Proceedings of the ECCS 2005 Satellite Workshop: *Embracing Complexity in Design*, 17th November 2005, Cité Internationale Universitaire de Paris.
Edited by Jeffrey Johnson, Theodore Zamenopoulos and Katerina Alexiou, Department of Design and Innovation, The Open University, UK.

Multidimensional Multilevel Networks in the Science of the Design of Complex Systems

Jeffrey Johnson

Department of Design and Innovation, The Open University
Milton Keynes, MK7 6AA, UK
j.h.johnson@open.ac.uk

ABSTRACT

Assembling parts under relations to form wholes is the *Fundamental Construct of Multilevel Systems*. Intermediate assemblies themselves become assemblies, defining multilevel structure. If the set of parts exists at one level, the set structured by an assembly relation exists at a higher level. The structured set with its assembly relation defines a multidimensional simplex at this higher level. A simplex is a multidimensional generalisation of a link in a network: a relation between three things being a triangle (two-dimensional simplex), a relation between four things being a tetrahedron (three-dimensional simplex), and so on. Sets of simplices have a multidimensional connectivity that can be analysed by *Q-analysis*, and a more refined method called *star-hub* analysis. Star-hub systems also have a Galois lattice structure. Connectivity between simplices acts as a kind of relatively static *backcloth* for more dynamic patterns of numbers called the *system traffic*. Relationships between numerical mappings constitute the *Order-I* dynamics of the system, while changes in the backcloth constitute *Order-II* dynamics in the system. Multidimensional connectivity constrains the horizontal intra-level Order-I dynamics and the Order-I inter-level dynamics. Order-II dynamics concern the *building* of structure and the *annihilation* of structure, and are discrete and non-linear. A theory of design is presented using this multidimensional multilevel network theory. Designers *build* structures in bottom-up and top-down fashion. Top-down involves hypothesising sets of parts and relationship to aggregate into higher level abstract constructs. Bottom-up involves assembling real things into realisable structures under explicit relations. As top-down meets bottom-up, abstractions are instantiated with tangibilities, and eventually the whole design becomes grounded in tangible things. This is the *blueprint* stage at which the design can be fabricated. To achieve the blueprint it is necessary to follow a dynamic creative process, the *design process*, which is sensitive to initial conditions, computationally irreducible, path-dependent and characterised by emergence and coevolution between the designed system and the requirements and specification of that system. This is a *science of the artificial*: if the designer creates a system that did not exist before, they are the first person to accumulate and synthesise knowledge about that system. Thus the designer acts as a scientist, by building the *representation* of the system, making *hypotheses* about the system within the *language* being constructed, performing *experiments* on the system, and synthesising this into a *theory* of the system and its dynamics. Many scientists interested in complex artificial systems are motivated by the possibility of using that scientific knowledge to manipulate the system, either by designing new systems, or modifying and managing the behaviour of existing systems. Thus not only are the designers of artificial systems scientists, but the *scientists of artificial systems are designers*. During the meetings of the *Embracing Complexity in Design* cluster it has become clear that designers across the disciplines share a culture based on the creation of new systems and the management of existing systems. In particular the *design process* is common to all design domains, from graphic design through architecture through software to engineering design. This culture informs the particular design process, supporting creativity and divergence, and leading to convergence and delivery of results. It is suggested that scientists of the artificial would benefit from accepting that they are acting as designers, and that complexity science has much to learn from the design community.

Keywords: multilevel, multidimensional, systems, hierarchy, networks, simplicial complexes, Q-analysis, backcloth, traffic, design, artificial systems, synthetic systems, complex systems.

1. INTRODUCTION

This paper will develop the argument that design is at the heart of creating a science of complex systems. Let us make a distinction between natural systems and artificial systems: *natural systems* will be those that exist without human intervention; *artificial systems* will be those that exist as a result of accidental or deliberate human intervention. Natural systems are studied as they are. Artificial systems can be physical objects such as the products one buys, they can be human systems such as a choir, and they can be socio-technical systems such as cities, universities, armies, and businesses which are made up of physical and human parts. Artificial systems can be studied as they are with no intention of introducing change, as in anthropology, but more often they are studied in the context of what they *ought* to be[1]. Disciplines such as anthropology, psychology, and sociology can be free of judgements of what human systems ought to be, but often they feed into policy inducing change. Disciplines such as city planning and peace studies are explicitly linked to policy and managing socio-technical dynamics. Systems that do not already exist are created by *design*.

The designer of a completely new system is the first person to know anything about that system. The design process involves collecting information as the system is created, and the designer is the first person to bring that knowledge together into a *theory* of the new system. *Designers are the first scientists to investigate the systems they create.*

Although the motivation in studying complex systems may involve pure scientific curiosity, in many cases it includes the desire to change systems and system behaviour. In other words, for artificial systems *science proceeds through design*. There are many examples: the biochemist *designs* the molecule; the physician *designs* the treatment; the roboticist *designs* the robot; the planner *designs* the city; the administrator *designs* the organisation; and so on.

At the heart of design is the idea that one can take the 'right' set of parts and put them together the 'right' way to produce a system with certain pre-specified properties[2]. As will be seen, finding the right parts and the right way to put them together is a non-trivial process, involving the construction of an explicit *description* of the system and the accumulation of knowledge about the system.

The design process involves clarifying requirements and specifications, using various methods to find the 'right' set of parts and the 'right' way to put them together to satisfy the specification. At the end of the process there is a *blueprint*, detailing all the parts and the way they must be put together.

A scientist can make three kinds of observation:

(1) observing that something *exists*

(2) counting and assigning *numbers* to things

(3) observing *relationships* between things

Traditional science has focused on the first two of these, but increasing it is realised that *relationships* are important in complex systems. For many years relationships in human systems have been studied using network theory, and it is now clear that network properties play a fundamental role in the behaviour of complex systems[3]. Mathematical relations play a fundamental role in this paper, as a means of making the multilevel nature of systems well defined, and defining the multidimensional spaces on which patterns of numbers can be defined. The dynamics of systems can be played out both through changes in the numbers (Order-I dynamics) or changes in the multidimensional structure (Order-II dynamics)[4].

In design one often *observes in the mind's eye* things that do not exist, *i.e.* designers envisage things that *could* exist but do not yet exist. But designers do not do this instantaneously: *e.g.* aeronautical engineers do not suddenly find fully instantiated designs for a new aeroplane in their heads, architects do not suddenly find fully instantiated designs for a building in their heads, and electrical engineers do not suddenly find fully instantiated designs for circuit boards in their heads. These designs emerge from a process – the *design process*.

In this paper I will sketch a theory of how the creative design process works. For complex systems it involves building a language to represent that system. Often this involves creating new parts and giving them names, and assembling them to create a multi-level ensemble. Part of the process is bottom-up – putting together existing things under new or known relationships. Another fundamental part of the process is top-down, replacing high level uninstantiated abstractions with more detailed abstractions at a lower level. Eventually the bottom-up and top-down meet, and the design becomes a fully instantiated blueprint, with every construct at every level *grounded* in elements that exist in some observable way. To achieve this involves a lot of reasoning about how the parts might interact and what their emergent properties might be. Occasionally higher level reasoning in abstractions leads to errors. As the design is instantiated at lower levels things that are expected to fit fail to do so. Such design problems may be solved by ingenuity, but when they cannot, the design has to back-up with higher level assumptions being modified. Generally this is highly undesirable, implying loss of time, disruption to schedule, and related expense.

In the context of this theory of design, I will claim that the science of artificial systems follows the path of the designer. It is clear that designers are the first scientists to build theories of the artificial systems they create. I will argue that scientists investigating artificial systems with a view to applying that knowledge *are* designers, whether they know it or not. In other words, the design process *is* the scientific process, and the systematic body of knowledge known as 'design research' can be extremely valuable to those engaged in the science of artificial systems.

2. MULTILEVEL SYSTEMS

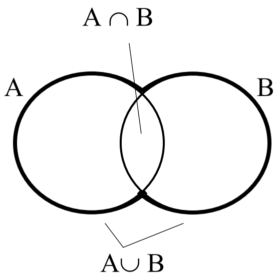


Figure 1. Euler circles representing sets and set operations

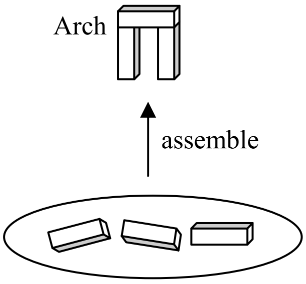


Figure 2. Assembling a set of parts to form a whole

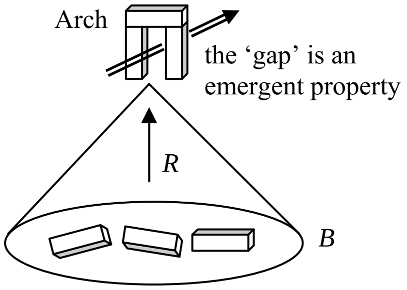


Figure 3. A hierarchical cone

Figure 1 shows the use of *Euler circles* to represent sets and their intersection and union. These diagrams are used extensively in set theory, and we use them to represent sets of parts. In our diagrams the Euler circles will usually be draw in perspective, and so appear as *Euler ellipses*. Figure 2 uses this idea to show how the elements of a set of a set of blocks can be assembled to form a structured object called an *Arch*. The set of parts is represented by the blocks enclosed in an Euler ellipse. These are then assembled to form a structure represented by its *name*, here Arch. Using the convention that higher level objects appear higher on the page than lower level objects, one can draw a *hierarchical cone*, as illustrated in Figure 3, showing how the parts aggregate into the structure under the *assembly relation*, R. If B is the set of blocks, we can write $R(B)$ as the result of applying R to B, so that $R(B)$ = Arch.

Figure 4 shows the *Fundamental Construction of Multilevel Systems* in which wholes are assembled from sets of parts under an *assembly relation*. The whole may have properties not possessed by its parts, *e.g.* the arch has a 'gap' between its vertical support blocks. These 'emerge' from the construction and are called *emergent properties*. The arch is a structured set, denoted $\langle b_1, b_2, b_3 ; R \rangle$. We use the convention that an object and its name can be used interchangeably, so $R(B) = \langle b_1, b_2, b_3 ; R \rangle = Arch$.

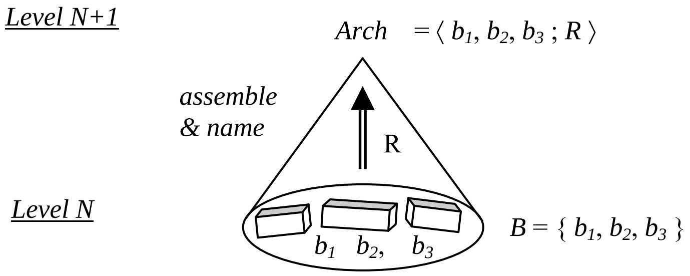


Figure 4. The Fundamental Construction of Multilevel Systems

In general systems have many levels, as illustrated in Figure 5. Here *lines* at the lowest atomic level, which is denoted *Level N*, are assembled under the relations R_1, R_2, and R_3 to form blocks of various kinds at a *higher hierarchical level*, denoted *Level N+1*. These blocks are assembled by relations R_4, R_5, and R_6 to form shapes of various kinds at another higher level, denoted *Level N+2*. Finally these three shapes are assembled by the relation R_7 into the final shape at what is called *Level N+3*.

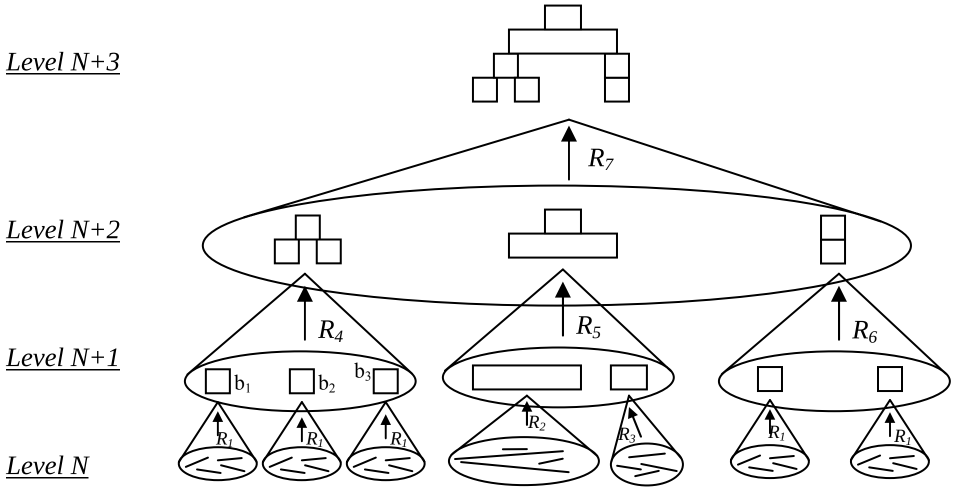


Figure 5. Hierarchical aggregation in multilevel systems

The notation *N+k* is used to show that the levels are not absolute. In general there are many ways to aggregate through the hierarchy of assembly, and it common to introduce a new level between two existing levels to represent intermediate assemblies. In design there may be many ways to put things together, but usually just one of them is selected for the final blueprint.

3. RELATIONAL STRUCTURE

R	a_1	a_2	a_3	a_4
a_1	0	1	1	0
a_2	0	0	1	1
a_3	1	0	0	0
a_4	0	0	0	0

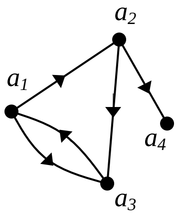


Figure 6. The graph of a relation from a set to itself

R	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
a_1	1	1	1	0	0	0	0	0
a_2	0	1	1	1	0	0	0	0
a_3	0	0	1	1	1	1	0	0
a_4	0	0	0	0	0	1	1	1

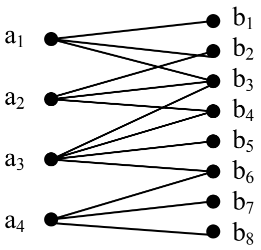


Figure 7. A bipartite graph between sets *A* and *B*

A *relation*, *R*, between two sets *A* and *B* is an operational rule for deciding of each *a* in *A* and each *b* in *B* whether or not *a* is related to *b*. A relation can be recorded as an *incidence matrix*, as illustrated in Figures 6 and 7. In Figure 6 the relation is between a set and itself, while in Figure 7 it is between two different sets. Next to each matrix is a diagram showing the elements as points and the relationships by lines linking points. Technically, a *graph* is a set of *vertices*, *V*, and a set of pairs of elements of *V*, denoted $\langle v_1, v_2\rangle$, called *edges*. In Figure 6 the graph has the vertices $\{ a_1, a_2, a_3, a_4 \}$ and the edges $\{ \langle a_1, a_2\rangle, \langle a_1, a_3\rangle, \langle a_2, a_3\rangle, \langle a_2, a_4\rangle, \langle a_3, a_1\rangle \}$. When the edges are *directed* with $\langle v_1, v_2\rangle \neq \langle v_2, v_1\rangle$ they may be drawn as arrows, as shown in Figure 6. Such directed graphs are often called networks. Graphs and networks have many useful properties. For example, the *degree* of a vertex is the number of incident edges, and is a direct measure of how many things that vertex is related to. One of the most important features of networks is the *connectivity*, which allows things to flow through the system, and can be responsible for one part of the system influencing another.

In general relations are defined between different sets, as illustrated in Figure 7. This gives rise to what is called a *bipartite graph*. In Figure 7 the vertices are $\{ a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8 \}$ and the edges are $\{ \langle a_1, b_1\rangle, \langle a_1, b_2\rangle, \langle a_1, b_3\rangle, \langle a_2, b_2\rangle, \langle a_2, b_3\rangle, \langle a_2, b_4\rangle, \langle a_3, b_3\rangle, \langle a_3, b_4\rangle, \langle a_3, b_5\rangle, \langle a_3, b_6\rangle, \langle a_4, b_6\rangle, \langle a_4, b_7\rangle, \langle a_4, b_8\rangle \}$.

The bipartite graph of a relation between two different sets may appear rather uninteresting, but the degrees of the vertices are related to a much richer structure, namely a *hypergraph*. Given a set V of vertices, a *hypergraph* is that set of vertices together with a class of subsets of the vertices, called *hyper-edges*. For example, the relation in Figure 7 defines a class of subsets of B given by $\{ a_1 \rightarrow \{b_1, b_2, b_3\}, a_2 \rightarrow \{b_2, b_3, b_4\}, a_3 \rightarrow \{ b_3, b_4, b_5, b_6\}, a_4 \rightarrow \{b_6, b_7, b_8\} \}$, which can be considered to be a hypergraph. A more interesting hypergraph also includes the intersections of the sets, as shown in Figure 8. Let this hypergraph be denoted $\mathrm{H_A}(B, R)$. This hypergraph is defined by the elements of A being related to subsets of B. The *conjugate hypergraph*, $\mathrm{H_B}(A, R)$, is defined by the elements of B being related to subsets of A, as shown in Figure 9.

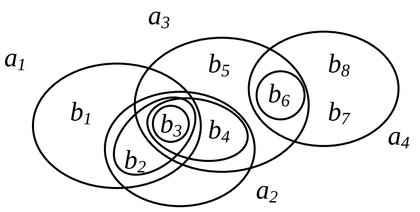


Figure 8. The hypergraph $\mathrm{H_A}(B, R)$

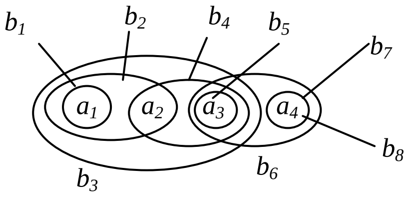


Figure 9. The hypergraph $\mathrm{H_B}(A, R)$

These two hypergraphs have an intimate interrelationship. Every hyper-edge of $\mathrm{H_A}(B, R)$ is associated with a hyper-edge in $\mathrm{H_B}(A, R)$, with all elements related. For example $\{ a_1, a_2 \}$ is associated with $\{b_2, b_3\}$, with both of a_1 and a_2 related to both of b_1 and b_2. These pairs of sets can be arranged in what is called a *Galois lattice*, illustrated in Figure 10. Here $\varnothing$ is the empty set, and the expressions $(\{ a_1, a_2, a_3, a_4 \}, \varnothing)$ means that no member of B is related to every member of A, while and $(\varnothing, \{b_3, b_6, b_7, b_8\})$ means that no member of A is related to every member of B.

A *lattice* is a partially ordered set is which every two elements have a supremum and an infimum. In Figure 10 we say that $(x, y) < (x', y')$ if $x \subset x'$ and $y \supset y'$. For any (x, y) and (x', y'), $(x \cup x', y \cap y')$ exists by construction. It is the supremum of (x, y) and (x', y'), since $(x, y) < (x \cup x', y \cap y')$ and $(x', y') < (x \cup x', y \cap y')$. Similarly $(x \cap x', y \cup y')$ belongs to the system, and is the infimum of (x, y) and (x', y'), since $(x \cap x', y \cup y') < (x, y)$ and $(x \cap x', y \cup y') < (x', y')$. In Figure 9 a line is drawn between each pair of sets and their supremum and infimum, to produce the lattice.

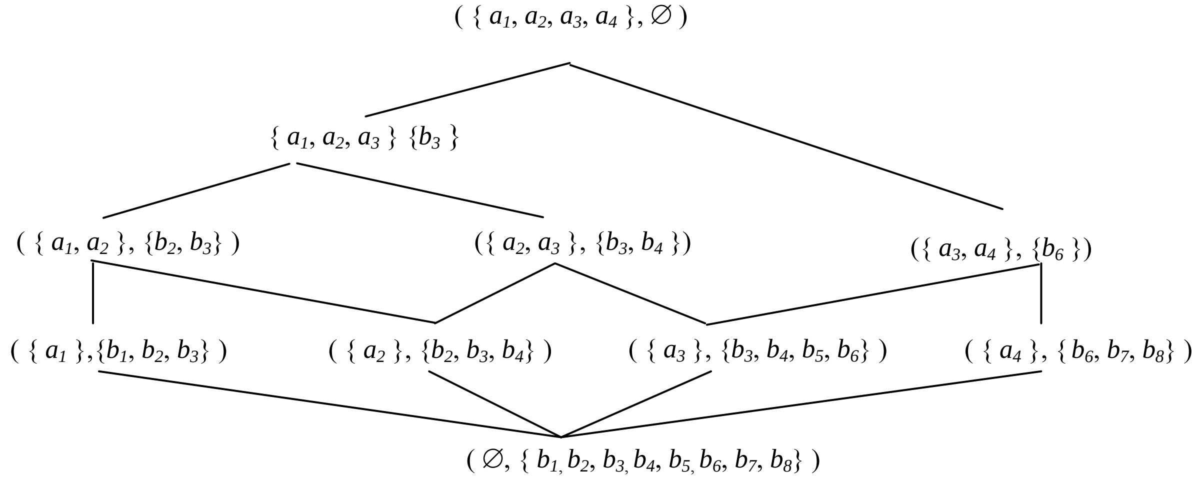
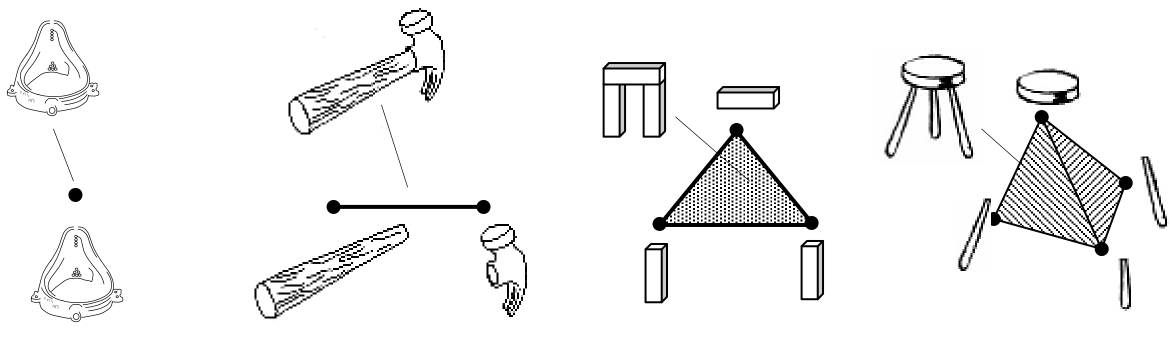


Figure 10. The Galois lattice of the relation R between A and B.

4. MULTIDIMENSIONAL NETWORKS AND Q-ANALYSIS

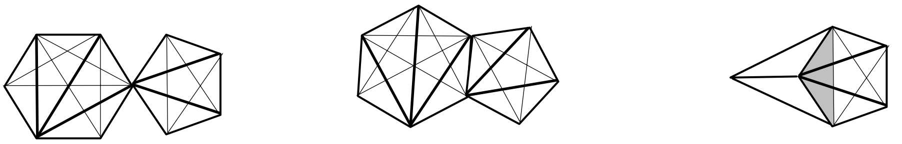
The Galois lattice construction sketched in the previous section is a powerful way of investigating relations in systems. However it is essentially *set theoretic*. In the light of Section 2, the underlying hypergraphs can represent the *sets* of components in multilevel systems, but they cannot also represent the *structured sets of components*. For this we need to make the assembly relations explicit, and develop the algebraic properties of the relations as a natural multidimensional generalisation of network theory.



(a) vertex or 0-simplex (b) line or 1-simplex (c) triangle or 2-simplex (d) tetrahedron or 3-simplex

Figure 11. Multidimensional simplices as the generalisation of links in a network

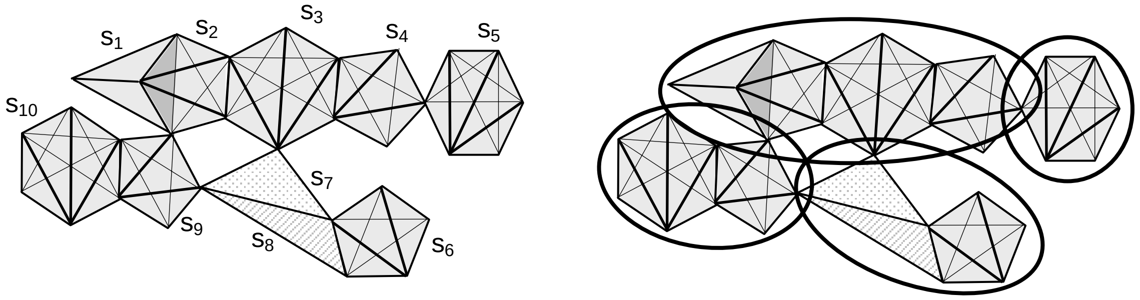
In Section 2 we introduced the notation $\langle b_1, b_2, b_3 ; R \rangle$ to represent a set of three blocks assembled into an arch. This is clearly more that the set $\{ b_1, b_2, b_3 \}$ which we drew as an Euler circle. In Figure 11(c) we show $\langle b_1, b_2, b_3 ; R \rangle$ drawn as a *triangle*, a two-dimensional object. This is the natural generalisation of using a one-dimensional *line* to represent a relation between two things, and a zero-dimensional point, or vertex, to represent a single thing. The structure $\langle b_1, b_2, b_3 ; R \rangle$ is called a *simplex* and it has *dimension* two. Figure 11(d) shows an object made up of four parts, being represented by a three-dimensional tetrahedron. In general a relation between n vertices is represented by an (n-1)-dimensional polyhedron in a multidimensional space. A simplex is a *face* of another if its vertex set is a subset of that simplex. A set of simplices with all its faces is called a *simplicial complex*.



(a) 1 shared vertex (b) 2 shared vertices (c) 3 shared vertices

Figure 12. Simplices can be connected at different dimensions

Simplices have interesting connectivity properties. Figure 12 shows how simplices can share different numbers of vertices, and that the more vertices they share, the more *highly connected* they are. The intersection of two simplices is called their *shared face*. If the shared face has dimension q, the simplices are said to b q-*near*. Thus the simplices in Figure 12(a) are 0-near (a single vertex has dimension zero), those in Figure 12(b) are 1-near (two vertices make a one-dimensional line), and those in Figure 12(c) are 2-near (three vertices make a two-dimensional triangle). We say two simplices are q-*connected* if there is a chain of pairwise q-*near* simplices between them.



(a) the simplicial complex

(b) components connected by lines

Figure 13. A simplicial complex of connected simplices[4].

A *Q-analysis* of a simplicial complex is a listing of its *q-connected components*. For example, for Figure 13(a):

$q = 5$ $\{s_3\}, \{s_5\}, \{s_{10}\}$

$q = 4$ $\{s_2\}, \{s_3\}, \{s_4\}, \{s_5\}, \{s_6\}, \{s_9\}, \{s_{10}\}$

$q = 3$ $\{s_1\}, \{s_2\}, \{s_3\}, \{s_4\}, \{s_5\}, \{s_6\}, \{s_9\}, \{s_{10}\}$

$q = 2$ $\{s_1, s_2\}, \{s_3\}, \{s_4\}, \{s_5\}, \{s_6\}, \{s_7\}, \{s_8\}, \{s_9\}, \{s_{10}\}$

$q = 1$ $\{s_1, s_2, s_3, s_4\}, \{s_5\}, \{s_6, s_7, s_8\}, \{s_9, s_{10}\}$ (illustrated in Figure 13(b))

$q = 0$ $\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}$

Figure 14 shows the relationship between four English public houses and the customers that frequent them. Typically people like to go to more than one pub on different days for the variety it brings. Suppose that someone who likes The Swan, the simplex on the left, knows a very good joke. When he gets to the Swan pub he tells it to the people who happen to be in that day. They may tell the joke to other people in the pub, and it is likely to be transmitted to everyone in the Swan before the day is finished. The next day, one of those people in the Swan might visit the Anchor, and tell the joke there. Again the joke gets transmitted within the pub. The next day one of the people from the Anchor might visit the Goat pub, and tell the story there. In this way the joke can get transmitted from the Swan pub to the Bull pub, even though they have no customers in common. This illustrates how information can pass through social structure determined by relations. In general the more highly connected the structure, the more rapidly information is transmitted[4].

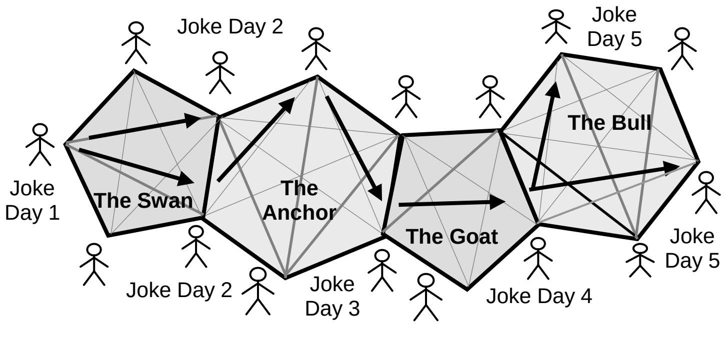


Figure 14. The transmission of information on the pub-customer structure[4]

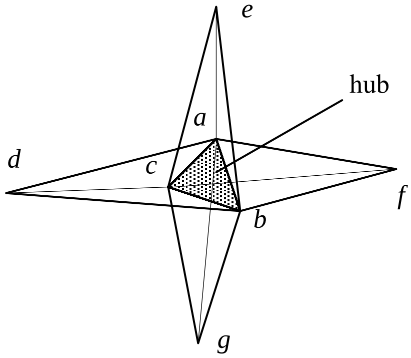


Figure 15. A star-hub configuration

Q-analysis is based on the connectivity of pairs of simplices, but it is more general to consider the intersection of sets of simplices. This leads to the concept of stars and hubs, as illustrated in Figure 15 where the simplices, $\langle a, b, c, d \rangle$, $\langle a, b, c, e \rangle$, $\langle a, b, c, f \rangle$, and $\langle a, b, c, g \rangle$ share the face $\langle a, b, c \rangle$. The set of the four simplices is called a *star* and their intersection is called their *hub*.

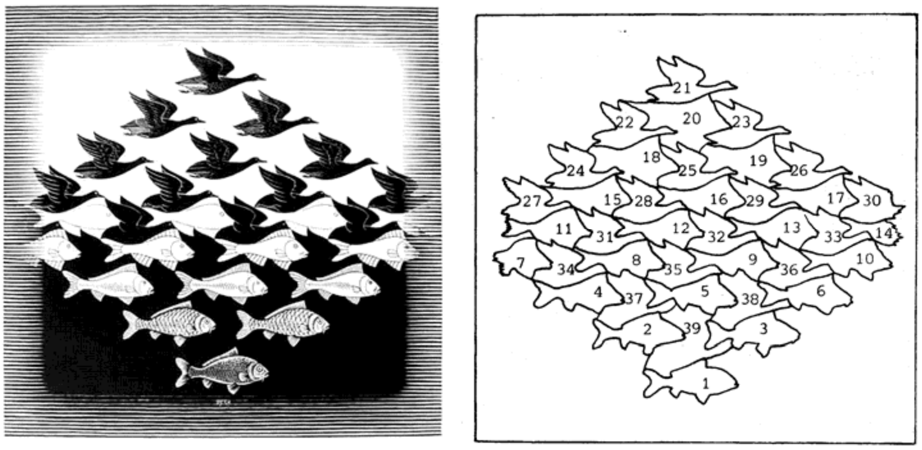


Figure 16. An analysis of Escher's Sky and Water[5]

If the rows and columns of an incidence matrix are arranged appropriately, hubs correspond to rectangular regions in which the values are all 1. For example, Figure 16shows thirty nine shapes abstracted from Escher's Sky and Water. The relation between these and the set of visual features {scales, mouth, gills, fish-tail, fins, fish-shape, eye, duck-shape, two-wings, feathers, beak, legs} is given by the incidence matrix below.

	1	2	3	4	5	6	8	9	10	11	12	13	7	21	22	23	24	25	26	28	29	31	32	33	27	30	34	35	36	37	38	14	15	16	17	18	19	20	39
scales	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mouth	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gills	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fish-tail	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
fins	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fish-shape	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
eye	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
duck-shape	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0
two-wings	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
feathers	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
beak	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
legs	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The matrix shows that the simplices for the shapes numbered 1 to 6 are all the same, with a block of 1s corresponding to the features { scales, mouth, gills, fish-tail, fins, fish-shape, eye },with all the shapes being good examples of fish. There is another group, shapes 1 to 13 all related to the hub features {fins, fish-shape, eye}, which includes the less perfect fish shapes 8 to 13. The following major star-hub pairs can be abstracted as follows:

$$\langle 1, 2, 3, 4, 5, 6 \rangle \leftrightarrow \langle \text{scales, mouth, gills, fish-tail, fins, fish-shape, eye} \rangle$$

$$\langle 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13 \rangle \leftrightarrow \langle \text{fish-tail, fins, fish-shape, eye} \rangle$$

$$\langle 21, 22, 23, 24, 25, 26, 28, 29 \rangle \leftrightarrow \langle \text{eye, duck shape, two wings, feathers, beak, legs} \rangle$$

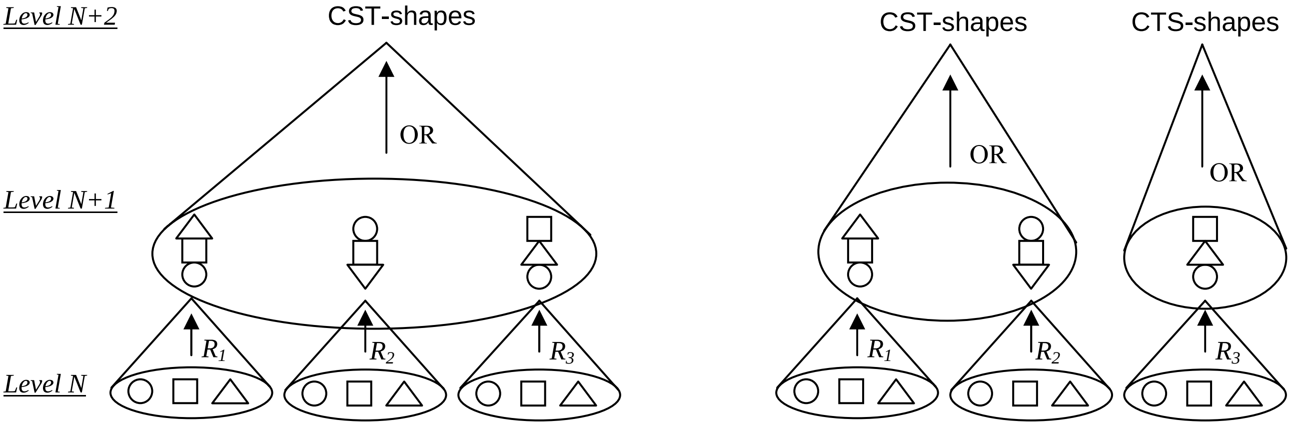
$$\langle 21, 22, 23, 24, 25, 26, 28, 29, 31, 32, 33, 27 \rangle \leftrightarrow \langle \text{eye, duck shape, two wings} \rangle$$

The final weak structure is given by the shapes 15-19 which are related only to fish-shape, and 34-38 which are only related to duck shape. Thus Escher has the fish shapes highly connected at the bottom of the picture and the bird shapes highly connected at the top. Moving towards the centre the connectivity is still high but less, and it gets relatively low as the shapes lose features and become the background.

5. OR - AGGREGATION IN MULTILEVEL SYSTEMS

The assembly relations discussed in the previous sections required *all* the elements to be present for the relation to hold between them, and introduced other conditions too. For example, in Figure 4 we require b_1 AND b_2 AND b_3 for the assembly into an arch. We call this kind of hierarchical aggregation an *AND aggregation*. This is different to another kind of aggregation, the *OR aggregation*, in which just one element is sufficient to move up the hierarchy.

For example, consider the Circle, Square and Triangle shapes in Figure 17. Let them be denoted by C, S, and T respectively. These have been assembled in three different ways by the relations R_1, R_2, and R_3. Are the structures the same, *i.e.* is $\langle C, S, T ; R_1 \rangle = \langle C, S, T ; R_2 \rangle = \langle C, S, T ; R_3 \rangle$? In some obvious sense they are not all equal because they are all different. However, for the purpose in hand it may not matter the centre shape an upside-down version of the leftmost shape. Indeed it may not matter that the rightmost shape has the blocks arranged vertically in a different order. In other words, for the particular purpose in hand all these shapes might be considered *equivalent*, each being an example of what might be called a CST-shape.



(a) any one of the shapes is sufficient for OR-aggregation

(b) OR-classes are usually defined by purpose

Figure 17. Assembling shapes under an OR aggregation

For another purpose it may indeed matter that the blocks are assembled in a different order, leading to two classes: the CST-shapes = $\{ \langle C, S, T ; R_1 \rangle , \langle C, S, T ; R_2 \rangle \}$ and the CTS-shapes = $\{ \langle C, S, T ; R_3 \rangle \}$. This illustrates the general point that classifications are usually motivated by some purpose. Generally OR aggregations are a matter of definition to achieve a particular purpose.

There are two ways to define a class. The first is by *extension* or by listing the elements. For example, let *X* be the set {a, e, i, o, u}. Then some people call this the set of vowels. Others argue that *X'* = {a, e, i, o, u, y} should be called the set of vowels. The important thing is that by listing the elements, both *X* and *X'* are well defined. Which of them should be called the set of vowels is a matter of definition by *intension*.

Definition by *intension* involves giving a defining property(s). For example, Wikipedia defines vowel as follows: "In phonetics, a vowel is a sound in spoken language that is characterized by an open configuration of the vocal tract where there is no build-up of air pressure above the glottis. This contrasts with consonants, which are characterized by a constriction or closure at one or more points along the vocal tract. The additional requirement is that vowels function as syllabic units: it is this criterion that helps distinguish vowels from approximants (in some languages approximants may be slightly more constricted or less intense)", where "Approximants are speech sounds that could be regarded as intermediate between vowels and typical consonants. In the articulation of approximants, articulatory organs produce a narrowing of the vocal tract, but leave enough space for air to flow without much audible turbulence. Approximants are therefore more open than fricatives. This class of sounds includes lateral approximants like [l], as in lip, and the so-called semivowels [y] and [w] in yes and well." (http://en.wikipedia.org/wiki/Vowels, accessed 30-10-05).

In contrast to Wikipedia, http://arapaho.nsuok.edu/~gieseb/4323/phonicsterminology.html (accessed 30-10-05) gives an extensional definition "The letters a, e, i, o, and u represent vowel sounds", and then muddies the water by adding "and the letters w and y take on the characteristics of vowels when they appear in the final position in a word of syllable. The letter y also has the characteristics of a vowel in the medial (middle) position in a word of syllable."

These definitions are summarized in Figure 18.

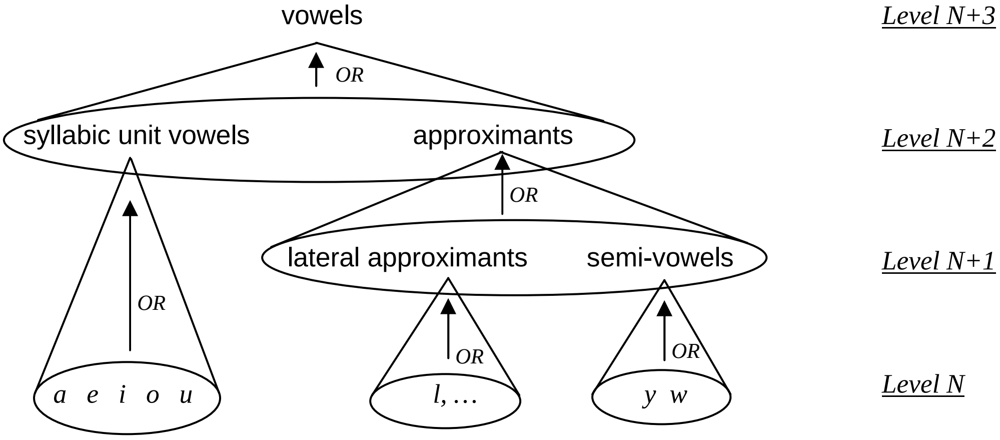


Figure 18. Higher level intensional definitions grounded in instantiated extensional definitions at lower levels.

Set definition by intension can be written in the form $X = \{\ x \mid P(x) = \text{TRUE}\ \}$, where P is a proposition specifying the properties that candidates for membership of X must satisfy to be members. Thus we have vowels = $\{\ x \mid x$ is a syllabic unit vowel OR x is an approximant $\}$, which resolves the term 'vowel' at a lower level. Going down further levels we have vowels = $\{\ x \mid x = a$ OR $x = e$ OR $x = i$ OR $x = o$ OR $x = u$ OR x is a lateral approximant OR x is a semi-vowel), and finally vowels = $\{\ x \mid x = a$ OR $x = e$ OR $x = i$ OR $x = o$ OR $x = u$ OR $x = l$ OR $x = y$ OR $x = w\ \}$.

We say a proposition P is *well-defined* if there is an operational procedure for deciding of any potential candidate for membership, x, that $P(x)$ is *well-formed* in a logical sense, and that there is an operational procedure for deciding whether or not $P(x)$ is true. To illustrate this, consider the set $X = \{\ x \mid x$ is a house extension that is exempt from the building regulations $\}$. The proposition $P(x)$ is true if "x is a house extension that is exempt from the building regulations". This requires an operational way of deciding if x is a house extension. Assuming that x is indeed a house extension, the operational procedures require the application of the following:

$P(x)$ = True according to the Building Regulations 1991 (as amended) if

x has a completely transparent or translucent roof
AND x has extension walls that are substantially glazed
AND x has a floor area not exceeding 30m squared.
AND x is sited at ground level.
AND x is permanently separated from the remainder of the property by means of a door.
AND x has separately controllable radiator (if fitted)
AND x has glazing satisfying the requirements of part N, Schedule 1 (toughened/safety glass).
AND x does not contain any drainage facilities. (i.e. sink, WC, or washing machine)
(Source: http://www.conservatoriesonline.com/planperm.htm, accessed 30-10-05)

Suppose x were an elephant. Then x fails the test that it is a house extension, and does not belong to the set of things exempt from building regulations. Suppose x is a 'my glass conservatory'. Then each of the conditions can be applied meaningfully to 'my glass conservatory' and decided to be true or false. In this case all of the sub-propositions needs to be true for $P(x)$ to be true.

Conventionally hierarchical structure is associated with classification and OR-aggregations. Related to this are the many methods of clustering concepts to form higher level constructs, *e.g.* houses and cathedrals are buildings. Star-hub analysis is very useful for this, but that is outside the scope of this paper. Important through the OR-aggregation is, this paper argues that AND-aggregations are absolutely fundamental in defining the vocabulary of multilevel systems.

6. PART – WHOLE PARADOXES AND RELATIONAL STRUCTURE

The relationship between parts and wholes causes endless unnecessary confusion unless it is made mathematically well-defined. For example, Figure 19 illustrates the conundrum that “Simpson’s Finger belongs to Simpson. Simpson belongs to the Philosophy Department. Simpson’s Finger belongs to the Philosophy Department”. This vernacular way of expressing things seems to produce the very odd conclusion that Simpson’s finger belongs to the Philosophy Department.

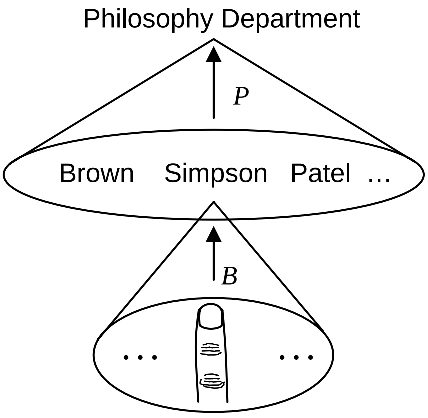


Figure 19. Simpson’s finger belongs to the Philosophy Department.

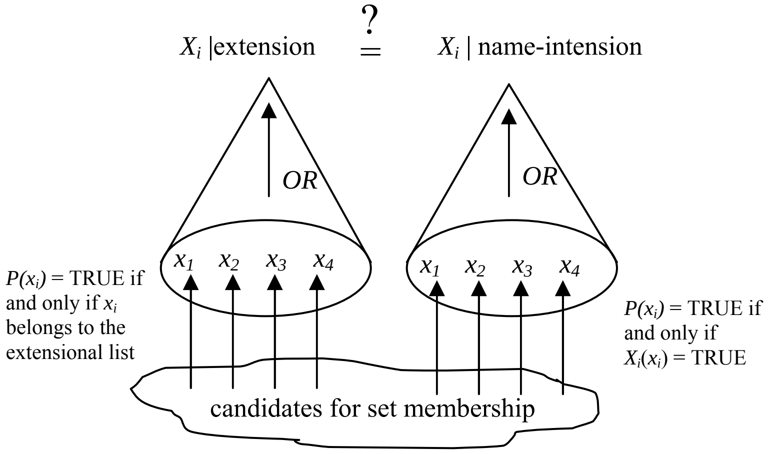


Figure 20 Set names define intensional proposition that may be inconsistent with an extensional definition of the set

Let the assembly relation R take a set of parts, S, to the higher level structure x, R: $S \longrightarrow R(S) = x$. Then define the *R-base of* x, $\text{base}_R(x)$, to be the set S, and write $\text{base}_R(x) = S$. We will also use the notation $R^{-1}(x) = S$, to suggest *disassembling* x into its constituent parts. In other words the *R-base* of a structure, x, is a set of component parts, S, at a lower level. The particular lower level and higher level are implicit in the relation R.

Suppose $R^{-1}(x) = S$, and $S = \{ s_1, s_2, s_3, s_4, \ldots \}$. Suppose that there are component sets C_i and assembly relations R_i, with $R_i(C_i) = s_i$, for $i = 1, \ldots, n$. Then we write $R_i^{-1} \circ R^{-1}(S) = R_i^{-1}(s_1, \ldots, s_i, \ldots, s_n) = R_i^{-1}(s_i) = C_i$. Using this notation, Simpson belongs to P^{-1}(Philosophy Department) and Simpson’s Finger belongs to B^{-1}(Simpson). Nothing surprising there. We also have Simpson’s Finger belongs to $B^{-1} \circ P^{-1}$ (Philosophy Department), which it indeed does. It multilevel systems there are many such compositions, and they reflect relationships, *e.g.* if Simpson hurts his finger he may not attend a departmental meeting, even though his finger normally has no impact on the department.

The ‘paradox’ of Simpson’s finger is artificial and arises from equating the relations B and P under the single term ‘belongs to’. The relation B only has meaning in the context of the other parts of Simpson’s body, because ‘Simpson’ is a meaningless construct unless all his parts are present. Similarly, the relation P is part of the *definition* of the philosophy department.

Generally a relation like P does two things. First it identifies the *set* of component parts, either by extension or by intension. Thus to form the simplex $\langle x_1, x_2, \ldots, x_n ; R \rangle$ it is necessary to have *decision functions* $D_R : x_i \longrightarrow$ {True, False}. When the sets are defined extensionally, the decision function will be of the form $D_R(x_i)$ = True if and only x_i belongs to a given list, X_i. When the sets are defined intensionally, the decision function will be of the form $D_i(x_i)$ = True if and only if $P_i(x_i)$ = True, where P_i is a proposition about x_i with an operational procedure for determining the truth value.

A common source of inconsistency and apparent paradox can arises when sets are given names that attempt to describe their members (Figure 20). For example, a television programme classification scheme[6] had the class ‘sports not requiring equipment’, and gave boxing and wrestling as examples. Unfortunately this produces an inconsistency since { x | x is a sport not requiring equipment } ≠ { boxing, wrestling, …. } because both boxing and wrestling do require equipment.

When a set has an operational definition for determining its members in terms of lower level sets its defining proposition will be said to be *instantiated.* When there is an operational definition determining some of the elements the set will be said to be *partly instantiated*. Once the component sets are instantiated, P has to test if the n-ary relation holds between the vertices. Generally this is much more difficult than testing for parts. If the relation holds, P is said to be instantiated.

7. BACKCLOTH AND TRAFFIC

Apart from relational structure, most multilevel systems have patterns of numerical properties distributed across them, as illustrated in Figure 21. Generally the numbers at lower level aggregate up the hierarchy. This can be by simple linear addition, as shown between *Levels N* and *N+1*, or by non-linear functions, as illustrated between *Levels N+1* and *N+2*. Generally the relational structure of systems is relatively fixed, while the numerical values are relatively dynamic. For this reason we refer to the relational structure as the *relatively static backcloth* and say that it supports the *relatively dynamic traffic* of activity on the system. For example, the topology of a motor car is relatively fixed, while its speed may change considerably over time. Similarly, the infrastructure of the stock market involves many relatively fixed relationship, and this acts as a backcloth supporting the highly dynamic traffic of trades and prices.

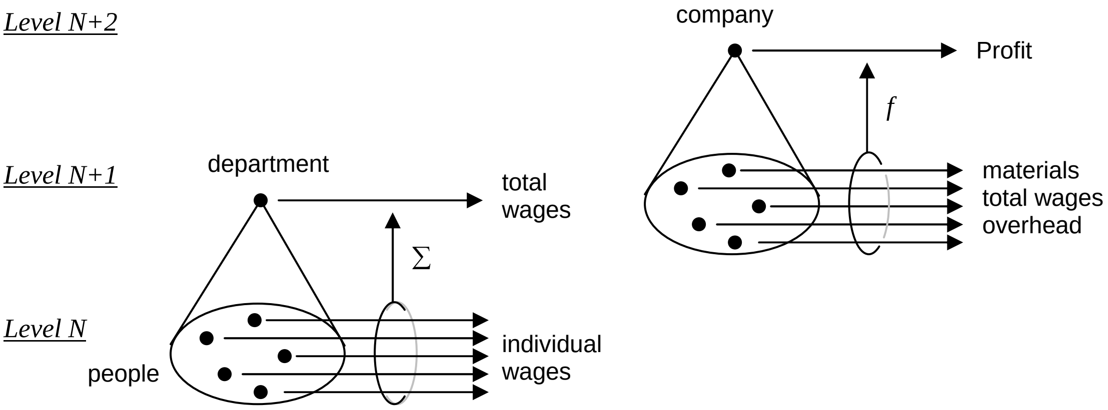


Figure 21. Multidimensional traffic on the multidimensional backcloth[4].

Suppose that a system has been designed and that there is a blueprint with sufficient information to fabricate it. Then that blueprint will specify all the necessary parts at *Level N* and explicitly give all the assembly relations enabling the atomic parts to be built into intermediate level structure or components at *Level N+1*. Similarly the blueprint will give all the assembly information necessary to assemble the system through all the levels, in a bottom-up fashion.

This brings us to the fundamental question: where do the 'right' atomic sets come from, and where do the 'right' assembly relations come from? These are scientific questions, of the same kind that Galileo answered by defining sets of time intervals and distances along an inclined plane, and that Harvey answered by defining sets of arteries and veins. In neither case was defining the 'right' sets a trivial matter, both having occupied other scientists for centuries. In both cases the breakthrough came from postulating the nature of the relations that assemble the parts into the whole. Thus in formulating a theory about a multilevel system, the scientist has to find an answer to the *Intermediate Word Problem*, illustrated in Figure 22.

The Intermediate Word Problem reflects the *top-down* aspects of understanding systems. If the system already exists, one looks at it and tries to see how it is made up of parts.

This *reductionist* approach, contrary to complaints that wholes cannot be understood in terms of their parts, is an essential part of the scientific process. If it were otherwise it would mean that no information about parts of the system would be useful in understanding it. If this were so the system would only have one relevant level, that of the 'The System'. Some systems are like this, *e.g.* the volume-temperature-pressure relationships on a 'fixed mass of gas system'. This system could be represented by a cone, with the set of gas molecules in an Euler circle aggregating into 'fixed mass of gas system', and in principle knowledge of the molecules could be used to calculate the volume, temperature and pressure. In practice the lower level structure is ignored, because the Gas Laws do not require them to be explicit.

8. TOP-DOWN AND BOTTOM-UP REASONING IN DESIGN

If the system does not already exist, the designer has to suggest the parts and how they aggregate between levels. In this paper we will describe this process for design, and argue that it is also the process that has to be used in formulating a science of artificial systems.

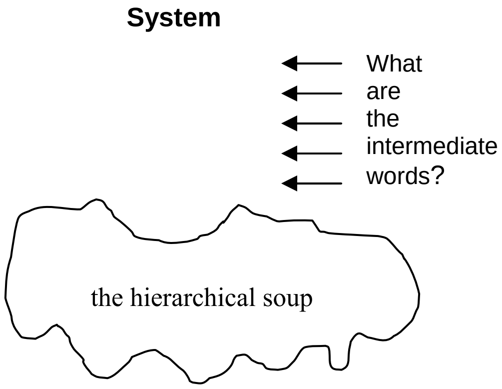


Figure 22. The Intermediate Word problem

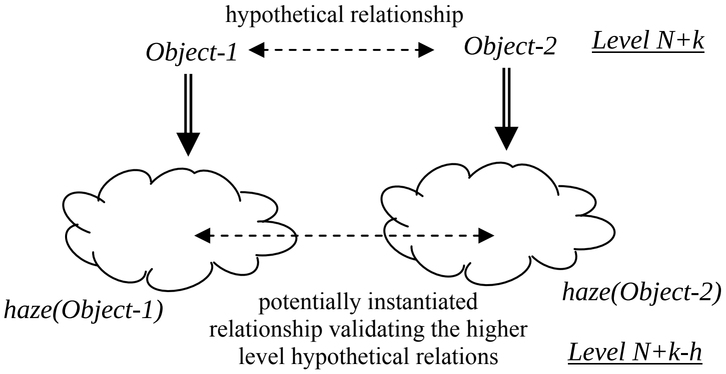


Figure 23. A top-down *hierarchical haze*

The *bottom-up* part-whole assembly approach is complemented by a top-down approach in which designers *analyse* systems in terms of their parts and characteristics. To do this the designer addresses the Intermediate Word Problem (Figure 22). When we look at any system as a whole, our impression of it is a Gestalt. Alongside this, there may be prior knowledge of the system, and we are usually able to see the constituent parts and substructures.

Thus we experience the whole in some sensory way, and we make associations between the whole and pre-existing things in our minds. The prior knowledge that we have forms a 'soup' of information, "a pre-logical primordial source containing the building blocks of all subsequent substructures" [6]. Thus, when analysing a system, the designer has to abstract a vocabulary to represent the system between the uninformative highest level term, "the system", and this vernacular hierarchical soup (Figure 22).

As a design progresses, some things will be totally instantiated at lower level, but some will not. Those things that are not instantiated are *hazy* abstractions, important parts of the design awaiting further information. Even though a part of the system is not instantiated, it may play an important part in reasoning about the system. Relationships are hypothesized between objects whether or not they are instantiated. These hypotheses may or may not be validated as the design becomes grounded and higher level constructs are instantiated in terms of testable lower levels (Figure 23).

These idea are illustrated in Figure 24 for a helicopter. Here the engineer has identified parts of the system and given them *names*. These names can then become part of a formal vocabulary used to describe the system and reason about it. In this hierarchical decomposition, the lowest level is *grounded* in the soup.

Any designed and manufactured object can be represented in terms of a hierarchy of assembly like this. The use of numbers to define levels requires some justification, because this type of hierarchy may not be linear but more like a tree or a lattice. For our purposes it is sufficient to note that if x aggregates into y, then x is at a lower level than y in the representation. Thus when the designer begins, even though nothing may have been decided in terms of actual component parts or assemblies, he or she knows that when they have finished the system will be represented in terms of a specification of its parts and precise instructions on how to assemble them at every level. This is the *blueprint* that the designer hands over for fabrication. The blueprint communicates the necessary and sufficient information to build the system. When it is complete every part in the hierarchy is explicitly identified and named.

The design process can be viewed as the process of creating the blueprint. Initially the designer starts with some hazy idea of the 'the system' at some high level, in the context of a soup of terms and prior knowledge from which the explicit vocabulary of the design must be abstracted (Figure 22). On reflection the designer will postulate various subsystems and components as intermediate words. Some of these may be completely unspecified in terms of their details, and they are represented in Figure 23 by hazy cloud-like symbols. Some parts of the system may be entirely specified from the start, such as the tail rotor in Figure 25(a). For example, this may be a legacy subsystem that must be used in the design as part of the specification.

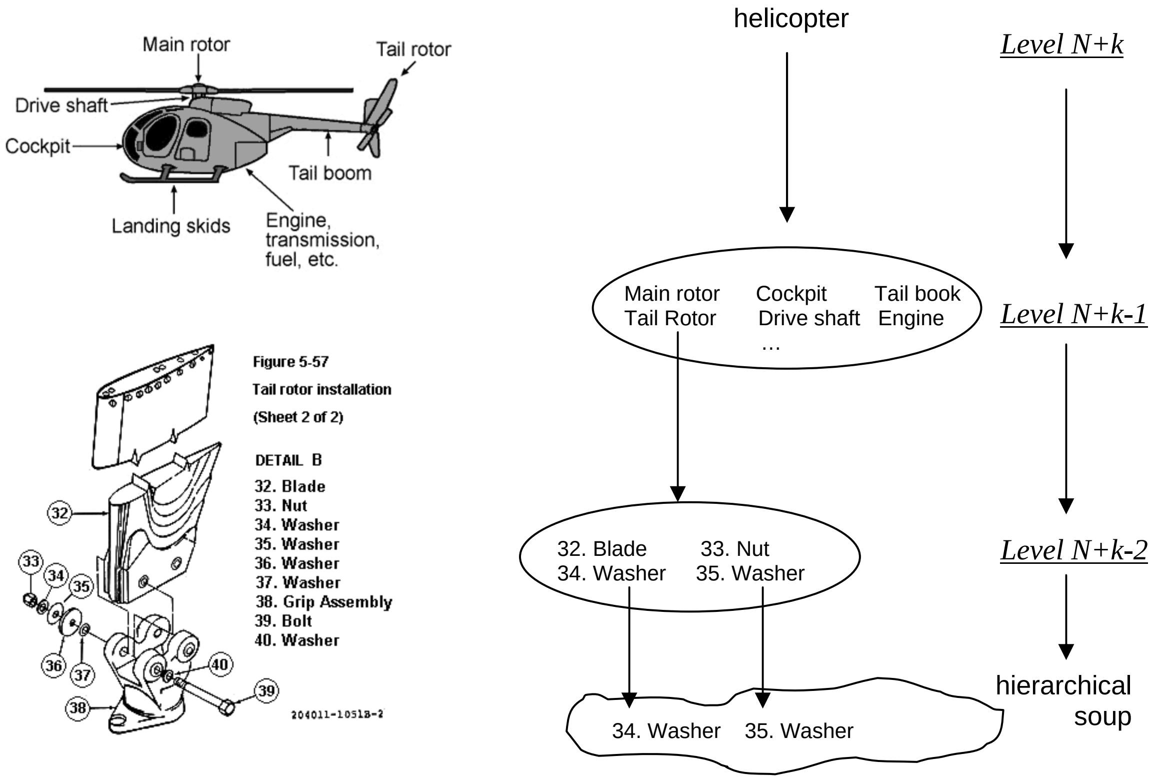
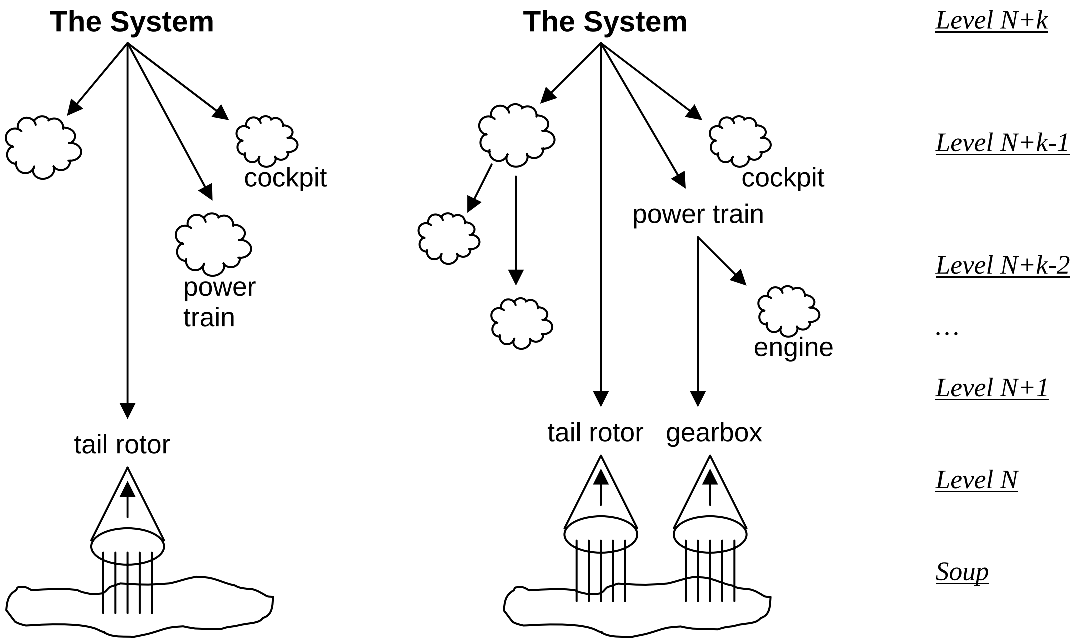


Figure 24. Abstracting intermediate words for a helicopter (Source: Johnson, 2005)



(a) Initial sketch (b) partial instantiation

Figure 25. Design as the process of building an ontology (Source: Johnson, 2005)

As the design proceeds, the hazy subsystems may be worked up in more detail. This could include postulates of other hazy clouds at lower levels, or it could include fully instantiated components, as illustrated by the gearbox in Figure 25(b). In principle the design proceeds by the designer working down the hierarchy, adding more explicit detail until the whole design is instantiated by particular components, and subsystems built from explicitly specified components and sub-assemblies.

Some part of the design process may be bottom-up, as the designer creates new assemblies with the necessary properties to be components at higher level. In practice one can imagine designers constantly scanning the design in both top-down and bottom-up modes, trying to connect the higher level abstractions to the lower level realities.

As the design proceeds and more of the hazy clouds become instantiated with fully defined objects, implicit assumptions may turn out to be problematic. For example, in Figure 25(b), the particular the rotor and the particular gearbox may have undesirable interactions. This unexpected emergent property becomes a design problem. Perhaps the designer can find some new way of putting the components together that obviates the problem. If not, it may be necessary to back-track, abandoning the particular tail rotor or the particular gearbox. In extreme cases the original specification that the particular tail rotor must be used may have to be changed, so that the fully specified component has to be replaced by a hazy cloud at a higher level in the representation.

9. THE DESIGN CYCLE

The design process has been described as that of building a multilevel language to represent the artefact being designed, with top-down – bottom-up interactions until both meet and a consistent blue print is achieved. In practice the process is more complicated than this. The design process begins with a perceived need or *requirements*. These are translated into a *specification* which the designed system should meet. Even at the blueprint stage, when fully instantiated, the design may be evaluated and found deficient. Thus everything said so far has to be put in the context of the *design cycle*. There are many models of the design cycle in the literature, but they all have the generate-evaluate loop shown in Figure 26.

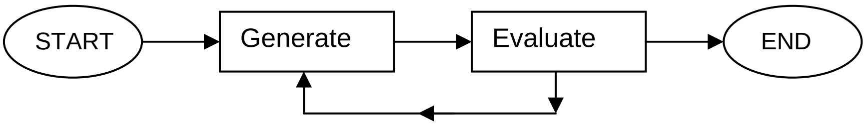


Figure 26. A simple representation of the design cycle

In practice the design process is even more complicated than this. Design can be thought of as a search for a solution to the problem “find a systems that satisfies these specification”. Very often the specifications are over-constrained or under constrained, so there are either no solutions, or too many solutions. In the former case the specifications have to be relaxed, with some constraints being removed or eased. For example, some specified property may be abandoned as desirable but not essential, or cost constraints may be eased by agreeing a higher price. In the latter case, new constraints may be added to force a solution, usually seeking a more optimal solution. Thus the goalposts are moved, and any future solution will be a solution to a different problem to that originally specified. In this one can see that the design process is a co-evolution between specification and proposed solution, until a problem-solution pair is found that is considered to be satisfactory.

In the context of the previous sections, it would be rare for a complete design to be evaluated in fully instantiated form and rejected. This is because the design process can be very expensive in terms of people’s time and the collection of information at various levels throughout the process. Often designs or parts of designs can be rejected as the design process proceeds, with the focus at any time being on the most critical parts, however defined.

10. CONCLUSIONS

The majority of this paper has been devoted to sketching out a mathematical theory of multilevel systems and its related multidimensional structure. The whole enterprise is based on what was called the *Fundamental Construction of Multilevel Systems*, where a set of component parts at a given hierarchical level is assembled into a whole at a higher hierarchical level, in bottom-up fashion. In principle this process is grounded at the level of concrete objects and builds up increasingly more complicated structure in a combinatorial fashion. In contrast to this 'put one brick on top of another' approach is a top-down approach in which the system is dissembled into its parts. When the system does not exist, as it often does not in design, the top-down approach is dealing with abstractions. These are instantiated at lower level in a hazy way, and all reasoning at the higher levels is contingent on lower level instantiation.

A theory of design has been proposed in which the designer creates new artefacts and the multilevel vocabulary to describe them. Initially the design is ill-specified, being sketched in terms of hazy high-level constructs and some more concrete structures at lower levels. The design proceeds by hypothesis-making at all level, and periodically subjecting hypotheses to validation as the design is instantiated. The generate-evaluate cycle operates at all levels, with a premium on rejecting incorrect hypotheses before too much resource is expended on consequent structures. The outcome of this design process is a fully instantiated blueprint, in which the vocabulary necessary to describe the system has been built, and all hypotheses about the system have been tested, leading to a 'theory' of the system.

In this context, it has been argued that the designers of new systems are the first scientists to accumulate knowledge about those systems. Furthermore, it has been argued that scientists exploring artificial systems must follow the path trodden by designers, especially if they have ambitions to change and control those systems as engineers, planners, or administrators. In summary, it has been argued that, if design is the science of the artificial, then the science of the artificial is design.

ACKNOWLEDGMENTS

The ideas in this paper have progressed considerably as a result of being able to discuss them with members of the Embracing Complexity in Design Cluster. I am grateful to other members of the cluster for many fascinating discussions on complexity and multilevel systems in design, and to both the Arts and Humanities Research Council and Engineering and Physical Science Research Council for funding the cluster.

REFERENCES

1. Simon, H. *The sciences of the artificial*, MIT Press (Cambridge, Mass), 1969.
2. Johnson, J.H., 'Relational structure in planning and design', *Environment and Planning B: Planning and Design* Anniversary Edition, 57-67, 1998.
3. Barabási, A-L, *Linked*, PLUME (Penguin Group, New York), 2003.
4. Johnson, J.H., 'Exploring Complexity in Collaborative Design Solutions', Proc. Int. Workshop on Collaborative Design, Qin and Johnson (eds), Brunel University, 23-24 June 2005.
5. Johnson, J.H., 'Combinatorial Structure in Digital Pictures', NERC Remote Sensing Project Discussion Paper 1,Mimeo, Centre for Configurational Studies, The Open University, January 1985.
6. Gould, P., Johnson, J.H., Chapman, G.P., *The structure of television*. PION (London), 1984.
7. Johnson, J.H., 'Some structures and notation of Q-analysis', *Environment and Planning B*, **8**, 73-86, 1981.
8. Johnson, J.H., 'A language of structure in the science of complexity', *Complexity*, **1**(3), 22-29, 1995.
9. Johnson, J.H., 'Q-transmission in simplicial complexes', *I.J. Man-Machine Studies*, **16**, 351-377.