? We take a symbol to mean the tight binding between a label and a large set of consciously accessible experiential patterns. The label is usually a communicative term, and constitutes its phonological pole. Occasionally, however, for example while "talking" to a sketch, a designer may get a conscious awareness of a constraint without verbalizing it - and even these are symbols. Or it may be verbalized, and acquire a name. Either process (called *reification* results in a new symbol. At the same time, amorphous implicit schemas, which are formed well before we are aware of them (see the book Blink [Gladwell *et al.*, 2005] for examples of this) are incipient symbols, but not quite there yet. Still, they are useful, and we do use them, but they need to prove their mettle before they become true symbols. This interpretation is in line with a long tradition in psychology and linguistics [Mandler, 2004].

Since we are concerned with discovering patterns that apply to the set of *good* designs, we must have a way of assessing whether a design is good or not. This implies that some set of functional criteria must be available; even if it is not quantitative, it must be capable of ordering a wide variety of design instances, and given a design instances (a set of values for the design variables) it must be possible to determine the degree of functional feasibility. Such a model of function is usually not available in the very earliest stages of design; indeed even for human designers, insights usually arise while fiddling with instances, when suddenly a pattern may light up. So we are considering this type of emergence only after an embodiment is available, when the nature of its function can be related to the set of design variables. As in [Moss *et al.*, 2004] we consider "good designs" as those arising from multi-criteria optimization or based on user-defined minimal functional criteria.

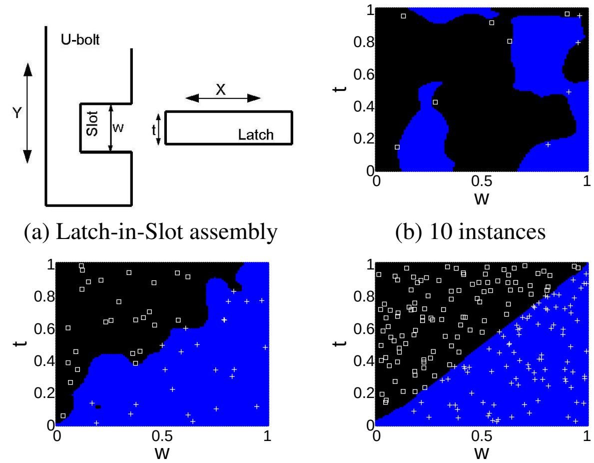
The next sections outlines the development of the infant designer into a (very) early designer; the idea is that computational models capable of human-like ability would be possible based on the equivalent of years of exposure to such situations. In the end, we may expect to have a grounded symbol system that can reason about the flow of energy and take it down to a detailed design, but that end, even if it is achievable, is clearly far away. At present, let us start with the infant designer.

2 Infant designer

An infant designer is really the baby who is first discovering regularity of object behaviour in the world and is one who is just beginning to form her knowledge of the world. She can make various choices, and evaluate them based on some notion of function. Considering the peg-in-hole task just alluded to, we see how she might learn the concept that a peg must be smaller than a hole.

The functional model considered is simple - the design is functionally feasible if the peg can go in (actually our system computes the configuration space - the penetration region disappears when w¿t - but this is not relevant here). We consider a horizontal version of the peg-in-hole - a latch is entering a slot on a bolt, say. Figure 3 shows how after evaluating a number of instances in the design space of latch-widths w and slot-widths t; in (w, t) space, a clear 45 degree line emerges, separating the "good designs" from the bad.

Does this constitute symbolic knowledge for the infant designer? Most likely not. However, it is something that might become a symbol as she acquires other concepts that she can refer to. An important aspect of symbols is that they are defined in relation to other symbols, e.g. if we consider vocabulary A which has both the terms "slot" and "slit", versus system B, which has only "slot", then we might expect the semantic pole of "slot" to more narrowly defined in A than in B.



(a) Latch-in-Slot assembly (b) 10 instances

(c) 50 instances (d) 200 instances

Figure 3: *Learning through experience that latch-must-be-smaller-than-slot ($w > t$).* (a) A latch of thickness t is fitted to a slot of width w. The learned patterns are shown in (w, t)-space in (b)-(d). The quality of the learned pattern varies greatly with degree of experience: results shown for a multi-layer perceptron after experiencing 10,50, and 200 design instances.

Similarly for the infant, it needs a certain density of putative symbols before making the symbolic jump.

What is interesting in the results of figure 3 is how, after experiencing just a few instances, the pattern is inchoate, so the baby keeps trying to insert the fat square into the smaller circle, filling up the negative (black) area of the figure. Eventually the defining boundary becomes sharper, and at some point it can be said to knows the principle, at least implicitly.

At the next step for our infant designer, we consider the concept that a designer knows as "fit". By now our infant learner will attempt to insert pegs only if they are smaller than the slot. The function is defined in terms of the degree of *fit* - how much does it wiggle? Defining the wiggle in terms of the area of the free-space in the configuration space, we see that if the wiggle desired is very small, we get the situation on the left, and if it is very large, we get the situation on the right. Eventually, the learner learns the concept of "fit" as a chunk (composed as $w - t$) - thus, given a level of fit, it imposes a constraint where w and t are related in a manner where they constitute a one-dimensional chunk instead of two independent variable.

Of course, from a machine learning perspective, both these examples are rather elementary. Our objective in presenting it is merely to emphasize the role of even the earliest knowledge in many advanced design situations. These two concepts are also among our earliest knowledge achievements; typically, infants learn containment (peg in hole) by about 3 months, and tight vs loose by 5 months [Casasola *et al.*, 2003]. Many cognitive scientists believe that our concepts of abstraction, including the *is-a* crucial to constructing hierarchies, is a metaphorical extension of containment [Lakoff and Johnson, 1999].

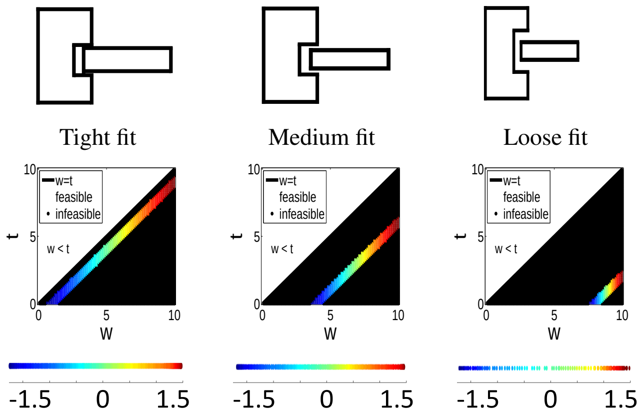


Figure 4: *Birth of the image-schema for "fit":* An insertion task with different kinds of fit are shown in the top row and the corresponding design spaces (w, t) with feasible and infeasible regions are shown below. The function is given as the amount of play available (amount of free-motion or wiggle). If the desirable wiggle is specified, the two-dimensional design space is effectively reduced to one since a relation emerges between the feasible w and t. This mapping or image schema is a early prototype of the concept of "fit".

3 Symbol emergence

As the designer matures from infancy, we can consider the more general process by which symbols form. These correspond to the stages shown in figure 5. At first, the designer explores with instances in the design space, distinguishing the good designs from the bad. Eventually a region in the design space emerges as that containing mostly the feasible designs - this is the Functionally Feasible region (FFR), or the space of "good designs". At least implicitly, this region is being continuously abstracted in terms of function. If the region is a very simple, one may even get an explicit characterization for it. However, in many high dimensional design spaces (even in the 8-dimensional example next) such relations are far from obvious.

The other interesting aspect of the FFRs is that they often correspond to narrow bands of functional feasibility. This may be because they are the result of (possibly unconscious) multi-objective optimization - thus, if there are k design objectives, then they constitute a $k-1$ surface in the objective space. When the function measures that map from the design variable space to the objective space are at least continuous, their Jacobians would be well-posed, and the near neighbours in the objective space may correspond to near neighbours in the design space. If this is actually correct, we may expect the designs to lie along a $k-1$ surface (or "manifold") in the objecitve space (shown as a folded patch in the figure). More generally, the mapping from the design space to the functional objective space is not so well-posed, but the dimensionality reduction, while not quite as pronounced, may still be very significant.

Each dimension in this reduced dimensional map reflects an inter-relation between many independent design parameters. Subsequently, many other design situations also reveal such reduced dimensional hyper surfaces, with different dimensionalities. Sometimes, some of these dimensional mappings or chunks may recur in many design situations - this makes the chunk useful, which is an important criteria for becoming a symbol. In the interim, the designer may use these chunks with a dim awareness of it for a long period, even several years. Meanwhile, as other instances are being explored, and the confidence in the manifold mapping increases, the designer may eventually articulate it (at least to herself), which would be the birth of a true symbolic representation for this concept. At this point, a label may get attached to it, and many other associations would eventually accrue to this term / image-schema pair; it would then constitute a truly reified symbol in the sense that most cognitive scientists would understand it.

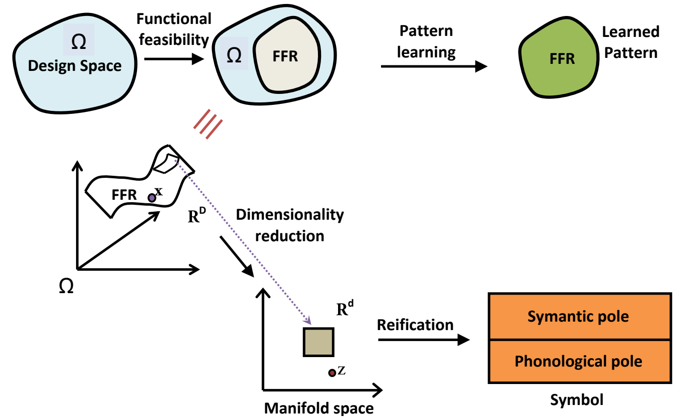


Figure 5: *The symbol emergence process*: our main interest is to discover and learn structural or behavioral chunks that result in good designs, corresponding to functionally feasible regions (FFRs) in the designs space. FFRs typically reflect multiple functional criteria, and may be obtained from some approximate optimization, or from user specified minimal functional criteria. A set of FFR instances can be used to learn a pattern of functional feasibility, the quality of this pattern improves with experience as earlier. Once the FFR is sufficiently rich, one may also discover that they lie along some low-dimensional manifold (R^d) embedded in the high-dimensional design space R^D ($d \ll D$). The lower dimensional space is then a chunked representation for the initial design space. If this relation becomes conscious, it may then become a design symbol.

3.1 Dimensionality Reduction

To determine the inter-relations between many independent design parameters , now we present the algorithm to obtain "good designs" constrained to a much lower dimensional manifold in the design space.

One of the main strategies to handle high dimensional design data is *diminsionality reduction*, involves finding low-dimensional structures in high-dimensional space. Though there is a large body of work is concentrated in determining these low-dimensional representations [Bishop, 2006] like linear methods PCA, ICA, etc, but these linear methods fails when the data lies on nonlinear manifold.in such situations the linear algorithms give the smallest convex subspace encapsulating the manifold, which is often of a much higher dimension.Approaches for obtaining the non-linear representation of the data include Global methods (Isomaps [Tenenbaum *et al.*, 2000]) and Local methods (Locally Linear Embedding [Saul and Roweis, 2003] and Laplacian Eigenmaps [Belkin and Niyogi, 2002]).

---

1. `Compute the neighbors` X_j `of each data point,`X_i`.`
2. `Compute the weights` W_{ij} `that best reconstruct each data point` X_i `from its neighbors, minimizing the reconstruction error` $(\epsilon(W) = \sum_i |X_i - \sum_j W_{ij} X_j|^2)$ `by constrained linear fits.`
3. `Compute the vectors` Γ_i `best reconstructed by the weights` W_{ij}`, minimizing the quadratic form` $(\Phi(\Gamma) = \sum_i |\Gamma_i - \sum_j W_{ij} \Gamma_j|^2)$ `by its bottom nonzero eigenvectors.`

In this study, we use LLE, which is an eigen vector method for the problem of nonlinear dimensionality reduction [Saul and Roweis, 2003; Roweis and Saul, 2000] to identify the underlying non-linear manifold in high-dimensional design space. The LLE algorithm is described in algorithm 3.1.

3.2 Universal Motor example

Next we consider a well-known problem in the design literature, that of designing an Universal Motor. Universal motors have been well studied in the product family design literature [Simpson, 1998], where the physical description and schematic of the universal motor have been presented. The design space for embodiment design consists of 10 design variables $\vec{v} = \{N_c, N_s, A_{wa}, A_{wf}, r_o, t, l_{gap}, I, V_t, L\}$ where N_c: number of wire turns on the armature, N_s: number of wire turns on each pole on the field , A_{wa}: cross-sectional area of the wire on the armature, A_{wf}: cross-sectional area of the wire on the field, r_o:radius of the motor, t: thickness of the stator , I: current drawn by the motor, L: stack length, and the performance behaviors are taken as strength, mass, energy and efficiency and the corresponding performance metrics interms of these design variables can be $\pi_{torque}(\vec{v}) = \frac{N_c \phi I}{\Pi}$, $\pi_{mass}(\vec{v}) = mass_{windings} + mass_{armature} + mass_{windings}$, $\pi_{power}(\vec{v}) = V_t I - I^2(R_a + R_s) - 2I$, and $\pi_{efficiency}(\vec{v}) = \frac{\pi_{power}}{V_t I}$. The derivation and the related equations are presented well in [Simpson, 1998]. And the main constraints we considered here for the feasible motors are (i) the magnetizing intensity $H < 5000$ and (ii) the outer radius of the stator r_o must be greater than the thickness of the stator t.

A minimal parameter set for the universal motors may be considered in terms of the two design parameters L and I, while keeping other parameters constant [Simpson, 1998].

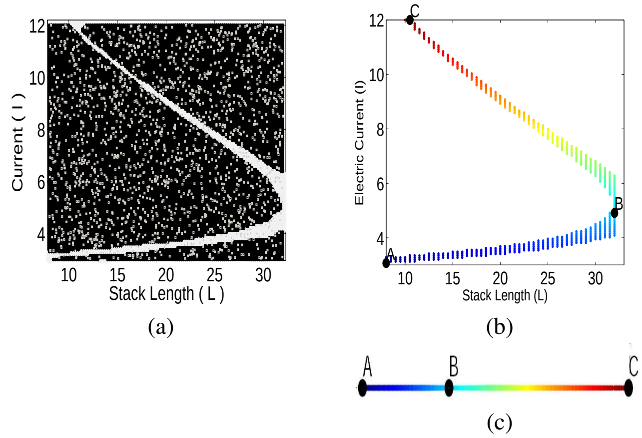


Figure 6: *Chunking on the L, I subspace for Universal Motors*: (a) The implicit constraint on the L, I subspace of the Design Space is learned for 2000 design instances under the functional specification 280 W< π_{power} < 295 W. (b) The feasible designs in the L, I subspace, in which each design instance can be identified based on two parameters L and I. This subspace is mapped into a low-dimensional (one-dimension) shown in (c). A,B, and C are three different design instances mapped from L, I space to one-dimensional space.

To obtain the feasible universal motors for the desired functional range of power 280 W< π_{power} < 295. Figure 6(a) shows the result of learning the FFR (the valid designs resulting from this constraint). These lie along a small band, which can be thought of as a curved 1-D manifold (with a slight thickness). 6(b).

Next, we demonstrate the manifold nature of the FFR using the eigenvector method called locally linear embedding or *LLE*, explained in the next section. The mapping between the nonlinear feasible region (Fig. 6 (b)) and the one-dimensional chunk for it below (Fig. 6 (c)) shows the continuity of mapping between these. If we take three data points A,B, and C in L, I space. Let us say X = [A B C], each data point is a real-valued vector, with of dimensionality 2. With the help of Local Linear Embedding (LLE) algorithm [Roweis and Saul, 2000], we construct a neighborhood preserving mapping from L, I space to Γ. The three points A= $(32.0, 4.09)$, B= $(22.5, 3.5455)$ and C= $(10.5, 12.000)$ and their corresponding mappings in the lower-dimensional manifold are $\gamma_A = -0.2102$, $\gamma_B = -0.1430$ and $\gamma_C = 0.0007$.

After this γ chunk is discovered in this Universal motor situation, it is possible that it may also emerge in other situations. If so, a designer may communicate the idea of this chunk to another; let us say he calls it "gavagai". Then as the term "gavagai" spreads in the design community, it would occur in many other situations, and each association would form part of the semantics of the term gavagai.

Indeed, different generations of designers would discover their own image schemas for gavagai, and the term may be applied to wider contexts ("generalization") or more specific ones ("specialization"); thus, it would slowly change its semantics over time. This is another way in which static programmed machine semantics, even if they can capture all the usages at a given point of time, would not be able to keep up with human usage unless it adopts an experiential learning modality as proposed here.

4 Chunking from Multi-Objective Optimization

Until now, we have considered the FFRs as arising from user defined functional desiderata. However, how does the user arrive at these functional bounds? Here, we need not know the bounds on the performance, just the set of performance measures, computable given an instance of the design, are to be specified. In practice, a more useful approach towards finding FFRs is to consider them as the estimated pareto fronts arising from Multi-objective optimization. Most computational multi-objective optimization (MOO) algorithms provide estimates for the non-dominated front [Deb, 2001]. For our purposes, the non-dominated front constitutes a description of "good designs". The front is obtained in the space of objective functions, and our first task would be to map it to the design space. We would then consider these "good designs" in the design space and try to find a manifold representation for it.

4.1 Example A : Universal Electric Motors

Let us now formulate the design of Universal motors as a multi-objective optimization problem to identify FFRs. Here we consider eight design parameters $\vec{v} = \{N_c, N_s, A_{wa}, A_{wf}, r_o, t, I, L\}$ and their ranges are, N_c: number of wire turns on the armature $[100, 1500]$, N_s: number of wire turns on each pole on the field $[1, 500]$, A_{wa}: cross-sectional area of the wire on the armature $[0.01, 1.0mm^2]$, A_{wf}: cross-sectional area of the wire on the field $[0.01, 1.0]mm^2$, r_o:radius of the motor $[1.0, 10.0]cm$, t: thickness of the stator $[0.5, 10.0]mm$, I: current drawn by the motor $[0.1, 6.0]A$, L: stack length $[0.057, 5.18]cm$The mathematical formulation of multi-objective optimization problem as follows:

Multi-Objective Optimization

$$\begin{array}{ll} \text{Minimize} & \pi_{mass}(\underline{v}) \\ \text{Maximize} & \pi_{efficiency}(\underline{v}) \\ \text{Maximize} & \pi_{torque}(\underline{v}) \\ \text{Subject to} & g_1(\underline{v}) \equiv r - t > 0 \\ & g_2(\underline{v}) \equiv 5000 - H > 0, \\ & g_3(\underline{v}) \equiv 2.0 - \pi_{Mass} \geq 0, \\ & g_4(\underline{v}) \equiv 0.5 \leq \pi_{torque} \leq 5.0, \\ & g_5(\underline{v}) \equiv 300 \leq \pi_{Power} \leq 600 \\ & g_6(\underline{v}) \equiv \pi_{efficiency} - 0.15 \geq 0 \end{array} \tag{1}$$

The multi-objective optimization (MOO) technique used here is NSGA-II [Deb, 2001], a well known evolutionary algorithm. The NSGA-II parameters used in this study are as follows: population size = 2000, maximum generations = 500, probability of crossover for both real-valued and binary variables = 0.8, probability of mutation 0.33 and 0.1 for real and binary variables and the distribution index for crossover and mutation are 16 and 30. The Pareto optimal front obtained using NSGA-II is shown in Fig. 7(a). The Pareto-optimal surface is for maximizing both the torque (π_{torque}) and efficiency ($\pi_{efficiency}$) while minimizing the mass (π_{mass}).Having obtained non-dominated sets of designs, and mapping these to the design space reveals that the good designs (FFRs) are often restricted to a few patches on a low-dimensional manifold, thus resulting in significant dimensionality reductions for the design space.

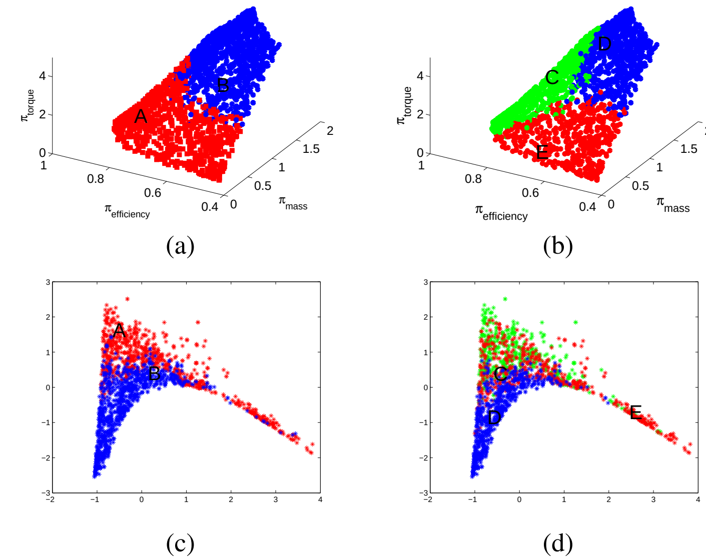


Figure 7: *Clusters in the non-dominated space for Universal motor.* (a) The non-dominated solutions (pareto-front) in the 3-objective space of mass, efficiency and torque. By considering the feasible designs in the design space, we obtain two clusters based on unsupervised clsutering in the design space. (c) The manifold space corresponding to the map from the high-dimensional design space $D = 8$ to low-dimensional design space $d = 2$ obtained with the help of LLE. Similary three clusters are formed with varying unsupervised clustering input parameters (b) and the corresponding low-dimensional manifold is shown in (d).

5 Intrinsic Dimension

The main parameters in this reduction algorithm is estimating (i) number of neighborhoods (K) for each data point in the higher dimension (D), and (ii) the lower dimensionality (d). The estimated dimension d of d^* is provided by the user as a parameter to the algorithm. If d is an under estimate of d^*, there is a loss of information and if d is an over estimate of d^*, then LLE will include arbitrary dimension. In PCA, the dimensionality is estimated by the number of eigen values of the simple covariance matrix comparable in magnitude to the largest eigen value. In LLE, to estimate the intrinsic dimensionality (d^*), one may count the number of eigen values of the covariance matrix ($M = (I - W)^T(I - W)$ (see [Saul and Roweis, 2003]) comparable in magnitude to the lowest

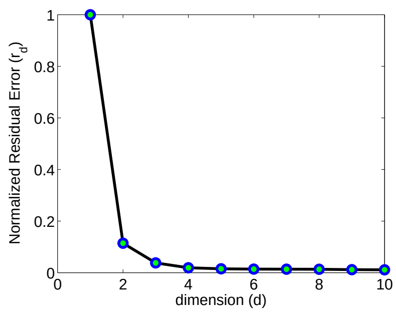


Figure 8: *Dimensionality of manifold for Universal Motors based on* . The FFR data is mapped onto manifolds of different dimensions, and then mapped back to the original design space and the error is estimated. The error drops sharply from 1-D to 2-D manifold, and then less sharply. The knee of the curve at "2" is indicative of the intrinsic dimensionality of the space. A separate maximum likelihood method estimates the dimensionality of the Universal Motor space to be 2.6.

non-zero eigen values [Polito and Perona, 2002]. But these eigen values are not informative based on the work [Saul and Roweis, 2003].

To obtain an estimate of the dimension of the manifold for our data set, we use the technique based on the idea that a dimensionality reduction algorithm should preserve information on a global scale, as measured using bijection [Martin and Backer, 2005]. For a given input dataset $X = \{X_1, \ldots X_N\} \subset R^D$, the dimensional reduction algorithm such as LLE provide a reduced dimensional representation $Y = \{Y_1, \ldots Y_N\} \subset R^d$ of the original data set X. It is assumed that X lies on a manifold M embedded in R^D with intrinsic dimensionality d^*. The estimated dimension d of d^* is a parameter to the algorithm. As per [Martin and Backer, 2005], a good low dimensional representation of X can be expected only if we should be able to go back and forth from X and Y with no loss of information. Let us say, this algorithm provides a map $f : X \rightarrow Y$, mathematically an inverse function $f^{-1} : Y \rightarrow X$ exists such that $f^{-1}(f(X_i)) = X_i$ for all i and exists only if the estimated dimension $d \geq d^*$ of the manifold. To determine the existence of f^{-1}, we have considered the same measure as proposed by [Martin and Backer, 2005], squared residual error $(r_d) = \sum_i ||f_d^{-1}(f_d(X_i - X_i)||$, where $f_d : X \rightarrow Y$ is map produced by LLE, which depends on the estimate d of the intrinsic dimensionality and $f_d^{-1} = Y \rightarrow X$ is a proposed inverse to f_d. To estimate the actual dimension d^* of the manifold M, we found r_d for different values of d. This residual error should be zero when $d \geq d^*$ so that we may determine d^* by finding the smallest value of d such that $r_d = 0$. Hence we can predict the intrinsic dimension by minimizing both r_d and d. By observing the behavior of r_d for different values of d shown in Fig. 8 we can predict the intrinsic dimension can be 2.

Also, we have used a method proposed by [Levina and Bickel, 2004] for estimating the intrinsic dimension of a dataset by applying the principle of maximum likelihood to the distances between the neighbors. With this method we have obtained the intrinsic dimension value $d^* = 2.6$.

6 Conclusion

In this work we have presented an approach to the learning of design symbols which are understood to be a tight binding between a term or label and a semantic representation or image schema formed by an extensive set of experiential associations. These term-meaning pairs constitute an important part of the cognitive repertoire of the human designer, but the semantic pole is completely lacking or severely bleached in current machine systems. This makes it impossible to refer to instances of abstract symbols (defined in terms of other symbols) which is what gives the human systems the plasticity in deploying these symbols.

In language learning by humans, only a few hundred symbols are learned *ab initio*, in the pre-linguistic stage. The remaining tens of thousands of lexical units in the adult vocabulary are learned by exposure to language, typically through reading. What this means is that symbols are ultimately defined in terms of other symbols, but even for these later symbols, instances are available as defined by the linguistic context. Symbol combinations are understood through a process of *composition* where the syntactic structure involving different symbols are combined to construct a more complex symbol [Langacker, 1986].

This process is clearly at work in the design process as well. Once the infant designer has learned a few symbols, these may be reinforced by communication, and be used to define other symbols. For example, it may be told that a "peg" is the thing that fits tightly in a "hole". However, the symbol "fit-tightly-in-a-hole" must be an experienced symbol for this schema to have the plasticity needed in future applications, without it it would be subject to brittle failures. In design terms, "syntax" may constitute the elements of a design and the modalities of connecting them, its semantics then is the overall behaviour and associations of the assembly. Thus, the semantics of fitting an electric plug into a socket may inherit some aspects of the symbol "fit-tightly-in-a-hole" but also some aspects of "energy-flow-electrical-connection". The appearance of the typical plug, its prong structure, and many other aspects would also be part of the semantics. Clearly, defining all of these experiential aspects would be a taxing task. Furthermore, with the evolution of technology or human communication, the semantics may change, and maintaining such systems would be eventually a doomed enterprise.

This work is clearly exploratory, and much work is needed to define the problem more clearly. Among the contributions of this work, we propose non-linear manifold learning as an important step in discovering latent relationships among the many design parameters that permits defining this objective in terms of chunks, which constitute the first step towards forming symbols. Among the work that would need to be done next is to the conjoints of more than one symbol; i.e. given the design elements each as an individual symbol, we need to be able to say what the conjunction of these elements (the syntax) will do, and whether the resulting object - a design instance - will be adequate to meet the design task or not. Again, depending on the "good designs" that emerge in the process, a combination of symbols may come to be designated as a symbol on its own right, leading to the birth of abstract symbols.

At the same time, there is an important role for high-level symbol ontologies. Just as designers can be taught certain relations, even computer functional models may benefit from explicit awareness of pre-programmed relations, especially at levels of abstraction that would be hard to grasp otherwise. However, it is essential that the semantic poles for these instructions be defined in terms of its own internal image schemas; without this, brittleness may again prevail. Also, there remains a role for these in human communication process. However, for the purposes of low-level symbol development, it would be futile to try to define these semantics in terms of other symbols and rules, or even in terms of a few definitions.

In the final analysis, the argument presented here implies that in the long run, to create viable computer vocabularies for design, we must train the systems to learn these relationships, by experiencing many design and real world situations. This may be done in an accelerated manner, but the system must be exposed to something like the vast array of experiences of a human - or possibly many more, since the abstraction processes as we understand it so far may not be as efficient. As different systems are deployed in solving different problems, their somewhat differing input sets would result in somewhat different abstractions for the same symbols. These resulting design agents may therefore be somewhat less predictable than current computers, but such is the price of flexibility.

References

[Belkin and Niyogi, 2002] M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. *Advances in Neural Information Processing Systems*, 1:585–592, 2002.

[Bishop, 2006] C.M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[Bohm *et al.*, 2005] M.R. Bohm, R.B. Stone, and S. Szykman. Enhancing Virtual Product Representations for Advanced Design Repository Systems. *Journal of Computing and Information Science in Engineering*, 5:360, 2005.

[Campbell *et al.*, 2000] M.I. Campbell, J. Cagan, and K. Kotovsky. Agent-based synthesis of electromechanical design configurations. *Journal of Mechanical Design*, 122:61, 2000.

[Casasola *et al.*, 2003] Marianella Casasola, Leslie B. Cohen, and Elizabeth Chiarello. Six-month-old infants' categorization of containment spatial relations. *Child Development*, 74:679–693, 2003.

[Chakrabarti *et al.*, 2005] A. Chakrabarti, P. Sarkar, B. Leelavathamma, and B S Nataraju. A functional representation for aiding biomimetic and artificial inspiration of new ideas. *AI EDAM*, 19(02):113–132, 2005.

[De Saussure, 19161986] F. De Saussure. *Course in general linguistics*. Open Court, 1916/1986.

[Deb, 2001] Kalyanmoy Deb. *Multi-Objective imization using Evolutionary Algorithms*. Chichester, John Wiley and Sons, Ltd., 1 edition, 2001.

[Gero and Fujii, 2000] JS Gero and H. Fujii. A computational framework for concept formation for a situated design agent. *Knowledge-Based Systems*, 13(6):361–368, 2000.

[Gladwell *et al.*, 2005] M. Gladwell, H. Finder, and C-SPAN (Television network. *Blink: The power of thinking without thinking*. Penguin Books, 2005.

[Gobet *et al.*, 2001] F. Gobet, P.C.R. Lane, S. Croker, P.C.H. Cheng, G. Jones, I. Oliver, and J.M. Pine. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5(6):236–243, 2001.

[Goldschmidt, 2003] G. Goldschmidt. The backtalk of self-generated sketches. *Design Issues*, 19(1):72–88, 2003.

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

[Gross, 1986] Mark Donald Gross. *Design as Exploring Constraints*. PhD thesis, Department of Architecture ,Massachusetts Institute of Technology, February 1986.

[Hirtz *et al.*, 2002] J. Hirtz, R.B. Stone, D.A. McAdams, S. Szykman, and K.L. Wood. A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design*, 13(2):65–82, 2002.

[Kurtoglu *et al.*, 2005] T. Kurtoglu, M. Campbell, J. Gonzales, C. Bryant, and R. Stone. Capturing empirically derived design knowledge for creating conceptual design configurations. In *Proceedings of the ASME Design Engineering Technical Conferences And Computers In Engineering Conference. DETC2005-84405, in review, Long Beach, CA*, 2005.

[Lakoff and Johnson, 1999] George Lakoff and Mark Johnson. *Philosophy in the Flesh: The embodied mind and its challenge to Western thought*. Basic Books, New York, 1999.

[Langacker, 1986] Ronald W. Langacker. An introduction to cognitive grammar. *Cognitive Science*, (10):l–40, 1986.

[Levina and Bickel, 2004] E. Levina and P.J. Bickel. Maximum likelihood estimation of intrinsic dimension. *Ann Arbor MI*, 48109:1092, 2004.

[Mandler, 2004] Jean Matter Mandler. *Foundations of Mind : Origins of conceptual thought*. Oxford University Press, New York, 2004.

[Martin and Backer, 2005] S Martin and A Backer. Estimating manifold dimension by inversion error. In *ACM Symposium on Applied Computing*, pages 22–26, 2005.

[Moss *et al.*, 2004] J. Moss, J. Cagan, and K. Kotovsky. Learning from design experience in an agent-based design system. *Research in Engineering Design*, 15(2):77–92, 2004.

[Nanda *et al.*, 2007] J Nanda, H.J. Thevenot, T.W. Simpson, R.B. Stone, M. Bohm, and S.B. Shooter. Product family design knowledge representation, aggregation, reuse, and analysis. *AI EDAM*, 21(02):173–192, 2007.

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

[Park and Gero, 1999] SH Park and J.S. Gero. Qualitative representation and reasoning about shapes. In *Visual and Spatial Reasoning in Design*, volume 99, pages 55–68, 1999.

[Polito and Perona, 2002] M. Polito and P. Perona. Grouping and Dimensionality Reduction by Locally Linear Embedding. *Advances in Neural Information Processing Systems*, 2:1255–1262, 2002.

[Richard and Dixon, 1994] Welch V Richard and John R Dixon. Guiding conceptual design through behavioural reasoning. *Research in Engineering*, 6:169–188, 1994.

[Roweis and Saul, 2000] S.T. Roweis and L.K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding, 2000.

[Sarkar *et al.*, 2008] Somwrita Sarkar, Andy Dong, and John S Gero. A learning and inference mechanism for design optimization problem (re)-formulation using singular value decomposition. In *Proceedings of DETC'08, ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, August 2008.

[Saul and Roweis, 2003] L K Saul and S T Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4:119–155, 2003.

[Simpson, 1998] Timothy W. Simpson. *A Concept Exploration Method for Product Family Design*. PhD thesis, Georgia Tech University, Dept Mechanical Engineering, 1998.

[Szykman *et al.*, 2001] S. Szykman, R.D. Sriram, and W.C. Regli. The role of knowledge in next-generation product development systems. *Journal of computing and information Science in Engineering*, 1:3, 2001.

[Tenenbaum *et al.*, 2000] J.B. Tenenbaum, V. Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction, 2000.