

# Multidimensional Events in the Design and Management of Manufacturing Systems

Jeffrey Johnson<sup>1</sup>, Nick Scott<sup>2</sup>, Gerry Frizelle<sup>3</sup>

Open University<sup>1</sup>, Linx plc<sup>2</sup>, Cambridge University<sup>3</sup>  
[j.h.johnson@open.ac.uk](mailto:j.h.johnson@open.ac.uk), [NScott@linx.co.uk](mailto:NScott@linx.co.uk), [gdmf@cam.ac.uk](mailto:gdmf@cam.ac.uk)

## ABSTRACT

Design, manufacturing, and management involve generating and manipulating relational structure. This leads to a theory of structural system events and time, that can be applied to industrial manufacturing systems. The approach extends traditional planning tools by providing a richer and more coherent system picture, and new analytic functionality that can be computer-implemented.

## 1. INTRODUCTION

When manufacturers assemble components to make a product, they impose *relationships* between the components. Design is the process that results in the specification of these relationships.

When an operator machines a workpiece, there have to maintain various *relationships* between themselves, the workpiece and the machine. Production management involves specifying and implementing these relationships. When a salesman wins a new customer, he establishes *relationships* between himself, his organisation, the customer, and the customer's organisation. His job is to identify appropriate relationships, and implement them. When company directors manage their companies, they establish or maintain, *relationships* internally and externally.

These examples illustrate the general proposition that *design and manufacturing management are essentially the creation and maintenance of dynamic relationships*.

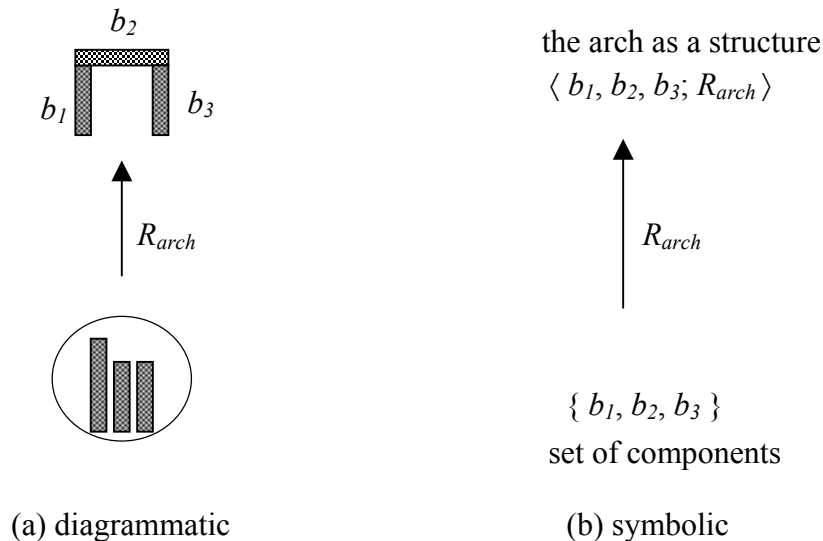
The relationships of design, manufacture and management are too complex to be represented by graphs and networks alone. This paper presents an essential extension to the better known structures of graphs and networks. With this extension a theory of structural time can be developed, in which the formation of relations can be regarded as natural *system events* that mark the passage of system time. In this context, management involves planning the creation of events and causing them to happen at appropriate sequences in clock time.

Sometimes events can be very simple and *e.g.* "a hole has been drilled", and sometimes they can be very complex, *e.g.* "the new product is ready for market!". Generally small events evolve on short time scales, while large events evolve on much longer time scales. Some large events can be decomposed into sub-events that then can be assembled on shorter time scales.

Events generally have *names* so that they can be planned and managed. These names belong to structured vocabularies reflecting the relations that define the events, and the event - sub-event hierarchical structure. *Constructing this vocabulary is an important part of the design and management process.*

Systems get complex when there are many time-dependent relationships between many things, and many relationships between many events. The new theory of relational structure gives a framework that enables all these dynamic relational entities to be represented together in a coherent overall structure that supports the design, control and management of system events through time.

## 2. REPRESENTING RELATIONSHIPS



**Figure 1. Representing an arch construction.**

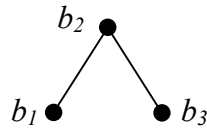
Consider a manufactured product, such as an aeroplane. When it is delivered to the customer it is made up of many thousands of components and sub-assemblies that have been assembled ‘in the right way’. In other words, all of the parts hold the ‘right’ relationships to each other. How can such relational complexity be represented?

As a simpler example, consider assembling an arch out of a set of building blocks (Fig. 2). Let  $b_1$ ,  $b_2$ , and  $b_3$  be the three blocks shown in the circle<sup>1</sup> on the left of Figure 1. The symbol  $R_{arch}$  is intended to represent the relationship between the blocks defining this type of arch. How it does this will be explained.

Let the *touching relation* between the blocks of the arch be denoted  $R_{touching}$ . This is a *binary* relation, and it can be drawn as a graph, with three vertices and two edges (Fig. 2). In graph theory, the edges are drawn as lines and denoted by the pair of vertices at the ends. Thus the graph in Figure 2 has edges  $(b_1, b_2)$  and  $(b_2, b_3)$ , representing the fact that  $b_1$  touches  $b_2$  and  $b_3$  touches  $b_2$ .

Sometimes it is useful to make the relationship that defines edges in graphs explicit. We will use the notation  $\langle b_1, b_2; R_{touching} \rangle$  instead of the less informative  $(b_1, b_2)$ .

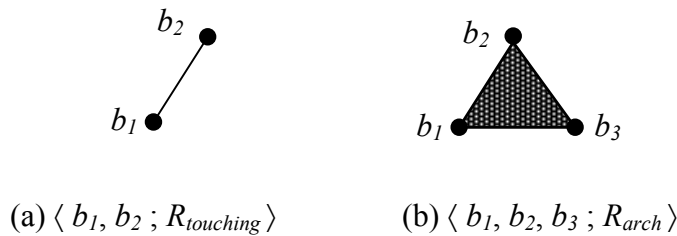
<sup>1</sup> In set theory, this is called an *Euler circle*. Euler showed how simple properties of set intersection and union could be illustrated diagrammatically. John Venn extended this in the nineteenth century, and Venn diagrams with sets represented by circles and other shapes are widely used in elementary set theory and its applications.



**Figure 2.** The graph of the ‘touching’ relation,  $R_{touching}$ , between the building blocks.

Graphs give a powerful way of representing relationships between pairs of things. They can be extended to relationships between more than two things, as follows.

Let the notation  $\langle b_1, b_2, b_3 ; R_{arch} \rangle$  mean that the set of blocks represented by the symbols  $b_1, b_2$ , and  $b_3$  are configured in a way that satisfies the definition of the relationship represented by the symbol  $R_{arch}$ . This involves all three blocks, and is a *3-ary relation*.



**Figure 3.** Representing  $n$ -ary relations,  $n = 2$  and  $n = 3$ .

In Figure 3, the 2-ary relation is represented by a line with two vertices, and the 3-ary relation is represented by a triangle with three vertices. In general an  $n$ -ary relation is represented by a polyhedron with  $n$  vertices.

The 3-ary relation,  $R_{arch}$ , could be defined as follows:

$b_1, b_2$ , and  $b_3$  are related under  $R_{arch}$  if  $b_1$  and  $b_3$  are vertical and  $b_3$  is horizontal.  $b_2$  must rest on top of  $b_1$  and  $b_3$ . The left edge of  $b_2$  must be aligned with the left edge of  $b_1$ . The right edge of  $b_2$  must be aligned with  $b_3$ .

An alternative definition could be:

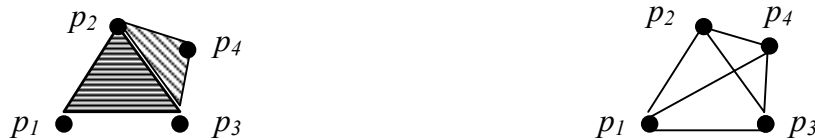
$b_1, b_2$ , and  $b_3$  are related under  $R_{arch}$  if they are put together in a way that looks like the object at the top of Figure 1(a).

In the first definition, the 3-ary relation is given in terms of a set of binary relations (2-ary relations). When it is possible to reduce the relational structure to composite binary relations, it makes specification of general  $n$ -ary relations relatively easy.

In the second definition, the test for the relationship involves comparing it with a template. This is particularly possible for geometric objects, and the relationships in many designs are recorded and communicated by drawings. It is less common to be able to specify more abstract objects and relationships in this way.

Consider four people discussing something on the telephone. They could phone each other pairwise, as  $\langle p_1, p_2 \rangle, \langle p_1, p_3 \rangle, \langle p_1, p_4 \rangle, \langle p_2, p_3 \rangle, \langle p_2, p_4 \rangle$ , and  $\langle p_3, p_4 \rangle$ . This experience would be quite different from them having a conference telephone call in which they all simultaneously listen and have the chance to speak. The notation  $\langle p_1, p_2, p_3, p_4; R_{conf} \rangle$  facilitates making such important distinctions.

In the same way that it is sometimes useful to draw binary relationships as lines in graphs, it is also sometimes useful to portray  $n$ -ary relationships in a graphical way. For example, the relationship between the three blocks was drawn as a triangle in Figure 3. By extension, the 4-ary relationship denoted by  $\langle p_1, p_2, p_3, p_4; R_{conf} \rangle$  is portrayed as a tetrahedron in Figure 4.



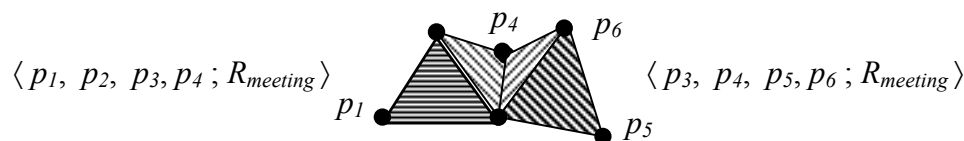
(a) portraying a 4-ary relationship by a tetrahedron

(b) portraying a set of 2-ary relationships by lines

**Figure 4. A 4-way conference phone call is more than a set of two-way phone calls.**

The relation  $R_{conf}$  is an example of an  $n$ -ary relation that cannot be built up from binary relations. Either you are having a conference call or you are not. The experience is not the same as having a series of two-way calls.

The generalisation of lines, triangles, and tetrahedra is the *polyhedron*. For example, when they work together as they should, the players on a football team form an 11-hedron. Of course drawings have limited use. Some graphs get too spaghetti-like to convey useful information, and then one must use symbolic methods of analysis.



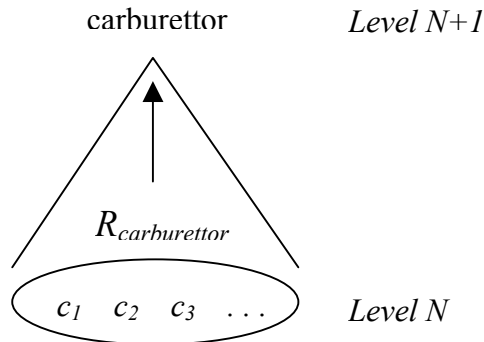
**Figure 5. Connected relational structures**

Figure 5 illustrates the possibility that polyhedra may be *connected*. Here the structure  $\langle p_1, p_2, p_3, p_4; R_{meeting} \rangle$  intersects the structure  $\langle p_3, p_4, p_5, p_6; R_{meeting} \rangle$ , with  $p_3$  and  $p_4$  attending both meetings.

Generally intersections between structures are *sites of interaction*. Sometimes this can be good, when it is useful to facilitate information and other flows. Sometimes it can be bad, since parts of the system may become inappropriately coupled.

In this case, having  $p_3$  and  $p_4$  on both committees will probably facilitate communication and understanding between the committees.

### 3. HIERARCHIES OF ASSEMBLY

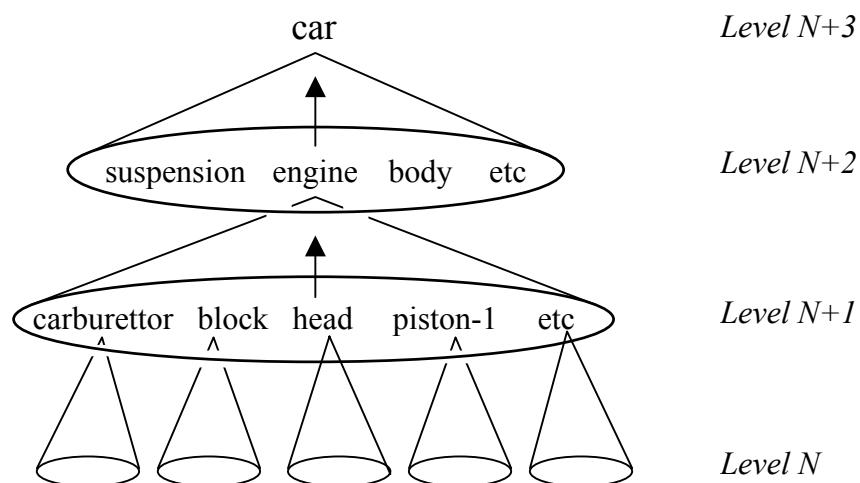


**Figure 6. The cone construction**

Generally, objects are manufactured in hierarchies of components and sub-assemblies. In our terms, each assembly between  $n$  things is characterised by an  $n$ -ary relation.

Let  $C = \{c_1, c_2, c_3, \dots\}$  be the set of component parts of a carburettor. These must be assembled according to the specified relationship,  $R_{carburettor}$ . Then the assembled carburettor will be represented by the notation  $\langle c_1, c_2, c_3, \dots; R_{carburettor} \rangle$ , or simply the word 'carburettor'.

Figure 6 illustrates the *cone construction*, in which the assembly relation maps the component parts to the assembled whole. An essential element of this construction is that the whole exists at a higher level of assembly than its parts. As shown the parts exist at so-called *Level-N*, while the assembly exists at *Level N+1*.



**Figure 7. A simplified hierarchy of assembly for a motor car**

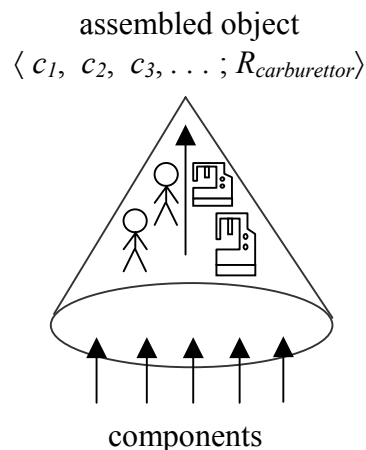
Then the carburettor is assembled into the engine, and the engine is assembled with other subsystems to make the final product of a motor car. Figure 7 shows this simplified four-level hierarchy of assembly represented by cones. In practice the assembly hierarchy is likely have to seven or more levels.

Figure 7 represents a complex structure that *is* a motor car. When companies build motor cars they know *all* the assembly relations necessary to build the components, put those components to form more complex sub-assemblies, and so on until the whole vehicle is constructed.

Where did all this assembly information come from? Some of it is inherited as previously existing components and sub-assemblies are re-used. Some of the information is created by the designer. In general, the assembly information is the final product of the design process. When the design is finished, all the parts and all the assembly relations are specified.

Making explicit the information of Figure 7 is the *target* of the design process. Thus design involves building a hierarchical language to represent the artefacts to be constructed. This language involves both vocabulary to *name* the intermediate parts, and vocabulary to describe the assembly those parts into the whole.

#### 4. ASSEMBLY, HUMANS, AND MACHINES



**Figure 8. Assembling humans and machines to assemble components into products.**

The design process specifies the assembly relation. This then has to be realised in practice by a combination of men and machines, as suggested in Figure 8.

Where do these combinations of men and machines come from? Someone, such as an engineering manager, *designs* the production system by specifying the components (humans, skills, machines), and specifying their dynamic inter-relationships.

Designing optimal socio-technical systems can be much more difficult than designing entirely physical systems such as a car or an aeroplane.

## 5. STRUCTURES, EVENTS, AND TIME

Figure 7 provides a *static* description of the assembly relationships of a manufactured product. The *dynamics* of bringing an artefact into being are also important. If the process takes too long, involves too many people, or too many expensive machines, the commercial viability of the product may be compromised.

To understand the dynamics of a system requires an appropriate theory of time. Let *clock time* be the usual time of clocks and calendars. Clock time is measured by periodic events, such as the swing of a pendulum, or electrons oscillating in a crystal. There are other kinds of time measured by the significant *events* of a system. Here we develop the idea that *structural events are the heartbeat of socio-technical systems*.

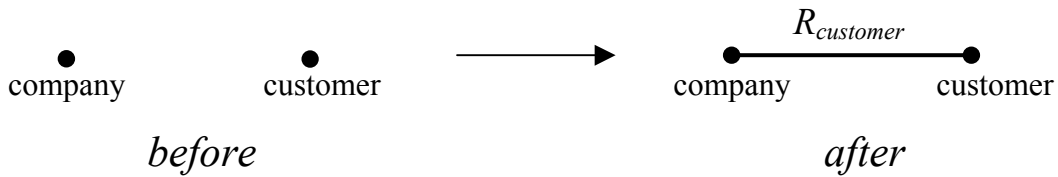


Figure 9. A 'winning a new customer' event

Gaining a new customer is an important kind of event for a business. It can be represented graphically as in Figure 9. Before the event there is no relationship between the company and its potential customer. After the event, the relationship is established. The event is characterised by the creation of the *link* in the graph.

Thus a *structural event* is observed as the moment at which a relation holds, when previously it did not. For example, scoring a goal is a significant event in football, and getting married is a significant event in many peoples' lives.

Events have *gestation periods*, of course. For example, the event of having raw materials delivered may be preceded by the sub-event of the order being placed, and the intermediate gestation time that involves the goods being located, packed, transport being arranged, etc. This gestation time corresponds to the on Gantt charts, that show when activities are live, before the completion event (Figs 14 & 16).

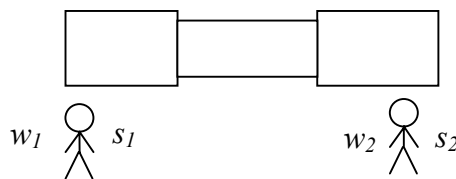


Figure 10. The assembled components of a simple production line

Consider a company setting up a new production line, involving machines  $m_1$  and  $m_2$ , conveyor belt  $b$  and workers  $w_1$  and  $w_2$  with skills  $s_1$  and  $s_2$ . These 'components' must be assembled appropriately for the production line to work (Fig. 10). The assembly relation,  $R$ , on the set of *components*,  $\{m_1, m_2, w_1, w_2, s_1, s_2, b\}$ , can be given as a set of binary relationships illustrated in Figure 11. For example, worker  $w_1$  has to be related to skill  $s_1$  by possessing it. He has to be related to machine  $m_1$  to operate it. Machine  $m_1$  is related to machine  $m_2$  by supplying it with parts. To do this it needs to be related to the conveyor belt,  $b$ , as does  $m_2$ . And so on.

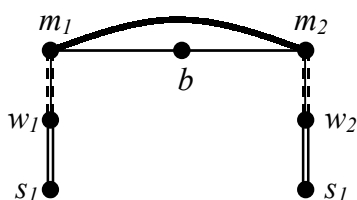


Figure 11. A set of binary relations defines  $R$ .

By definition, when all these binary relations hold, the relation  $R$  holds and the expression  $\langle m_1, m_2, w_1, w_2, s_1, s_2, b; R \rangle$  denotes the whole structure.

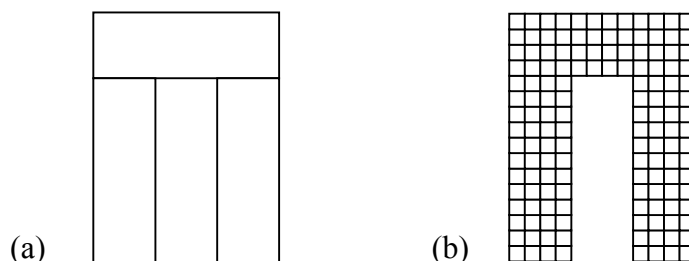
In order for the production line to work, the whole  $\langle m_1, m_2, w_1, w_2, s_1, s_2, b; R \rangle$  structure must be in place. The managers planning the new line might have to order new machines, take on new workers with the necessary skills or retrain their existing workers. In other words, they have to *build* the structure to support the traffic of production. When the structure is built, production can start. Before the *whole* structure is in place, production cannot start.

The formation of this structure is an important defining *event* for the company. This event is characterised by  $\langle m_1, m_2, w_1, w_2, s_1, s_2, b; R \rangle$  not existing before the event, and existing after the event.

Although structural events mark the natural time of systems, many resources are consumed linearly in clock time. Employees expect to be paid every day, week or month, whether or not they are productive. Similarly, rent and other outgoings have to be paid in clock time. Thus being able to establish relationships between structural events and clock time is essential.

## 6. MAPPING EVENTS INTO CLOCK TIME

The arch in Figure 12(a) is a very simple structure with three components. It would take a short interval of clock time to build it from toy blocks, perhaps a few minutes. If it had a hundred or more component blocks, Fig. 12(b), it could take much longer,

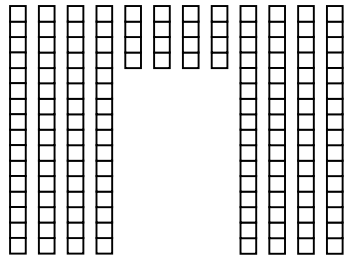


**Figure 12. Building a 144-block arch takes much longer than building a 3-block arch.**

Apart from having to handle more blocks, there is a combinatorial aspect. Many of the blocks in Figure 12 will have to be glued to all their neighbours, an average of approximately two gluings per block. So it won't just take  $144/3 = 48$  times longer to handle the block, it will also take  $144/3 \times 2 = 96$  times longer to glue them.

This illustrates the general principle that events with large numbers of parts may take combinatorially longer to occur in clock time: *events with many vertices usually have a longer clock time gestation period than events with few vertices.*

This idea of structural time and events is due to R.H. Atkin, who gives some very interesting statistics on the combinatorial nature of events based on the number of elements in them (Atkin, 1981).

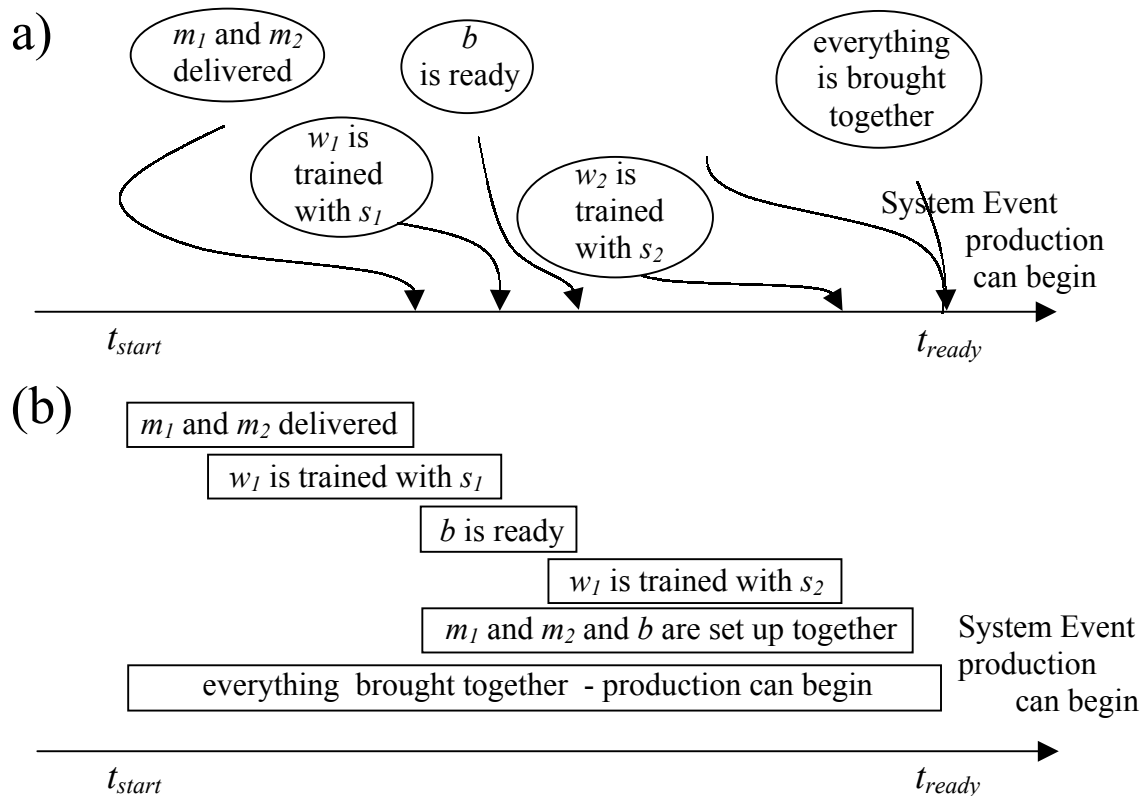


**Figure 13. Creating twelve sub-events for the arch assembling task.**

Figure 13 illustrates an idea that might reduce the assembly time. Suppose that twelve strips of eight or four blocks were formed, as shown in Figure 13. This involves 132 gluing sub-events. Eleven further gluing sub-events between the strips will complete the arch. So now the gluing takes only  $143/3 = 41$  times longer. Thus breaking the overall assembly event down into the strip formation sub-events reduces the overall assembly time by a half. Of course, manufacturers of complex products already know that breaking down complex events with many vertices into simpler events with fewer vertices may reduce the overall clock time for assembly.

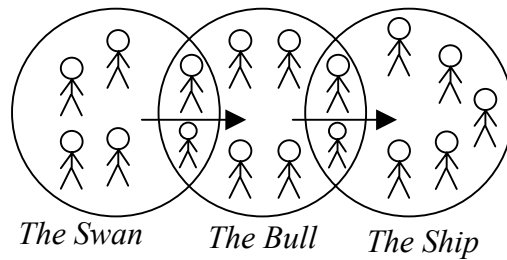
## 7. PLANNING AND MANAGEMENT

Planning involves mapping sequences of events into clock time, as shown in Fig. 14. Drawing conventional Gantt charts amounts to laying out the event gestation times. Although not shown on the diagram, some events may have to precede others, and the diagram can take on the nature of a critical path diagram.



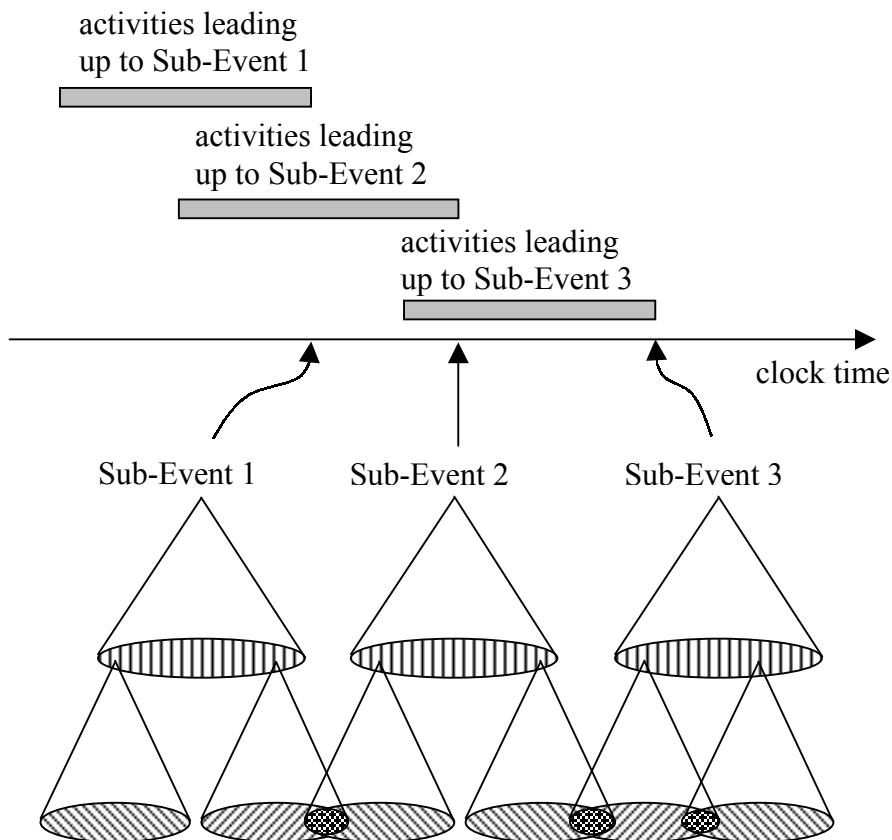
**Figure 14. Mapping system events into clock time. (a) planning system events in clock time. (b) Gantt chart of event gestation times.**

What does our analysis add that is missing from conventional planning tools? One of the characteristics of complex systems is that they have many elements and combinatorially many relations between those elements. Furthermore the relational connectivities need not be direct for interaction, and dependencies may not be obvious at every level



**Figure 15. A joke can be transmitted from Pub A to Pub C, even though they are not directly connected through any shared clients.**

Figure 15 shows the clients of three hypothetical British public houses named the Swan, The Bull, and Ship. One of the joys of pubs is hearing and retelling jokes. Suppose someone tells a joke in the Swan. Then a person who also frequents the Bull may retell it there. Subsequently a person who also frequents the Ship may retell it there. In this way the event of the joke being told can be transmitted from The Swan to The Ship, even though they have no clients in common.



**Figure 16. Low-level connectivities may connect apparently independent events making them interdependent.**

Figure 16 shows a more subtle form of transmission. Although the three sub-events appear to be well spaced in time, at a more detailed lower level they are connected. Thus unexpected interaction between Sub-Events 1 and 2 may delay the completion of either or both. Similarly, unexpected interaction between Sub-Events 2 and 3 may exacerbate the delay, and the plan may go badly wrong with long clock-time delays. Thus, discrete event simulation and other computerised means for managing manufacturing systems could benefit from this theory of structure and connectivity, by investigating whether key events are connected through lower level dependencies.

## 8. BUILDING THE SYSTEM VOCABULARY

In a study involving classification of television programmes (Gould *et al* 1984), it was found that every broadcasting organisation used a different programme classification method when compiling statistics, *e.g.* a programme classified as comedy for one organisation might be classified as light entertainment for another. Why didn't they all use the same standard scheme? Often the reason was that the organisation-specific vocabulary reflected internal structure. One organisation might have a department called Light Entertainment, while the other might have had a department called Comedy. The statistics were a measure of department's output and performance, and the classifications played a crucial role in the allocation of resources.

Most companies and organisations have their own unique ways of doing things, and the *vocabulary* used for planning and management must reflect this structure. Historically, a company's vocabulary is *constructed* by the people who run it, as they build the infrastructure that will enable the company to manufacture new products.

To see how this works, consider a company working on a 'a new product', code-named Product-X. Initially the definition of Product-X might be 'a new printing machine', as discussed at a board meeting. This general proposal then goes to a design team meeting, and a process begins that will end some months or years later with a precisely specified machine that will be manufactured and sold.

Initially what is meant by Product-X is very ill-specified and the design team must explore the possibilities. In so-doing they introduce vocabulary, which may or may not become part of the final specification. Figure 17 shows some possibilities.

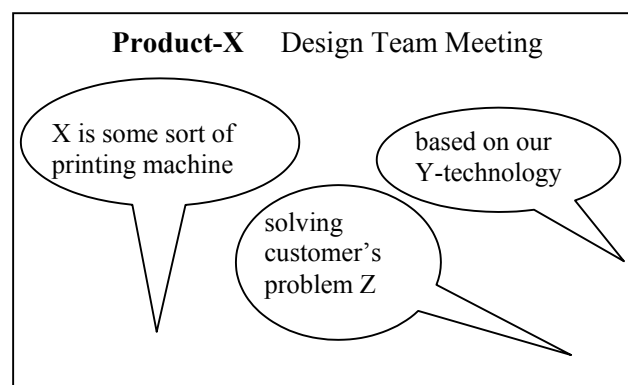
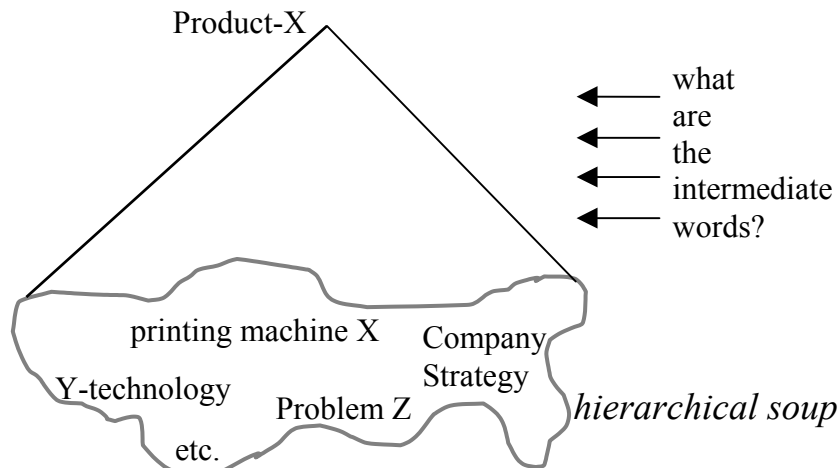


Figure 17. Beginning the process of instantiating 'Product-X'

The outcome of this process is usually a mass of words, phrases and, possibly, some boxes and arrows exploring dependencies. At this open-ended stage none of this need

make much sense, and there are probably many inconsistencies and anomalies. We call this collection the *hierarchical soup*, since it will be a mixture of ideas from many different levels of aggregation.

As time goes on, things must become more precise. In part, the designers are facing what Gould *et al* called the *Intermediate Word Problem* in which they try to abstract a consistent vocabulary out of the soup (Fig. 18). For example, Figure 7 showed four hierarchical levels abstracted from the hierarchical soup.



**Figure 18. The intermediate word problem**

## 9. NUMBERS VERSUS STRUCTURAL EVENTS

In this paper we have stressed the importance of structural events, at the expense of not discussing numerical or statistical data. We have done this to keep the presentation simple and short. There is, of course, an extensive theory of the way numbers are distributed over the events, and some events can be purely numerical. Indeed there is theory to maintain coherence between hierarchical structures and the aggregation of numbers over them [Johnson, 200].

## 10. TAKING CONTROL OF EVENTS

In the terms of this paper, desirable events don't just happen, they are *made* to happen by far-sighted and creative managers. How can the theory sketched here help in this process?

We have defined an event to be a relational structure. The event is the transition from 'relation does not hold' to 'relation does hold'.

The abstract polyhedra we have defined have vertices and an  $n$ -ary relation. Therefore the questions that managers should always be asking are "what are the possible vertices of my events, and what are the relationships that will define the events".

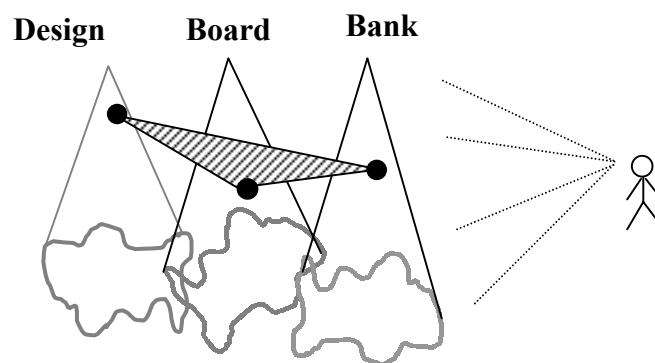
To do this, managers must be aware of the hierarchical soup that provides the potential vocabulary for building events. Managers who are out of touch, who do not know what plant they have, who don't know the attributes of the staff they have, or who have other gaps in their general knowledge of the system will be ill-equipped to generate the events that are essential for the business to thrive.

Within the hierarchical soup there may be high-level, ill-specified events, that require instantiation before they can be considered in policy terms. Working this up into something better defined involves what we have called the intermediate word problem.

The soup may also contain *latency*, with the possibility of relating previously unrelated low-level elements to construct useful higher level events. This might involve observing possible synergy between production lines, for example by standardising some parts.

Because managers have to have an overview of the whole organisation, they often deal with great complexity. There will be many hierarchies of assembly in any reasonably large organisation. These will reflect the many products and processes, and the administrative structure of the organisation.

It is not true that managers sit at the top of their organisations and pass high-level instructions down to be interpreted at lower levels. Managers must be aware of all the aspects of the organisation, and the interdependencies between them (Fig. 19).



**Figure 19. Managers have to build events from many hierarchies**

The principles of building events are the same for the company managers as those for designers. The managers too must expect to build vocabulary to express the structural events they are planning. By making this vocabulary explicit they are in a much better position to understand the lower level connectivities that may jeopardise their plans.

## **11. IMPLEMENTATION THROUGH COMPUTER TOOLS**

The ultimate test for the ideas in this paper is that they should be useful to people managing manufacture systems. Many computer tools already exist to help managers, but the theory developed here suggests new tools can usefully be developed. The computer tools to implement this theory must

- assist in the formulation of partly instantiated hierarchical vocabularies
- allow the numerical data of the system to be coherently distributed over the hierarchical vocabulary
- make operational the pattern recognition for observing that events have happened
- allow connectivities between events to be investigated, especially through less obvious lower level connections.
- allow structural events to be assigned gestation windows of appropriate length in clock time

- allow structural event to be laid out in time as multi-hierarchical Gantt charts.

## 12. CONCLUSIONS

In the paper we have introduced the following ideas

- design, manufacturing, and management involve generating and manipulating relational structure
- graphs and networks can be extended from binary relations to  $n$ -ary relations by polyhedra
- polyhedra can have intersections, and these are the sites of interactions
- the formation of relations determines *system events*
- it is necessary to build hierarchical vocabulary to represent system events
- system events can be mapped into clock time for planning purposes
- large events take longer to gestate than simple events
- large events can sometimes be usefully assembled from sub-events
- events may interact through lower level connectivities

The approach sketched here has the potential to extend traditional planning tools by providing a richer and more coherent picture of the system, and new analytic functionality that can be computer-implemented.

## REFERENCES

- Atkin, R.H., (1981) *Multidimensional Man*, Penguin Books (Harmondsworth).
  - Gould, P., Johnson, J, Chapman, G, (1984) *The structure of television*, Pion (London).
  - Johnson, J. H.(2001) ‘The “Can You Trust It?” Problem of Simulation Science in the Design of Socio-technical Systems’, *Complexity*, **6**(2), 34-40.
- Johnson, J, H., (2000) ‘Coherence Theory in the Science of Complexity’, *Proc. Conf. Complexity and Complex Systems in Industry*, I.P.McCarthy and T. Rakotobe-Joel (eds), Warwick University, 19-20 Sept 2000.