WILLIAMS, T.M., Eden, C.L., Ackermann, F.R., and Tait, A. (1995). The effects of design changes and delays on project costs. *Journal of the Operational Research Society 46*, 7, 809-818

The effects of design changes and delays on project costs

Terry Williams, Colin Eden, Fran Ackermann, Andrew Tait
Department of Management Science, Strathclyde University, Glasgow

This paper describes a study of a large design and manufacture engineering project, undertaken as part of a Delay and Disruption litigation. Design changes and delays in design approval would have caused delay to the project; in order to fulfil a tight time-constraint, management had to increase parallel development in the network logic, reducing delay but setting up feed-back loops that markedly increased total project spend. Cognitive mapping was used to elicit the relationships, which suggested the use of System Dynamics to quantify the effects. Results are described which show the effect of levels of design changes and approval delays, and their compounding effect. The wider implications on modelling projects are also discussed.

Keywords: project management, system dynamics

THE CASE STUDY

The authors were commissioned to investigate the reasons for delay and disruption in a major engineering project, and to quantify the effects by an auditable model. The work was in support of a considerable claim against the project client towards the end of a project, this study in particular being intended to support the part of the claim which was for Delay and Disruption.

The project was to design a few related versions of a specialised vehicle at the leading-edge of development, and to manufacture around 250 in total of the vehicles. The project as originally envisaged had a constraining time-limit, so the design management used a phased plan of parallel design of related components, and manufacture of components was planned to start before final completion of the design ("concurrent engineering"). The project had resulted in considerable overspend, and some lateness, and hence the formal claim.

The majority of the claim was for design changes to the product requested by the purchaser, but not the subject of a formal Contract Change or Variation Order (this was called the "Direct Claim"). However, it was felt that the totality of these design changes caused an overspend greater than the sum of the effects that could be assigned to the individual changes. Furthermore, there was some thousands of items of design documentation, which contractually had to be approved within a certain time-limit; from study of a (specially drawn-up) data-base containing details of these documents, it was known (and could be proved) that the project client's average approval time was well in excess of this contractual limit, with some instances of documents taking many times the limit to gain approval; it was felt that these delays contributed strongly to the overspend. Finally, it was felt that many comments on design documents were invalid, serving no valid design purpose but slowing down the design process as the comments had to be answered and the documents re-entered into the approval process (an example might be requiring proof of a self-evident assertion). (The lawyers gave legal definitions of concepts such as an "invalid comment".)

There were further causes to the Delay and Disruption experienced, and claimed for, in the project, but this paper focuses on the three main factors described above: changes to the design, document approval delays, and extra (invalid) comments. The questions that needed to be answered were: what were the effects of these factors (qualitatively), and what was the extent of the effects (quantitatively). The second question was essential as the work was part of a formal legal claim to which a sum of money had to be assigned; the first question was an essential pre-cursor to understand what the effects were, how they related to each other, and how they could be modelled.

QUALITATIVE MODELLING

Senior members of the project team, and the various managers involved, were interviewed by the authors. The key technique used both within the interviews and subsequently to model the explanations given for the various circumstances of the project was "cognitive mapping", which structures the way in which humans construe and make sense of their experiences. Eden[1] gives a general description of this technique, and Ackermann and Tait[2] discuss its use within this case study.

Specialist computer software called "Graphics COPE" was used to record and analyse the extensive maps developed[*]. Each interviewee's cognitive map was input, and these were then combined (through

[*] Graphics COPE is developed and supplied by the Department of Management Science at the University of Strathclyde, and runs in the Windows environment on a PC.

cross-relationships and the merging of identical ideas) into a single model which gave an overall representative view. This model was developed and validated working in a visual interactive mode with groups of senior members of the project team[3].

The cognitive map generated was large, containing 760 concepts and 900 links. This map was reduced to leave only those elements relevant to analysing the overall project spend. The analysis and clustering methods within Graphics COPE were then used to identify all the positive feedback loops, in order to understand how delay and disruption was generated by the dynamic impact of the known effects. The overall feedback loop structure was still complex (with over 90 feedback loops), as was the dynamics of the real situation - the overall behaviour of inter-connected and nested feedback loops is characteristically difficult to discern subjectively.

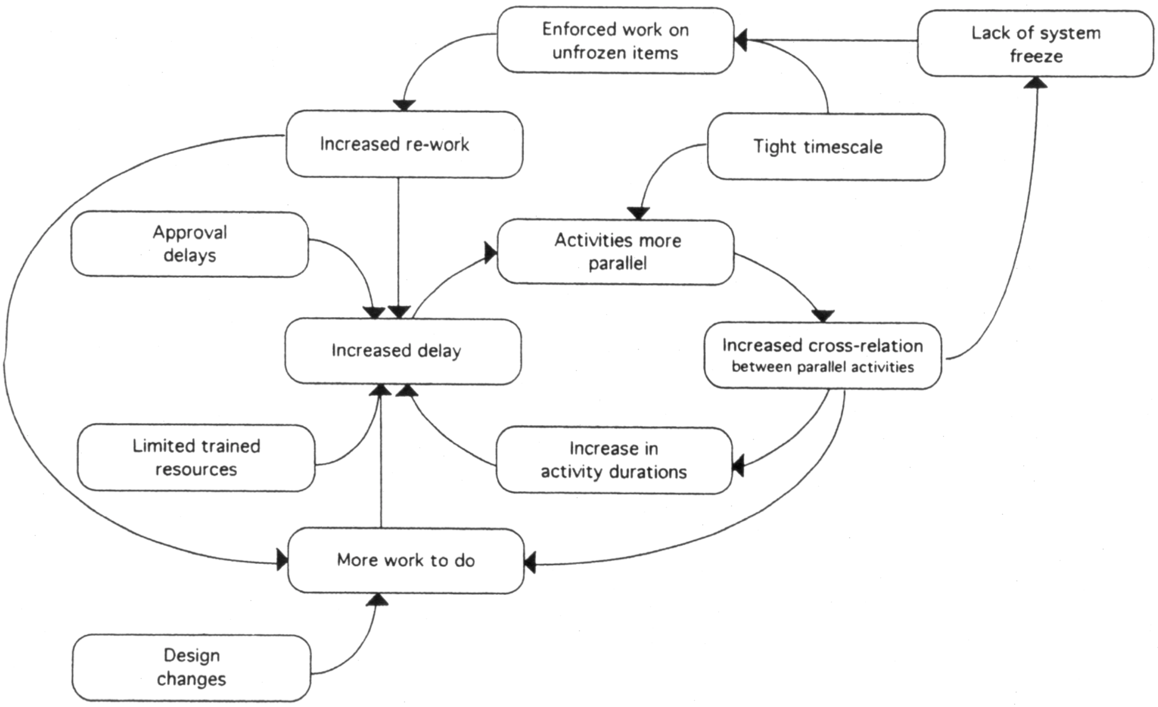

Figure 1: Main influences

Some key feedback loops are shown in Figure 1 (this Figure, and its constituent parts, are discussed more fully in Williams et al.[4]; note that this figure is designed to show the general effects operating; it is not an Influence Diagram in the technical (System Dynamics) sense). There were two exogenous inputs to the system: firstly, comment on design documentation by the purchaser took longer than planned; secondly, when the documentation was commented upon, there were substantive comments requiring re-work more often than planned (and, it was claimed, more often than reasonable or contractual).

Delays to approval beyond what had been planned meant that individual activities were delayed. However, there was a tight timescale-constraint, so that the project could not simply be extended; this meant that management had to make the project more parallel, by overlapping activities planned in the future with the delayed activities. This increased the extent to which design of inter-related components was occurring in parallel. This caused individual design activities to take longer, since each activity has to take cognisance of the others. Thus was one positive feedback-loop set up.

Furthermore, increasing the amount of parallel development of cross-related components implied increasing difficulty in providing a system freeze. Items should not be designed until the surrounding system has been defined and will not changed (termed "frozen"), otherwise changes to the system might mean that the specification of the item will change. In this case, increasing the amount of parallel development meant that changes in one component increasingly cross-impacted other components, and so on rippling throughout the system. When again combined with a tight timescale-constraint, this forced work to begin on components for which the surrounding system was not yet frozen, that is, components on which work had not been planned to start. This led to an increasing amount of re-work as changes to the surrounding (not yet frozen) system required changes to a component whose design had been started before plan.

The second exogenous factor, which formed the basis of the "Direct Claim", was the extent to which design changes were required by the project client, beyond both what had been planned and what was thought reasonable. Some of these changes simply required part of the design to be re-worked, while others required not only a re-work but a substantially greater amount of design work; these two types of change had to be treated separately in the quantitative model, but for the purposes of the qualitative modelling, they both input extra work into the system (as indeed did the re-work identified previously, and the increasing use of parallel development of related components). Since there were limited trained resources (the supply of design manpower trained in the particular domain, either for direct recruitment or sub-contract, began to be exhausted in the whole of the geographic region), this caused further delay, again adding to the positive feedback.

These factors combine to give a mutually reinforcing loop-structure with four positive feedback loops. Added to this, of course, is the effect of the extra invalid comments, which reinforces the loops.

Furthermore, there were a considerable number of other elements studied, but these show the major effects in the Design phase.

Further loops were set up when a concurrent Manufacturing phase was considered, both because design activities finish later and thus increase concurrency (and so on), and also because items begin manufacture and are then changed, which leads to retrofit, degradation of manufacturing learning[5] etc. This paper shows only the effects of the Design phase, firstly because the model is a self-contained model which demonstrates the points this papers seeks to make; secondly because the primary effects this paper seeks to demonstrate are within the Design phase; thirdly because the effects within the Design phase and the knock-on effects on the Manufacturing were more important in practice than the feedback from Manufacturing into Design. However, in the study carried out the two formed a single model and feedbacks from Manufacturing into Design played their part.

QUANTITATIVE TECHNIQUE

Having analysed the systemic effects, it was necessary to quantify these feedback effects, to provide a "quantum" for the legal claim. More generally, a model of the effects would be useful to planners and analysts.

The natural tool for studying feedback dynamics is System Dynamics (SD); indeed, SD was designed for exploring such effects. Furthermore, the use of this method followed naturally from the use of cognitive mapping and "Graphics COPE" software, which provides the initial structure of a influence diagram, which in turn provides the initial structure of a full SD model.

SD modelling was originally developed in the 1960's at MIT by Forrester[6]; Wolstenholme gives a good overview of the current state of the art[7]. SD methodology can be summarised as constructing a model by considering the way in which the <u>state</u> of the system changes with the <u>rate</u> of input and output of each variable which can be monitored. Changes in these rates depends upon an evaluation of the last monitoring of the system; this monitoring is modelled by the <u>auxiliary</u> variables which are used to represent the information and decision-making procedures. When constructing such a model for a project, the <u>state</u> variables monitored would be those things that could be counted if the system stood still - for example, in a plant the number of people working on site can be counted, the extent of completion of a product can be evaluated, and this monitoring used to determine whether extra staff should be hired. If staff are to be hired or fired then staff levels will subsequently change depending upon the rate of hiring or firing. In this way the simulation model replicates movement over time. The modelling approach focuses upon an understanding of feedback and feedforward relationships, and the model construction requires the analyst to construct the relationships between the various state variables and rate variables (flows).

The use of SD to study project behaviour is not new: SD has been used to explain the general effect of re-work[8], and indeed has contributed to a similar litigation case[9]. However, this study varies from the current emphasis on the use of SD. Such methods are often used in archetypical manner as part of organisational learning[10] where it is not particularly important that they are validated at the level of detailed output; they are used to demonstrate patterns of behaviour. In contrast, this model was required to report the quantum of behaviour under a variety of circumstances (e.g. the project as originally anticipated, the project as completed, and various different profiles of client behaviour and liability). Forensic modelling, in which the output was compared with what had actually happened, meant that it was essential to show that the model was valid; actuality couldn't be replicated exactly (this would lead to a model as complex as reality) but model had to show general outcomes "reasonably well".

Furthermore, this model went further than previous studies of the behaviour of projects using SD, as it attempted to analyse the combined effect of a multiplicity of loops, augmenting and exacerbating each other (a review of the use of SD to analyse project behaviour is given in Rodrigues[11]).

System Dynamics assumes that the system being modelled is continuous. Clearly, at the lowest level this system was not continuous, as it involved discrete documents and drawings. The team had to be convinced that the continuous approximation was good (indeed, the first author was initially of the opinion that it was insufficient), and needed to be prepared to defend this. It was found that the model seemed to exhibit the sort of behaviour the system had actually experienced (discrete models of similar systems had difficulty in modelling this sort of behaviour) and it replicated the known scenarios (i.e. planned and actuality) well. Alternative modelling techniques were exploited in an attempt to triangulate both these results and also those of intermediate scenarios, where there was no data available to check validity (Eden and Huxham[12]). Having said all this, there were some minor points at which the continuous modelling appeared unnatural, in particular the modelling of cross-impacts between design components.

A visual interactive modelling package known as "Stella"** was employed ("PowerSim"*** is similar). One of the most attractive advantages of such packages was their auditability: an analyst from outside the team could look at the model and know exactly how it worked, and there was no possibility of hidden "fiddles" or "fudge-factors". In the environment of a legal case, where the derivation of evidence had to be transparent to the other side, this was an essential requirement of a modelling package, which contrasts markedly with discrete-event models written in a base-language such as Pascal. Furthermore, the model was large with many input variables, implying that the project team could be accused of having considerable scope for fitting the model to the desired answer; therefore the input parameters had to be visible, and each supported by firm evidence (or at least the legal idea of 'best evidence': data superior (or at least equal) in quality to any other data which could be provided) in order to refute this accusation.

The SD model was developed in parallel with the COPE model. That is, the influences shown by the COPE model directly led to parts of the SD model. Conversely, the requirement to produce a coherent, explicit SD model threw up many questions about the influences, which required discussion and clarification, and hence informed the COPE model. The Macintosh computer on which the SD model was developed was kept in the same room as a large-screen display of the COPE model, so that this two-way interaction could indeed take place.

MODEL

Not surprisingly, the Stella model developed for the case was sizeable. The model consisted of two inter-related parts, one dealing with Design and the second with Manufacture: as discussed above, it is only the former part with which this paper is dealing. In this Design part, 29 states or stocks (the rectangular symbol in Stella) were used, with 43 flow-rates (the valve symbol); in addition, 120 auxiliary variables (denoted by a circle in Stella) were used to evaluate or store numerical values or intermediate calculations.

** Stella is a registered trademark of High Performance Systems, Inc, Hanover NH, USA. The package runs on Macintosh computers.

*** PowerSim v 1.0, developed 1993, is available from ModellData AS, Bergen, Norway. It runs in the Windows environment.

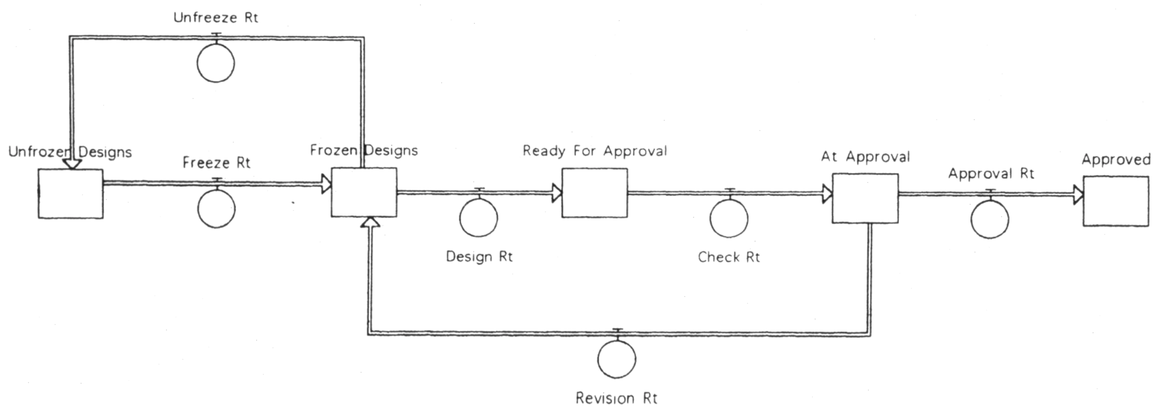

Figure 2: Extract from Stella model (1)

It would clearly be unprofitable to repeat this whole model here, but a simplified model showing the key chain of flow is shown in Figure 2. At the beginning of the project, certain of the work can be started since the system is well-enough defined (*Frozen designs*); the majority of the work, however, ought to wait until other work has been complete since there is insufficient information to enable a designer to start (*Unfrozen designs*). As the project progresses, and design-documents are approved, work flows from the second of these categories into the first (this controls the *Freeze Rt* in Figure 2, "Rt" being shorthand for "Rate"). The work that can be started is done at a rate subject to the available resources (*Design Rt*), and is then *Ready For Approval*. When the Client has checked the design (again at a certain rate *Check Rt*), the work is either approved or is required to be revised (work flows in and immediately out of the *At Approval* stock, with the two rates *Approval Rt* and *Revision Rt* merely dividing the flow in the required proportions). Revised work returns to have some more work done on it (not necessarily the same amount as the original design). Although in this simple case it is not relevant, also shown is a flow back from *Frozen Designs* into *Unfrozen Designs*, where revisions have cross impacts that mean that other systems are no longer frozen, as occurs in the full model.

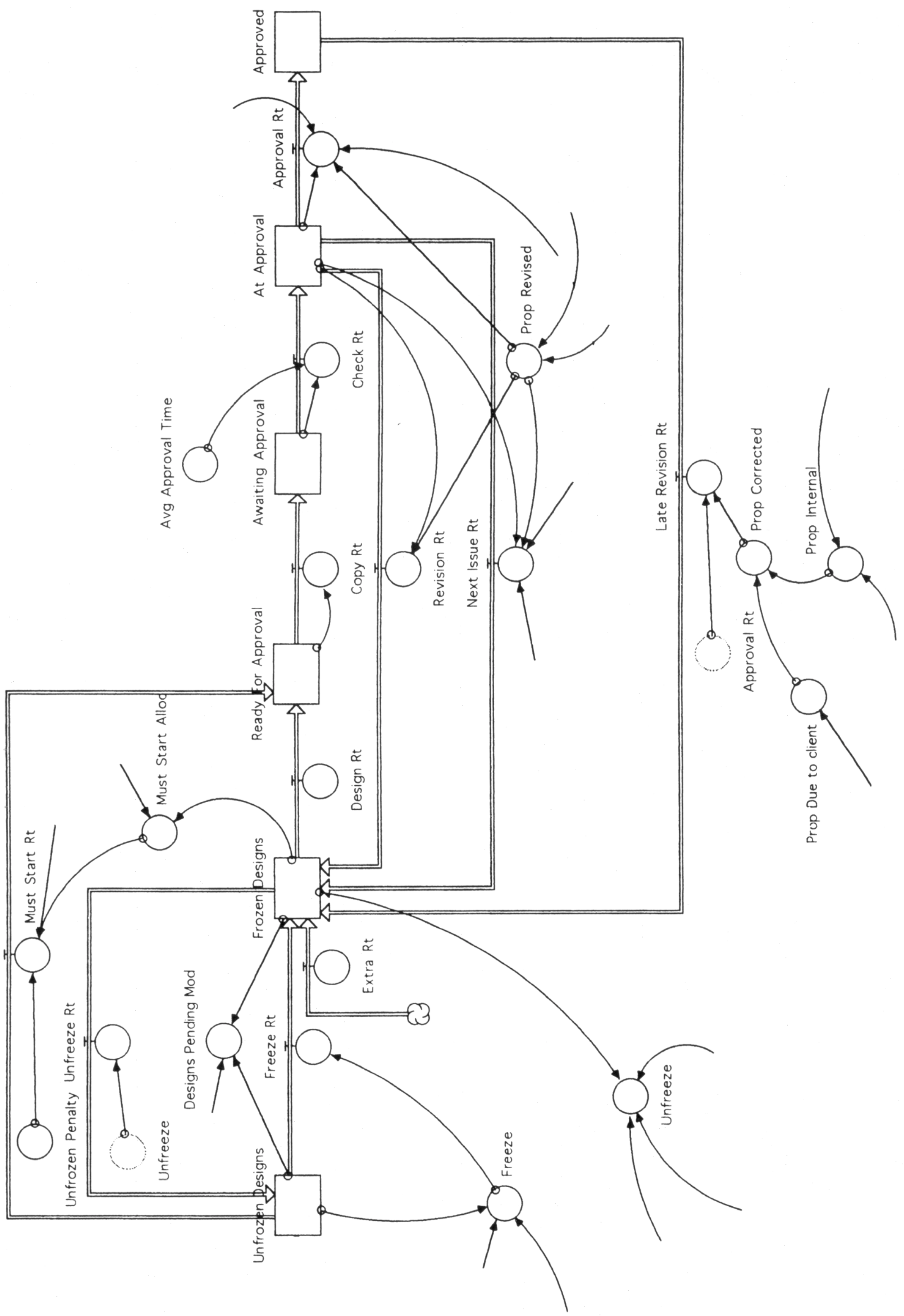

Figure 3: Extract from Stella model (2)

Figure 3 puts the model of Figure 2 into a wider context, although there are still many parts of the model omitted. This allows work to be done on the *Unfrozen Designs* if the time-targets governing the scheduling require it, when delays in the system mean that the design work-force runs out of frozen designs to work upon (*Must Start*). Such work is carried out less efficiently than work on frozen designs. More importantly, it is possible that such work will be done and approved, then changes to the surrounding system (which had not been frozen) mean that the design is incorrect. Thus there is a further feedback loop from the *Approved* stock due to internal corrections. This feedback loop is additionally due to comments by the Client after he had approved the document, and *Late Revision Rt* can be seen to be influenced by these two factors ("Prop" is short for "Proportion"). Work becoming thus "un-*Approved*" can cause previously frozen designs to *Unfreeze*, as their surrounding system is no longer agreed and approved.

Figure 3 also shows *Extra* work entering the system, when comments by the client (either made at the proper time or late revisions) required substantive changes or (as often occurred) required major enhancements to the product.

There is a considerable amount of the Stella model not shown in Figure 3. In particular, there was a large part of the model dealing with managing a workforce consisting of designers, free-lance designers, and a capacity for recruiting designers inexperienced in the domain. These last had been used in practice, but in the event were barely used in the model; this was because the workforce-management rules put into the model were made efficient in order to provide a conservative estimate of over-run; this was a point at which replication of actual policy was problematic. Other parts of the model not shown here tracked aspects such as the type of work in the system, cross-impacts between systems etc.

Since this was supporting a legal case, a considerable amount of checking was carried out. The modellers themselves carried out checks such as the dimensional consistency of each equation, the conservation of material during a run, and tests for consistency when varying the time-interval δt (a particular problem in SD). Furthermore, a separate modeller was employed to audit the model and its supporting data. Finally, Prof. Wolstenholme (quoted above) provided expert advice to the team.

RESULTS

The model was run for the case under a variety of circumstances: firstly, the project as originally anticipated, then adding in the various factors as experienced to see their differential effect. The results are confidential and are not replicated exactly here. For the purposes of illustration, the model has been run with closely approximating numbers, and the resulting number of man-hours scaled to a proportion of the original budget. The model was run firstly roughly as originally anticipated, then varying the levels of three factors (the term "level" here is used in the natural sense of the value of a parameter, and is not to be confused with the concept of a level as a state variable in system dynamics).

(1) Increasing the average time for a document to gain client approval/comment (*Avg Approval Time*) from the level contractually agreed (referred to below as the Low level) to a level nearer to that experienced in the project (referred to below as the High level). In fact, it is not only the average approval time but also the distribution of approval-time that has an effect, since a small number of crucial documents held up for a very long time - as indeed happened - have the effect of stopping much of the system being frozen. However, for simplicity this paper only considers the effect of increasing the average.

(2) Increasing the proportion of documents the client commented upon (but without requiring extra-contractual modifications to the product) from a Low level to a High Level: in fact, the first of these two levels was the proportion commented upon correctly by the client, and to this was added the proportion given invalid or incorrect comments to give the second level (as calculated by independent design engineers).

(3) Increasing the proportion of documents the client required extra-contractual work on, either unjustifiable changes or major enhancements to the product, including comments made both at the proper time and late revisions.

The last two contributed to the proportion of documents revised (*Prop Revised*) in Figure 3.

The results of the eight runs carried out, with the three variables each at two levels, were as shown in Table 1, where the last column shows the number of man-hours of designers used, normalised to 100 at the base case where all factors are Low. When the equivalent runs were carried out in practice, so that "Low" meant the anticipated value of the parameter, and "High" the actual value, the first run corresponded to the anticipated case (i.e. as budgeted, excluding contingency-allowances) and the last of these corresponded to the actual case. The results found in the actual case were not identical to those below, but they were similar.

(1) Average approval time	(2) Proportion Comments	(3) Proportion Extra work	Total man-hours used
Low	Low	Low	100.0
Low	Low	High	192.0
Low	High	Low	111.8
Low	High	High	267.9
High	Low	Low	100.4
High	Low	High	190.3
High	High	Low	113.1
High	High	High	325.7

Table 1: Results of model runs

The design of an experiment to establish the single and interaction effects arising from these parameters is not obvious. Given a set of input data, the simulations are long enough that there is very little uncertainty in the final spend. There are a variety of ways of approaching an analysis of the above base results. One possible (and perhaps the simplest) interpretation is to suppose that the total man-hours used (T) is 100 multiplied by parameters depending on which factor is there (an analysis-of-variance-type approach). Thus, the first experiment results in a T of 100; the second in a T of 100 * X_3, where X_3 is the effect due to (3); the third similarly gives T = 100 * X_2; the fourth has two factors applied, so T = 100 * X_2 * X_3 * X_{23}, where X_{23} is the compounding effect of having both (2) and (3) applied together. The last experiment would have 100 multiplied by seven multiplicative parameters, the three single-factor parameters, three double-factor parameters, and one triple-factor parameter X_{123}. Solving for the X's gives:

X_1	=	1.004
X_2	=	1.118
X_3	=	1.920
X_{12}	=	1.008
X_{13}	=	0.986
X_{23}	=	1.248
X_{123}	=	1.218

The actual size of the base effects is, to a certain extent, determined by the size of the High factors chosen in this study. However, the compounding effect of the factors can be seen. When comments increase (factor (2)), they add a certain proportion to the cost of a slack project (12%); however, when extra work is put into the system (factor (3)) and the comments increase *as well*, their combined multiplicative effect is bigger than the two individual factors combined (25% (X_{23}) bigger). If to this you then add in approval delays, which have little effect on a slack project, there is an effect 22% beyond what is explained simply by that additional multiplication. In fact, the effect of all three factors together is equal to the individual factors multiplied by 1.008 * 0.987 * 1.248 * 1.218 of the individual factors, or an extra 51%.

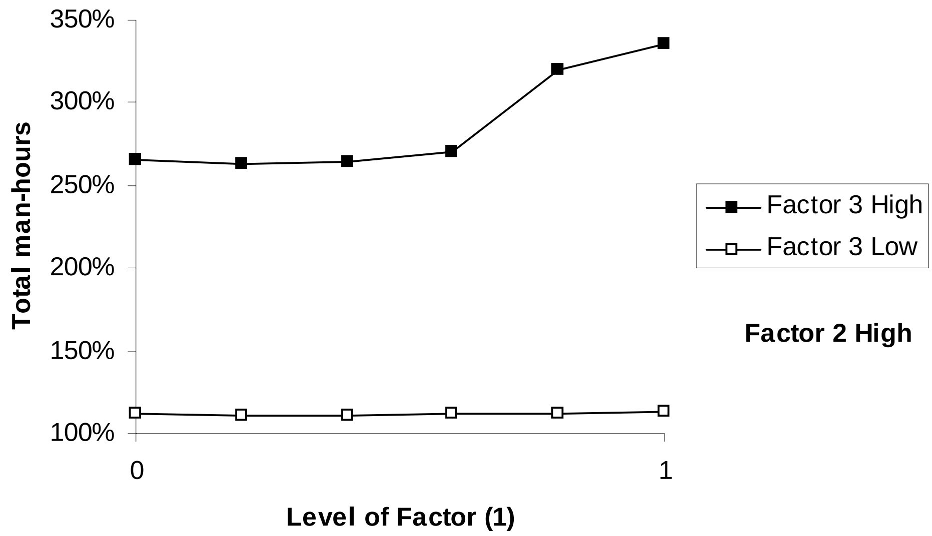

Figure 4: Effect of varying average approval delays (Factor (1))

This can also be illustrated graphically. When factor (2) is fixed at a High level, Figure 4 shows the effect of varying factor (1) between its two limits, with both factor (3) High and Low shown. When the project stays roughly the same size (factor 3 is low), increasing the approval delays (factor 1) has little effect; however, when the project is increased in size (factor 3 is high), the effect of increasing the approval delays is exacerbated.

The three parameters varied in Table 1 have here assumed to be constant over the whole simulation. Clearly this was only a first assumption, and further work was just starting to look at the time-varying behaviour of the parameters based on the data-base of client comments (Stella allows time-varying inputs), although no indication had been found when the case was settled that such secondary analysis would change the results markedly.

In the context of a legal case, the analysis above implied that the effects of the causes being claimed for, even when restricted to the three main factors above, were not additive. Thus, for example, if it were agreed that the client was responsible for the approval delays, the question of how much cost the client is liable for, is ambiguous. Taking the figures in Table 1 and adding approval delays to the project as originally envisaged adds only 0.4% to the cost; however, again taking the figures in Table 1, removing approval delays from the project as it actually occurred, reduces costs from 325.7 to 267.9, implying that the delays added 22% to the project cost. It is therefore difficult to express these items in the legal format of a claim, by which costs are broken down and each item given a label, since there is no suitable legal label to attach to the costs of factors compounding. (Similarly, in the legal domain, traditional "Extension of Time" claims methods[13], which only look at how individual activities have become extended, do not represent the full extent of the delay and disruption caused.)

WIDER IMPLICATIONS FOR NETWORK PLANNING

There are also wide implications for the use of network planning of projects. Standard network modelling would not have forecast, nor could have explained, the effects described here, which were crucial to the project outcome. Standard network modelling takes no account of management actions within the network, so that as feedback loops take effect, the network structure itself changes: activities become longer, and therefore become more and more concurrent, and the inter-relationships between activities change.

The System Dynamics ideas described here might provide an aggregate analysis method which would take into account the feedback effects, but it would not be appropriate as a detailed planning tool, since the individual activities and Work Breakdown Structure elements are not distinguishable. However, improvements to network methodology could be considered using the lessons of the SD model. These would essentially cover two areas. Firstly, there is a need for better modelling of activity inter-relationships, including recognising feed-back loops (i.e. if Activity X affects Activity Y, Y might also directly or indirectly impact X), perhaps following RiskNet[14] ideas. Secondly there is a need to recognise management intervention within the network, which might change resource allocation (i.e. resources allocated will depend on how late the project is) or even the network structure itself (perhaps increasing the extent to which activities occur in parallel, increasing the relationship between activities and setting up further feedbacks). Research is currently under way by one of the authors into such *Robust networks*. The SD model would provide a check on such a modified network analysis, since it is able to take into account the compounding effects as the perturbations and uncertainties combine.

REFERENCES

1. C.L.EDEN (1988) Cognitive mapping: a review *Eur.J.Opl.Res.* **36** 1, 1-13
2. F.ACKERMANN AND A.TAIT (1994) COPE-ing with System Dynamics - a story about soft and hard OR. Presented at the Young O.R. Conference, York, UK, March 1994.
3. C.L.EDEN AND F.ACKERMANN (1992) Strategy development and implementation - the role of a Group Decision Support System, in, R. BOSTROM, R. WATSON AND S. KINNEY (eds) *Computer Augmented Teamwork - A Guided Tour*, Van Nostrand Reinhold, New York
4. T.M.WILLIAMS, C.L.EDEN, F.ACKERMANN AND A.TAIT (1994) The vicious circles of parallelism. To appear in *Int.J.Proj.Mgmt.*
5. C.L.EDEN, T.M.WILLIAMS, A.TAIT and F.ACKERMANN Dismantling the Learning Curve: the role of learning in understanding disruption. EURO XIII Conference on O.R., University of Strathclyde, Glasgow 1994
6. J.W.FORRESTER (1961) *Industrial Dynamics* MIT Press
7. E.F.WOLSTENHOLME (1990) *System enquiry: a system dynamics approach*. Wiley.
8. K.G.COOPER (1993) The rework cycle: benchmarks for the project manager *Proj.Mgmt.J.* **24** No. 1
9. H.B.WEIL and R.L.ETHERTON (1990) System Dynamics in dispute resolution. Proc. 1990 International Systems Dynamics Conf., pgs 1311-1324.
10. P.M.SENGE (1990) *The fifth discipline: the art and practise of the learning organization* Doubleday Currency, New York.
11. A.RODRIGUES (1994) The role of system dynamics in project management: a comparative analysis with traditional models. Proc. 1994 International System Dynamics Conf.
12. C.EDEN and C.HUXHAM. (1993) 'Distinguishing Action Research' Presented at Brit.Acad.Mgmt. conf., Milton Keynes UK, Sept 1993, and available as Working Paper 93/18, Department of Management Science, University of Strathclyde.
13. S.SCOTT (1993) Dealing with delay claims - a survey. *Int.J.Proj.Mgmt.* **11**, 3, 143-154.
14. T.M.WILLIAMS (1990) Risk analysis using an embedded CPA package . *Int.J.Proj.Mgmt.* **8**, 84-88.