Symbol emergence in design

Amitabha Mukerjee, Madan M Dabbeeru
Indian Institute of Technology Kanpur
amit@cse.iitk.ac.in, mmadan@iitk.ac.in

Abstract

A key step in mapping the more conceptual stages of design onto computational systems involves identifying a vocabulary and ontology. 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 with the help of universal motor design.

1 Efforts towards standardizing the design vocabulary

Evolving a standardized vocabulary for design has emerged as an important focus in engineering design. Possible applications include developing design repositories [Bohm *et al.*, 2005], computer assisted conceptual design [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, 19881996] 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].

The above represents the human-engineered approach to defining symbols. This type of approach is initially tempting because it tends to meet immediate applications, but a long history in knowledge-based systems has shown it to be brittle, i.e. subject to failure under even minor deviations in the domain. In general, it may be that symbols are more meaningfully developed by abstracting from existing data. The novel contribution of this paper is to show that at least in certain types of design tasks, lower-dimensional surfaces are revealed by multi-objective optimization. The intrinsic dimensions in these pareto-surfaces might constitute one approach to obtaining “symbols” directly from experiential data as opposed to engineering them by programming definitions / rules. These approaches are detailed further in section 1.2 and section 3, but first we look more closely at the term “symbol”, and what is understood by its semantics.

1.1 The semantics of design symbols

Unfortunately the term “symbol”, as it is used in the logic and computational theory 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 such tokens. If we present an analogy, a blind man knows “red” is a different color from “blue” and “green” but his understanding of red 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

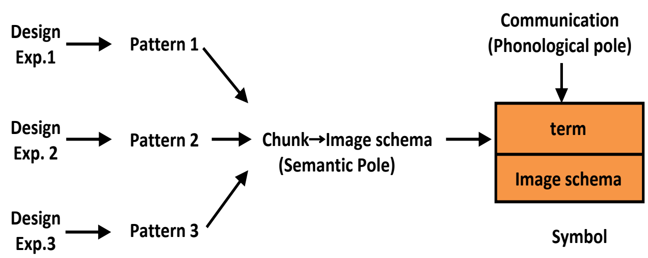


Figure 1: *Emergence of symbols based on experience*: Often the same abstract pattern (or *chunk*) appears in many experiences (e.g. the notion of "containment" for peg in hole, bolt in latch, plug in sink, etc.). If a chunk is valuable in compactly representing many situations, it has a higher likelihood 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*.

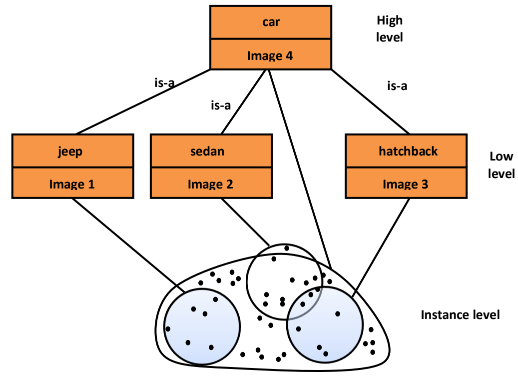


Figure 2: *Abstraction starts with ground instances*: Symbols like "hatchback", "sedan", or "jeep" may correspond an abstract pattern or "image schema", which is used to identify instances as belonging to a symbol category, but also in composing symbols, and in interpreting higher abstractions. Primitive design ontologies like *is-a* arise when instances already known as sedans or hatchbacks are also labelled as "car" by a trusted user. Similarly, other relations e.g. "jeeps can drive over rough terrain" 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.

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 classes, and eventually, it may reflect changes in how we think. For instance, Korean language makes a distinction between spatial tight-fit situations, *kkita*, (as in "put the cap on the pen", "hand in glove") from other usages of "in" or "on". Infants growing up in English and Korean linguistic environments were sensitive to both contrasts, but English children appear to lose this sensitivity around the time they start acquiring language, suggesting that the language construct may have weakened their sensitivity to these changes [McDonough *et al.*, 2003].

On the other hand, incompatibility of design vocabulary is rarely a problem between humans (that's why exceptions often become memorable). If designers A and B are talking, and A does not have a particular symbol λ, its image-schema may emerge through a small amount of discussion; in many cases, just a single example may be enough to stretch an existing concept $\lambda\prime$ in A to the current one. Of course, the new symbol $\lambda\prime$ remains imprecise, and designer A is aware of it, and subsequent uses of $\lambda\prime$ will serve to ground it. All this is possible because the semantic pole for the human is a complex, elastic set of associations that cannot be defined in terms of a single predicate or even a range, it is the set of all situations where the symbol may be encountered (figure 2). All these associations need to be learned, and cannot be inferred based on a single definition (not to mention issues such as nonmonotonicity); hence the programmer-given single definition, usually created to demonstrate the example at hand, is a hopelessly inadequate semantics for a design symbol; and that is why we need bottom-up symbol discovery in order to ground a design vocabulary.

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].

For example, in designing a padlock, we may learn that the shackle diameter increases roughly in proportion with body size. Thus these two parameters can then be brought down to a single chunk. These chunks, which limit the choices used in "good designs", may be what are used by expert designers [Gross, 1986].

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 a design observer agent considers trends among good designs and try to extracts chunks. Similarly a recent approach by [Sarkar *et al.*, 2008], considers Singular Value Decomposition (SVD) on a co-occurance matrix of matrix of variables and constraints to identify the relations between different variable groups.

However, none of these proposals attempt to learn their symbols 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. In the end, many concepts are grounded in terms of a number of experiential instances. For a human designer, this learning cannot be limited to the years of training as a designer, but must include all of her knowledge about the world, the so called commonsense knowledge. Thus, the fact that a fat peg will not go into a thin hole is part of her prior knowledge. Indeed, it is likely that the process by which she acquires these patterns, built upon many layers of pre-existing knowledge, may be similar in some salient ways with her earliest learning.

In this work, we propose to take the first step towards building such a grounded semantics, which we call the birth of symbols. In a human design scenario, say while "talking" to a sketch, a designer may get a conscious awareness of a constraint without verbalizing it - this is referred to as *reification*, becoming real - and is a key step in forming new symbols. Sometimes, amorphous implicit schemas, which are formed well before we are aware of them [Gladwell *et al.*, 2005] are incipient symbols, 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, that symbols are "aware" or conscious [Mandler, 2004].

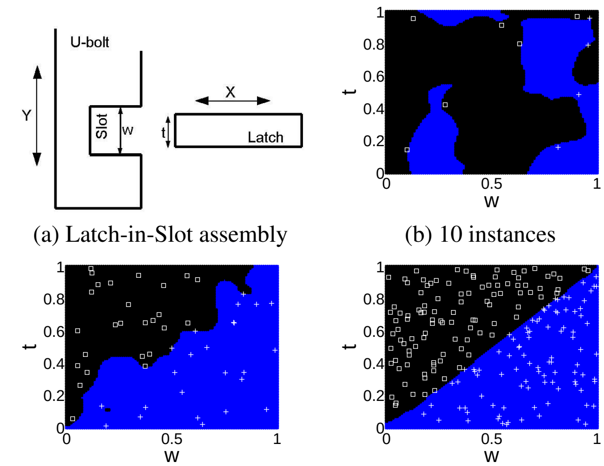
2 Infant designer

A system learning symbols is like a baby who is first discovering regularity of object behaviour in 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 $w > t$). 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. 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.



(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.

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].

3 Symbol emergence

As the designer matures from infancy, we can consider the more general process by which symbols form. These may 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 subset of the design space emerges as the Functionally Feasible region (FFR), or the space of "good designs". Often, FFRs correspond to narrow bands of functional feasibility. This

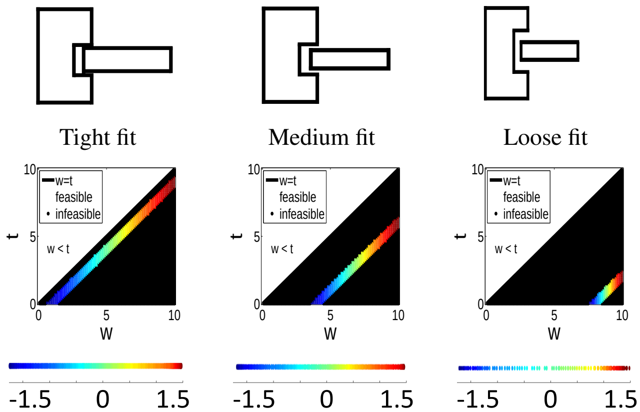


Figure 4: *Birth of the image-schema for "fit":* An insertion task with different kinds of fit are shown in the top row and the corresponding design spaces (w, t) with feasible and infeasible regions are shown below. The function is given as the amount of play available (amount of free-motion or wiggle). 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".

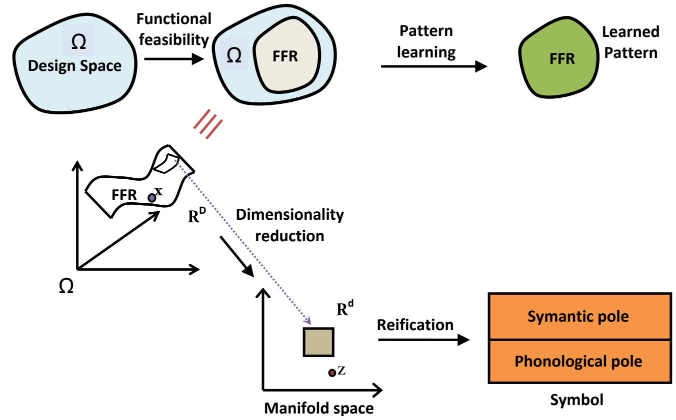


Figure 5: *The symbol emergence process*: our main interest is to discover and learn structural or behavioral chunks that result in good designs, corresponding to functionally feasible regions (FFRs) in the designs space. FFRs typically reflect multiple functional criteria, and may be obtained from some 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.

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. In continuum design situations (i.e. the search space is continuous and not combinatorial), if the function measures that map from the design variable space to the objective space are continuous, their Jacobians would be well-posed, and the near neighbours in the objective space may correspond to near neighbours in the design space. While this assumption is flawed for a large class of difficult optimization problems (e.g. Quadratic assignment), it often holds for a large if not preponderant fraction of real tasks. Thus, in such situations, we may designs that lie along a $k-1$ pareto-surface (or "manifold") in the objective space (shown as a folded patch in the figure), and a similar lower-dimensional manifold in the design space as well. Each dimension of this lower dimensional space reflects an inter-relation between independent design parameters (e.g. the shackle diameter and the lock size). 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. Later, 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.

Thus a key aspect of design symbol formation is *dimensionality reduction*, - i.e. finding low-dimensional patterns in high-dimensional space. There are two classes of dimensionality reduction algorithms - linear methods like PCA or ICA [Bishop, 2006], or nonlinear approaches, which may be global (Isomaps [Tenenbaum *et al.*, 2000]) or local (Locally Linear Embedding or LLE [Saul and Roweis, 2003] and Laplacian Eigenmaps [Belkin and Niyogi, 2002]). Here we present some results based on the LLE algorithm, which is an eigenvector method that works based on the assumption that the same weighted sum between neighbours would hold both in the high and the low dimensional spaces (algorithm 1).

3.1 Universal Motor example

We illustrate the working of the process based on the Universal Motor,which has been well studied in the product family design literature [Simpson, 1998]. The design space consists of eight design variables: N_c (number of wire turns on armature) N_s (number of turns on each field pole), A_{wa} (cross-section area of armature wire), A_{wf} (cross-section area of the field wire), r_o (radius of motor), t (thickness of stator) , I: (current drawn by motor), L (stack length). Function is measured through a set of performance behaviors: strength, mass, energy and efficiency. The corresponding performance metrics in terms 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 $\pi_{efficiency}(\vec{v}) = \frac{\pi_{power}}{V_t I}$. (following [Simpson, 1998]). We may now consider that the feasible designs have (i) the magnetizing intensity $H < 5000$ and (ii) the outer

Algorithm 1 Local Linear Embedding

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.`

radius of the stator r_o greater than the thickness of the stator t.

We next outline two experiments designed to reveal the inter-relationships in the parameter space when it comes to the optimized designs. The results suggest that the optimized designs are not scattered uniformly across the design space, but reveal certain inter-relations between the design parameters. Thus, the initial parameter space of 8 parameters may actually constitute only two independent parameters when it comes to the optimized designs. While these results hold only for these design classes, the implications might be more general, and imply far-reaching consequences in obtaining symbols as dimension-reducing patterns in continuous parameter space of a wide ranging set of problems. However, whether these results will scale up to other remains a subject of considerably more research; the results below only indicate that this may be so.

3.2 Two-dimensional design space

In an initial experiment, we consider a minimal parameter set for the universal motor - modeling the design variability in terms of only two design parameters L and I, while keeping other parameters constant [Simpson, 1998]. For a desired functional range of power 280 W< $\pi_{power} < 295$, the FFR (the valid designs resulting from this constraint) is shown in Figure 6(a). These lie along a small band, which can be thought of as a curved 1-D manifold (with a slight thickness). 6(b).

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 $\gamma_C = 0.0007$.

This reduction of the two design parameters to a single γ represents the first stage of symbol formation. If, later, this γ chunk is discovered in other situations, then a label, say "gavagai", may attach to it. Then as the term "gavagai" may spread in the design community, and might occur in many

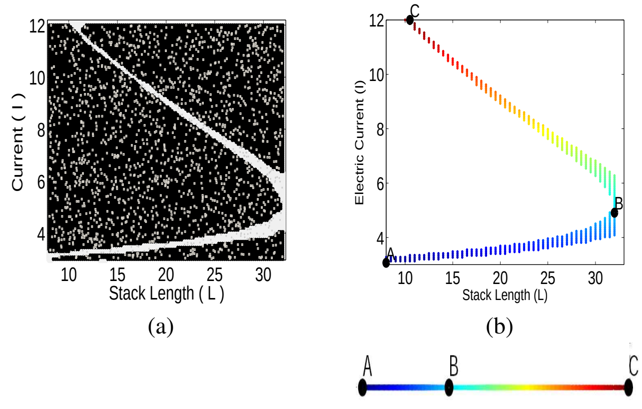


Figure 6: *Chunking on the L, I subspace for Universal Motors*: (a) The implicit constraint on the L, I subspace of the Design Space is learned for 2000 design instances under the functional specification 280 W< $\pi_{power} < 295$ W. (b) The feasible designs in the L, I subspace. (c) The mapping onto a low-dimensional (1-D) space; this reveals that for good designs, the stack-length L and the motor current I are related. A,B, C: individual design instances in (b) and (c).

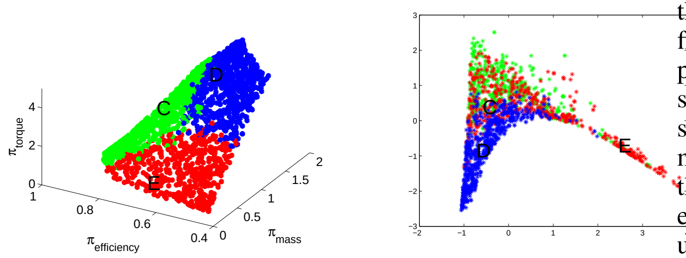
other situations, and each such association would form part of the semantics of the term *gavagai*. A computational system that learns this term in this way would need to participate in such discussions in the design community to keep its semantics current. This is another reason why static programmed machine semantics, even if they can capture all the usages at a given point of time, fail in the long run as human usage changes.

3.3 High-dimensional spaces: Multi-Objective Optimization

If we are to consider the eight-dimensional design space for the Universal motor, a more useful approach towards finding FFRs may be to consider a multi-objective optimization problem based on a set of performance metrics. If design solution A is better than solution B in all the functional criteria, we say that A *dominates*— B. The set of all non-dominated solutions is the non-dominated front or *pareto-front*, and usually lies along a surface in the space of objective functions. For the Universal motors example, the multi-objective optimization problem may be formulated as follows:

Multi-Objective Optimization

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



(a) (b)

Figure 7: *The non-dominated front for the Universal motor.* (a) The non-dominated solutions (pareto-front) in the 3-objective space of mass, efficiency and torque. (b) The manifold space corresponding to the map from the high-dimensional design space $D = 8$ to low-dimensional design space $d = 2$ obtained by LLE. Note that the distribution of colours are non-uniform in the two maps, but they remain segregated (with some noise).

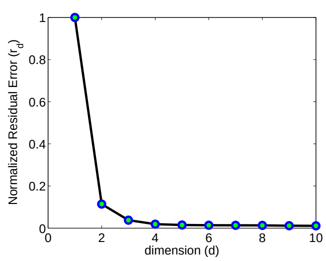


Figure 8: *Dimensionality of manifold for Universal Motors based on* . The FFR data is mapped onto manifolds of different dimensions, and then mapped back to the original design space and the error is estimated. The error drops sharply from 1-D to 2-D manifold, and then less sharply. The knee of the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

We use the well known NSGA-II [Deb, 2001] evolutionary algorithm, with population size 2000, and probability of crossover 0.8, mutation probability 0.33 and 0.1 (for real/ binary). The estimated pareto front for maximizing both the torque (π_{torque}) and efficiency ($\pi_{efficiency}$) while minimizing the mass (π_{mass}) is shown in Fig. 7(a). The designs in this non-dominated front in objective space are identified in the original 8-parameter design space. We now attempt to see if these 8-D points actually constitute a lower dimensionality manifold, by considering the reconstruction error when mapped to differing dimensionalities from 2 to 8 (figure Fig. 8; the sharp knee at 2 indicates considerable information abstraction, and Fig. 7(b) shows the mapping to a 2-dimensional space obtained by LLE. This mapping reveals that neighbours in the high dimensional space remain nearby in the lower-dimensional space at least for this universal motor problem.

The results here signify that for the universal motor, obtaining the FFR as a 2-dimensional non-dominated surface in objective space can lead to a dimensionality reduction to 2 in the design space as well. These two dimensions possibly reflect inter-relations between the original eight parameters that pertain to the better designs in the design space. In terms of symbol formation, these two dimensions ("XX" and "YY", say), if they are found repeatedly in other domains as well, may eventually become symbols. With sufficient experience, the relation between these two parameters and the design may eventually be encoded into design rules: e.g. "higher YY is usually associated with the more efficient designs". Subsequent experience may also alter the way we understand these chunks, and therefore rules like the above that are built on it; through this demonstration we are primarily arguing that by keeping these symbols grounded, it would be possible to keep updating their semantics *and* their inter-relations (the rules), thus providing a truly flexible symbol system, in contrast to static symbol systems.

We must be careful to point however, that in general a $k-1$-dimensional pareto-surface in objective space may not map to an equivalent manifold in design space - there are a large number of situations where the performance metrics mapping from design space to objective space are not so well-behaved, and such results may not hold. Nonetheless, even if a subset of design parameters are well-behaved, at least some dimensionality reduction may occur in these spaces. 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, so that the inverse mapping error should be minimal. 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. How to determine the reduced-dimensionality d is not clear; one approach may be to consider several d's and select that which minimizes the residual bijection error $(r_d) = \sum_i ||f_d^{-1}(f_d(X_i - X_i)||$,[Martin and Backer, 2005] where $f_d : X \rightarrow Y$ is the map produced by LLE. By observing the behavior of r_d for different values of d shown in Fig. 8 we can suggest the intrinsic dimension for the universal motor is most likely 2; i.e. the initial space of 8 parameters can, given these optimization conditions, be reduced to two incipient "symbols".

4 Conclusion

The main contributions of this work is the proposal that non-linear manifold learning may constitute an important step in discovering latent relationships among the many parameters that define how the world works. A key constraint is our incomplete characterization of the situations in which such a lower-dimensional characterization would exist.

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.

The argument presented here implies that in the long run, to create viable computer vocabularies for design or AI, 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 computationally available today 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.

[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.

[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.

[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.

[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.

[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.

[McDonough *et al.*, 2003] L. McDonough, S. Choi, and J.M. Mandler. Understanding spatial relations: Flexible infants, lexical adults. *Cognitive Psychology*, 46(3):229–259, 2003.

[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, 19881996] G Pahl and W Beitz. *Engineering Design:A Systematic Approach*, pages 199–400. The Design Council/Springer-Verlag, London/Berlin, 1988/1996.

[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.

[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.