CoDesign, Vol. 1, No. 4, December 2005, 223–242

Taylor & Francis
Taylor & Francis Group

Complexity science in collaborative design

JEFFREY JOHNSON*

Department of Design and Innovation, The Open University,
Milton Keynes MK7 6AA UK

(*Received 23 June 2005; in final form 20 November 2005*)

Bringing designers together as successful design teams is investigated in the context of the science of complex systems. The relational structure inherent in design defines multilevel multidimensional networks, which can represent part-whole hierarchies in design. Design is presented as an iterative top-down and bottom-up process that induces the emergence of desirable properties. This involves the construction and instantiation of multilevel representations. The implications of this are discussed for computer supported collaborative design, and how interaction structures can be designed to support design collaboration. The paper is aimed at a general reader and is self-contained.

Keywords: Complex systems; Collaborative design; Multilevel systems; Multidimensional networks

1. Introduction

This paper will investigate what the emerging science of complex systems can contribute to the theory and practice of collaborative design. It concerns with the way sets of individual designers can be brought together to form successful design teams. Conventionally designers were brought together by working in the same offices at the same time. Such physical proximity supports a complex dynamic of formal and informal meetings between groups who need to communicate and exchange ideas and information of many kinds. A small organisation with just ten people has thousands of ways of combining into groups. Rather than prescribe ways for each group to come together, it may sometimes be more flexible to let them *self-organise* into meetings involving those who are necessary to achieve some purposes.

Short *ad hoc* two-person meetings are common in the workplace, with one person asking another a question, and the other person giving an answer. Such groups may extend to three people, as a nearby person adds their comments. Larger meetings are usually organised more formally, with specified participants, rooms, times, and agendas.

---

*Corresponding author. Email: j.h.johnson@open.ac.uk

CoDesign
ISSN 1571-0882 Print/ISSN 1745-3755 online
http://www.tandf.co.uk/journals
DOI: 10.1080/15710880500478346

Informal meetings can be very important, such as those that exchange information around the water cooler, in the corridor, or at lunch. Physical presence and the right environment are very important in human communication, evidenced by meetings at comfortable hotels and in other attractive environments. This suggests that the social infrastructure for collaboration in general is *designed*, and that this applies to collaborative design in particular.

Over the past few decades it has become increasingly apparent that traditional scientific approaches do not work for a wide range of systems across a wide range of domains. These include physical systems encountered in physics, chemistry, and biology; social systems encountered in psychology, sociology, and politics; and artificial systems encountered in design, manufacture and engineering. Somehow the systems under study are too 'complex' for conventional approaches. Paradoxically, there is no universally accepted satisfactory definition of what it means for a system to be complex. However there are some widely accepted characteristics of complex system systems, including:

- some deterministic systems are inherently unpredictable over long periods of time
    - chaotic systems are very sensitive to initial conditions
    - computationally irreducible systems cannot be modelled by solving equations
    - path dependent systems depend on their particular history
    - systems may have co-evolving subsystems, or co-evolve with their environment
    - emergence: new order may emerge from existing system states

- some systems are discrete and cannot be represented by numerical equations
    - topology: the dynamics of systems may be constrained by network connectivity
    - self-organisation: system behaviour can emerge from autonomous agent interactions
    - multilevel dynamics: the behaviour of systems may depend on many discrete levels
    - emergence: wholes can have properties not possessed by their parts

This paper will begin by introducing some of these ideas from complex systems, and discuss their relevance to design. In this context:

- many artificial systems that we design are complex, *e.g.* cities, drugs, the Internet.
- many manufacturing processes are complex, *e.g.* factory design, supply chains
- the environment of design is complex, *e.g.* markets, fashion, social mood
- the design process can be a complex human system, *e.g.* couture, designing a new jet engine

The rest of this paper addresses the last of these, in particular the first of the design activities suggested by Pahl and Beitz (1996):

- conceptualising, *i.e.* searching for solution principles,
- embodying, *i.e.* engineering a solution principle by determining the general arrangement and preliminary shapes and materials of all components
- detailing, *i.e.* finalising production and operating details, and
- computing, drawing and information collecting which occur at all phases of the design process.

In this it will develop some new ideas based on a recent theory of discrete multilevel systems and their dynamics (Johnson, 2005).

2. Dynamics in complex systems

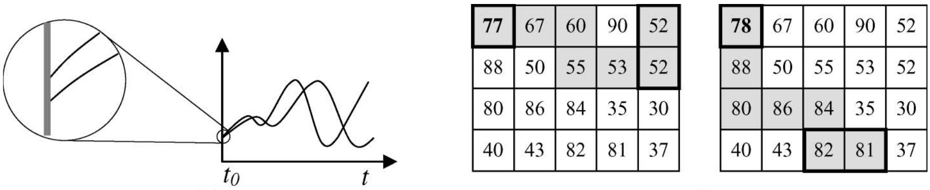
In the nineteen sixties the weather scientist, Lorenz, discovered what is now called *deterministic chaos* (Gleick, 1988). Technically a chaotic system is one that is bounded and sensitive to initial conditions. The term 'bounded' means that the system does not fly off to infinity, or in practical terms, does not disintegrate. The phrase 'sensitive to initial conditions' means that if the system is started again from the 'same' position, its trajectory will deviate from previously observed trajectories. This is illustrated in Figure 1(a). Although the two curves appear to start with the same value at t_0 the magnified area shows there is a slight difference, and this is enough to make the system diverge widely at t.

Figure 1(b) shows a system with a robot starting in the top left corner. The robot always moves to the horizontally or vertically adjacent square with the number most similar to the one it currently occupies. In the two cases shown the only difference in the environment is that the top left corner value of 77 increases to 78 on the right. This small change is sufficient for the robot to follow a completely different path. On the left it finishes up in an 'attractor', oscillating between the two squares with value 52. On the right it oscillates between values 81 and 82. It can be very difficult to predict the behaviour of systems that are sensitive to initial conditions, making them difficult to design, manage and control.

One of the triumphs of conventional science is to reduce system dynamics to formulae that allow the future state of the system at time t to be computed from the present state. For example, if a car is travelling at 50 km/hour, the distance it has travelled after t hours is $50 \times t$ km. Thus after 2 hours it has travelled 100 km, and this can be calculated without calculating how far it had travelled in 1 hour. For any t, determining the distance can be reduced to a single calculation. In contrast, dynamics like those in Figure 1(b) have to have every step computed, and they are *computationally irreducible*.

Usually the design process is *computationally irreducible*. Every step has to be executed, since what happens next depends on the outcome of the present step. The outcome of the present step is 'computed' by executing it. It could be an evaluation, or it could be trying a new idea to generate new candidate designs.

Design is usually *path dependent*, with the history of previous steps sometimes playing a part in the outcome of the next step. For example, the designer might have previously tried an idea and abandoned it, but much later find that it can be revived and brought back.



(a)

77	67	60	90	52
88	50	55	53	52
80	86	84	35	30
40	43	82	81	37

78	67	60	90	52
88	50	55	53	52
80	86	84	35	30
40	43	82	81	37

(b)

Figure 1. In deterministic systems, small changes in initial conditions can give different outcomes. (a) measurement of initial conditions always has errors; (b) slight changes in initial conditons can produce large changes in outcome.

Of many *co-evolutions* in design, the most obvious is that between problem and solution. Often specifications result in design problems that either under-constrained with too many solutions or over-constrained with no solutions. In the first case more constraints may be added, possibly making the solution more optimal. In the second case, some of the constraints must be relaxed. Either way, the specification changes. Seen this way, design can be viewed as a process in which specification and candidate solutions co-evolve until a satisfactory problem-solution pair has been found.

3. Parts and wholes in design

Figure 2(a) shows a stack of lines. The stack has the emergent property of occupying a rectangular area. Figure 2(b) shows the *sun illusion*. This has the remarkable emergent property that most people see a white circle at the centre of the lines, even though that circle does not exist for any of the lines. Figure 2(c) shows sets of bricks assembled into arches, each with the emergent property of having a gap between the sides and the top. None of the parts has such a gap.

Figure 2(d) shows a set of people. Under the right circumstances these people can be assembled into a *design team* with the desirable emergent property of being able to work together to create innovative high added-value designs. To do this one has to understand the social and professional *networks* of the designers and how they are *connected* to each other. Some designers complement others and work well in teams. Some designers have bad relationships with others, making productive teamwork more difficult. Collaborative design can be seen in network terms, and later in this paper we will investigate how network-theoretic ideas may be used to plan and manage the process.

Most designers have a degree of autonomy in how they work, and how they interact with other designers. Thus they can be viewed as autonomous *agents* in multi-agent systems. It is possible that the way designers collaborate can be investigated using the methods of complex systems, including discrete multi-agent computer simulation.

Making things is the bottom-up process of taking the right set of parts and assembling them in the right way to create systems with desirable *emergence* of properties and behaviours. Figure 3(a) shows a set of blocks being assembled to make an arch. The graphic objects on the left represent the 'real' system which is mapped into a symbolic representation on the right. Thus one can write R:{part-1, part-2, part-3} → *Arch*. The notation ⟨part-1, part-2, part-3; R⟩ is used to represent the structured set, and we write *Arch* = ⟨part-1, part-2, part-3; R⟩. Figure 3 suggests that assembling elements at one level of representation results in a structured object at a higher level of representation.

Assembling parts into wholes is one way to move up a hierarchy of representation. This process will be called *AND-aggregation* because it requires all the parts. For example, to make a gin-and-tonic one needs gin AND tonic AND ice AND lemon. There is another way of moving up hierarchies, using what we will call OR-aggregations. This is illustrated

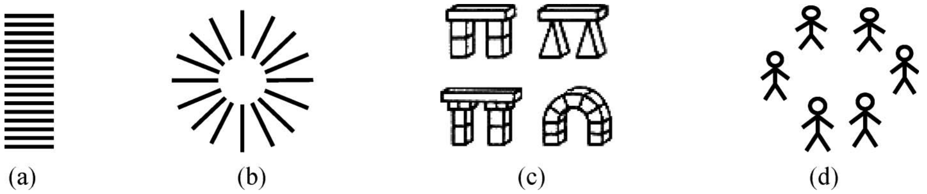
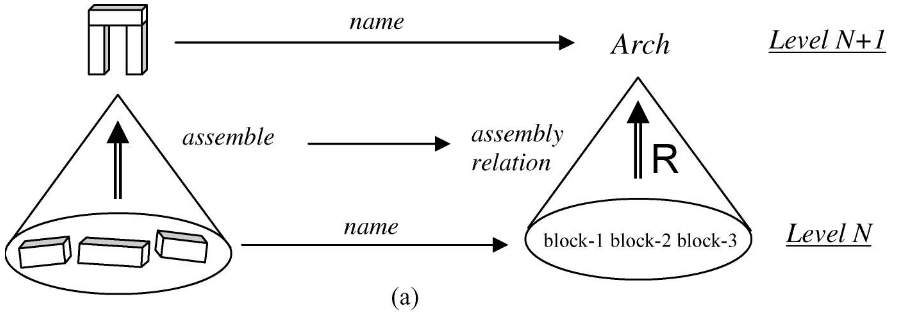


Figure 2. Assembling structures from sets of parts.



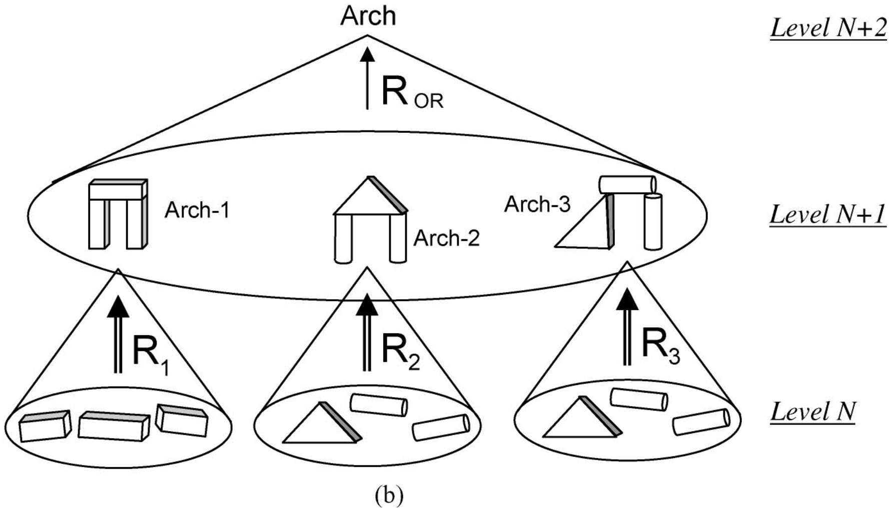


Figure 3. Parts and wholes: representing the assembly a set of blocks into an arch. (a) representing the assembly a set of blocks into an arch; (b) Building sets of alternative structures using an OR-aggregation.

in Figure 3(b) where there are three types of arch. According to this aggregation an arch can be any of the arches Arch-1 OR Arch-2 OR Arch-3. Such AND-OR aggregations are the basic building blocks of multilevel systems.

Compared to making things, which is the bottom-up process of taking the right set of parts and assembling them in the right way to achieve desirable *emergence*, *design* is the process of deciding what kind of emergence is desirable, and finding sets of parts and ways of assembling them to create wholes with desirable emergent properties and without undesirable emergent properties. When designing complex systems it is common to use a top-down divide and rule strategy, with systems described as part-whole subsystems at many levels. Later we will investigate how this approach involves the creation of multilevel representations, as a fundamental part of the design process itself.

4. Asembling teams of collaborative designers

Social structures are also assemblies. Conventionally the structure of social systems has been analysed using network theory, but the binary relations of social networks are not rich enough to represent social structures such as teams of designers. This is illustrated in Figure 4 by communication between three people. The communication enabled by the conference call between all three people in Figure 4(b) is clearly different to the communication enabled by a series of pair-wise telephone calls in Figure 4(a). Pairwise calls are well represented by lines (links in a network or edges in a graph). By extension, the relation between three people, a *3-ary relation*, can be represented by a *triangle*.

Figure 5 illustrates the generality of representing *n*-ary relations graphically. Lines (1-dimension) can represent binary relations, triangles (2-dimensions) can represent 3-ary relations, tetrahedra (4-dimensions) can represent 4-ary relations, and in general *n*-hedra can represent *n*-ary relations. In general an *n*-ary relation can be represented by an *n*-hedron with *n vertices* in $(n-1)$-dimensional space.

The pictures of polyhedra represent more abstract objects called *simplices*. A *p*-dimensional simplex has $p+1$vertices. If $\{v_1, v_2, \ldots, v_n\}$ is a set of vertices, we use the notation $\langle v_1, v_2, \ldots, v_n \rangle$ to show that they are related under an *n*-ary relation, and form a simplex. In the special case of $n = 2$, the simplex $\langle v_1, v_2 \rangle$ is a network-like link with vertices $\langle v_1 \rangle$ and $\langle v_2 \rangle$. A set of simplices with all their faces is called a *simplicial complex*. When the relation, *R*, is known we use the notation $\langle v_1, v_2, \ldots, v_n; R \rangle$ to represent the simplex. In general, structures formed from *n*-ary relations have *emergent properties* that are not possessed by their elements or sub-structures. For example, when coffee, milk, and sugar are mixed they form a drink with a taste which is different from the tastes of the constituent elements. Thus {coffee, milk, sugar} $\neq$ ⟨coffee, milk, sugar⟩.

Figure 4. Binary relations are not rich enough to represent all relational structure. (a) three 2-ary relations; (b) a 3-ary relation.

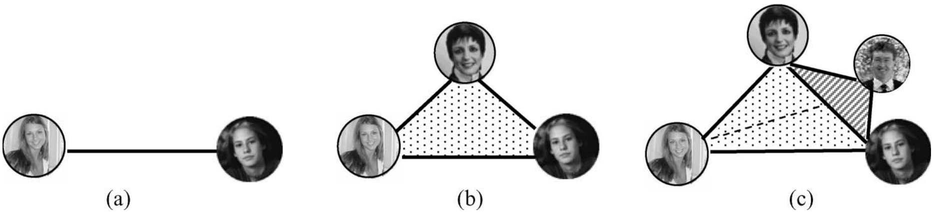


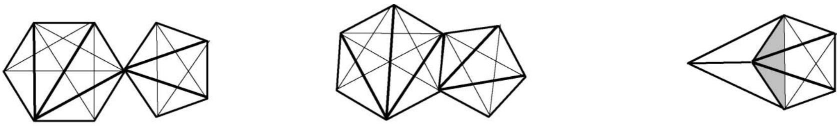
Figure 5. *n*-ary relations represented by multidimensional polyhedra in multidimensional space. (a) binary relation (line); (b) 3-ary relation (triangle); (c) 4-ary relation (tetrahedron).

5. Connectivity and communication in collaborative design

Relational structure is everywhere, simplices are everywhere, and simplices have interesting connectivity properties. Figure 6 shows how simplices can share different numbers of vertices, and that the more vertices they share, the more *highly connected* they are.

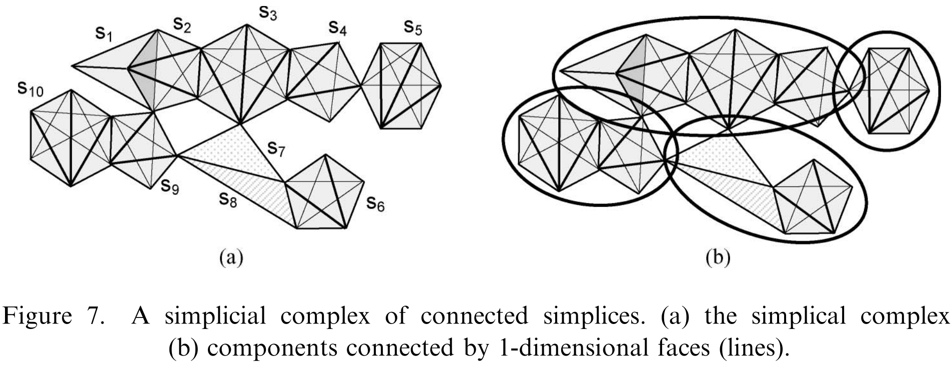
The intersection of two simplices is called their *shared face*. If the shared face has dimension q, the simplices are said to be *q-near*. Thus the simplices in Figure 6(a) are 0-near (a single vertex has dimension zero), those in Figure 6(b) are 1-near (two vertices make a one-dimensional line), and those in Figure 6(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. Figure 7(b) shows the *q-connected* components of the simplicial complex in Figure 7(a).

Figure 8 shows the relationship between four English public houses and the customers that frequent them. Typically people like to go to more than one pub 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, such as that in collaborative design, can pass through social structure determined by relations. In general the more highly connected the structure, the more rapidly information is transmitted.



(a) (b) (c)

Figure 6. Simplices can be connected at different dimensions. (a) 1 shared vertex; (b) 2 shared vertices; (c) 3 shared vertices.



(a) (b)

Figure 7. A simplicial complex of connected simplices. (a) the simplical complex; (b) components connected by 1-dimensional faces (lines).

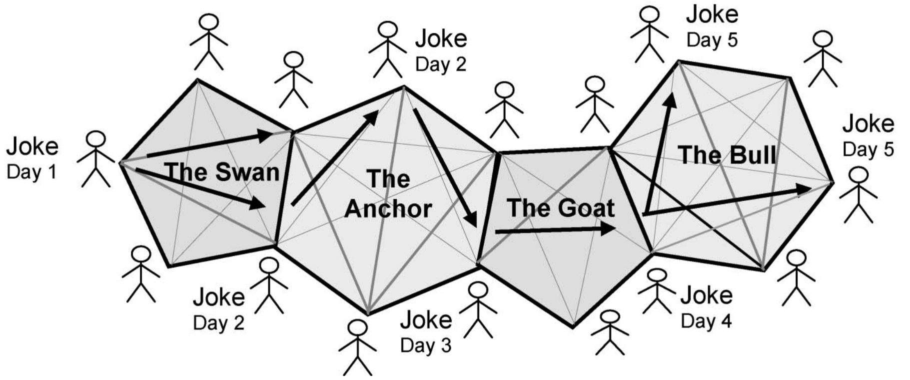


Figure 8. The transmission of jokes and information on the pub-customer structure.

Connectivity is essential for the communication of design information, and this extends to connectivity through shared interpretations. People have different experience, expertise, and expectations that can lead to disconnections making communication difficult. In the next section it will be suggested that designers *build* the multilevel language necessary to represent what they are designing. In doing this they use many vernacular terms, often giving them specialised meanings in the particular context.

It is very easy for different people to attribute different meaning to the same words and phrases. For example, when building a classification scheme for television programmes. Gould and co-workers (1984) often experienced disagreements in the ways that words should be used to describe programmes. Twenty years later it is more obvious that words do not carry absolute meanings and that ontologies evolve as communities and their environments change.

6. Part-whole multilevel systems in design

Pahl and Beitz (1996) write

> *Analysis* is the resolution of anything complex into its elements and the study of these elements and of their interrelationships. It calls for identification, definition, structuring and arrangement. The acquired information is transformed into knowledge.
>
> *Abstraction* gives the possibility of finding a higher level interrelationship, that is, one which is more generic and comprehensive. Such a process reduces complexity and emphasises the essential characteristics of the problem and thereby provides an opportunity to search for and find other solutions containing the identified characteristics. At the same time new structures emerge in the minds of designers and these assist with the organisation and retrieval of the many ideas and representations.
>
> *Synthesis* is the putting together of parts or elements to produce new effects and to demonstrate that these effects create an overall order. It involves search and discovery, and also composition and combination. An essential feature of all design

> work is the combination of individual findings or sub-solutions into an overall working system – that is the association of components to form a whole.

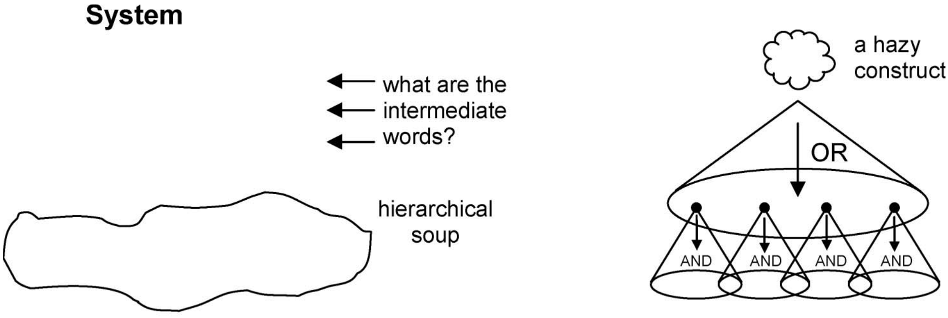
Thus *synthesis*, a *bottom-up* part-whole assembly process, is complemented by a top-down process in which designers *analyse* systems in terms of their parts and characteristics. In our terms, abstraction will be seen as an oscillation between the top-down and bottom-up approaches, involving what will be called the *Intermediate Word Problem* (Figure 9(a)).

When we look at an existing system as a whole, our impression of it is a Gestalt. Alongside this, there may be prior knowledge of the system, and the ability 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" (Gould *et al*, 1984). 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 9(a)).

Designers contemplate what might be called *could-be systems*, things that don't currently exist, but that could exist. Thus the designer is contemplating things that, if they did exist, would be sets of components assembled under appropriate relations to form a whole (synthesis). However, the precise set of components is generally unknown in systems that are multiple assemblies of assemblies. Thus the best the designer can do is to contemplate the abstract whole and hypothesise a set of unspecified component parts (analysis). They do this by creating intermediate word vocabulary to label *hazy constructs*, to be instantiated with lower level detail, as shown in Figure 9(b).

This is illustrated in Figure 10 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 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



(a) (b)

Figure 9. Building a vocabulary to represent could-be objects and systems in design. (a) the intermediate word problem; (b) instantiating a hazy construct.

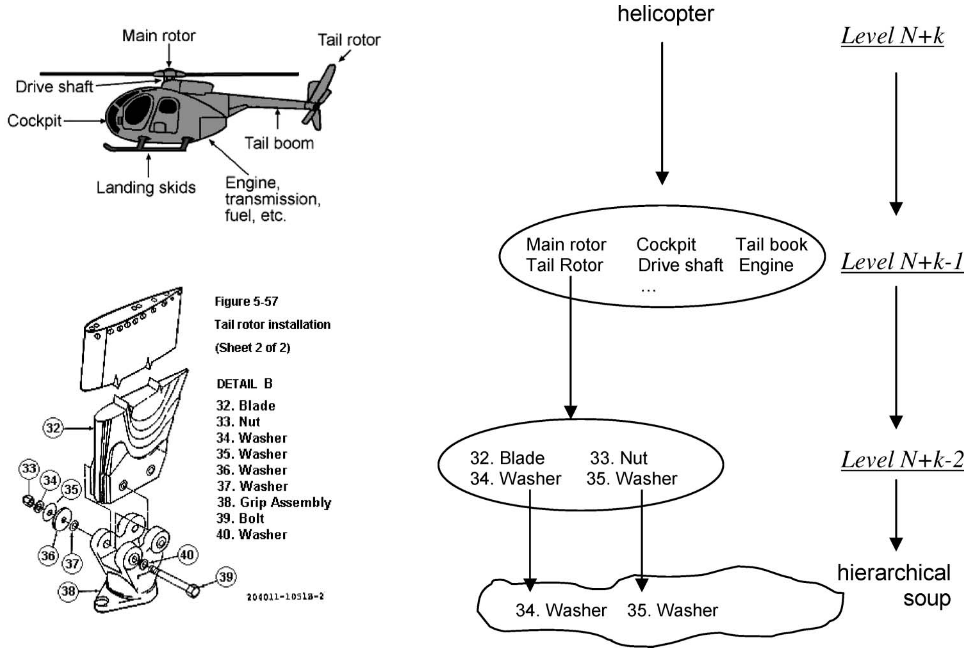


Figure 10. Abstracting intermediate words for a helicopter.

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 9). 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 by hazy cloud-like symbols. Some parts of the system may be entirely specified from the start, such as the tail rotor in Figure 11(a). For example, this may be a legacy subsystem that must be used in the design.

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 11(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 a higher level. In practice one can imagine designers constantly scanning the design in both top-down and bottom-up modes, try to connect the higher level abstractions to the lower level realities.

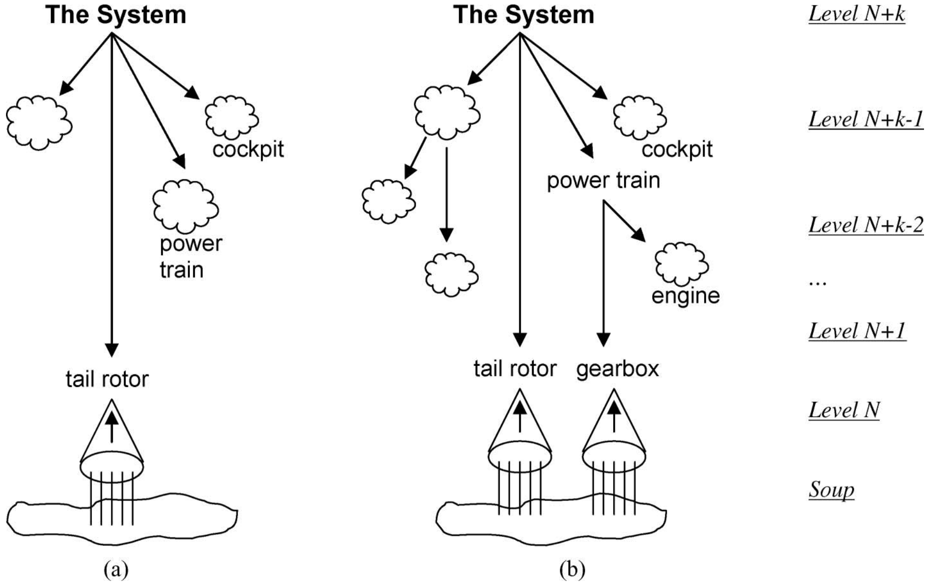


Figure 11. Design as the top-down and bottom-up process of building an ontology. (a) Initial sketch; (b) partial instantiation.

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 11(b), the particular 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.

It is important to note that building the multilevel representation of a system during is a social process, where the terms used to label things may carry different meaning to different people. Thus these have to be negotiated and care has to be taken that hazy concepts do not lead to misunderstandings.

7. Computer support for building representions in collaborative design

There is of course more to the design process than suggested in the previous section, but building a representation of the system is essential if the design is to be communicated to third parties to be manufactured. Furthermore, in collaborative design, the construction of the blueprint and its associated representation must be shared by the design team to support collective reasoning and debate.

In this context it is easy to understand why computer aided tools work so well in the final stages of design, but generally are less helpful in the early stages. In many CAD systems the components are represented geometrically as mathematical objects. Generally there is poorer support for the *ontology* being constructed and used by the designers, since these are much more hazy, and in general computer support for the kind of language used in everyday speech is poor.

Sketching may be supported in CAD systems in terms of the creation of graphic objects, but currently pattern recognition is poorly supported, and the human ability to see interesting things in roughly sketched lines and tones far surpasses what can be done by computers.

Increasingly remote collaborative design is being supported by Internet-enabled communication tools. One example of this is *Flash Meeting* developed on a recent European project [2]. *Flash Meeting* enables up to fifty people to meet on line, as illustrated in Figure 12. Simple 'raise hand' protocols allow individuals to take turns in holding the floor. At any time the current speaker holds down a button, selecting their webcam image and sound to be received by everyone else. On releasing the button the floor passes to next person with their hand raised. The experience is that people find this mode of interaction very satisfactory.

The bottom of Figure 12 shows a graphic record of the meeting as it progresses. Each participant has a timeline showing the intervals during which they held the floor. *Flash Meeting* stores the images and sound of all the participants, providing a random access record of the meeting that can be accessed through the time lines. This system also has many other features for annotating and retrieving information about the meeting which will not be discussed here.

In general, interactions between designers will be situated in linear time, illustrated by *Flash Meeting*. Let R be the relation on the set of designers with them able to collaborate. If all the designers have access to the Internet via the right hardware and software, then each designer will be able to collaborate with all the others, they will form a *clique* as illustrated in Figure 13(a).

However, as discussed in Section 3, subsets of n designers may interact forming n-ary polyhedra. This is illustrated in Figure 13(b) where, for example the interaction polyhedron group $\langle d_1, d_2, d_3 \rangle$ may work on the design during time interval Δt_1, with d_1


11-06-04 Prolearn TV revisited

2 of 56

13:30:19 Martin Wolpers
13:30:32 Peter and Kevin
13:30:40 Juan Quemada
13:30:44 Denis Gillet
13:31:22 Peter and Kevin
13:32:02 Peter and Kevin
13:32:37 Joachim Wetterling
13:33:01 Peter and Kevin
13:33:08 Effie Law
13:36:34 Peter and Kevin
13:36:43 Juan Quemada
13:40:10 Peter and Kevin

http://cnm.open.ac.uk/projects/hexagon

Figure 12. The Flash Meeting systems supports dynamic multidimensional interactions.

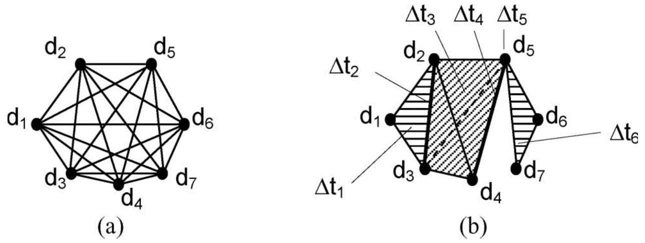


Figure 13. Collaborative design interaction structures. (a) designers forming a clique in a graph; (b) designers forming polyhedra through time.

dropping out, leaving the pair $\langle d_1, d_2 \rangle$ working together throughout Δt_2. These two may then be joined by d_4 and d_5 during Δt_3 to form the interaction polyhedron $\langle d_2, d_3, d_4, d_5 \rangle$. During Δt_4, d_2 and d_3 drop out leaving the interaction pair $\langle d_4, d_5 \rangle$, which becomes the vertex $\langle d_5 \rangle$ when d_4 drops out during Δt_5. Subsequently d_5 is joined by d_6 and d_7 during Δt_6 to form the interaction polyhedron $\langle d_5, d_6, d_7 \rangle$. Thus the design information flows through the system by a process of transmission, dependent on the underlying connectivity structure.

8. Designing the interaction backcloth for design traffic

In the study of complex systems, a distinction can be made *slow processes*, such as the building of a cathedral, and *fast processes* such as a religious service within a cathedral. In this case the cathedral provides the infrastructure, or *backcloth* for various kinds of *traffic* of human activity. This analogy comes from the idea of dynamic flows through relatively fixed channels. Examples of traffic and backcloth include cars on roads, conversations in pubs, calls across telephone network, and money across banking networks. In the case of collaborative design, the relational structure is provided by the ability to interact. This can include

- individuals communicating verbal statements that others can hear
- individuals communicating sketches that others can see
- individuals communicating drawings that others can see
- individual communicating text that other can see
- individuals communicating calculations that others can see
- individuals communicating tabulated data that others can see
- individuals communicating non-verbal signs of their emotional state

This communication can be viewed as a *traffic of information* on the design communication backcloth, as illustrated in Figure 14. In this case five designers are engaged in a meeting with a 5-ary communication relation between them. This relation could be established by them meeting in an office, or it could be established by them meeting over the Internet using tools such as *Flash Meeting*.

The traffic consists of two major parts. At the lower level there are the statements and other information offered to the meeting by individuals. These are received by everyone by virtue of the 5-ary relation between them, and at the higher level they fuse as a synthesis of the contributions of the group.

The backcloth might take some time to establish, possibly with a convenor approaching each participant to find their availability, and booking the necessary channels in cyberspace. Getting everyone mutually on-line is a necessary condition for the polyhedron to form.

It is easy to understand that information is combinatorially combined at meetings to produce a synthesis that could not be obtained by individual designers. However, the information flows can get more complex, as illustrated in Figure 15. Here it is supposed that three teams are working on parts of the design, with each producing transcripts and minutes of their deliberations. It is further supposed that an integration group is taking that information, sending it down to its members, and assuming that each will synthesise the three information sources in their own way. Then when the Integration Group meets, the individual interpretations can be combined to give an overview of the segmented design. In practice the integration group is likely to include the chairs of each design group, as another means of facilitating information flows.

Once the polyhedral backcloth is in place, the traffic of communication can begin. In the terms complexity science, meetings like this are *computationally irreducible*, since they involve a co-evolution between the ideas of the members of the team. A comment by one

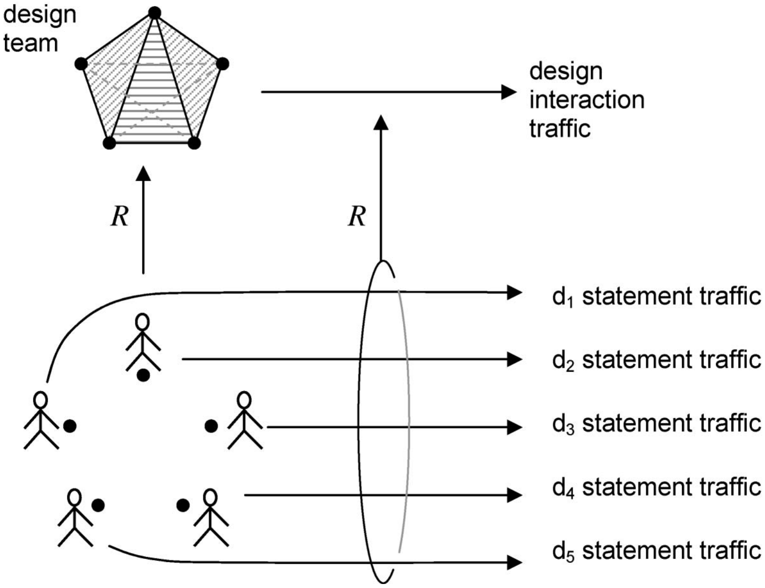


Figure 14. Microlevel communication traffic on the hierarchical backcloth.

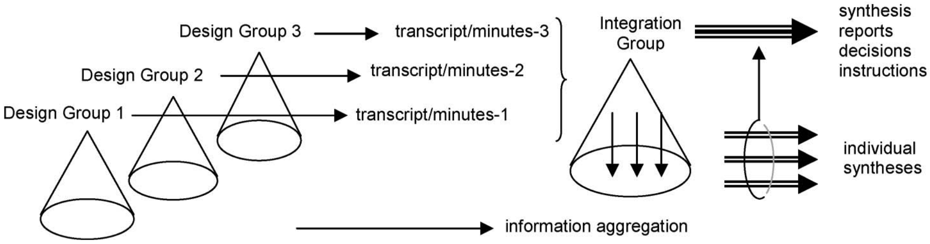


Figure 15. Information aggregation over the designer-team structure.

member of the team may send the discussion in completely unexpected directions. Sometimes this is good, as when new aspects of problems and opportunities are discussed, and sometimes it is bad, as when people embark on long digressions which ultimately contribute little to the matter in hand.

Sometimes such meetings are chaired, and sometimes every participant has an equal authority to take the meeting wherever it must go. In the former case, the chair might have already set an agenda. In the latter case, it is necessary that people behave in disciplined ways, recognising what is useful to the group, and gently steering each other away from less relevant issues. In future this kind of management of the meeting traffic may be computer assisted, with flags being raised when the main issues of the meeting are being ignored. But to be really useful, especially when designers are being divergent, this would require levels of language understanding that are far beyond current capabilities.

At any time, members of the design team(s) are discussing the design in terms of its representation (Figure 16). Some parts of this representation will be geometric within CAD systems, with parts fully instantiated. Other parts will be in natural language, which may or may not be explicit in the representation. The ontology of the design may be represented at higher levels by *schematics*, which are often block-diagrams containing words, and links between them representing relationships. These diagrams are graphs representing the topological constraints of the design at intermediate levels. The ontology may also be contained in reports and other documents such as part-lists and specifications for manufacture and assembly of components and subsystems.

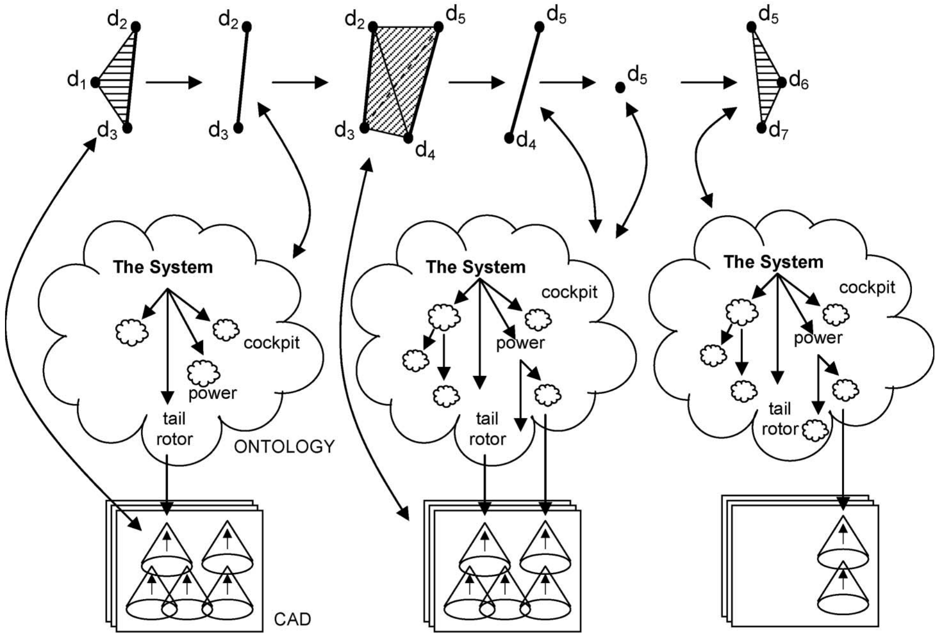


Figure 16. Relational dynamics as design interaction polyhedra change the design representation.

9. Example: collaborative design of open courses for the European complex systems community

In 2005 the Information, Society and Technology unit of the Future and Emerging Technology Division of the European Commission (FET/IST) funded a Coordination Action called ONCE-CS (Open Network of Centres of Excellence in Complex Systems). One of the objectives of that project is to create open courses on various aspects of complex systems for scientists, business people, and government administrators across Europe.

It is estimated that the European complex systems community has between 1,000 and 10,000 scientists working across the disciplines in the natural and social sciences. The budget available to ONCE-CS makes delivering the required educational courses face-to-face impractical, and they must all delivered using the Internet. Furthermore, it is expected that the authors of these courses will come from the complex systems community across Europe. Again the available budget precludes the geographically wide spread course teams having regular face-to-face meetings.

These constraints require that the courses are designed by collaborative working over the Internet, and that they are presented and studied by students over the Internet.

In terms of the theory developed in this paper, a course is a structured set of *learning objects*. The multimedia learning objects could be pieces of text, images, video clips, animations, audio clips, and so on. They could be assembled as books, PDF electronic files, web pages, movies, and so on. In turn these could be assembled to form the course. As can be seen from Figures 17 and 18, this is a relatively simple system in terms of part-whole assemblies.

This part-whole approach is supported by the *Connexions* system developed at Rice University in the USA and supported on ONCE-CS by International University Bremen [3]. Connexions supports learning objects at all these levels. Furthermore it is an *Open Commons* system that allows learning objects to be modified and re-used by the community under copyleft principles [6].

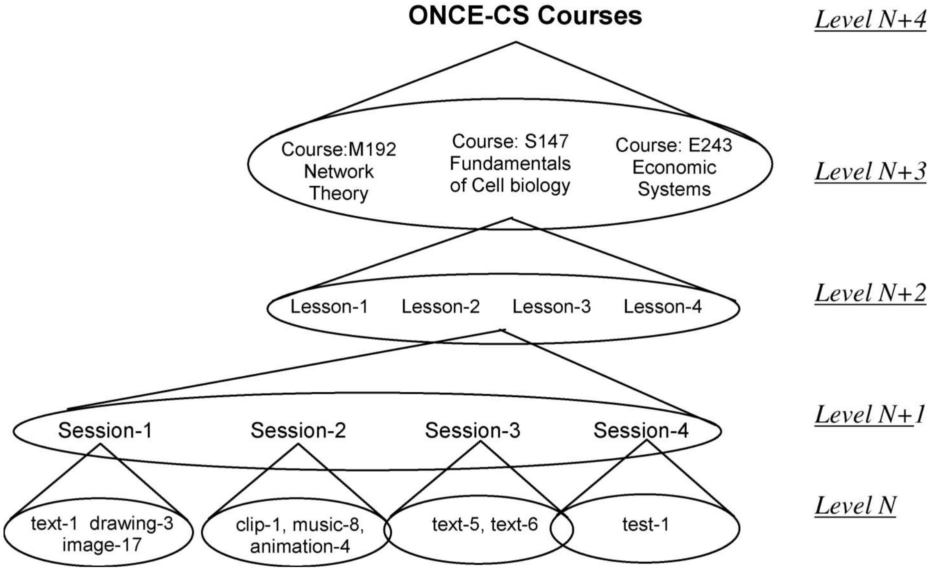


Figure 17. A part-whole hierarchy of courses.

Generally, the people producing the courses will be members of the complex systems community distributed across Europe, including support groups at the Open University (OU) in the UK, International University Bremen (IUB) in Germany, and the Institute for Scientific Interchange Foundation (ISI) in Italy. These three institutions form a core structure, $\langle$OU, IUB, ISI$\rangle$, to define a European PhD in complex systems science, to develop a curriculum for it, and to support course production and delivery.

To illustrate possible course production models, consider scientists working at Genopole, the 'Gene and Biotech City' at Evry to the South of Paris [4]. For this we will form a course team structure of people based on the institutions $\langle$OU, IUB, ISI, Genopole$\rangle$. For simplicity we will assume that OU means a person(s) from the OU, IUB means a person from IUB, and so on.

The simplex $\langle$OU, IUB, ISI, Genopole; $R_{\text{face-to-face}}\rangle$ has been under construction over the past year, and already there has been considerable discussion traffic on this structure. From this we already know that a lot of materials already exist to be used as resources on this project. Thus the hierarchical soup is already richly endowed with useful materials, which is one of the reasons this has been chosen to be the first course produced.

At some stage it will be necessary to examine these materials, and lift them out of the soup to become more explicit as components in the courses. This will require that all the partners see the materials and discuss them as traffic on a structure $\langle$OU, IUB, ISI, Genopole; $R_{\text{to-be-decided}}\rangle$. Conventionally, one would arrange a meeting at Genopole to be attended by OU, IUB, and ISI, with $R_{\text{to-be-decided}} = R_{\text{Genopole}}$. However this would be quite expensive, and it is much more desirable that $R_{\text{to-be-decided}} = R_{\text{Flash-Meeting}}$.

It is an open question is whether $\langle$OU, IUB, ISI, Genopole; $R_{\text{Flash-Meeting}}\rangle$ can support the necessary traffic of documents and materials. Certainly whatever materials already exist in electronic form can, in principle, be exchanged by email on the structure $\langle$OU, IUB, ISI, Genopole; $R_{\text{email}}\rangle$. One mechanism for this is to have the materials flow from Genopole to the OU, and for the OU to distribute them to IUB and ISI (Figure 19). This distribution could electronic or it could hardcopy by post.

In this particular case it is likely that an OU person will go to Genopole because experience shows that one gains a lot of serendipity information at such meetings. For example, the Genopole person may have materials in the office or lab that they don't appreciate could be used for the course. In this case the OU person is gathering potential components, in the same way that a designer would see new components at trade shows. At the moment electronic communication is not well adapted for this 'site-visit' part of the design process.

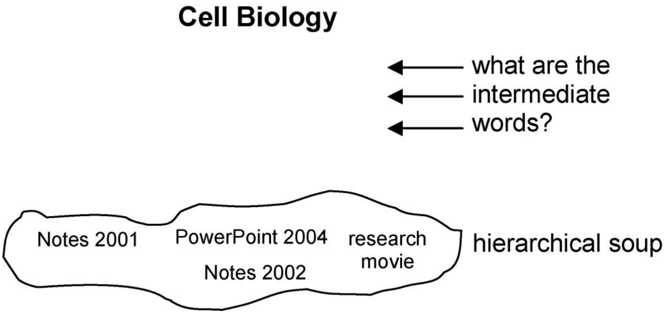


Figure 18. The Intermediate Word Problem for making a course on cell biology.

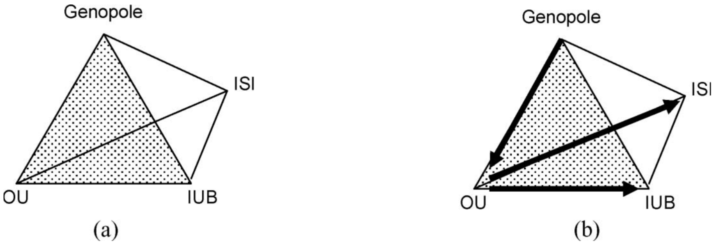


Figure 19. The human structure necessary to design the cell biology course. (a) the institution and person structure; (b) using the structure to communicate materials.

Once the basic materials for the course are available to all the course designers by file sharing, it is likely that *Flash Meeting* communication structure will support the levels of interaction needed for this relatively simple project.

10. Collaborative design futures

It can be expected that collaborative design will be greatly affected by the possibilities opened up by the Internet. The Internet itself is a massive self-organised system and the science of complex systems can play a major role in the way that it will support completely new modes of collaborative design.

In this paper we conclude that CAD in general must be extended to include ontological representation rather than just geometric representation. This is essential if machines are going to support reasoning as it occurs in design, and support collaborative reasoning as designers work through open-ended problems together. Thus it is likely that the computers of the future will provide tools for the dynamic development of multilevel ontologies within the design process, as suggested in this paper.

In future many relationships will be relevant in collaborative design. These include

Designers and their Geographical Locations
Designers and their Skills
Components and the Skills necessary to design them
Components and the Designers authorised to change them
Designers able to collaborate with other Designers
Designers liking to collaborate with other Designers

When designing a backcloth structure for collaborative design, all kinds of relations like this become relevant. The fundamental structure is combinations of designers able to communicate with each other. Underlying this is the question of which designers could work together profitably under what circumstances.

One possibility, reflecting common practice today, is that designers should self-organise by making their own decisions who they should communicate with. Computer tools to support this include finding dates in diaries, computer-aided interaction dynamics management, or setting up the necessary technical resources.

Another possibility is that designers are assisted in the self-organisation by computer aids. For example, how should design teams form to optimise the likelihood of a successful outcome? In organisations with many designers, it may be impractical for every designer to know every other designer. Also it is possible that the design teams of the future will be drawn from many organisations, for example, with small companies offering specialised services.

In cases like this it is possible that the design support systems of the future will find potential collaborators and investigate the possible interaction dynamics. For example, each designer could be an agent in a simulation, using information about their technical abilities, personal preferences, known successful past collaborations, and so on. Such a system could *generate* many possible design team combinations and *evaluate* them on the basis of the distribution of simulated outcomes. No doubt, individuals would always want to have the last word, but this kind of computer aid could search the space of possible collaborations much more effectively than any individual designer.

Other possibilities include self-organising databases, overcoming the indexing problem that stands between data and its potential users. Possibly there will be an extension of this with machines all over the word self-organising to service a distributed multi-organisation design team for a particular project. Such services could go beyond present computer aids, with machines generating designs and helping to evaluate the designs of other machines – themselves being part of the creative design process.

Thus it is possible that new kinds of design practice will emerge in the future, with designers acting much more as autonomous agents forming opportunistic structures. This could be much more flexible since design practices might not need large numbers of permanent employees – designers can self-organise, join in and drop out.

The mathematical theory sketched in this paper applies to the design of any system, and includes the early stages of the design process. The concept of 'haze' has been introduced to represent abstract design ideas which are not fully instantiated. The concept of hazy idea developed in this way is close to being implemented in computers. This may enable the early stages of the design process to be better supported in computers, and in turn enable designers to collaborate more naturally throughout the design process.

Acknowledgement

I am very grateful to an anonymous referee who pointed out some shortcomings in an earlier version of this paper and suggested ways of improving it. Their comments have given me new insights and taken this research forward.

References

Atkin, R.H. (1974) *Mathematical structure in human affairs*. Heinemann Education Books. London.

Atkin, R.H. (1981) Multidimensional Man, Penguin, (London).

Gleick, J. (1988) *Chaos: Making a New Science*. Penguin Books (New York).

Gould, P, Johnson, J, Chapman, G, (1984) *The structure of television*, PION (London).

Johnson, J.H. (1983) Hierarchical set definition by Q-analysis. Part I: The hierarchical backcloth', Part II: Traffic on the hierarchical backcloth', I.J. Man-Machine Studies, **18**, 337/359, 467/487.

Johnson, J.H. (1998) Relational structure in planning and design, *Environment & Planning B/Planning & Design, Anniversary Issue*, 57/67.

Johnson, J.H., 'Multidimensional multilevel networks in the science of the design of complex systems', *Proc. Satellite Workshop on Embracing Complexity in Design, ECCS'05*, *European Conference on Complex Systems, ECCS'05*, Cité Universitaire de Paris, 17-Nov-2005, Johnson, Zamenopoulis, Alexiou (eds), The Open University (Milton Keynes).

Lane, D. (November 2004) 'Ontological uncertainty', University of Moderna, Italy.

Pahl, G., Beitz, W. (1996) *Engineering design: a systematic approach*, Springer-Verlag (London).

[1] http://www.centennialofflight.gov/essay/Dictionary/helicopter/DI27.htm (Accessed 21 June 2005)

[2] http://cnm.open.ac.uk/projects/flashmeeting/ (Accessed 21 June 2005)
[3] http://cnx.rice.edu/
[4] http://www.genopole.org/index_flash.htm
[5] http://www.cordis.lu/ist/fet/co.htm
[6] http://www.gnu.org/copyleft/copyleft.html